E-Assessment of Free-Text Answers Based on Domain Specific Sublanguages and Knowledge Representation

Victor Gonzalez-Barbone

Grado de Doctor por la Universidad de Vigo con Mención Internacional
(PhD Degree by the University of Vigo with International Mention)

Thesis Supervisor:
Martin Llamas-Nistal

Universidade de Vigo

Universidad de Vigo
Departamento de Ingeniería Telemática
Programa de Doctorado en Ingeniería Telemática

2013
ACADEMIC THESIS

to obtain the degree of

Doctor en Ingeniería Telemática

at the Universidad de Vigo, Spain.

Under the supervision of

Martin Llamas-Nistal.

Doctorate Thesis Committee:

Manuel Castro-Gil (UNED, Spain)
Luis Anido-Rifón (Universidad de Vigo, Spain)
Pablo Belzarena (Universidad de la República, Uruguay)

Defended in public on Monday 17th June, 2013 at 12:00

by

Victor Gonzalez Barbone

at the University of Vigo, Spain.

Qualification obtained: apto cum laude.
I hear and I forget.
I see and I remember.
I do and I understand.

Confucius (551–479 BC)

Vita brevis,
ars longa,
occasio praeceps,
experimentum periculorum,
iudicium difficile.

Hippocrates (c. 460 BC – c. 370 BC)
Acknowledgements

It was quite late in my professional life that I turned to research. For nearly thirty years I had been devoted to teaching and working, mainly in computer systems engineering. In 1999 I took a small position as an assistant teacher in the Institute of Electrical Engineering, from the Universidad de la Republica, Uruguay. I got gradually more involved in University teaching activities. Being research one of these activities, I naturally made some steps into it. An email told me of a Doctorate programme in Engineering at the Universidad de Vigo, Spain, that could be taken as distance learning. I got into it. My first formal activities in research were carried on under the guidance of Luis Anido Rifon. Later on, Martin Llamas Nistal accepted to be my supervisor for a Doctorate thesis. I am indebted to both of them for their guide, for their friendly treatment, for having always punctually answered my mails, and for having accepted me as a student much older than is usual for these studies. Pablo Belzarena was both a friend and a reference along the years of thesis work, always interested in my progress, ready to support me on moments of doubt, discussing with me some difficult points. Javier Couto read the thesis and provided valuable assessment and suggestions on Natural Language Processing related items. Cristina Ayçaguer read parts of the manuscript with her outstanding care for detail. Many other friends and relations did their bit to help me on this task, consciously or unconsciously. To them all, my thanks. As a last word in these acknowledgements, I would like to thank my father, Atilio Gonzalez Rivero, who in some subtle way led me to like knowledge for knowledge itself.
Abstract

Free-text answers can broadly be assessed from two viewpoints: correctness of knowledge content and quality of writing. Content evaluation ultimately results in a comparison of students’ answers against a reference answer or model. The reference may be obtained from teachers’ answers, manually marked students’ answers, a corpus of reference texts, or structures such as a semantic network. Techniques used for free-text assessment include keyword analysis, latent semantic analysis (LSA), surface linguistic features, text categorization, information extraction, and clustering. Many different systems exist, applied to different subject areas, using different evaluation metrics, which makes them very difficult to compare. The main weakness of most of these systems is the lack of a large enough corpus of marked answers or some other reliable reference. Dispersion among teachers has also proved to be a factor of distortion. In this work, the reference model is a semantic network inferred from text written in an easy to use sublanguage, built by the students themselves as a learning activity. This is expected to improve on the drawbacks of existing systems, since no training set is required, the reference is agreed upon by teachers and students, texts are written in an predictable way, terms are uniformly used, and the assessment system works in the same way the reference was created. Moreover, the proposal in this thesis is conceived mainly as a learning activity, supported by a learning tool, which can be used in assessment. The construction of a sublanguage usable by secondary school students is discussed, and some experimental sublanguages developed. Answers written in the sublanguage can be converted into a semantic network, which can be compared against a reference semantic network. The reference semantic network has been gradually put together by the students themselves, as an activity in the learning process, similar to taking notes, writing a summary or building a concept map. A proof of concept prototype includes facilities for guiding the writing in a sublanguage, transforming the resulting text into a semantic network, and colouring the reference semantic network to show what parts of the answer have been recognized, together with a mark. Assessment is prepared in a learning activity, marking is automatic, and feedback to students immediate. Several lexical resources are identified, tested and used to build a general purpose lexicon apt for different domains. Several versions of generative grammars are proposed and analyzed. A methodology for sublanguage development is proposed. Knowledge Representation is studied in its applicability to Education and Learning, identifying the most promising techniques. Several possible schemes of sublanguage and knowledge representation are analyzed, each targeted to a different purpose. A complete example of a syntax
based sublanguage and a semantic network as a knowledge representation model is
presented, and tested using the prototype application developed. Original contribu-
tions of this thesis result from the combination of assessment, sublanguages and
knowledge representation models in a practicable proposal that effectively integrates
assessment into the learning process, and a satisfactory end to end testing of the
system that proves its feasibility.

Keywords: eLearning, eAssessment, free-text answers, Knowledge Representation, Sublan-
guages, Education.
# Contents

4.4. Why use restricted languages .................................................. 64
4.5. Some existing proposals .......................................................... 66
  4.5.1. Machine oriented restricted languages ............................... 67
  4.5.2. Human oriented restricted languages for technical use .......... 68
  4.5.3. Human oriented restricted languages for general use ........... 69
4.6. A restricted language for Education ........................................ 70
4.7. Research lines .......................................................................... 71
4.8. Conclusions ............................................................................. 72

5. Construction of a Sublanguage .................................................... 75
  5.1. Introduction ........................................................................... 76
  5.2. Terms and tasks in NLP ......................................................... 76
  5.3. Design considerations .......................................................... 77
  5.4. Writing in a sublanguage ....................................................... 78
  5.5. From text to graph ................................................................... 81
  5.6. Tagsets .................................................................................. 83
  5.7. Lexicon .................................................................................. 84
  5.8. Rules ...................................................................................... 85
  5.9. Preprocessing ......................................................................... 86
  5.10. Enhancements ....................................................................... 88
  5.11. Conclusions ......................................................................... 89

6. Lexical Resources .......................................................................... 91
  6.1. Introduction ........................................................................... 92
  6.2. Terms ..................................................................................... 92
  6.3. Lexical Resources ................................................................. 93
    6.3.1. Word Lists Content and Uses ....................................... 94
    6.3.2. The General Service List of English Words .................. 95
    6.3.3. Simple English Wikipedia ........................................... 96
    6.3.4. VOA Special English ..................................................... 96
    6.3.5. Longman Communication 3000 .................................... 97
    6.3.6. BNC Word Frequencies ............................................... 97
    6.3.7. COCA ........................................................................... 97
    6.3.8. SCOWL, AGID and PoS DB ......................................... 98
    6.3.9. Wordnet ......................................................................... 98
  6.4. Evaluation and Comparison .................................................. 99
    6.4.1. The GSL versions ............................................................. 100
    6.4.2. The other short lists ........................................................ 102
    6.4.3. Comparison of the short lists ....................................... 103
    6.4.4. The long lists ................................................................. 104
    6.4.5. Comparison of the long lists ....................................... 105
    6.4.6. Coverage of the short lists in the long lists .................. 106
    6.4.7. Wordnet in numbers ...................................................... 106
  6.5. Conclusions ........................................................................... 108

7. Compilation of a Lexicon ............................................................. 109
  7.1. Introduction ........................................................................... 109
## Contents

7.2. Approach .................................................. 110
7.3. A data structure for compiled lexicons .................... 110
7.4. A GSL based experimental lexicon .......................... 112
   7.4.1. Aims and constraints ................................ 112
   7.4.2. Function words ........................................ 113
   7.4.3. Content words .......................................... 116
   7.4.4. Adjustments and additions to the lexicon ............... 117
7.5. Adding sense ................................................ 120
   7.5.1. GSL Gilmore in Wordnet ............................... 120
   7.5.2. From words to meaning ................................ 121
   7.5.3. Further exploitation of the Wordnet .................... 122
7.6. A note on Methodology ...................................... 123
7.7. Validating a compiled lexicon .............................. 125
   7.7.1. Coverage in the Brown Corpus ......................... 126
   7.7.2. A Longman based experimental lexicon ................. 127
7.8. Discussion .................................................. 127
7.9. Conclusions ................................................ 128

8. Syntax based sublanguages .................................. 129
   8.1. Introduction ............................................... 130
   8.2. Lexicon ................................................... 130
   8.3. Design considerations .................................... 130
   8.4. Tagset ..................................................... 131
   8.5. Procedure ................................................ 133
   8.6. Syntax-based Spanish grammar versions .................... 135
      8.6.1. Spanish01 ........................................... 136
      8.6.2. Spanish02 ........................................... 136
      8.6.3. Spanish03 ........................................... 136
      8.6.4. Spanish04 ........................................... 137
      8.6.5. Spanish05 ........................................... 138
   8.7. A syntax-based English grammar ........................... 138
   8.8. Discussion ................................................ 142
   8.9. Conclusions ............................................... 143

IV. Knowledge .................................................. 145

9. Knowledge Representation, state of the art .................. 149
   9.1. Terms ..................................................... 149
   9.2. Knowledge Representation and Reasoning .................... 150
   9.3. Knowledge Representation techniques ....................... 152
   9.4. Languages for Knowledge Representation ..................... 152
      9.4.1. Logic based ........................................... 152
      9.4.2. Rule based ........................................... 153
      9.4.3. Frame based ........................................... 154
      9.4.4. Visual Languages ..................................... 155
      9.4.5. Natural Languages .................................... 156
# Contents

- 12.7. Concept maps, mind maps .............................................. 214  
- 12.8. Topic Maps network .................................................. 215  
- 12.9. Conclusions ............................................................. 216  

## V. Solution Proposal

- 13. Overview of the Proposed Solution ..................................... 221  
  - 13.1. Thesis contents ..................................................... 221  
  - 13.2. Feasibility ............................................................ 223  
  - 13.3. View of the System ................................................ 224  
  - 13.4. How it works ........................................................ 226  
  - 13.5. Discussion ............................................................ 227  
  - 13.6. Conclusions .......................................................... 228  

- 14. The KLEAR project ......................................................... 229  
  - 14.1. The KLEAR project ................................................ 229  
  - 14.2. The Lexicon module ............................................... 230  
  - 14.3. Demo application .................................................. 231  
  - 14.4. Semantic graph module .......................................... 232  
  - 14.5. Dependency tree modules ....................................... 233  
  - 14.6. The Demo modules ................................................. 233  
  - 14.7. Conclusions .......................................................... 234  

- 15. Deployment considerations ............................................. 237  
  - 15.1. Purpose ............................................................... 237  
  - 15.2. A learning activity ............................................... 238  
  - 15.3. A suitable application ............................................ 238  
  - 15.4. Learning units ...................................................... 239  
  - 15.5. The representation scheme ..................................... 240  
  - 15.6. Documentation and tutorials .................................. 240  
  - 15.7. Basic knowledge and skills .................................... 241  
  - 15.8. Engagement of the actors ...................................... 241  
  - 15.9. Conclusions .......................................................... 242  

## VI. Final

- 16. Evaluation ..................................................................... 247  
  - 16.1. Introduction ............................................................ 248  
  - 16.2. Evaluation of assessment systems .............................. 248  
  - 16.3. Requirements for validation ..................................... 248  
  - 16.4. A classroom test ..................................................... 249  
  - 16.5. Proof of concept validation ..................................... 250  
  - 16.6. Sublanguage ............................................................ 250  
    - 16.6.1. Lexicon .......................................................... 250  
    - 16.6.2. Production rules ............................................... 251  
    - 16.6.3. The KLEAR editor ............................................ 251
22.1.4. Prototipos y desarrollo .................................................. 303
22.2. Conclusiones ................................................................. 304

23. Conclusiones ................................................................. 305
  23.1. Cumplimiento .............................................................. 305
  23.2. Una propuesta practicable ............................................... 306
  23.3. Una actividad de aprendizaje .......................................... 307
  23.4. La cuestión del lenguaje ................................................ 308
  23.5. Adquisición de conocimiento .......................................... 309
  23.6. Síntesis ........................................................................ 310
  23.7. Una última palabra ........................................................ 311

VIII. Appendices ................................................................. 313

Bibliography ........................................................................ 316

A. The Example Sublanguages ............................................... 327
  A.1. Tagsets ......................................................................... 327
  A.2. Spanish01 ..................................................................... 328
  A.3. Spanish03 ..................................................................... 331
  A.4. Spanish04 ..................................................................... 335
  A.5. Spanish05 ..................................................................... 339
  A.6. English05 ..................................................................... 343

B. The KLEAR Project Documentation ...................................... 349

C. Glossary ........................................................................... 351
Part I.

Introduction
This part describes the research problem undertaken, why it is considered to deserve attention, and the perspective under which it is treated in this work.

Chapter 1 describes the motivation which led to consider the problem of the automatic marking of free text answers, defines the purpose and scope of research undertaken, defines some guiding principles for the solution to be simple and practicable, describes the engineering research methodology applied, and summarizes the organization of this document.
1. Statement of the Project

Abstract. This thesis intends to propose a feasible system for the computer assisted marking of free text answers by guiding the writing of the answers into a deterministic subset of the language, and representing knowledge in data structures among which a comparison can be established. The solution must be simple enough to be within reach of a small team of instructors. The sublanguage must be natural enough to be considered correct normal writing. The knowledge representation scheme must be intuitive enough to be of use both for assessment and for learning. Students can create the reference knowledge representation instances as one of their learning activities, thus becoming involved in the preparation of the assessment instance. Existing proposals are usually based on the statistical treatment of the answers, requiring a number of answers to be manually marked or reserved for training. These approaches exhibit an inherent dispersion which widens the range of acceptance; unforeseen correct answers may go unnoticed or marked wrong; statistical uncertainty or different treatment of answers may be disputed by students. Existing developments in lexicons and wordnets, available toolkits for Natural Language Processing, and several practicable schemes for Knowledge Representation give a hint on the feasibility of the project. The improvement of writing abilities and the engagement of the students in knowledge representation activities add to the main interest of computer assisted marking. Characteristics of the solution are formulated as guiding principles. Adopted methodology is the engineering method proposed by Adrion in 1993; a traditional development cycle of requirements, design, implementation and testing is rather loosely followed for prototyping. This document is organized in seven parts: Introduction explains motivation and purpose; Assessment deals with state of the art in the eAssessment of free text answers; Language deals with the state of the art, existing resources, and the construction of a sublanguage; Knowledge reviews state of the art in Knowledge Representation, selects some models adequate for Education, develops a syntax based language and knowledge representation scheme, and analyzes other representation schemes; Solution Proposal provides an overview of the proposed solution, describes the system and testing code developed, and makes some considerations on deployment; a final part contains evaluation, contributions, future work and conclusions; Appendices include example sublanguages, documentation, glossary, and more technical material.

The automatic marking of free text answers or essays is a long standing problem. A number of solutions have been attempted, and promising results have been obtained, but no satisfactory solution has to date been found. It is a difficult problem, where the complexities of natural language add to the complexities of knowledge extraction and knowledge representation. This thesis proposes a solution based on guiding the writing of the answers to use only a subset of natural language, and representing knowledge in data structures among which comparisons
1. Statement of the Project

can be established. Both the subset of natural language and the data structures for knowledge representation must be intuitive enough to be of value both for assessment and learning, and hence equally useful for instructors and students. This chapter accounts for the motivation of the project, the foundations for the perspective adopted, and why this course of research is considered worthwhile to follow.

1.1. What this thesis is about

Probably the oldest and historically most popular way of determining what a learner knows about a subject is the word, written or spoken. In the last decades, the universalization of education and the extension of the learning period in a typical lifespan resulted in a nearly always insufficient and overworked staff of teachers and instructors at all levels. The task of knowledge assessment has always been heavy, tedious and time consuming. The increasing availability of computers and connectivity allowed to implement several forms of “closed questions”, nowadays commonly found in any modern learning platform. The main reason for the adoption of these forms of assessment is that they can be corrected automatically with very little or no uncertainty in the marking. However, even the most carefully designed set of closed questions falls short of the possibilities offered by a single open question. The open question directs the attention of the student to some aspects, facts or situations of a subject, and then the student is left alone to produce his knowledge in free text, writing as she wishes. The answer may be a short sentence or a long essay.

A naive approach to the automatic marking of free text answers would process the text of the answers trying to extract the concepts and relations therein recorded, and verify if those concepts and relations are correct, i.e. if they are also recorded in what is considered an authoritative reference. Hence, the problem becomes one of understanding text written in natural language. Natural language understanding is a topic in Natural Language Processing (NLP) and Artificial Intelligence (AI). The general problem of language understanding is considered AI-complete, which is an informal way of saying that its difficulty is equivalent to building a computer as intelligent as a human being.

The marking of free text answers is not so wide, though: knowledge is limited to some domain, and discourse occurs within this domain; many aspects of the use of language are predictable, at least to some extent. Not all types of question are suitable for automatic marking; questions related to factual information where no two opinions may be held seem the more tractable. Restricting the types of questions, the extension of the answers, and to some extent the style of writing, several strategies have been tried, and promising results have been obtained.

The most common approach involves the application of NLP and Statistics to the answers and also to some reference material, which may be a set of marked answers, a set of correct answers written by the instructors, a collection of reference texts, or some kind of data structure. All these proposals suffer from one or more of these drawbacks: a significant part of the students’ answers must be manually corrected to act as a reference; answers written by teachers do not always agree among themselves, thus unduly widening the range of acceptance; manually corrected answers may also suffer this kind of dispersion, also widening acceptance; an unforeseen correct answer goes unnoticed, and may be marked as wrong; most systems are not adaptable enough to assimilate a new correct answer or correct an error without starting all over again; statistical results, even with a confidence as high as 95%, may not be acceptable
1.1. What this thesis is about

for education instances, and be disputed by the students.

Although some other approaches have been tried, three stages may be uniformly recognized
in existing proposals: a language processing of the answers to detect knowledge, the recording
or recognition of this knowledge in some information structure or text corpus, and a comparison
of knowledge derived from the answers to knowledge derived from the reference material and
considered correct. The comparison results in the mark, generally a number, assigned to the
answer.

The proposal of this thesis follows this general way, but attempts to guide the writing of the
answers so that the resulting answer text is unambiguous within the knowledge domain, and
hence more tractable. Knowledge can be extracted from these texts in a predictable way, and
stored in some form of knowledge representation. A knowledge representation instance derived
from an answer can be compared to a knowledge representation instance of the same type taken
as a reference of the correct answer. A distance measure between these two instances results
in the mark given to the answer.

In this thesis, purely automatic assessment of free text answers is considered inadequate for
most educational purposes. Computer assisted assessment is proposed instead. The knowledge
representation instance taken as reference is designed to be dynamically adjusted: when some
structure not present in the reference is found in an answer, the instructor can include this
new structure in the reference, either as correct or incorrect. The next time this structure
appears in an answer, it will be recognized in the reference, and marked accordingly. A system
implementing this scheme can “be taught” by the instructor; the knowledge representation
instance taken as reference may be created before marking begins, and adjusted as marking
is being carried on.

The purpose of this thesis is the study and design of a feasible system of computer as-
sisted assessment of free text answers by guiding the use of language in the answers so as to
enable a deterministic knowledge extraction and comparison against a reference knowledge
representation instance.

The two main areas engaged in this project are Natural Language Processing (NLP) and
Knowledge Representation (KR). Both these areas are vast and complex, and both provide
a variety of strategies and tools which may be successfully used in the design and imple-
mentation of the proposed solution. Though proof of concept and prototyping require that
options be made among these possible strategies and tools, this thesis intends to examine
the most promising of them, trying to assess their effectiveness and the effort required for
their development. In other words, though the thesis restricts itself to proof of concept and
prototyping, some insight into the design and development of a would-be production system
is given through the evaluation of alternative strategies and tools. Consequently, a production
system may, and will most surely be, very differently implemented.

Another principle along which this thesis develops is keeping the usability of its proposed
system within reach of instructors only moderately skilled in Information Technologies. In
the general case, the design of a sublanguage requires a linguist; the design of a knowledge
representation calls for a knowledge engineer. This professionalized approach will surely result
in a much more powerful and widely usable system, but requires at least an institutional
consensus, the contribution of a number of experts, and financial support to develop. All this
may prove very difficult to obtain, even more if there is not a clear evidence of the usability of
the system. The proposed system must be simple enough to allow a small group of instructors
not only to use it, but also to understand what they are doing and the subsidiary benefits of
deterministic writing.
1. Statement of the Project

Assessment can be more fully integrated into the learning process and made less painful to students if they are given some part in the preparation process. The sublanguage and knowledge representation scheme can be used as teaching tools, students can actively take part in the construction of the knowledge structures that will be used as reference for the assessment of their answers. This parallels the proposal of using concept maps, mind maps, brainstorming or similar schemes for the acquisition and organization of knowledge as learning strategies. These same structures created for learning can be used as the reference structures for assessment. This thesis proposes a collaborative approach involving students and teachers in the construction of the reference structures which will be ultimately used in assessment, but which are created primarily for learning as normal activities of a course.

The rest of this chapter goes further into the motivation for this thesis, formalizes its purpose, defines its scope, explains the methodological approach, states some principles under which the thesis was developed, and describes how this document is organized.

1.2. Motivation

Automatic grading of essays, or more modestly, of short free text answers, has always been a highly rated prize among teachers and educators. Attempts to solve the problems posed by the automatic assessment of free text answers date from 60 years ago [Whittington and Hunt, 1999].

A number of different strategies have been employed, from surface analysis of style to complex knowledge representation structures, mostly using Natural Language Processing (NLP) and Statistics. Though much progress has been made, a comparison of a number of existing systems proves very difficult to determine which is best [Pérez-Marín et al., 2009]. To achieve a usable, reliable system, there is still a long way to go.

Most existing solutions are statistical in nature, and results are assumed to have an error. This error may be as small as 5%, but may be very difficult to diminish, since it arises from the nature of statistical processing itself.

Another limitation arises from the reference material against which answers are compared. In some subject areas, answers written by the teachers show a high dispersion among themselves, which widens the acceptance of students’ answers. Manually marked answers require separating the answers in two sets, one to correct manually and the other to mark automatically. The number of manually corrected answers may be a significant portion of the whole, reducing the benefit of the system. This also creates an uncomfortable difference between two groups of students, those whose answers were manually marked and those whose answers were automatically marked. Systems based on machine learning technologies exhibit the same limitations, since answers must be separated into a training set and a marking set. Since use of language is so variable and idiosyncratic, using marked answers from other years, schools or groups may add another source of dissent. Use of reference texts such as textbooks are not so prone to the former errors, and are known by the students, which may have read them to prepare for the test. Unfortunately it may prove very difficult to take them as references for correct answers: the use of language in textbooks differs widely from the typical use of language in a student’s answer, and a deep semantic or discourse analysis of these texts may be called for. As for references, different knowledge representation strategies have also been tried, the difficulties here being knowledge extraction from the answers (a well known, difficult NLP problem), the creation of the knowledge representation instance, and the inherent complexity of the language or data structure required to support the knowledge representation
1.2. Motivation

type selected.

This thesis sets out to explore the subject of free text answer marking from a different perspective, namely guiding the student’s writing to produce deterministic, more tractable text, and using a fairly intuitive but also deterministic knowledge representation scheme. Tractable text is obtained by the use of a sublanguage, a subset of natural language including only predetermined rules and words. This sublanguage must necessarily be domain specific, to avoid one of the many sources of ambiguity inherent to unrestricted natural language. A carefully correlated design of sublanguage and knowledge representation scheme allows for a quite straightforward knowledge extraction and recording into a knowledge representation instance. Two knowledge representations instances of the same type can be compared for coincidences and differences, weights can be assigned, and a distance measure defined. The distance separating the KR instance of the answer and the KR instance of the reference may be taken as a mark to be assigned to the answer.

This thesis addresses a difficult problem, which has been tackled by many researchers since a long time ago; no impressive findings can be reasonably expected. However, the perspective adopted is considered promising in several aspects:

- guiding the writing of answers to conform to a carefully designed sublanguage, domain specific and reasonably expressive, can do more good than harm, teaching students to write in a simple, clear way. Existing proposals in sublanguages and natural controlled languages have a reputation for clarity and simplicity. An application in computer assisted writing is certainly required, but such an application is feasible; a prototype has been developed for this thesis.

- there are usable NLP tools to help in the tailoring of a sublanguage for a specific domain. Vocabularies can be extracted from reference texts or commonly used word lists, Wordnets are at hand to select canonical wordforms for a concept and resolve polysemy and homonymy. Though this standardization is no light work, subject areas tend to have their own lists of terms and glossaries.

- among the many knowledge representation schemes possible, some are powerful enough and simple enough to be managed by non experts. Different forms of presentation, including visualization, are possible. Changes and additions can be done at any time, backtracking is possible and a set of answers can be corrected again with little effort.

- work on the Semantic Web has led to the development of knowledge representation in the form of ontologies, languages and schemes to support them, which offer a new range of possibilities as reference models and machine processable data repositories. Though the use of these resources may require a specific application or interface, there is considerable activity in the field, and a practicable solution may already exist.

- the proposed system is felt to be manageable and productive for the instructors: the use of adequate knowledge representation types allows the visualization of concepts and relations in a topic, and can be used not only in assessment but also in teaching. Students may know and contribute to the knowledge base which will be used as a reference when assessment time comes.

The former considerations have led us to consider the project described in this thesis as worthwhile. We all know by now that there is no silver bullet to defend us from the many
1. Statement of the Project

evils a software research project may have to face [Brooks, 1987], but we also know a wide perspective and careful consideration of all possibilities may lead us a long way on. The difficulties of the problem make any promising way deserve some exploration.

1.3. Purpose and scope

The purpose of this thesis can be succinctly formulated as follows:

This thesis intends to develop a proposal and build a software application prototype to assist in the assessment of free text answers, based on the use of a domain specific sublanguage to write the answer, the extraction of a knowledge representation instance from the text of the answer, and a comparison of this knowledge representation instance against a knowledge representation instance of the correct answer. A distance measure defined between two instances of the same knowledge representation type determines the mark given to the answer. The whole scheme should be simple enough to be understood, managed and applied by teachers with little more than current abilities in the use of Information Technologies. This thesis also intends to explore the use of tools and produce recommendations to build a domain specific sublanguage, to select and manage a knowledge representation, to define a distance measure, and to give guidelines to bring the system into the classroom.

The purpose of this thesis can be explained in the following categories and goals:

State of the Art

- to review the state of the art in the eAssessment of free text answers, evaluate existing solutions, and try to improve on their limitations.

- to review the state of the art in Natural Language Processing, in the fields of controlled languages and sublanguages, to know existing proposals, evaluate their usability in Education for the writing of processable texts, and to pinpoint techniques, resources and tools that may help in the adoption, adaptation or construction of a domain specific sublanguage, and also in knowledge extraction from the texts written in such sublanguage.

- to review the state of the art in Knowledge Representation, to determine which techniques and languages can be most appropriate for representing knowledge in an educational environment, to evaluate their usability for knowledge recognition in an existing knowledge structure, and the possibilities of defining a distance measure between to instances of knowledge representation structures.

Assessment and eAssessment

- study the requirements for an assessment system acceptable both for teachers and students. Most existing eAssessment systems have not succeeded in gaining the students’ acceptance, and exhibit a margin of approximation or error inadequate for educational purposes.
1.3. Purpose and scope

- whatever the solution proposed, evaluate its applicability in real teaching, and provide instructions and recommendations as deemed necessary to put the system to work. This includes but is not limited to specific course preparation, a suitable software application to support the system, development of the necessary skills to understand and use the application, tutorials and reference material.

- a prototype application to demonstrate the use of the system and the feasibility of its implementation.

Language

- adopt, adapt or develop at least one example sublanguage for the writing of free text answers, on which knowledge extraction can be deterministically performed. A sublanguage for use in teaching and learning must be restricted to be processable, but must also allow for correct writing: a piece of text written in the sublanguage must be as readable as any text written in the language, sound natural, and never be perceived as an artificial construction.

- detect, analyze and evaluate existing lexical and syntactical resources that may help towards the compilation of a lexicon and a set of rules of production for the definition of a sublanguage, or the complementation of a basic sublanguage with domain specific vocabulary, or even specific syntactic structures.

- adopt, adapt or develop a way of extracting knowledge from text written in the sublanguage into a knowledge structure. The conversion from text to knowledge structure must be deterministic: though knowledge structure instances need not be exactly the same, the knowledge conveyed by each must be the same, and it must be possible to compare them to determine if a piece of knowledge in one instance is present in the other.

Knowledge

- adopt, adapt or develop a knowledge representation language apt for use in teaching and learning, and usable also in assessment. Though concept maps and graphs are the usual choice, they may be conceived very differently; some kind of formalization or restriction may be required. Besides, other less common knowledge representation languages may be of interest for some kinds of knowledge (i.e frames or rules).

- define a distance measure among different instances of knowledge representations of the same type, e.g. some formalized version of concept maps. The distance between two knowledge representations instances is taken as the mark given to the text answer; hence, it must be something within control of the teachers, and it must be easy to assign weights to each concept or relation present.

Concerning scope, the following limitations are perceived:

- the solution proposed will be limited to factual knowledge, i.e. that kind of knowledge that can be expressed as assertive sentences, or imperative sentences for instructions. Depending on the knowledge representation language chosen, specific syntactic structures may be called for, but in any case knowledge is expected to be limited to the statement of facts.
1. Statement of the Project

- the solution proposed will be applied for the teaching and learning of some specific subject taught by some instructors in a certain school; no attempt will be made to build an ontology of the field, nor even to produce material to be used in a large scale. Though sharability and reuse are always possible, and hence the scaling of the system, it is initially proposed as a one classroom experience led by a small team of teachers.

- though the writing in a sublanguage is expected to help improve the quality of texts written by the students, the solution proposed should not be taken as a computer assisted writing system. Though this idea is in the basis of the proposal, such a system would require a very carefully developed sublanguage, a sophisticated application, and the coordinated effort of experts in several different fields.

1.4. Guiding Principles

Work in this thesis has been guided by some principles in regard to the solution. These principles act as non-functional requirements or constraints in a software development process. The proposed solution must:

- be easy to understand and easy to manage, not requiring much further mastering of information technologies than the ordinary abilities expected from present day teachers.

- be actually useful to teaching and learning, not only to assessment. Writing correct (though limited) natural language, selecting and organizing knowledge in semantic graphs (concept maps) or similar knowledge representation types, are all learning activities.

- lend itself to collaborative learning, engaging the students in the writing of text and the construction of graphs as they become familiar with the subject of study.

- be practicable, within the working capacity of a small team of instructors giving a course in a certain subject; parts of this subject become the specific domain where assessment will be applied.

- be designed with an engineering perspective: its ultimate purpose is to provide the foundations for the building of an application to effectively assist instructors in the marking of free text answers, and at the same time useful as a tool for the recording and visualization of knowledge.

- follow an empirical design view, steering clear of the many subtleties and theoretical discussions in Education, Language and Knowledge. The final vision is an application usable in the classroom, managed from the teacher’s desk, and shared with the students.

- limited to factual knowledge, expressed as assertive or imperative sentences.

This thesis intends to show its proposed scheme for marking free text answers is within reach of a small group of instructors, that the devices used are understood by the students, and are useful to learning. If the project escalates, the contribution of a linguist will surely add value to sublanguage definition, to the exploitation of linguistic resources, and the design of the rules. The same can be said for experts in NLP or KR, or professional software developers. For the time being, a set of prototypes for proof of concept is presented.
1.5. Methodology

The proposed system is not intended for literature, philosophy or the arts, where ample vocabularies and elaborate structures may be desirable or appreciated, but rather for the sciences, where conciseness and precision are required.

This work was carried on in the belief that knowledge should always be for the public benefit: all the software, recommendations and work in this thesis is intended to be usable and available to anyone.

1.5. Methodology

As early as 1993 W. Richards Andion reports on four identified methodologies for research in Software Engineering [Andion, 1993]:

- the **scientific method** starts with the observation of the real world, then builds a model or theory, and afterwards tries to validate this model or theory by measurement or analysis.

- the **engineering method** is an evolutionary paradigm: it starts with the study of existing solutions, tries to build a better solution, submits it to testing, and starts all over again until no further improvement can be achieved or is required.

- the **empirical method** is termed a revolutionary paradigm; it starts by proposing a model, then applies it to case studies, and tries to validate it by measurement, analysis or statistics; then this cycle is repeated.

- the **analytical method** proposes a formal theory or set of axioms, develops a new theory from them, derives results from this new theory and tries to validate these results against empirical observations.

Andion [op.cit.] also states that part of the difficulties in Software Engineering research methodology is a blurred frontier between Software Engineering itself and its scientific foundations in Computer Science (programming languages, data structures, algorithms).

A number of research methods exist and have been applied in Software Engineering, for different topics, involving different reference disciplines. A classification of these can be found in [Vessey et al., 2005].

Compared with other research methods applied in Software Engineering, the schematic proposals of Andion seem too sketchy and subject to different interpretations according to the problem, topic or reference discipline. However, this generality and openness are considered an advantage in the context of this project. This thesis looks for a practicable solution to the automatic marking of free text answers. Ultimately, this solution is expected to be the basis for the design of a suite of applications to manage all the assessment process: sublanguage definition, knowledge representation, questionnaires, computer assisted answering by the students, computer assisted marking by the instructors. Though a methodology of software engineering research and development is necessary and must be adopted, the main contributions of this thesis are not in the development of software applications, but in the judicious selection and application of knowledge and techniques from other disciplines, namely Assessment, Natural Language Processing and Knowledge Representation. Software development is subsidiary to the theoretical solution proposed, and limited to prototypes used for demonstration and proof of concept.

Research and development methodologies are to be selected or even adapted to the problem domain [Glass, 2004]. In the context of this thesis, the engineering method proposed by
1. Statement of the Project

Andrion is considered the most adequate to follow. A traditional software development cycle of requirements, design, implementation and testing, is adopted and rather loosely followed, in successive iterations, to develop prototype tools and applications, but essentially to confirm that the proposed ideas work in practice. Software development is, consequently, a research tool used to assess feasibility, achieve proof of concept, and demonstrate in practice the main ideas of this thesis.

A prototype end to end solution is provided, but many enhancements are possible. The technical areas on which this project is based, Natural Language Processing and Knowledge Representation, provide a number of tools and strategies to improve on a prototype, offer more help to users, and generally make matters simpler for the user. Whenever possible, these enhancements are indicated, and to a certain extent tried, so as to be applied incrementally in the future design of an improved system.

1.6. Organization of this document

This section describes the contents of the parts and chapters contained in this document. A summary version of this thesis has been translated into Spanish. The translated chapters can be taken as a quick reading guide to the whole thesis. These chapters are:

- Statement of the project, this chapter.
- Overview of the proposed system, chapter 13
- Contributions and future work, chapter 17
- Conclusions, chapter 18

Part I, Introduction, describes the research problem undertaken, why it is considered to deserve attention, summarizes current approaches, and explains the particular perspective adopted in this work towards a possible solution.

Chapter 1 describes the motivation which led to consider the problem of the automatic marking of free text answers, defines the purpose and scope of research undertaken, defines some guiding principles for the solution to be simple and practicable, describes the engineering research methodology applied, and summarizes the organization of this document.

Part II, Assessment, deals with Assessment and eAssessment, both in a general perspective to consider some educational aspects of interest for an acceptable solution, and in the specific subject of the automatic marking of free text answers.

Chapter 2 discusses several aspects of assessment, such as the educative expectations of the actors, the limitations of closed questions, the possibilities of free text answers, and their difficulties for automatic marking; reasons are given to prefer computer assisted marking over purely automatic marking, and the characteristics of an ideal system are sketched.

Chapter 3 reviews state of the art in eAssessment, describing some existing system for the automatic or computer assisted marking of free text answers, followed by a critical appreciation of their possibilities. This chapter concludes by placing the present proposal in the wider perspective of a learning activity of which the assessment instance is the final step.

Part III, Language, deals with the language related problems involved in the automatic marking of free text answers. Underlying discipline is Natural Language Processing, from which knowledge, resources, and tools are engaged towards a solution.
1.6. Organization of this document

Chapter 4 deals with state of the art in restricted languages, considering both controlled languages and sublanguages. Restricted languages are examined from an educational perspective, focusing in what they have to offer, not only to simplify the computational treatment of texts, but also as a subset of natural language supportive of simple, correct writing.

Chapter 5 analyzes the construction of a sublanguage appropriate for the writing of free text answers in an educational environment. Some design considerations are discussed, together with issues to consider when writing in a sublanguage, and the transformation of the text into a graph of some type. Tagsets, lexicon and rules are key components to consider.

Chapter 6 identifies, characterizes and evaluates freely available lexical resources towards their potential use in the construction of a sublanguage appropriate for educational use.

Chapter 7 describes the compilation of a lexicon of commonly used words based on the lexical resources formerly identified and evaluated; domain specific vocabulary can then be added to this basic lexicon.

Chapter 8 describes the construction of a sublanguage based on a general purpose lexicon and a set of production rules for assertive sentences. Rules for both Spanish and English are provided. These experimental sublanguages are designed to be processable by syntactic analysis, from which a knowledge structure may be inferred.

Part IV, Knowledge, deals with Knowledge and Knowledge Representation, studied from the perspective of Education and Assessment.

Chapter 9 reviews state of the art in Knowledge Representation. The main techniques and languages are examined, with a view to their potential application to learning and assessment.

Chapter 10 deals with issues related to the application of Knowledge Representation to Education. The five traditional roles of Knowledge Representation are discussed in their relative importance for learning and assessment, trying to determine what to look for when choosing a Knowledge Representation language to support educational tasks.

Chapter 11 proposes a syntax based sublanguage and knowledge representation scheme. The example sublanguages are taken as a start point to arrive into a knowledge representation instance in the form of a graph. Several potential difficulties and ideas on how to overcome them, are considered. A distance measure between knowledge representations instances is proposed, which accounts for a mark given to the free text answer.

Chapter 12 starts with an analysis of types of knowledge and knowledge topologies to represent them. Requirements for a sublanguage and knowledge representation scheme are identified. Some possible schemes for different types of knowledge are considered, such as object oriented for category and individual representation, imperative models for instructions, a sequential model for time related knowledge such as historical events, together with widely known concept maps, mind maps, and the more formal and promising Topic Maps standard.

Part V, Solution Proposal, describes the solution proposed in this work, the prototypes built for proof of concept, and deployment considerations for trial in the classroom.

Chapter 13 describes the proposed system, discusses its feasibility, gives a view of the system, and provides an example of its use. An analysis of the learning and assessment process the proposed solution implies is also given, describing the necessary steps to take when teaching a learning unit following the principles proposed in this thesis.

Chapter 14 describes the KLEAR project; KLEAR stands for Knowledge and Language for Education, Assessment and Research, and is the name under which testing and prototype software tools were developed. A Lexicon module defines a common data structure for lexicon
testing and support; a Semantic Graph module handles graphs and their visualization; Dependency Tree modules bridge the gap between the sublanguage and the graph representing the knowledge inferred from a text. A demonstrative application allows to write some text, eventually add to the lexicon, represent concepts and relations in a graph, and recognize the knowledge contained in a sentence as a part of the graph.

Chapter 13 goes into the requirements, preparation and general conception of teaching and learning involved in bringing the proposed solution into the classroom. Besides a suitable, friendly application, the proposal requires adequate planning, division of learning content into learning units, acquisition of some skills on the part of the students and teachers, and a collaborative approach engaging both students and teachers into the learning and assessment process.

Part VI includes evaluation of the proposed solution, contributions of this thesis, future work, and conclusions.

Chapter 16 discusses some issues related to the evaluation of assessment systems, requirements for their validation, how to conduct a classroom evaluation, proof of concept evaluation, and more specific aspects of evaluation related to the sublanguage and knowledge representation scheme adopted.

Chapter 17 describes the contributions made in this thesis, in the areas of Education, Assessment, Language, and Knowledge Representation, as well as the software tools developed, which may be taken as a start point for a production quality application to support the learning and assessment model proposed in the solution. Several directions of future work are also identified.

Chapter 18 discusses the conclusions inferred along this work. The chapter starts with an analysis of the extent of accomplishment of the objectives originally proposed in this work. Conclusions range through the three main fields involved in this work, namely Education and Assessment, Language and Knowledge Representation. The solution proposed is more like a complete learning activity than just as an assessment system; it is a learning activity that includes assessment as the final step of the learning process.

Part VII, Appendices, includes the following appendices:

- Bibliography,
- rules and example sentences for the various example sublanguages developed,
- a summary of the KLEAR project documentation,
- glossary of terms as used in this work.

1.7. About this document

In the writing of this document, several sources were consulted on how to write a dissertation thesis. The ones considered most helpful were [Turabian, 2007], [Levine, 2011] and [Chimneck, 1999]. These sources were also helpful on how to conduct research work. A thesis dissertation akin to the subject of this one, by Diana Perez-Marin, taken as a reference in this project, was also helpful to define the structure of this document [Perez-Marin, 2007].
1.7. About this document

Across the literature, technical terms are often not so uniformly interpreted as desirable: superposition, ambiguities, common words used for specific constructs, are a source of confusion to the newcomer. Whenever deemed necessary, terms are explained. No attempt at a definition is made; just the sense in which the term is used in the context of this work is stated as clearly as possible. Some words or phrases may refer to a discipline, to a class of things, or to an instance of this class. Knowledge Representation is a discipline, there are several types of knowledge representations (semantic networks, frames, rules), and a diagram representing a family tree is a knowledge representation instance. We reserve the capitals for the discipline, and refer to classes or instances in lowercase.

In this document, these words are taken as synonyms: teacher, instructor; student, pupil.

Gender neutral language is honored by using ‘she’, which must be understood as ‘he or she’.

Perhaps contrary to what is usual in the field, acronyms and abbreviations are very sparingly used, limited only to well known or persistently used concepts such as KR for Knowledge Representation or FOL for First Order Logic. Readability has been a major concern in writing.

Some controlled redundancy is included in the text, with the purpose of making the flow of ideas easier to follow, not forcing the reader to remember all particulars of formerly discussed material. Hopefully it has been done sufficiently enough to help the casual reader, and sparingly enough not to tire the attentive reader.

This document was written in \LaTeX\ using the LYX editor. Thesis format style is book Koma-Script. Bibliography was created using Bib\TeX\ and presented in APA like style. Diagrams were made with the Dia diagram editor. Underlying operating system was Ubuntu Desktop, Long Term Support (LTS) versions.
Part II.

Assessment
This part deals with Assessment and eAssessment, both in a general perspective to consider some educational aspects of interest for an acceptable solution, and in the specific subject of the automatic marking of free text answers.

Chapter 2 discusses several aspects of assessment, such as the educative expectations of the actors, the limitations of closed questions, the possibilities of free text answers, and their difficulties for automatic marking; reasons are given to prefer computer assisted marking over purely automatic marking, and the characteristics of an ideal system are sketched.

Chapter 3 reviews state of the art in eAssessment, describing some existing system for the automatic or computer assisted marking of free text answers, followed by a critical appreciation of their possibilities. This chapter concludes by placing the present proposal in the wider perspective of a learning activity of which the assessment instance is the final step.
2. Assessment and eAssessment

**Abstract.** Free text answers provides an excellent means for assessment, but it is hard work to mark this type of material. The automatic assessment of free text answers can be applied successfully to carefully written literal text explicit and literal text implicit questions, both categories very frequent in factual knowledge. Existing automatic marking systems exhibit results comparable to manual marking, but most of them fall short of educators’ expectations. Teachers like to keep control of marking, judge by themselves unusual answers, correct their own model answer on the fly. Hence, Computer Assisted Assessment (CAA) is preferred to fully automatic assessment. The inherent ambiguity of natural language and the difficulties of knowledge representation add to the constraints imposed by teachers’ requirements. If some limits could be put on the use of the language in the answers, natural language processing techniques could be used to transform texts into knowledge representation structures among which a comparison might be made, and a mark obtained.

Assessment is a sensitive matter. Students depend on assessment to advance in their careers; a failed exam means time and effort lost, sometimes money, and nearly always an ungrateful moment. For teachers, assessment is generally hard work, many times a wearing task. The different expectations of teachers, students, institutions, make it difficult to design an assessment system to conform all. With the advent of the computer, the possibilities of automatic marking are no longer a dream, causing both joy and worries. This chapter defines some terms related to assessment, and discusses some issues related to the use of computers in assessment: expectations of the actors, the use of free text answers, the types of assessment and questions, the requirements of an ideal system. Most of the material in this chapter was published in article [González-Barbone and Llamas-Nistal, 2008].

2.1. Terms

Use of words in a subject area is frequently different from their use in everyday language. Even within the same subject area, terms may be used differently by different authors. Standardization of lexicon is frequently a concern, but some degree of imprecision persists. In this section, as well as in the other sections on terms in the following chapters, the explanations given on terms just try to state the meaning of each term in the context of this work. No attempt at formal definitions has been made, but to explain as clearly as possible what each term refers to. For this section, the main sources for terms and definitions have been JISC E-Assessment Glossary [Jisc, 2006], the various dictionary entries collected by The Free Online Dictionary [Farlex Inc., 2010], and Wikipedia [Wikipedia, 2012b]; the wording is mostly ours.

EAssessment is the use of Information and Communications Technologies for assessment. EAssessment comprises the presentation of assessment activities and the recording of responses. According to JISC, eAssessment is an end to end process for learners, tutors, learning
2. Assessment and eAssessment

establishments, awarding bodies and regulators, and also for the public general. An assessment specification is a detailed description of methods, processes, tasks and criteria used to assess a learning objective.

An assessment objective is a single unit of knowledge, skills or understanding that a test is designed to assess in a candidate. This single unit of knowledge, skills or understanding defines a learning objective. Learning objectives and assessment objectives are usually part of a program of study.

Assessment criteria define what the learner must do during the assessment instance in order to demonstrate that a learning objective has been achieved. Assessment determines to what extent a candidate has achieved the assessment criteria.

Closed questions are a type of question or task in which the range of possible responses or outcomes the student can give are limited. Assessment by closed questions is called objective assessment. Closed questions are apt for automatic marking, because the given answers can be compared to a correct answer. Open questions are a type of question or task with no predetermined response or outcome. Assessment by open questions is called subjective assessment. Open questions are difficult to mark automatically. An essay question is a type of question which calls for a written answer of some specified length, usually a short essay. An essay is a short literary composition dealing with a subject, usually presenting the personal views of an author. Free text answers are candidates’ responses consisting of pieces of text of some length. Words or short phrases are not considered free text answers; these short pieces of text can be marked automatically by much simpler techniques such as pattern matching. Free text answers are long phrases or sentences, paragraphs, or a short essay.

Computer Assisted Assessment is considered broadly similar to eAssessment [Jisc, 2006]. In this work, we make a difference between Computer Assisted Assessment and Automatic Assessment: in Automatic Assessment the marks are given by a computer system, with no intervention of the teacher. In Computer Assisted Assessment, the computer system is a tool which helps the teacher at her request and under her control; in Computer Assisted Assessment the teacher can see and modify the markings according to her own criteria.

2.2. Free text answers

The most common form of open questions are those which call for free text answers; in the literature, open questions are sometimes understood as those requiring a free text answer. Automatic Assessment can be successfully carried out on closed questions such as [Castro, 2004]:

- Multiple Choice, with one or more correct answers.
- True or False.
- Short Answer, a word or simple phrase from a list.
- Numerical, an exact number or a number within a range.
- Matching, a two-column concept matching question.
- Calculated Questions: random values are placed into an equation, result differs each time the question is seen.
2.3. Educativ e expectations

Though useful to check whether students have grasped the essentials, closed questions offer poor information on the student’s ability to actively manage and apply their recently acquired knowledge. Automatic marking of closed questions does not require much work, but preparing closed questions tests remains a tedious, exacting, time consuming chore which must, moreover, be carefully undertaken.

Open questions are a common way to get an insight into the conceptual maturity a student may have achieved after a learning period. In a free text answer the student must produce a piece of text using her own choice of words and form of expression. Questions can be very specific, with only some phrases or a sentence for a correct answer, or require the student to write a brief essay on the subject. Even for short answer questions, the marking of open questions tests proves much harder than that of closed questions tests. The automatic marking of free text answers is still a field of research, and the problem remains unsolved in the general case. Subject areas of factual nature, such as the natural sciences, or human sciences such as history or anthropology, offer better conditions for automatic marking, since most of their knowledge can be expressed in assertive sentences, and refer to objective knowledge.

Teachers keep reluctant to the use of free text automatic assessment systems [Pérez-Marin et al., 2006]. Besides the difficulties of learning to use a new tool, and choosing one in the first place, some educative goals may not have been adequately addressed so far. A brilliant talk or piece of writing full of meaning to a human audience may be marked as complete nonsense by an automatic system. Not any type of question may be automatically assessed. A statistical probability will not be acceptable as a mark unless endorsed by a human teacher who has seen the answers by himself. Automatic marking depends on reference material not always available, or a carefully constructed reference answer. Teachers would not like to lose contact with their students. Computer Assisted Assessment, instead of purely automatic marking, is closer to teachers’ and students’ expectations of a reliable assessment system.

Closed questions may be used cost effectively to verify comprehension, within their known limitations. Generally, asking students to construct something provides better assessment than asking them to select among a small set of alternatives [Sargeant et al., 2004]. The use of open questions may provide benefits in at least two dimensions of assessment: conceptual maturity and ability of communication. Being able to select concepts from reality (or a question), search for related concepts in the student’s mental body of knowledge, establish adequate relations, is closer to the creative process involved in the retrieval and application of knowledge. Ability to clearly express ideas, choose the adequate words, combine them in a bunch of readable sentences, cannot but help students become familiar with the objective, concise, matter of fact style most professional writing demands. Though training in report writing is included in most undergraduate courses one way or other, writing small pieces of free text in an exam will certainly do no harm.

2.3. Educativ e expectations

The expectations of students and teachers concerning assessment may vary wildly; the assessment of free text answers is a difficult problem which makes it very challenging for any system to conform all; requirements for a reliable, acceptable system, are not easy to meet. The following points reflect our view of the most relevant expectations and values which must be considered in the design of a successful system.
2. Assessment and eAssessment

In Education, the different actors have different expectations concerning automatic assessment:

- institutions look for a reliable, cost effective system.
- teachers try to lighten the burden of their work without resigning their own judgment of students' assessment work.
- students expect a fair system capable of reflecting their true knowledge and capabilities, with results available in a reasonably short time.
- developers attempt to restrict the generic, intractable problem of natural language understanding into a computationally feasible task, imposing as few limitations as possible on the other actors’ expectations.

A successful system should be able to satisfy all these expectations to some degree. In open answer assessment several ambitious educative values come into play:

- *the language as a means of expression*. In written or oral work, a student is expected to exhibit a mastering of language adequate to the subject in question: a clear expression, richness of vocabulary, adequate use of sentence structure, a measured use of simile and metaphor, concision without awkwardness, a style adequate to the subject, some ambiguity to pique the reader’s imagination, a cold determinism to report facts. Richness of language is that of life itself. In factual writing, however, clearness of expression, precision in choice of words, conciseness and matter-of-fact style are the most outstanding virtues. Mastering of language will help or hinder a student’s career to a greater extent than is usually seen.

- *original ideas*. No true educator would accept to let go inadvertently a student’s original idea. Creative, original ideas are in the core of human progress. It is not known how much originality is being quenched by our present assessment systems, each day more structured.

- *the value of concept*. A human educator can easily differentiate a badly expressed correct concept from an erroneous concept, thus rewarding the idea and penalizing the expression.

- *error correcting*. An assessment instance is also an educative instance: a student’s misunderstanding can be detected and corrected during the assessment process.

- *learning process correction*. The presence of an error, especially if recurrent, may indicate a flaw in the assessment or the teaching process.

A first limit emerges on automatic assessment. It would not be possible to assess automatically:

- the use of language and expressiveness of an answer, in the general case. In disciplines where technical and factual matters demand a uniform, strict style, this is not as severe a limitation as in subjects where a more narrative style is required or possible.

- literary, philosophical, religious or creative areas where originality in style and expression may be close to the essence of the subject. Again, this is not a limitation for technical disciplines.
2.4. The challenge of free text answers automatic assessment

- well conceived but wrongly expressed ideas.
- automatic or assisted assessment would be generally limited to the factual aspects of a subject, in the best case to opinions or pieces of criticism of wide acceptance, already present in the assessment reference material.

Most teachers would require several facilities from an automatic or assisted assessment system:

- see by themselves uncommon answers, were they original in their concepts or simply wrong.
- be able to change a mark given automatically, to reflect the teacher’s valuation of the answer.
- be able to change on the fly the marking criteria on coming to a yet unregistered error or a possibly correct answer not included in the model answer, without be compelled to backward revise all the marking.

These reasonable, desirable features rule out, in Education, a purely automatic assessment system, at least in the present state of the art. A Computer Assisted Assessment system would carry most of the marking burden, automatically marking or suggesting marks for the most common situations, but keeping the active role of the teacher, drawing her attention to the infrequent answers, were they erroneous or original. The system would learn from the teacher’s judgment by including a new type of error or a new correct or partially correct answer, modifying the model answer accordingly [Jones, 2005].

2.4. The challenge of free text answers automatic assessment

The automatic assessment of free text answers as such, with no limitations, is an intractable problem in the present state of the art. Natural language is inherently ambiguous, both in word meaning and discourse; the same ideas can be expressed with very different words and syntax; cross reference among sentences may be difficult to infer; "knowledge of the world" is frequently required to understand the question and provide an answer. In the last decades, Natural Language Processing (NLP), a field of Artificial Intelligence (AI), has achieved outstanding results by the use of Statistics in the analysis of large collections of annotated texts (a corpus) [Hermet et al., 2006]. Considering open questions addressing summative assessment, where emphasis is on content and not on style or other formal properties, the automatic marking of free text answers to the level of accuracy required in Education is recognized as an AI-complete problem [Sargeant et al., 2004]. Such problems are intractable with present day means and knowledge. Even trivial free text answers may prove very difficult to mark. A question on probability where the correct answer could be expressed as 0.18, 18% or the calculation leading to the result, produced 117 different answers in 144, when answers were treated as text strings [Sargeant et al., 2004]. Not all questions are so demanding, but other problems persist. Misspelling of correct answers is very common, and can be dealt with accepting words within an "edit distance". Edit distance can be defined as the minimum number of character substitutions required to convert a word into another. Accepting words at an edit distance of 2 reduces considerably the number of answers which must be seen. In short words, however, even an edit distance of 2 may be too much [Jones, 2005] [Sargeant et al., 2004].
2. Assessment and eAssessment

A more difficult problem to deal with is that of context dependent synonyms, the collection of words that can be considered as reasonable substitutes of the ones mostly accepted within a particular field of knowledge. Context synonyms are very frequent; a number of them may turn up even in a simple question. An attempt to predict them may prove not worth the effort; in the end context synonyms must be dealt with when they turn up in the answers, by human intervention [Sargent et al., 2004].

2.5. Types of assessment and questions

Assessment may be summative or formative. **Summative assessment** attempts to verify if the intended educative goals have been achieved, typically in the form of an exam or similar instance. This type of assessment has also diagnostic value, since it helps detect deficiencies in the learning or the teaching. **Formative assessment** aims at producing achievements on the part of the students; it is a bidirectional process between the student and the teacher to enlarge, recognize and respond to learning [Wikipedia, 2012b, Educational Assessment][FSU, 2011, chapter 2].

Summative assessment may be attempted by means of Computer Assisted Assessment. Formative assessment is a complex, interactive process; the computer can only provide a modest support [Hermet et al., 2006].

Questions can be classified in two dimensions: cognitive and formal. In the cognitive dimension three categories are recognized, each one associated to a different form of comprehension: literal, interpreted and critical [Hermet et al., 2006]:

- **in literal questions** meaning is directly conveyed in the words and expression.
- **interpreted questions** require to elaborate on the words to get the meaning, which must be inferred from the text.
- **critical questions** demand careful judgment or judicious evaluation on the part of the student, sometimes according to his own notions on the subject.

Only literal questions can be put to automatic marking with present day means. Literal comprehension questions deal with definitions and causal relations in texts [Hermet and Szpakowicz, 2006].

In the formal dimension questions can be classified as text explicit, text implicit and “script implicit”:

- **in text explicit questions** the answer is right in the texts available to the students.
- **in text implicit questions** the student is required to search for information and make links and inferences across the text.
- **in script implicit questions** the student is required to draw on his own knowledge, besides any text, to build an answer. This last category requires the student to perform an inference involving her "knowledge of the world"; these questions are very difficult to mark automatically [Hermet and Szpakowicz, 2006].

Factual knowledge can be assessed to a great extent with text explicit or text implicit questions. These types of questions can be treated by recovering, at least partially, the necessary fragments from one or several sentences in the answer text, and comparing to a reference answer, a reference text or a knowledge structure. Considering only literal text explicit
2.6. The case for Computer Assisted Assessment

and literal text implicit questions automatic evaluation of free text answers becomes possible [Hermet and Szpakowicz, 2006] [Hermet et al., 2006].

2.5.1. The assessment process

This section provides a short revision of the assessment process considering the possibilities and limitations of Computer Assisted Assessment (CAA).

- Preparation.
  - question types. Knowledge of the type of question and its possible answers simplifies its preparation and enables for automatic marking.
  - question validation. Tests should be proposed to other teachers or assistants to ensure adequacy of the questions and their expected answers.

- Application.
  - the assessment instance. Students answer the questions on their computers, in the school premises or remotely.
  - feedback to student. Computer Assisted Assessment does not allow for immediate feedback to the student; a teacher must go through the answers, with the help of the system. For immediate feedback closed answers offer a better alternative.

- Marking.
  - from reference texts. There is a variety of ways in which a collection of reference texts can be used in automatic marking. Atypical answers are presented to the teacher, who marks them indicating correct or incorrect concepts and relationships. The Computer Assisted Assessment system "learns" from the marked answer. As the marking goes on less and less atypical answers are found, as the system registers more of the possible variants.
  - by dynamic construction of the model answer. The answers are presented to the teacher, who marks correct and incorrect concepts and relations. The system registers one or both. On finding similar situations the system marks new answers according to previously registered correct or incorrect answers.
  - marking revision. Exam results very near to the minimum score required for approval should be revised by the teacher. Ideally there should be very few, since the types of questions which can be automatically marked are little prone to differences in criteria.

2.6. The case for Computer Assisted Assessment

Computer Assisted Assessment or a variant called HCC (Human Computer Collaborative) assessment, have been proposed as alternatives to purely automatic systems [Jones, 2005]. Computer Assisted Assessment deliberately calls for human intervention or "moderation" in the marking process [Mitchell et al., 2003]. Human intervention makes the whole process more reliable and versatile, offers increased warranties and is closer to educators’ expectations, at
2. Assessment and eAssessment

the same time lightening the burden of routine work without losing too much contact with the students.

Automatic marking of free text answers has shown outstanding achievements by processing statistically large collections of text. It is not surprising to find coincidence with manual marking as high as 95%. Besides possible objections to the validity of the comparison because of the potential diversity of teachers’ answers or marking, no student will be satisfied to get a mark which is "95% correct": it would mean to accept a student among twenty has been wrongly marked. Perhaps human marking is not better than that, but the student will more easily accept human judgment than a machine mark, or at least is more in the habit of doing so. Perhaps in personnel selection or other areas this error might not be of major consequence, but in Education every effort must be done to avoid it, offering the student as much warranties as possible, specially if systems are known to be not fully reliable.

Computer Assisted Assessment allows to go without training material, enabling the dynamic creation of a knowledge representation or model answer during the marking process, showing the teacher several answers to the same question for comparison and judgment [Jones, 2005], or extracting from selected texts carefully worded text implicit or text explicit questions. Sound knowledge of the type of question proposed enables and simplifies the handling of the answer [Hermet et al., 2006].

2.7. An ideal system

Following the list of expectations and values formerly proposed, together with the analysis of the types of questions and the assessment process of free text answers, the characterization of an ideal system follows. Though arguable, the design of the system proposed in this thesis was oriented by the requirements included in this list.

A computer assisted software for open questions assessment would, ideally, comply with the following requirements:

- preferably not require the generation of reference material. Hard work will almost certainly be required to achieve the qualities of accuracy, determinism and completeness necessary for reliable correct answer references.

- preferably not be based on manually marked answers. Many teachers do no like to propose the same questions in different exams, the subject changes, the course contents change, questions become obsolete, new textbooks differ from former ones, students interchange information among them.

- be friendly, easy to use, when devising questions and model answers, perhaps through graphics of concepts and relations. An example of such a system, based on templates, can be seen in [Mitchell et al., 2003].

- transparent handling of lemmatization, syntax and other language aspects by means of adequate Natural Language Processing techniques, in an effort to recognize similar semantic content besides differences in language expression.

- effective handling of context synonyms within the subject area, were it by the use of specific dictionaries or the ability to create them, preferably dynamically, during the marking process [Mitchell et al., 2003].
2.8. Conclusions

- show the teacher all atypical answers for his direct assessment, allowing her to eventually include them in the model answer template or structure; do not show answers conveying the same knowledge and differing only in their language expression.

- show any answer at the teacher's request. Many teachers would like to see several answers at the same time for the sake of comparison [Sargeant et al., 2004].

- automatic backward marking when including a new right or wrong answer in the model template. This feature is unavoidable if the model answer is to be dynamically built during the marking process.

Many other desirable features come to mind. Only the most significant have been included, attending to the quality of the assessment, ease of use, clearness in presentation, and teachers' acceptance. Even within the perhaps limited set of features formerly outlined it may be very difficult to fully accomplish such a wish list.

2.8. Conclusions

Teachers, students, and studies in Education may point out many different requirements for an assessment system to conform all. Acceptance of an assessment system by both students and teachers is a major concern in this thesis; to this purpose, this chapter provided the following contribution:

- an evaluation of importance and a selection of requirements to be considered for a reliable and acceptable system. The expectations of actors, the educative values in an open answer system, the limits of automatic assessment, the facilities expected from a system, and as a consequence what may be considered an ideal system, have been analyzed, selected and organized according to our own view. Though they are by no means conclusive, they provided an orientation to our work.

Free text answers provide an insight into the conceptual maturity of students' knowledge and their communications abilities. Educators' expectations make Computer Assisted Assessment preferable to the purely automatic assessment of open questions. Computer Assisted Assessment of free text answers becomes possible if question types are limited to those apt for treatment, namely text explicit or text implicit questions. An ideal system must not depend on training material, must be reliable and easy to use, must allow for teacher intervention, adapt to new answers or criteria, backward mark already seen answers.

Assessment of free text answers implies natural language processing for knowledge extraction. This is a very difficult problem in the general case. If some kind of limitations could be posed on the language so that texts could be written in an unambiguous, predictable way, natural language processing techniques would be far more effective. This leads to our proposal of domain specific sublanguages.
3. EAssessment, state of the art

Abstract. Free-text answers can broadly be assessed from two viewpoints: correctness of knowledge content and technical writing. Content evaluation ultimately results in a comparison of students’ answers against a reference answer or model. The reference may be obtained from teachers’ answers, manually marked students’ answers, a corpus of reference tests, or structures such as a semantic network. Techniques used for free-text assessment include keyword analysis, latent semantic analysis (LSA), surface linguistic features, text categorization, information extraction, and clustering. In a study dated 2009, twenty two systems were described; their comparison proved very difficult because these systems were applied to different domains, used different corpora, and different evaluation metrics. No agreement could be reached as to which system was the best, but their main weakness was recognized to be the lack of a large enough corpus of marked answers to be used as a reference. Dispersion among teachers’ answers is also a factor of distortion; in a study dated 2010 a well trained machine showed to be more consistent in marking than human instructors. The Willow tools and Multinet Working Bench (MBR) are proposals akin to this work. Willow produces concept maps from text in unrestricted natural language; MBR creates a MultiNet semantic network from restricted text. In this work, the reference model is a semantic network inferred from text written in an easy to use sublanguage, built by the students themselves as a learning activity. This is expected to improve on the drawbacks of existing systems, since no training set is required, the reference is agreed upon by teachers and students, texts are written in a predictable way, terms are uniformly used, and the assessment system works in the same way the reference was created. Moreover, the proposal in this thesis is conceived mainly as a learning activity, supported by a learning tool, which can be used in assessment.

This chapter reviews the criteria and techniques employed by different systems for the assessment of free-text answers, briefly describing some existing systems. A critical appreciation of existing systems is made according to the stated purpose of this work. The conclusions include a summary on how the ideas in this work are expected to improve on existing systems.

3.1. Assessment criteria

A careful review on the state of the art of Computer Assisted Assessment of free text answers, dated 2009, can be read in Pérez-Marín et al., 2009. The authors point out most systems concern with the assessment of either short answers or essays; relatively few tools address both types. Two broad categories of assessment can be distinguished: essay content and technical writing quality. Essay content assessment tries to determine the accuracy of the concepts and relations contained in a text. Technical writing quality concerns the use of words, syntax, organization, and style. The traditional approach to technical writing quality assessment
3. EAssessment, state of the art

measures some specific qualities of the text, such as number and length of words, to infer more abstract qualities such as variety, fluency, or style. For content evaluation there are a variety of approaches, but most of them ultimately result in a comparison of the student’s answer against some reference, in the form of an ideal answer or a template. The reference answer may be obtained from a small set of teachers’ answers, a training set of manually corrected answers, a corpus of reference texts, or from some more elaborate structures such as a semantic network.

3.2. Techniques

Perez-Marín et al. provide a classification of techniques according to the type of natural language processing used [Pérez-Marín et al., 2009]. Shallow natural language processing (shallow NLP) uses statistical and machine learning techniques to analyze texts at the lexical level; typical tools are sentence splitters, word tokenizers, and part of speech taggers. Deep natural language processing (deep NLP) use manually developed grammars and language resources, such as syntactic analyzers to identify the constituents of a sentence and their syntactic dependencies, rhetorical parsers to determine the discourse structure of a text, semantic analyzers to determine the role of constituents in actions or states (patient, agent, location, and the like).

Shallow natural language processing systems rely on statistics to analyze some features of the texts, using different strategies [Pérez-Marín et al., 2009]:

- **keyword analysis**: detects keywords found in the answer which are also found in the reference. Analysis may be single keyword or N-grams, as in Willow.

  - **latent semantic analysis (LSA)** analyzes the relationship between terms and the documents in which they appear, assuming words close in meaning will appear in similar pieces of text. LSA is a complex technique supported by vector and matrix calculus and statistics; the similarity of words and contexts is determined by the cosine of the angles between vectors.

- **analysis of surface linguistic features**, in which the system is trained and calibrated on a list of features to measure.

- **text categorization techniques**, where text is classified into a discrete set of classes, each corresponding to a level or mark.

- **information extraction**, by name entity recognition, pattern matching, or some other natural language processing technique.

- **clustering** groups answers with similar word patterns into clusters, and assigns each of them a mark.

The use of full natural language processing has also been tried, attempting to improve on reliability when assessing the students’ answers, by performing a discourse and semantic analysis. Another approach consists in comparing the student’s answer expressed in a semantic network against a reference semantic network, thus transferring the problem to the construction of the semantic network from text, a problem of equivalent difficulty. Full natural language processing of a text is a very difficult task, and very dependent on the characteristics of the language, which makes it difficult to port to another language.
3.3. Existing systems

The cited work of Pérez-Marín et al. describe 22 systems of free text Computer Assisted Assessment [Pérez-Marín et al., 2009]. They also describe 7 different metrics employed to evaluate the results. The number of proposals, the different approaches and metrics, make it very difficult to compare the effective performance of all such systems. In what follows, some of these systems are briefly referred to, so as to give an idea of their variety.

Most open questions automatic assessment systems rely on some kind of reference material against which students’ answers are compared. Reference material may be classified in different, sometimes overlapping, categories.

- **Training set**, a number of correct answers or answers manually marked by teachers.
  
  - *Project Essay Grade (PEG)*, a pioneer work of the ’60s, obtains several values considering formal aspects of the answer. A correspondence between formal aspects and the quality of the answer is assumed [Whittington and Hunt, 1999], [Pérez-Marín et al., 2009].
  
  - *Educational Testing Service I (ETS I)*. Looks for some formal properties in fragments of 15 to 20 words. In a test instance 200 texts out of 378 were used to build the reference model [10]. ETS II, a later versión, led to E-rater, a system based on Statistics and NLP. New answers are compared to a set of training answers using two different similarity measures [Whittington and Hunt, 1999] [Pérez-Marín et al., 2006].
  
  - *C-rater* reduces the number of training texts focusing on specific information that must be present in a correct answer [Pérez-Marín et al., 2006].
  
  - *Paperless School free-text Marking Engine (PS-ME)* makes use of NLP starting from 30 texts manually marked by teachers; both right and wrong answers are included [Pérez-Marín et al., 2006].
  
  - *Atenea* uses NLP and statistics to compare a small set of answers written by several teachers. Marks short answer questions using the ERB module (Evaluating Responses with BLEU) and NLP [Pérez et al., 2005].
  
  - Pullman and Sukkarieh [Pullman and Sukkarieh, 2005] cautiously limit their work to short answers ranging from a few words to five lines, typical of factual sciences. They started using knowledge extraction by pattern search, but question preparation proved to be hard work, leading them to try machine learning techniques on a set of training texts.

- **Set of features**: features defined by the teachers, expressed in text or some structured form, which must be present in correct answers [Pérez-Marín et al., 2006].

  - *Larkey System*, uses text categorization techniques .
  
  - *Automated Text Marker (ATM)* searches for previously identified concepts and dependencies by means of information extraction .
  
  - *Auto Mark* uses NLP to analyze and compare text against predefined templates filled in by teachers. Schema Extract Analyze and Report (SEAR) and Auto-marking also use as reference material templates generated by teachers .
3. EAssessment, state of the art

- Text or set of texts: a model answer or a text on the subject or a set of texts representative of a language or subject area (corpus). This reference material may exist by itself, it is not necessarily prepared by the teachers.

  - Latent Semantic Analysis (LSA) is based on a representation of the contextual use of words. The reference text or "model answer" may be a textbook or an answer written by a teacher [Whittington and Hunt, 1999]. Intelligent Essay Assessor (IEA) obtains an LSA model from training texts [Pérez-Marín et al., 2006]. LSA was tested with a set of 1929 students’ answers and 142,580 texts in English on general subjects [Pérez et al., 2005].

  - The original ERB (Evaluating Responses with BLEU) proposal used the BLEU (Bilingual Evaluation Understudy) algorithm to mark answers against a set of reference texts [Pérez, 2005]. Comparison against a semantic structure representation of the correct answer.

  - Whittington and Hunt employ automatic translation techniques to draw from each answer a semantic representation which is then compared to another semantic representation obtained from a model answer [Whittington and Hunt, 1999].

  - Intelligent Assessment Technologies allows to mark short answers against a template created by means of a graphics tool [Mitchell et al., 2003].

An adaptive Computer Assisted Assessment system is proposed in Diana Pérez-Marin’s doctoral thesis [Pérez-Marín, 2007]. She uses shallow natural language processing and statistics to automatically build a concept map from the student’s text answer. The system is implemented through the Willow tools, a suite of applications consisting of a free-text adaptive computer assisted assessment system (Willow), an authoring tool (Willed), a configuration tool to select the NLP techniques used (Willoc), and a conceptual model viewer (COMOV). The system seeks to show up the differences between the concepts the teachers have taught and the concepts the students have actually learnt, both to keep track of students’ progress and to identify misunderstandings. The system provides students with instant feedback to their answers, and a knowledge representation on which they can identify and correct their misconceptions.

The teachers introduce questions and references on a subject using the Willed editor. A domain model is built through automatic identification of terms. The student answers the questions using Willow, eventually following a series of complementary questions for remedial work, proposed by the system on unsatisfactory answers. Through comparison of the answer and references, identification of concepts in the students’ answers, estimation of confidence values, and extraction of links among concepts, a conceptual model is built and displayed by the COMOV tool. The teacher can see each student’s conceptual model, and also the conceptual model of the whole class.

Instructors write the questions, and provide typically 3 or 4 answers written in unrestricted natural language. Students’ answers are compared to the correct answers and graded accordingly. Students must use the specific terms in the same way instructors use them in the reference answers. The domain is the course or area of knowledge under assessment. The proposal explicitly states that its goal is formative assessment, not teaching, which makes it unnecessary to store texts related to the domain. The core of the model are the questions and their correct answers. A student’s model is also built recording student’s preferences and personal features to adapt the assessment process, and customize the interface. On incomplete
3.4. Critical appreciation

or deficient answers, some types of predefined clarification or compensation questions may be proposed to the student.

A free-text assessment system based on semantic networks was developed at FernUniversitat in Germany [Lutticke, 2005]. The system uses the Multinet Working Bench (MWR), a graphical tool for the representation and edition of semantic networks in MultiNet form. MultiNet is a knowledge representation and a language for the meaning representation of natural language expressions, based on semantic networks and more than 100 predefined relations among concepts. Students can draw the semantic network directly using the graphical tool, and also write text which is converted into the graph. A comparison against a reference answer is presented both graphically and in statements such as “the entity 'blue' is missing”, “the property of an object is missing”, “the agent in your answer is wrong”. The system was tested in a course on semantic networks, in particular on the MultiNet paradigm; students who worked with the interactive environment were able to solve more exercises correctly and get better marks.

3.4. Critical appreciation

Most automatic marking systems show high scores of correctness when compared to manual marking. This is not surprising if training texts or teachers criteria show a relatively high degree of divergence, which is usually the case. Moreover, some comparisons are based on form, style or other highly subjective factors [Sargeant et al., 2004]. Pulman and Sukkarieh's very careful work on an automatic marking system developed at Oxford University and tested at Cambridge University attains 90% coincidence with manual marking of short textual answers of factual content (1-mark answers) when teachers agree on the answer. However, they honestly report some unexplained discrepancies on some particular questions. On more complex answers (2-mark answers), the system made more mistakes and results differed widely against human markers [Raikes, 2006]. Several weak points have been detected in purely automatic marking systems: the need for training material, the difficulty of marking even trivial questions, the handling of orthographic or expression errors, the difficult problem of interpreting context-dependent synonyms, the lack of certainty in statistical methods, among others [Sargeant et al., 2004, Hermet et al., 2006].

Computer Assisted Assessment (CAA) or a variant called HCC (Human Computer Collaborative) assessment, have been proposed as alternatives to purely automatic systems [Jones, 2005]. CAA deliberately calls for human intervention or "moderation" in the marking process [Mitchell et al., 2003]. Human intervention makes the whole process more reliable and versatile, offers increased warranties and is closer to educators' expectations, at the same time lightening the burden of routine work without losing too much contact with the students. Some CAA proposals even allow to go without training material, enabling the dynamic creation of a knowledge representation or model answer during the marking process, showing the teacher several answers to the same question for comparison and judgment [Jones, 2005], or extracting from selected texts carefully worded text implicit or text explicit questions. Sound knowledge of the type of question proposed enables and simplifies the handling of the answer [Hermet et al., 2006].

A comparison among the 22 systems described in [Pérez-Marín et al., 2009], and most probably among other systems as well, is very difficult due to the facts that they have been applied to different domains, used different corpora, and also different evaluation metrics. Most of these systems are evaluated by their agreement with scores given by human instructors, and
indeed it is generally considered that the best system will be the one which can achieve the best agreement with human marking.

No agreement has been reached among experts on which system is best, but the main weakness recognized in most systems is the lack of a large enough corpus of marked answers to be used as a reference [Pérez-Marín et al., 2009]. Besides, the dispersion of human marked answers has also been pointed out as a factor of distortion which may improve or impair the perceived performance of a system.

A recent work, dated 2010, showed computers could mark short answer questions as accurately as human teachers [Butcher and Jordan, 2010]. For large numbers of students, or when questions could be reused, marks given by machines were found more consistent than marks given by humans, due to the alarming frequency of errors due to misunderstandings among human markers. Responses were marked 0 or 1, with no credit for partly correct answers; in more extensive questions, or when partial credit is allowed, less agreement between machines and humans is to be expected. The study also compared three different systems: Intelligent Assessment Technologies (IAT) FreeText Author, OpenMark and Regular Expressions. IAT is based on computational linguistics; OpenMark and Regular Expressions on the algorithmic manipulation of keywords. All three systems exhibited at least 95% agreement with the question author, with OpenMark performing best. OpenMark also required less hours of training for the question authors. The number of marked answers required for training ranges from 50 to 200 according to different authors; the authors used training sets from 100 to 250, but indicate some questions required a significantly higher number [Butcher and Jordan, 2010].

The Willow tools use teachers’ answers as a reference, but implements an adaptive system that can rescue students on deficient answers by providing some complementary questions. The third and last rescue step is a question of the form “is is true that...”, which may come randomly negated; this last step seems too obvious, and prone to over reward students on defective answers. Moreover, the system seems not to make a difference between a student which went through the complementary questions and a student who answered the question right from the beginning [Pérez-Marín, 2007]. Anyway, the Willow tools come closer to educational needs, providing an adaptive system able to react on students’ answers, propose remedial questions, and bring back on track a number of students.

The Multinet Working Bench (MWR) restricts language and builds a semantic network from text [Lutticke, 2005], a scheme akin to the one proposed in this work. However, the proposal is oriented towards MultiNet, a powerful but relatively complex system of concepts and relations, which demands the student to be familiar with the MultiNet paradigm. This means no extra cost on a course on semantic networks and the MultiNet itself, but it will be a high cost for other subject areas. As with the Willow tools, the MultiNet Working Bench is able to present knowledge in a graphical way, a valuable aspect in learning.

3.5. Conclusions

The proposal for Computer Assisted Assessment presented in this work differs in many aspects from existing sytems. Probably the most outstanding difference is in purpose, since this proposal attempts to effectively integrate assessment into the learning process, while most of the systems formerly described confine themselves to assessment. In our proposal, assessment comes most naturally as a last step in the learning process, through an activity which is the same used by the students to generate the concept maps (semantic networks), carried out
3.5. Conclusions

when they were getting familiar with the subject. Assessment is targeted towards accuracy of content; technical writing quality is constrained, and at the same time guided, by the sublanguage. The use of a generative grammar and restricted vocabulary, and the tools developed to build a semantic network from text, place the present proposal in the category of deep natural language processing, plus knowledge representation for assessment, feedback to students, and support for learning.

The Willow proposal is close to the ideas on which this proposal is based. In Willow no constraint on language is imposed, and concept maps are created for assessment, not deliberately as a summary of the subject on hand. Though a general concept map is inferred from all the answers, there is no “official reference” known by the students beforehand.

In the MultiNet based proposal the students must become familiar with the MultiNet paradigm, know the different sorts of nodes, and their attributes and values corresponding to different semantic dimensions. The MWR tool is presented as a tool for the knowledge engineer; a look at the page describing MultiNet shows the system is not within reach of a typical student [Helbig, 2012].

Some shortcomings of the formerly analyzed systems are expected to be improved through the ideas put forward in this work. The reference model is naturally built by the students under the guide of the teacher, which does away with the following drawbacks:

- no training set is required, there is an undisputed reference, built along the course.
- no dispersion among human marked answers; the model built by the students and the teachers has been agreed upon through constructive discussion.
- no ambiguities or mistakes in processing text answers: students write in a predictable, machine processable sublanguage.

The use of a sublanguage ensures that:

- terms are uniformly understood in their meaning and use.
- a reference model (i.e. a semantic network) can be built, and answers compared against it, by extracting from the answer a model of the same type, in the same way.

On the whole, the proposal of this work is made as a learning activity, supported by a learning tool, which may at the same time be used in assessment.
Part III.

Language
This part deals with the language related problems involved in the automatic marking of free text answers. Underlying discipline is Natural Language Processing, from which knowledge, resources, and tools are engaged towards a solution.

Chapter 4 deals with state of the art in restricted languages, considering both controlled languages and sublanguages. Restricted languages are examined from an educational perspective, focusing in what they have to offer, not only to simplify the computational treatment of texts, but also as a subset of natural language supportive of simple, correct writing.

Chapter 5 analyzes the construction of a sublanguage appropriate for the writing of free text answers in an educational environment. Some design considerations are discussed, together with issues to consider when writing in a sublanguage, and the transformation of the text into a graph of some type. Tagsets, lexicon and rules are key components to consider.

Chapter 6 identifies, characterizes and evaluates freely available lexical resources towards their potential use in the construction of a sublanguage appropriate for educational use.

Chapter 7 describes the compilation of a lexicon of commonly used words based on the lexical resources formerly identified and evaluated; domain specific vocabulary can then be added to this basic lexicon.

Chapter 8 describes the construction of a sublanguage based on a general purpose lexicon and a set of production rules for assertive sentences. Rules for both Spanish and English are provided. These experimental sublanguages are designed to be processable by syntactic analysis, from which a knowledge structure may be inferred.
4. Sublanguages, state of the art

Abstract. Written text has a long tradition in Education. The outstanding progress in the Information and Communications Technologies (ICTs) has not changed this. Most information takes the form of text, and it will continue to be so. Text must be clear, concise, and easy to understand. In many areas of knowledge and culture, documents must also be consistent, uniform in style and aesthetically pleasant. Nowadays text should be apt for information retrieval, information extraction and knowledge representation. One way to deal effectively with the complexity, ambiguity, redundancy and casualness of natural languages is to prevent these problems at the moment of writing. Sublanguages are subsets of natural language used within a community or domain for specific needs. Controlled natural languages (CNLs) are subsets of natural languages based on a limited core vocabulary and simple grammar; specialized vocabulary may be added as needed. While CNLs are strictly regulated, sublanguages are more loose in their definition. A number of these restricted languages have been tried for different purposes: the production of machine processable texts, the writing of easily translatable technical manuals, the teaching of secondary languages, the broadcasting of news. The different types of restricted languages may be classified in machine oriented, human oriented for technical use, and human oriented for general use. A well designed restricted language for human communication allows freedom of expression within constrained rules, and results in texts easier to read and easier to understand, even for non native speakers. The use in Education of a well designed restricted language, supported by an intelligent editor, may help towards better writing as well as allow for information extraction, automatic assessment, teaching material comparison and synthesis. In the technical and scientific disciplines, the use of such a sublanguage is expected to have the same proven benefits technical sublanguages have shown in industry. Though never intended as a substitute for natural language, a well designed sublanguage can greatly help achieve a mastery in writing previous to or instead of the free full use of the language.

Processing of natural language is a very difficult problem. Different communities, specially those related to a specific subject area, tend to use a subset of all the sentences possible in natural languages. These sublanguages turn up almost spontaneously, without a definite intention. Controlled natural languages are deliberately designed for a purpose, regulating the lexicon and syntax of their sentences, sometimes adhering to a strict standard. Use of a sublanguage in free-text answers may allow for the deterministic extraction of knowledge from text, the building of a data structure to support this knowledge, and the comparison of knowledge data structures to mark students' answers. This chapter starts with a review of the challenges posed by the use of language and the written text in Education, examines the potential benefits of using restricted languages, lists the different types of sublanguages and controlled languages, and describes some existing proposals. The requirements for a
4. Sublanguages, state of the art

sublanguage to be usable in Education are put together and discussed.

4.1. Terms

This section provides an explanation of the meaning of some terms, as are used in this work.

The term language may be applied to the formal expressions of mathematics, logics or computer systems, the natural languages we speak and write in, and to any system of signals used for communication. A sublanguage refers to any proper subset of expressions in one of these languages which exhibits some systematic behavior, i.e. it is used like a language. Sublanguages tend to appear spontaneously in communities sharing a common interest and interchanging information on some specific subjects [Kittredge, 2003]. A controlled natural language is an engineered subset of natural languages where lexicon and grammar have been restricted to reduce the ambiguity and complexity of full natural languages [Schwitter, 2010]. The short form term “controlled language” has been superseded by “controlled natural language”, to make explicit that they are subsets of natural languages, and not related to languages associated with programming or logics, which are indeed controlled languages, not always “natural”.

Sublanguages have no strict rules as controlled natural languages have; sublanguages usually do not put any limit on the length or complexity of their expressions, whereas controlled natural languages do. Compilation of a controlled natural language takes considerable time and effort; domain experts, linguists and programmers must work together to produce a formal specification of the language; the users must be trained to develop the necessary skills [Kittredge, 2003], generally to write, since reading controlled natural languages usually requires no particular skills or training.

For the purpose of this thesis, a sublanguage for knowledge representation lies somewhere in the middle: though it is more formal than a sublanguage as formerly defined, since its vocabulary and grammar rules are determined, the users are allowed to modify them; there is not a formal specification to follow, and the scope of application may be very restricted, such as a bunch of learning units in a certain subject. Teachers adopting a sublanguage and students writing in it are expected to agree on modifications to the lexicon, and possibly also on some of the production rules in a simple generative grammar. The sublanguages proposed here, though not spontaneous in their origin, are expected to keep some of the characteristics of a language used within a community, and hence remain under their control. In a more ambitious project, an escalation of these ideas may lead to a sublanguage mature enough to be described in a formal specification; in that case, it would be better to term it a controlled natural language. For the time being, the term sublanguage is considered a better description of the restricted subsets of natural languages this project will be concerned with.

In this work, the term restricted language, short for restricted natural language, will be used to designate a sublanguage or a controlled natural language. A restricted language will always be a subset of a natural language, and will exhibit a behavior similar to a natural language.

Information retrieval consists of finding objects relevant to a user’s query, almost always expressed in words. The limits of statistically based methods seem to have been reached, and language oriented methods are presently being tried [Tzoukermann et al., 2003]. Information extraction allows for the identification of entities, relations and events in free text, by the automatic processing of extraction rules reflecting a user’s interest [Grisham, 2003].

An ontology is an explicit and formal specification of concepts and relations in a domain of
4.2. The challenge of the language

The written text in education has a long tradition. Advent of new facilities provided by the Information and Communications Technologies (ICTs) has proven complementary rather than substitutive of writing. Assessment in particular relies strongly on written texts, in the form of answers to questions or short essays.

Information, including educative material, is abundant and accessible as never before. Since primary school, teachers and pupils rely on the Internet to dig for material. Present day trends promise still more growth and widespread use. In Uruguay the Ceibal Plan has committed to the goal ‘one computer per child’ on a national scale [CEP, 2007].

Most information, educative or other, remains textual in nature. Sound or video tracks, animations, still photographs or drawings are an invaluable help in the understanding and comprehension of a number of subjects, but they rarely go without supporting or explanatory text.

The explosion of information made possible by ICTs poses challenges on our ability to profit from it: retrieval of most relevant and only relevant pieces of information, selection of qualified sources, comparison of conceptual content, evaluation of quality, seem too big tasks to undertake manually and too critical to be carried out by purely automatic means. These problems have led to the application of Natural Language Processing (NLP) techniques and the convergence of technologies known today as the Semantic Web.

Dealing with natural language in the general case is an Artificial Intelligence complete problem, or AI-complete [Sargeant et al., 2004]. A problem is said to be AI-complete when it is equivalent to solving the general problem of building a computer as intelligent as a human being [Shapiro, 1992]. In practical terms, this means the problem cannot be treated with present day means.

The exploitation of corpora, large collections of varied or domain specific texts, exhibit the limitations inherent to statistical applications. Establishing limits in language when it is written, though known since decades, is being revisited as a means of simplifying the processing of language, in particular for the generation of ontologies and as the underlying material in semantic web applications [Pool, 2006].

In this chapter, requirements, problems and proposals of sublanguages and controlled natural languages usable in Education for the representation of knowledge are examined. Sublanguages, though restricted, may be designed to be a subset of correct natural language. This opens a way for their use in Education, with a view of preserving and promoting correct writing, the automatic processing of text for assessment, teaching material preparation, and content comparison or synthesis. A restricted version of natural language supported by an intelligent editor may be a tool towards better writing. Several controlled natural languages exist for different purposes. Controlled natural languages are recognized as ‘machine oriented’ or ‘human oriented’. Human oriented controlled natural languages, or a less restricted sublanguage not necessarily regulated by a standard, are potentially useful for educational purposes, in particular to guide correct writing and to represent knowledge.
4.3. The written text in Education

In Education, the written text has not been substituted. The Internet has shown an outstanding growth in written material, as seen in the constantly increasing number of pages in the World Wide Web [Wikipedia, 2012b, WWW], most of which are text. Facilities like e-mail, wikis, blogs, are textual in nature. The boom of short messages in cell phones is not only textual in nature but promotes a differentiation of language which may well be considered a dialect. The wide use of these dialects by young people becomes an educational problem in itself: abbreviations and ‘codes’ used in cell phone messages cannot be admitted even in moderately informal writing. The language used in some popular blogs is also at a distance from what is considered correct use of the language. These dialects may be compared with teenagers’ ways of speaking, though amplified by the much wider reach of modern media.

The teaching of language, the teaching of written language in particular, offers new challenges. The need for clarity, precision, conciseness and consistency is a must in scientific, technical and legal writing, among others. ICTs offer not only huge amounts of information but the possibilities of generating more by anyone connected. The media, or the way people interact with media, exhibit some strong tendencies to distort the use of language, compromising its communication capabilities and confining its reach to groups of age or interest. Nowadays it is not uncommon to see, even at the University, written text pieces far from barely acceptable quality. Curiously enough, these drawbacks turn up in an age where international frontiers have almost ceased to be a barrier, when people can communicate with other people easily and cheaply, commonality of interest having surpassed all geographical and political frontiers.

Correct use of language, specially written language, has always been a major worry in Education, since early childhood well into adolescence. Present day challenges must be faced with present day solutions and support, both for teachers and students, but also for anyone in need of written communication.

Information retrieval and information extraction are becoming increasingly important in Education: information retrieval produces documents relevant to a query; information extraction allows comparing conceptual content of documents or generating a synthesis of the concepts and relations present in several documents. Quality of material found in the Web can, by the use of these facilities, be more easily evaluated and adapted as teaching material. Automatic assessment of free text answers may become possible by information extraction and comparison.

In all aspects outlined above, the permanence of the written text, the huge volumes in existence, the pressure to get fast responses, make the automatic processing of language a must. All tasks involving free text are inherently difficult due to the complexities of natural languages, their unlimited possibilities and the loose ways in which they can be used. Though not without costs and limits, restricted languages may help develop more effective solutions, in some cases providing a way to make a solution possible.

4.4. Why use restricted languages

The first controlled natural language was released in 1930 [Kittredge, 2003]. Charles Kay Ogden’s Basic English consisted of 850 words and a limited set of grammar patterns. Its purpose was mainly educational: the teaching of English as a foreign language. Basic English is a subset
of Standard English, no ‘unlearning’ is required, subsequent learning adds to this self-contained first stage of the language [BEI, 2012]. Today’s most outstanding use of controlled natural languages is the writing of technical documentation, easy to understand, easy to translate, no ambiguities allowed [Kittredge, 2003]. A recent development is the use of controlled natural languages to generate ontologies for the Semantic Web [Creagan et al., 2007] [Pool, 2006] [Funk et al., 2007] [Schmitter, 2010]. In the Web, ontologies provide a shared understanding of a domain, overcoming differences in terminology [Antoniou and Harmelen, 2008]. Using an appropriate controlled natural language allows for the generation of an ontology from a piece of written text.

Basic English was built on the belief that a speaker or writer of English could do well with only a thousand words instead of the 75000 available to a skilled speaker [Arnold, 2002]. Industry took on controlled natural languages to save on translation: operators and maintainers of industrial equipment could read and understand simple English; there was no need for a different translation for each language spoken in the country where machinery was exported. Two additional, unexpected benefits were found [Arnold, 2002]:

- texts written in a controlled natural language were clearer, easier to understand and much less prone to ambiguity than standard language texts.
- texts written in a controlled natural language performed significantly better in machine translation.

A more recent experiment on the readability of controlled natural language was carried out in 2008 at Dublin City University. The study showed controlled natural language texts to be more readable, more favorably regarded, and more helpful to remember keywords. Participants were submitted to both types of texts in different instances, and asked to complete a questionnaire each time. Though the samples were too small to generalize, a majority of participants judged the application of controlled rules to text improves readability [Cadwell, 2008].

Machine translation is only one of the language processing areas where determinism of controlled natural languages can be exploited. Programming languages are examples of formal, not always easily readable, controlled languages (not controlled natural languages). Ontology generation languages are also formal, but tolerably readable. Precision, clarity and readability of controlled natural languages have been proven by several proposals currently used in industry, finance, government, legal, and other areas.

Some features which make restricted languages interesting for Education are:

- **clarity, precision and readability**; these virtues allow for better and easier understanding, even of complex subjects.
- **consistency** in all documents, i.e. refer to the same things with the same words everywhere.
- **reuse of text units**; a document may be seen as a collection of information units, and the same information units may be used in different documents with different educational purposes.
- **suitability for automatic processing**, which enables tasks such as automatic assessment, information extraction, knowledge representation, teaching material comparison and synthesis.
4. Sublanguages, state of the art

Restricted languages allow for the construction of support tools for Computer Assisted Writing (CAW) in a much more effective and helpful way than unrestricted natural languages do. An intelligent editor application based on a restricted language can provide better aid to writing than editors for natural languages. Predictive editors are specially attractive, since they can limit the selection of possible words according to what has been previously written. A recent paper by Tobias Kuhn proposes a grammar notation for controlled natural languages to be used in predictive editors, able to support a large subset of ACE (Attempto Controlled English), one of the most advanced controlled natural languages to date [Kuhn, 2012a].

Students can be taught to write correctly using a restricted language prior to the full and free use of the language. Texts written in a restricted language are better suited to automatic assessment by knowledge extraction into a knowledge structure comparable to a reference structure of the same type. Teaching material written in a restricted language may be easily checked for consistency, redundancy, uniformity of style and presentation. Search among documents written in a restricted language will show better precision (exactness) and recall (completeness), owing to the restricted vocabulary. A restricted language may allow a number of syntactic and lexical variants, all traceable to canonical forms; a restricted language need not be "a poor language".

Restricted languages should not be seen as absolute substitutes of natural languages. Restricted languages have proved their benefits in specific domains (technical, scientific, educational, legal, broadcasting, newspapers), in certain levels of proficiency or learning (non native speakers, students learning to write), and for plain communication among people of different native languages. Complexities of natural languages reflect inner aspects of human nature which cannot and should not be subject to rules beyond those that naturally arise in the evolution of the languages and the communities which keep them alive.

4.5. Some existing proposals

According to their formality, restricted languages can be "tight" or "loose" in their conception and rules. On the "tight" side, computer programming languages and languages directly mapping to logical structures are very strict in their rules. They are "machine oriented", and, consequently, machine processable. They are apt for implementing an algorithm, execute a job, create a knowledge structure, and answer questions by logical inference on the data. On the "loose" side, restricted languages may consist of a list of recommended words, a list of words to avoid, syntactic and lexical rules to follow when writing, a model layout and presentation style. These restricted languages are "human oriented", and may be just a "style guide" which writers are encouraged or compelled to follow. Machine oriented restricted languages tend to be scarcely readable; human oriented restricted languages main purpose is readability and clarity. As usual, between the extremes many intermediate proposals exist. A timeline view of the development of controlled natural languages for English can be seen in [Kuhn, 2012b]; the poster shows the relative emphasis each proposal makes on precision, expressiveness, similarity to natural languages, and also gives a hint on their difficulty. The number of proposals gives an idea of the activity in the field: there are about 80 proposals, of which 26 are reported as continuing to date.
4.5. Some existing proposals

4.5.1. Machine oriented restricted languages

Attempts at a language both readable by humans and processable by machines can be traced back to the COBOL programming language, developed in 1959: 'Multiply Unit-Price by Quantity giving Line-total'. Though criticized by its verbosity, the extra typing was less than double when compared to a language like C. Its main limitation came from its fixed computer oriented ontology, which forces the language to emulate computer operations with no hope of application in other domains [Sowa, 2004]. Pure logic, not associated with any ontology, can be applied to any domain. A language that could be compiled to logic would be equally flexible in its application [Sowa, 2004].

A ttempto Controlled English (ACE) is a project of the University of Zurich [Attempto Project, 2011] to develop a controlled natural language usable as:

- a query language, to select and extract information from a database or an information system.
- a specification language, to formally describe information systems requirements, analysis and design.
- a knowledge representation language, to store knowledge in a structured, machine processable way apt to extract information, make logical inferences or emulate human intelligence.

Attempto recognizes a small, fixed set of function words and a large, extensible collection of content words. Function words are: a, an, the, every, some, and, or, not, if then, is, has, and some prepositions. Content words are most nouns, verbs, adjectives and adverbs. Content words are defined implicitly by statements [Sowa, 2004]. ACE offers synonymy both in syntax and lexicon: text can be written in different syntactically equivalent forms, and using lexical synonyms. These texts map unambiguously to Core ACE, a subset of ACE equivalent to full ACE without its variants. ACE texts map to Discourse Representation Structures (DRS) [Fuchs et al., 2006]. A DRS provides a level of representation where all textual references (pro-nouns, noun phrases) are resolved, were they within a single sentence or among sentences in the whole discourse [Hess, 1991]. A DRS shows relations among the entities, states and events of the application domain [Schwitter and Ljungberg, 2002]. A DRS can be unambiguously expressed in first-order logic as a formal deductive system. This makes DRSs independent of any ontology. ACE has been reported successful in various knowledge representation situations. In automatic reasoning ACE seems promising, but not enough convincing results have yet been produced. ACE has been successfully translated into and from several logic based languages, which accounts for interoperability among applications, particularly in the Web. ACE can be learned in two or three days with little previous knowledge, which shows its capabilities as a user interface. ACE is a promising alternative to bridge the gap between natural language and knowledge representation [Fuchs et al., 2006].

Processable English (PEng) is similar to ACE. Its purpose is the writing of formal specifications [PEng, 2007] [Schwitter and Ljungberg, 2002]. As ACE, PEng can create a DRS and can be translated into first-order logic. As ACE, PEng has a set of function words and a set of user-defined words. The latter can be extended while writing. To write in PEng the author is not required to know the lexicon and grammar restrictions in advance: an authoring tool is available. ECOLE is an intelligent look-ahead (predictive) editor capable of displaying possible syntactic choices after each typed word. The author may type a new word or choose it from
4. Sublanguages, state of the art

the words contained in one of the allowed syntactic categories. Only elementary knowledge of grammar is required to use the syntactic categories. The author can click on the categories displayed to ask for additional information [Schwitter et al., 2003] [Schwitter, 2007]. Other features of the ECOLE editor are:

- an entry window and a separate systems window to display complementary information concerning the entered text.
- paraphrase: the editor shows how the system interpreted the entered text, by showing the same text with annotations.
- a DRS and first-order logic inferred from the entered text can be displayed in the systems window.
- a spell checker and lexical analyzer allow detecting spelling errors or adding new words to the user defined set. Synonyms can be defined for existing or added words.
- a theorem prover can check the model for consistency and informativeness: no contradictory or redundant information is allowed.

PEng authors’ identify software engineering and the semantic Web as immediate domains of application. Other domains include business process modeling, database modeling, and legal reasoning. They also plan to teach logic and language technology with PEng [Schwitter et al., 2003]. A recent version called PEng Light can be used for knowledge representation; the input can be translated into first order logic or a variant of Description Logics [White and Schwitter, 2009].

Computer Processable Language (CPL), is a controlled natural language developed at Boeing Research. CPL uses heuristic rules for a variety of language processing activities. CPL accepts three types of sentences: ground, questions and rules. It has been used to encode common sense knowledge and to allow the formulation of queries in comprehensible way [Kuhn, 2010].

4.5.2. Human oriented restricted languages for technical use

In 1972 Caterpillar started using Caterpillar Fundamental English (CFE) in his technical manuals worldwide. CFE had a very restricted vocabulary and grammar. Its aim was to eliminate translation; CFE could be learned as a second language with little effort. CFE was discontinued in 1982 mainly due to the difficulties of producing really CFE compliant documents. The advent of word processors and language parsing technologies in the mid 1980s led to Caterpillar Technical English (CTE). CTE contains about 70,000 terms of precise meaning both for humans and machines, plus grammar rules and style patterns [Kamprath et al., 1998]. Benefits included:

- automatic translation with minimal post-editing.
- standard terminology and writing style for all documents.
- reuse of "information elements"; a document came to be considered a collection of information elements.
- consistency and standard appearance in all information and manuals, whatever the experience or expertise of authors.
CTE was made possible through computer systems support to authors. The CTE system provides:

- a centralized software environment for authors.
- online terminology definitions and usage information.
- interactive disambiguation, for accuracy and consistency.
- analysis of input sentences and sentences in documents.
- controlled sentence structures wide enough to support the grammar complexities required for writing in the domain.

CTE is reported to have a significant positive impact both in authoring quality and translation productivity [Kamprath et al., 1998] [Disborg, 2007].

ASD Simplified Technical English is a controlled natural language for the aerospace industry. The AECMA Simplified English Guide was released in 1986 as a result of the joint efforts of the European Association of Aerospace Industries (AECMA) and the Aerospace Industries Association (AIA) of America. In 2004, when the Aerospace and Defense Industries Association of Europe (ASD) was created, the standard was renamed ASD Simplified Technical English (STE). STE is defined by specification ASD-STE100. STE comprises a set of writing rules (grammar and style) and a dictionary of controlled vocabulary of about 1000 general words. The dictionary is conceived under the principle "one word, one meaning" as far as possible. ASD-STE100 is officially defined as an "International specification for the preparation of maintenance documentation" [ASD-STE100G, 2008]. The official name does not mention aerospace industries, and in fact the specification has been adopted by several other industries for their own documentation [Ledopres, 2007].

UN-Lencep is a controlled natural language for the automatic generation of software development artifacts. It has recently been applied to the automatic generation of Entity-Relationship (E-R) diagrams, defining an equivalence between several formal constructions and corresponding controlled natural language expressions [Zapata Jaramillo et al., 2011]. ucsCNL is a controlled natural language for use in requirements or use cases specification, to achieve a degree of standardization enough to allow the automatic generation of tests for automatic software testing [Barros et al., 2011]. These proposals are similar to those for knowledge representation analyzed in chapter 12.

A recent success story on the use of controlled natural languages in technical documentation concerns Japanese technology company Konica Minolta. In 2008 they found their English documentation was of low quality, lacked standardization, and led to misunderstandings originated in poor writing and inconsistencies. Konica Minolta bet on Acrolinx, an application software that uses natural language processing to guide authors in the creation of accurate, consistent content, easy to find by search engines, and apt for machine translations. The project produced both writing rules and a terminology. Tests showed these undertakings ensured high quality in English documentation, improved communication and reduced translation costs [Pabst and Siegel, 2009].

4.5.3. Human oriented restricted languages for general use

An attempt at standardization of language for general use is Plain English, a US government’s initiative started in 1995 by a group of federal employees. There are several US government
mandates for the use of plain language in communications to the public. Plain language aims at a simple writing that keeps in mind audience, clarity and comprehension. Its main benefits are faster writing, faster reading and better understanding. Plain language usage is not enforcing but an act of will on the part of the writer. To the purpose, guidelines, suggestions and word substitution lists are given for several fields. Guidelines go beyond the written text to include presentation aesthetics, style and tone [PLAIN and Network, 2012].

Voice of America has been broadcasting in Special English since 1959. Originally intended to reach non native language speakers in clear and simple English, Special English has helped people learn American English. Notwithstanding its simplicity, Special English proved to be enough to keep people current in world news, culture, and developments in science. Special English has a core vocabulary of 1500 words, uses simple sentences that express only one idea, avoids the passive voice, does not use idioms and is read in broadcast at a slower pace than standard English. Besides helping people learn English, Special English is reported to have helped fluent English speakers understand complex subjects [VOA, 2012].

Simple English Wikipedia is an easy to read encyclopedia written in Simple English, a version of English limited to the 1000-2000 most common English words, short sentences and simple grammar. Its target public is people who are learning English, children, students, adults with learning difficulties or anyone wishing to understand complex or unfamiliar topics [Wikipedia, 2012a]. Simple English writing is not enforced by mandatory rules. Instead, a guideline suggests to start writing in standard English and then proceed through several simplifying steps: change words to those found in basic word lists, substitute slang, idioms and jargon by explicit expressions, revert to active voice, use verbs in simple tenses [Wikipedia, 2012a]. A Simple English spell-checker for OpenOffice is available at the Basic English Institute [BEI, 2008]. By not sticking strictly to word lists such as Basic English 850, Simple English is not subject to awkwardness in expression.

### 4.6. A restricted language for Education

Except for second language teaching, not much use of restricted languages in Education has been made. There are several reasons which may account for this:

- restricted languages are difficult to design, build and test.
- according to their formalism, restricted languages may be difficult to learn.
- many current proposals are not readable enough (e.g. computer and specification languages), have been limited to a domain (e.g. technical writing), or are too restricted (e.g. Basic English 850).
- fear of limiting or distorting the learner’s creativity or ability of expression, belief that all restricted languages are "poor" languages.
- trying to apply a restricted language in a situation unsuitable for the restricted language design, or even for any restricted language.
- lack of really helpful tools for restricted language usage. Effective aiding tools may be a condition for restricted language acceptance.
4.7. Research lines

- lack of suitable restricted languages for specific purposes, lack of tools or techniques to adapt or build restricted languages in a cost effective way.

Modern proposals have shown most of the former difficulties and risks can be successfully coped with. Restricted languages remain difficult to design, build, and test. Fear of over constraining expression or thought is more of a feeling than a reason. Subsets of language have always been used in Education, especially in early school and second language teaching. Teachers trying to be clear and easy to understand look for common words, use simple sentences, avoid ambiguity. Teachers do, in an informal way, the same things restricted languages do in a deliberate way; they can be said to use a sublanguage in the small community of their classroom.

Formal controlled natural languages like ACE and PEng have achieved surprising readability considering a logic model is being built almost as words are being typed. But they are specification languages; they have been conceived with a purpose in mind which is not primarily human communication. It would be wonderful if an ontology could be deterministically built from a readable, clear, concise, aesthetically pleasant piece of writing, but this may still lay several steps ahead in technological evolution.

Restricted languages for technical documentation are definitely readable, easy to understand, and as concise as precision allows. They aim at human communication. They are domain specific: extension to other domains may not preserve their qualities. Writing in a restricted language for technical communication requires some training, but remains relatively easy to master. Training, tools and services are offered to companies and organizations. Tools include text mining to build dictionaries and thesaurus, syntax and spell checkers, editors, translators and quality measurement [Smart Communications, 2008] [Tedopres, 2007].

The examined restricted languages for general use, mostly based on style guides and simple recommendations, seem an acceptable basis for the building of a sublanguage aiming at Education. Many teachers would gladly accept rules such as "use active voice", "write short sentences", "use word meanings consistently", "only one idea in each sentence": they may even have taught these rules or their equivalents for decades. Present use and needs of written texts require more powerful action.

4.7. Research lines

Education is not foreign to the problems restricted languages address, and solutions found in other areas may be adapted for Education. The examination of existing restricted languages, taking into account their intended uses and limitations, help towards the identifications of requirements for a sublanguage useful in Education. There are several reasons to undertake research in sublanguages for Education:

- most information and communication takes the form of text, and it will continue to be so.

- text must be clear, concise, easy to understand. In many areas of knowledge and culture, documents must also be consistent, uniform in style and aesthetically pleasant.

- text should be apt for information retrieval, information extraction and knowledge representation.

To achieve the former qualities, some rules and formal restrictions must be imposed on text at the moment of writing. The use of natural language is normally imprecise, ambiguous,
4. Sublanguages, state of the art

redundant, and relies on the audience's "knowledge of the world", which may differ abruptly among groups or cultures.

Restricted languages exist and have proven successful both in narrow and wide domains; something in between should be possible and positive for Education.

A restricted language must be supported by an intelligent editor and other tools; aid in writing is a valid way of teaching to write.

No single restricted language is possible for all needs; a core restricted language supporting extension mechanisms may be a practicable alternative. Different degrees of formalism may also be called for.

More specifically, some desirable features of a restricted language for Education, its editor and other supporting tools are:

- a vocabulary inferred from existing texts; lexical ambiguity in a single domain is relatively low.

- ability to exploit lexical semantic resources, such as WordNets [Princeton University, 2011].

- error rescue: offer alternatives when a non valid word or construction is detected.

- predictive writing: offering words to choose from the correct syntactic categories, according to the grammar structures included in the restricted language and previously typed text.

- use of synonyms, both in lexicon and syntax (equivalent syntactic constructions).

- explanation of "what is going on": annotated text, syntactic trees, conceptual graphs, anything which shows the "consequences" of what is being written.

4.8. Conclusions

Having stated the differences among several types of restricted languages, in the scope of this project the term sublanguage is considered a better description of the restricted subsets of natural language which will best suit our purposes; controlled natural languages are usually accepted as languages regulated by a strict set of rules carefully put together, even to the point of reaching the quality of a standard. The term sublanguage is more apt for a restricted language used within a community, loosely or strictly specified, but agreed upon within the community of users, suitable for their purposes.

The former revision of restricted language proposals is far from exhaustive but enough to show that regulating the use of natural languages is seen as a necessity from several different viewpoints. Natural language remains the preferred if not the unique means of expression in several fields, which may be classified grossly in two groups: communications and information. Education is concerned with both. Quality writing, clear, precise, consistent, easy to read, easy to understand, aesthetically pleasant, can be achieved using carefully designed restricted languages. This has an impact both in teaching to write and in the generation of educational material. Predictability of written texts as conveyed by restricted languages allows better results in information retrieval and information extraction. This has an impact in the collection of material by both students and teachers. Knowledge representation may be used to compare
4.8. Conclusions

content among a set of documents, select from them, and generate new documents with specific contents, tailored to specific purposes.

A restricted language for use in Education may be envisioned as halfway between technical restricted languages and general purpose restricted languages. Such a sublanguage cannot be expected to exhibit the processable qualities of logic based languages; the complexities involved make them too difficult for application in an educational environment.

Success of a restricted language depends not only on its qualities but also on acceptance. An intelligent editor or some Computer Assisted Writing environment must be provided. Writing in a restricted language should be made as easy as possible, as it is easier to read and easier to understand when compared to natural language.

As the mass of text increases, as people feel more overwhelmed by data, better tools are needed for communication and information. Designing, building, testing and deploying an educational restricted language is a major effort. The need for a friendly, easy to use intelligent editor or Computer Assisted Writing environment means additional effort. Both endeavors should be carried on as part of the same research line. A restricted language and an intelligent editor are practicable tools to help people communicate and manage information. In an era of large population, blurring frontiers and overwhelming information, enhancing communication abilities and management of information may well be conceived a priority of Education.
5. Construction of a Sublanguage

Abstract. A sublanguage for knowledge representation consists of a lexicon apt for use in the target domain and a generative grammar whose constructs can be transformed into some kind of knowledge representation in a predictable way. A successful sublanguage must be readable, simple to write, unambiguous, and with a clear semantics. An author in a sublanguage must know its rules, have access to the lexicon, and adopt the sublanguage's style. Recommendations of style are similar to those given for simple writing, such as keep sentences short, use the active voice, keep to one topic per paragraph. Text in a sublanguage must be as readable as text in the natural language; a sublanguage is a subset of a natural language, not a different language. Documentation, tools and training are critical to help authors familiarize with the sublanguage as effortlessly as possible. Transition from text to graph requires a unique syntactic tree per sentence, identifying concepts, and to establish relations according to the meaning attributed to the sublanguage constructions. Operations of language processing towards a knowledge representation are sentence segmentation, tokenization, lemmatization, part of speech tagging, syntactic parsing, and word sense disambiguation. Authors must be trained in accepted forms of expression to avoid multiple ways of saying the same thing. Different requirements call for different tagsets to annotate text; this project is based on a simple tagset for English and another for Spanish. Wordforms similar to part of speech tags are used as substitute wordforms in a sentence to license the syntactic trees; in this way the production rules are decoupled from the lexicon, which may evolve on its own not affecting grammar and parsers. Compilation of a lexicon can be done from an existing corpus, or from a general service word list to which domain specific lexicon can be added. A carefully compiled lexicon can greatly simplify the tasks of lemmatization, part of speech tagging and word sense disambiguation. Production rules must consider both the processing of text and the knowledge representation scheme. Some language phenomena to consider include type of sentences, handling of negation, active voice only, syntactic unambiguity, avoidance or resolution of anaphora, simple sentences only, restricted use of conjunctions. Preprocessing the text can handle other phenomena such as multiple word units, proper nouns, numbers, dates, measurements and quantities. Once a basic sublanguage has been compiled and tested, many enhancements are possible to make it both more expressive and usable. An intelligent editor can offer word completion, partial parses, part of speech anticipation, lexical help and additions, word sense disambiguation. The tools offered by Natural Language Processing, several identified Knowledge Representation schemes, restriction to a specific domain, and the possibility of a gradual development, make the challenge of compiling a sublanguage a feasible task.

A sublanguage is needed for the deterministic construction of a knowledge data structure,
5. Construction of a Sublanguage

such as a semantic network, which can be used to compare students’ answers to an accepted reference of the same type. To be useful, this sublanguage must be simple and natural, posing only mild restrictions on natural language, but at the same time deterministic in their expression and meaning. This chapter discusses the main issues in the construction of a sublanguage for knowledge representation: requirements, problems, tools and possible solutions. This chapter is both an introduction and a summary of the following chapters, which deal with the compilation of a lexicon from word lists, and the construction of a syntax based sublanguage for knowledge representation.

5.1. Introduction

The construction of a sublanguage is no trivial task. In the design of an usable Domain Specific Sublanguage (DSS) expressivity must be balanced against simplicity. This balance must be kept both in the lexicon and in the production rules. This chapter discusses some issues in the construction of domain specific sublanguages with the purpose of writing texts which can be converted into a knowledge representation instance of some type.

The proposed sublanguages consist of a lexicon apt for use in the target domain, and a generative grammar whose constructs can be transformed into a knowledge representation instance in a predictable way.

The lexicon for a sublanguage can be very specific, limited to a small domain. Such a lexicon can be compiled from a collection of texts, from existing vocabularies, or from both. More challenging is the compilation of a “general purpose” lexicon to be used as a basic lexicon in several different domains, to which domain specific terms will eventually be added. Several word lists exist which can be used for this purpose. Testing or validation of the lexicon can be performed by determining coverage of the lexicon against a corpus of domain specific texts, or against an existing corpus of more general use. A manual revision and correction will be necessary in almost all cases, but this can be done gradually.

The compilation of the set of production rules will depend on the characteristics of the knowledge representation desired. As will be discussed, semantic networks are probably the most intuitive and most usable kind of knowledge representation. But a semantic network may be very differently conceived, and this will determine the characteristics of the required sublanguage. As an example, in some domains a class hierarchy will be most useful, while in others a sequential organization will be more natural, or there may be a central idea calling for a radial topology, as in mind maps. A knowledge representation scheme can also be based on the semantics defined by the syntactic structure of the sentence. This may require the users to be too “grammar conscious”, but a more pragmatic simplification is possible. A syntax oriented sublanguage and knowledge representation is discussed in chapter 8.

5.2. Terms and tasks in NLP

This section explains some terms used in Natural Language Programming (NLP), with an identification of some usual tasks in NLP of interest for sublanguage and knowledge representation, such as tokenization, part of speech tagging, and syntactic parsing. No attempt at formal definitions is made; the purpose of these explanations is only to state what is meant by each term. The main reference is Jurafsky and Martin, 2008, with occasional references to Wikipedia, 2012b and Farlex Inc., 2010, but the wording is mostly ours.
5.3. Design considerations

*Natural Language Processing (NLP)* tries to understand human language in such a way that a machine can extract meaningful information from texts or speech. It comprises several subfields: *morphology* studies the decomposition of words in meaningful pieces, *syntax* studies how one word relates to others in the structure of a sentence, *semantics* tries to determine the knowledge contained in a piece of text or speech, *pragmatics* deals with the somewhat elusive subject of determining the goals and intentions of the speaker, *discourse* studies how sentences relate to one another in a long piece of text or speech.

A *morpheme* is the minimum language unit with a meaning. A word may be formed by several morphemes. A *stem* is the main morpheme of a word, also called the *base* or *root* form of the word; an *affix* is a morpheme which may be added to a stem to complement the stem meaning. *Stemming* is the task of stripping affixes from a word leaving its stem. Decomposing a word into its morphemes is called *morphological parsing*. Stemming goes beyond variations in a word form to obtain the main meaning of the word.

The set of different forms of a word with the same meaning is called a *lexeme*; “find, finds, found, finding” are the forms in the English lexeme “find”. One form of the lexeme is selected as representative of the whole lexeme, and is called its *lemma*. The lemma is also called the *citation form*, *canonical form* or *dictionary form* of the lexeme. Word “find” is the lemma of lexeme “find, finds, found, finding”. *Lemmatization* is the task of determining the lemma corresponding to a given form of a lexeme. In some contexts, the lemma is called *headword*, to differentiate it from the *wordforms*, which are all the forms in a lexeme, including its headword.

*Sentence segmentation* is the task of dividing a text into sentences. *Word segmentation* or *tokenization* is the process of dividing a sentence into its compounding wordforms.

A *generative grammar* is a description of a language in terms of rules which define all the “well formed” strings in the language, i.e. the strings that comply with the rules. A language defined by a generative grammar is called a *formal language*.

A *token* is a string of symbols of an alphabet, formed according to certain rules. Though a token may be considered a word when dealing with text, it is a wider concept, since it may be just a sequence of characters, as in some formal languages.

Words may play different roles in a sentence, which leads to the notion of *lexical categories*, or *part of speech* (PoS) for that word. Lexical categories are further partitioned into *open word classes*, which constantly acquire new members while others fade away, and *closed word classes*, which exhibit little or null variation in time. For most western languages, open word classes are nouns, verbs, adjectives and adverbs; closed word classes are determiners, pronouns, prepositions, and conjunctions. Closed word classes are essentially *function words*, with little meaning themselves, but used to establish grammatical relations among words.

An *N-gram* is a subsequence of N items within a sequence, where N is an integer. A bigram is a 2-gram, and a trigram is a 3-gram. An N-gram model is a model which attempts to predict the next item in a sequence, given some subsequence. N-gram models are useful in PoS tagging, when a single wordform may belong in several part of speech categories: neighbour wordforms may help decide which part of speech the wordform is playing in a particular sequence of wordforms.

5.3. Design considerations

In the context of this work, a domain specific sublanguage is conceived as a means to create a knowledge representation instance from a piece of written text. This calls for a close
5. Construction of a Sublanguage

correspondence in design between the sublanguage and the knowledge representation. Both
depend ultimately on the target domain: the knowledge representation must be able to re-
fect the different types of concepts and associations proper of the area of knowledge, and the
sublanguage must be able to describe in text sentences how the concepts in the domain relate
to one another in the association types typical of the domain. In short, the design of the
sublanguage and the knowledge representation must be done in parallel, or at least, in close
co ordination.

The user’s perspective calls for a sublanguage which resembles natural language as closely
as possible: a text written in the sublanguage should be perceived as normal and correct use
of the natural language. In this way, the use of a sublanguage will be transparent to the
reader, though not to the writer, who must know and follow its rules. In this sense, it can be
said that the sublanguage is contained in the natural language: all sentences written in the
sublanguage are also correct sentences in the natural language. The reverse is, naturally, not
so; this is precisely the purpose of defining a sublanguage.

The requirements of a sublanguage for knowledge representation may be summarized as
follows:

- **readable**: text in a sublanguage should be both natural and correct use of the natural
  language.

- **easy to learn**: a post secondary student should be able to familiarize with the rules and
  keep within the lexicon with only a few hours instruction and practice.

- **simple to write**: once learnt, writing in the sublanguage should not require greater efort
  than writing in the natural language. An application may help towards this goal.

- **a clear semantics**: each structure in the language must lead to a structure in the knowl-
  edge representation scheme.

- **unambiguous**: no two different formal interpretations should be possible for each sentence
  in the text.

The most challenging requirement is determinism, to avoid ambiguities; this is the main prob-
lem in Natural Language Processing. Using a well defined sublanguage is a way to circumv en
an otherwise very difficult, and in many cases, intractable problem.

5.4. Writing in a sublanguage

Writing in a sublanguage requires some training. A successful sublanguage will most surely
depend on the quality and adequacy of documentation and training material. New users
should be clearly informed of the purpose, design, resources, tools and use of the sublanguage.
A careful set of example sentences illustrating the different cases will save time and effort.

There exist a number of guiles for writing, from simple recommendations of style to strict
instructions in controlled languages. Simple Wikipedia article *How to write Simple English
articles* [Wikipedia, 2012a] is an easy to follow guide to write articles in simple English for
contributors to the Simple English Wikipedia. Appendix 5 of Karin Disborg’s master thesis
[Disborg, 2007] contains an *SE Guide* to write in ASD Simplified Technical English, a con-
trolled language described in specification ASD-STE100 from ASD, the Aerospace and Defence
5.4. Writing in a sublanguage

Industries Association of Europe. The Plain English Campaign, an UK editing company, promotes Plain English, a style of writing which privileges simplicity and conciseness, mainly to be used by institutions when addressing the public; it is described in *How to write in plain English* [PlainLanguage, 2007].

The following points gather some writing rules for a sublanguage oriented towards knowledge representation. They are based on the former sources, plus some others specific for our purposes. Some of these rules can be enforced by the sublanguage generative grammar, others will depend on the writer. The rules given may be used as a first approach in the design of a sublanguage.

- Use only words from the lexicon. Additions or modifications to the lexicon should be done in conformance with the regulations issued by the community of users responsible for the lexicon.

- Proper nouns can be freely used, provided they are written as the sublanguage rules establish, so that they are effectively recognized as proper nouns. Technical names and acronyms may be treated as proper nouns, if they are not included in the domain specific vocabulary.

- Use words only as the part of speech given in the lexicon.

- Use words only in the senses recorded for that word in the lexicon.

- Keep clusters of nouns, adjectives and adverbs to no more than three.

- Use only the forms of the verbs found in the lexicon. Recommended verb forms to admit in a lexicon are: infinitive, imperative, simple present, simple past, and future. The past particle of a verb should be admitted in the lexicon only as an adjective, to be used with a noun, or after linking verbs like *be* or *become*.

- Use only the active voice. If the passive voice is admitted, it should be used sparingly, and only in descriptive writing, not in procedural writing (instructions).

- Keep to only one topic in a sentence (descriptive writing); keep to only one instruction in a sentence (procedural writing).

- Keep sentences short. Commonly recommended maximum lengths are 20-25 words for descriptive writing, 15-20 for procedural writing.

- Use the imperative for instructions (procedural writing).

- Try to vary length of sentences to make text readable. Recommended lengths are a maximum, shorter sentences are welcome.

- Keep to one topic per paragraph, or several related small topics.

- Write your paragraphs so that they show the organization of your document.

- Repeat words as necessary to relate one sentence to another.

- Keep the length of your paragraph to a maximum of 6 sentences.
5. Construction of a Sublanguage

- Use square brackets and punctuation marks according to the requirements of the sublanguage, to avoid ambiguities.

- Consult the lexicon to correctly get the words, senses and parts of speech accepted in the sublanguage.

Some of the preceding rules will most surely be present in most sublanguages, others may be included or not. These kind of rules are commonly applied in technical manuals and legal documents; the resulting texts are easier to read, easier to understand, and less prone to ambiguous interpretations. There is no extra effort on the reader of such texts; they are usually easier to read and simpler to understand. Effort on the writer is moderate to low, and after some practice the style is naturally adopted.

The kind of sublanguage needed for knowledge representation goes a step forward: the writer must be proficient in the application of grammar rules, she will be required to know the production rules of the generative grammar, and write according to them. This is not the same as being proficient in the grammar of the natural language, English, Spanish or other, but on the production rules of a particular sublanguage of the natural language. It is absolutely necessary that the production rules of a sublanguage produce grammatical constructions valid in the natural language. Not adopting this basic design principle will result in texts difficult to read and very difficult to write. A sublanguage is a subset of natural language, not a different language.

Writers in a sublanguage are expected to count on the following resources:

- a tutorial of the sublanguage, including
  - motivation and purpose of the sublanguage,
  - a review of the grammar structures used in the sublanguage, with plenty of examples.
  - the part of speech tagset and other symbols, explained.
  - a description of the lexicon, its sources, how to add to it if it is allowed.
  - the production rules, explained and illustrated with examples.
  - a “how to say” guide, to show how some common structures in natural language can be written in the structures accepted by the sublanguage.
  - examples of sentences and texts written in the sublanguage.

- a lexicon, easy to look up, with information on wordforms, headwords, parts of speech, and senses for each (headword, part of speech) pair.

- a sublanguage editor, with the following capabilities:
  - detect words not in the lexicon.
  - accept new words, if sublanguage allows; the writer must be ready to provide the necessary information, at least the headword and part of speech; for a headword, all its wordforms can be found in existing large lists of inflections.
  - build the syntactic tree for each sentence, inform the writer if no tree could be built, inform the writer if more than one tree was built, accept corrections to the sentence.
- context sensitive help, easy look up of words in the lexicon, access to example sentences.
- other more complex but useful facilities are: word completion, visualization of partial parses, anticipation of parts of speech allowed.

- additional documentation on the sublanguage, such as a specification for the construction and maintenance of the lexicon and production rules.

5.5. From text to graph

The transformation of a text written in a sublanguage into a knowledge representation instance will be done in these steps:

1. For each sentence in the text, build a unique syntactic tree. More than one syntactic tree for the same sentence would transfer ambiguity to the knowledge representation, which is assumed to be deterministic.

2. Recognize in the syntactic tree, or in a transformed version of it, the concepts and relationships allowed in the knowledge representation scheme corresponding to the sub-language.

3. In each new sentence, detect concepts already mentioned in former sentences, and hence already included in the knowledge representation instance; add new concepts and relationships as needed. Relationships may be established among new concepts, among a new concept and an existing one, or as a new relation among existing concepts.

One of the main purposes of using a sublanguage is to ensure only one syntactic tree may result from a sentence. To build the syntactic trees for the sentences in a text, the following tasks must be performed:

1. *Sentence segmentation:* separate the text in sentences. A sentence usually ends in a period, but a period may also be used in abbreviations. A solution is to recognize a sentence boundary as a sequence of characters (period, space) or (period, end of line).

2. *Tokenization:* separate the wordforms in a sentence. This is a harder than expected task in the general case; in a sublanguage, tokens must be wordforms or symbols in the lexicon.

3. *Lemmatization:* determine the lemma for a wordform. A wordform may lead to more than one lemma. The immediate solution is to allow only one lemma per wordform. NLP tools and resources exist to relax this requirement, such as the analysis of N-grams in comparison with those in an annotated corpus, or determining the possible parts of speech to see which suits in the sentence.

4. *PoS tagging:* assign a lexical category (part of speech) to each word in a sentence. This is a hard task in the general case. Besides the possible ambiguity in lemmatization, the same lemma may play as several parts of speech. Again, the immediate solution is allow only one part of speech per lemma, (besides one lemma per wordform). As in lemmatization, tools exist to relax this restriction.
5. Construction of a Sublanguage

5. Syntactic parsing: build a syntactic tree for the sentence. This should be straightforward on a generative grammar. However, more than one syntactic tree for a sentence may be licensed; syntactic ambiguity does not allow a deterministic knowledge representation. The rules of the grammar should allow for only one syntactic tree. This restriction can be relaxed by asking the writer to choose one syntactic tree when more than one are possible, according to her intentions.

6. Word sense disambiguation: determine the sense in which each word is used in a sentence. This is a major problem in the general case. In a sublanguage, the lexicon may be very specific, reducing drastically the meanings attached to a lemma (polysemy). Again, the author can be asked for her intended meaning.

Several of the former requirements can be made more flexible by asking the author to choose the adequate alternative according to what she wants to say. Though possible, this must be reserved as a last resort, or not used at all. Besides the bother for the author, accepting alternatives tends to weaken the sublanguage itself, opening it to ambiguities and imprecision.

Transformation of a syntactic tree into a knowledge representation instance can be done in many different ways. In a class hierarchy scheme, for instance, a sentence like “Dogs are animals” may result in the creation of Dogs and Animals as classes, with Dogs as a subclass of Animals; “Rufo is a brown dog” may result in the creation of Rufo as an object of class Dog with an unnamed instance property “brown”. In a purely syntactic interpretation, Dog will be related to Animal, Rufo to Dog and brown to Rufo, with no particular hierarchy relations, and perhaps even with no differentiation of relationship types. The relations involving sublanguage, knowledge representation language and this interpretation of the text are treated in Part IV, which deals with Knowledge Representation.

Besides the ambiguities already considered, natural language allows many different ways of conveying the same meaning, using different words and different structures, sometimes in very complicated circumlocutions. Even simple situations like “Rufo is a dog”, “Rufo is a brown dog”, “Rufo belongs to Jack”, “Jack’s dog is brown”, “Jack’s dog’s name is Rufo” may prove difficult to handle as such. An immediate approach is to recognize linking verb “be” as an adder of adjectives, in such a way that “the dog is brown”, “the dog chased the cat” can be recognized as equivalent to “the brown dog chased the cat”. When coming to the representation of knowledge, adjective “brown” will be attached to “dog”, and “dog” related to “cat” in a “chase” relation, whatever the sentences. The habit of writing in a straightforward way promoted by advocates of simple English in any of its forms does away with the most far fetched ways of writing: production rules enforce certain structures to establish relations, the lexicon limits the election of words. But all this may not be sufficient; even for a well designed sublanguage, there is always the risk of someone writing something in an unpredicted way, saying the correct thing. Stated in this way, there seems to be no way out of this problem. However, language is a social phenomenon, people learn its rules and go by them, so the others can understand. A sublanguage is a community endeavour, restricted to a group with some common interests, ready to make an effort towards a normalized, predictable expression of their shared knowledge. Variations in ways of saying the same thing are expected to be much less frequent within such a group. A well designed sublanguage, adequate training, effective tools, and engagement of the practitioners, are powerful allies towards a normalized communication.
5.6. Tagsets

A tagset is the collection of part of speech labels or "tags" used to annotate a text; tags indicate the lexical category of each word. There are many different tagsets, for different purposes, associated to different resources. Since tagsets are used in different ways for different tasks, no single tagset suits all, and variations in tagsets are unavoidable [Bird et al., 2011].

A simple, mnemonic tagset for English is given in the following boxes.

<table>
<thead>
<tr>
<th>Terminals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Np : noun, proper</td>
</tr>
<tr>
<td>Nc : noun, common</td>
</tr>
<tr>
<td>Adj : adjective</td>
</tr>
<tr>
<td>Adv : adverb</td>
</tr>
<tr>
<td>Vlnk : verb, linking</td>
</tr>
<tr>
<td>Vact : verb, action</td>
</tr>
<tr>
<td>Vintr : verb, intransitive</td>
</tr>
<tr>
<td>Vtran : verb, transitive</td>
</tr>
<tr>
<td>Prep : preposition</td>
</tr>
<tr>
<td>Det : determiner</td>
</tr>
<tr>
<td>P0 -&gt; '['</td>
</tr>
<tr>
<td>P1 -&gt; ']'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non terminals</th>
</tr>
</thead>
<tbody>
<tr>
<td>S : sentence, start symbol</td>
</tr>
<tr>
<td>NP : nominal phrase</td>
</tr>
<tr>
<td>NG : nominal group</td>
</tr>
<tr>
<td>AdjP : adjective phrase</td>
</tr>
<tr>
<td>AdjG : adjective group</td>
</tr>
<tr>
<td>AdvP : adverb phrase</td>
</tr>
<tr>
<td>AdvG : adverb group</td>
</tr>
<tr>
<td>PrepP : prepositional phrase</td>
</tr>
<tr>
<td>VP : verb phrase</td>
</tr>
<tr>
<td>VactP : active verb phrase</td>
</tr>
<tr>
<td>VtranP : transitive verb phrase</td>
</tr>
<tr>
<td>VPintr : intransitive verb phrase</td>
</tr>
<tr>
<td>VlnkP : linking verb phrase</td>
</tr>
<tr>
<td>Attr : attribute of linking verb</td>
</tr>
</tbody>
</table>

The following substitution tags are the names of terminals in lowercase, and are intended to be used instead of wordforms.
5. Construction of a Sublanguage

<table>
<thead>
<tr>
<th>Substitution tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Np -&gt; 'np'</td>
</tr>
<tr>
<td>Nc -&gt; 'nc'</td>
</tr>
<tr>
<td>Adj -&gt; 'adj'</td>
</tr>
<tr>
<td>Adv -&gt; 'adv'</td>
</tr>
<tr>
<td>Vlnk -&gt; 'vlnk'</td>
</tr>
<tr>
<td>Vintr -&gt; 'vintr'</td>
</tr>
<tr>
<td>Vtran -&gt; 'vtran'</td>
</tr>
<tr>
<td>Vact -&gt; 'vact'</td>
</tr>
<tr>
<td>Prep -&gt; 'prep'</td>
</tr>
<tr>
<td>Det -&gt; 'det'</td>
</tr>
</tbody>
</table>

A sentence like “Rufo is a big black dog” can be transformed into “np vlnk det adj adj nc”, its syntactic tree can be built from these “substitution wordforms”, and the transformation reversed to obtain the syntactic tree in the original wordforms. This allows to decouple the lexicon from the production rules: a set of production rules can be tested against these substitution wordforms, or a small lexicon prepared for better readability and used for testing instead of a full lexicon. This also allows the software tools to work on any combination of production rules and lexicon, provided a correspondence is established between the tagsets of both. The substitution tags can even act as an interlingua in the conversion of tagsets used in lexicon and in production rules, provided a mapping between these tagsets can be established.

5.7. Lexicon

A lexicon may be compiled from a corpus of domain specific texts or from existing lists of commonly used words. Probably the most cost effective approach is to adapt an existing word list and add to it domain specific vocabulary. Whatever the method, an ideal lexicon for a sublanguage is expected to have the following properties:

- return a unique headword (lemma) for a wordform.
- return a unique part of speech for a headword.
- return a unique sense for a headword.

Though these requirements can probably be realized in a strict domain specific lexicon, with no provision for “general use”, they are very restrictive, and difficult to reach in a real world application. Ambiguities are inherent to language, and difficult to eradicate. All ambiguities which cannot be solved by the sublanguage will end up on the writer, who is the last authority on her own writing. This may be a valid last chance resource, but the designers of a sublanguage will do as much as possible to solve all ambiguities within the sublanguage, first by not allowing them to happen. This is not an easy task, but is not far from the more general interest of having a domain specific vocabulary as clear and deterministic as possible for an area of study.

The difficulty of returning a unique headword for a wordform depends on how inflected is the natural language of which the sublanguage is a subset. English is relatively simple in this respect, while Spanish varies more. A general purpose lexicon plus domain specific vocabulary can be estimated to be around 3000 headwords; this seems a manageable size. An analysis of
5.8. Rules

non disjoint lexemes can be complemented with frequency of use counts to determine if some lexemes can be suppressed, or natural synonyms can be used instead.

Returning a single part of speech per headword is a similar problem, which can also be reduced by careful selection of headwords based on frequency of use and availability of synonyms. Context sensitive techniques can be applied to solve the ambiguities, such as N-gram analysis, in which the part of speech of a word may be inferred from the part of speech of some neighbour words; these techniques help towards a less restrictive lexicon.

Both problems, lemmatization and PoS tagging, can be dealt with simultaneously with some profit: an N-gram analysis may determine a unique PoS is possible for a wordform, which in turn may lead to a unique headword, even if the wordform in question could, alone by itself, lead to more than one headword.

Returning only one sense for a headword will be a desirable feature for any domain specific lexicon, but may be very difficult to achieve: polysemy is a very frequent linguistic phenomenon in most languages. Word sense disambiguation is a well known problem in NLP, and several techniques exist to deal with it. For knowledge representation, though, a unique sense per headword is not a crucial requirement: lemmatization and PoS tagging are necessary to ensure only one possible syntactic tree per sentence, but inferring the correct sense from a word, were it in a sentence or in a knowledge representation instance, will ultimately be done by a human mind. A carefully revised mapping of headwords to senses is a desirable feature in any lexicon for any area of knowledge. Frequency of use and availability of synonyms are once more the tools available to reduce the number of senses per headword.

5.8. Rules

For the compilation of a lexicon, a number of reliable resources are available. This is not so when putting together a set of production rules: tagsets may be very different, many proposals are presented only as examples, different purposes may lead to sets of rules with little in common among themselves.

Rules must comply with the same general requirements as lexicons and the sublanguage itself: be natural use of the language, be relatively easy to learn, be defined according to the intended knowledge representation. Rules must ensure any licensed sentence produces only one syntactic tree; syntactic ambiguity cannot be allowed in the construction of a knowledge representation. This requirement may be difficult to enforce and guarantee in all cases: it may pose restrictions on the sublanguage which may not be tolerable for the users. Ambiguity may come both from the lexicon, e.g. if a headword can play as different parts of speech, or from the rules themselves, e.g. if one group of words can be recognized as different syntactic constituents. There is a compromise between a single syntactic tree and a usable sublanguage. A pragmatic solution will do as much as possible to keep the sentence structure relatively simple, put a limit on the length of the sentence, use recursion judiciously, and ultimately accept that in some cases more than one tree may turn up, which will be cleared up by the author choosing the appropriate one according to what she wanted to say. Though it is not desirable to involve the writer in syntactic matters, ambiguity is a pervading phenomenon in the language, of which the author must be conscious. Writing in natural language frequently exhibits ambiguity, even to the point of confusing a human reader. Avoidance of ambiguity is a concern in educational and technical writing. Occasionally asking the writer to choose between a couple of alternatives to ensure determinism in her expression should not upset any
5. Construction of a Sublanguage

conscious author.

On the positive side, a set of production rules for a sublanguage may be used in different
disciplines, which makes its design more cost effective.

An approach to the requirements for a set of rules to be used in knowledge representation
follows:

- **Types of sentences.** The rules should admit only the types of sentences required by the
  purpose and knowledge representation scheme in mind. In most cases only assertive
  sentences will be allowed. Imperative sentences may be required for instructions.

- **Negation.** If negation is admitted, its forms and meaning must be adequately defined.
  Though easy to include in the rules, the semantic treatment of negation is not trivial in
  the general case.

- **Active voice.** The rules should admit only sentences in the active voice. Most passive
  voice sentences can be converted to active voice sentences in a straightforward way;
  others require a bit of effort to find a subject. Writing in the active voice is a usual
  recommendation for clear writing.

- **Syntactic determinism.** The rules should license only one syntactic tree per sentence, at
  least in most cases. Disambiguation by the author can be a last resort, tolerable if not
  very frequent.

- **Anaphora.** Avoid anaphora as far as possible. A first measure is not to accept pronouns;
  this leads to some repetition, but is simple to implement. There are NLP techniques
  for anaphora resolution which may be applied in further developments. Not all forms of
  anaphora can be automatically solved, though. The author must be conscious of this.
  Speaking of “John”, and then saying “the man” may be clear for a human reader, but
  not for a machine.

- **Simple sentences.** Admit only simple sentences, avoid compound sentences (sentences
  joined by a conjunction) and complex sentences (sentences joined by a subordinating
  conjunction). A knowledge representation instance can be built incrementally, adding
  one simple sentence after another. Though some repetition will be necessary, the text
  will be equally readable, and in many cases be clearer and less prone to ambiguities.

- **Conjunctions.** Admit conjunctions sparingly, only where repetition can be avoided with
  no loss in clearness, as in joining different subjects or objects: “Rufo, Wanda and Toppy
  are dogs”, “my brother met Julia, Davis and the carpenter at the seaside”.

The former requirements may be tightened or relaxed in a concrete proposal. A set of rules
developed along these lines is given in chapter 8 and a knowledge representation for these
rules is describe in chapter 11.

5.9. Preprocessing

A desirable feature for a successful sublanguage is to allow the writer to proceed much in the
same way as if she was writing in a natural language. A more precise, achievable goal is to
enable writing with only stylistic constraints, i.e. those recommendations for simple, clear
5.9. Preprocessing

Writing normally accepted even for writing in natural language, when efficient, unambiguous communication is at a premium. Preprocessing the text offers a way to hide from the author some of the constraints of the sublanguage, making writing look more natural. Preprocessing the text means scanning through it for the recognition of certain patterns not acceptable for the sublanguage, and transforming them accordingly before submitting the text to lexical validation and parsing. A discussion of some of these situations follows.

- **Units of several words.** Some concepts are expressed in more than one word; there is a group of words which acts as a single word. Phrasal verbs are an immediate example: “look up”, “look for”, “look after” are all different in meaning, and act as a single word. Though the two parts may be separated in a sentence like “look the address up in the directory”, the sublanguage may well call for phrasal verb components to be kept together. An immediate approach is to write these verbs with an underscore, as in “look_at”, “look_after”. Other groups of words which act as a unit can be identified in the same way. To avoid compelling the author to write the underscores, these word groups may be included in a complementary lexicon, the text explored for the sequence of words, and the underscores inserted before submitting the text to validation by the sublanguage. This requires that the sequence of words in a group is never valid as a sequence of separated words, but this is not difficult to achieve.

- **Proper nouns.** Proper nouns may be recognized for being in a complementary lexicon, identified by some particular way of writing, or by being so declared by the writer. Most languages start a new sentence with a capitalized word, but if the grammar always requires a common noun to be preceded by a determiner, at least at the start of the sentence, a capitalized word may be enough to identify a proper noun. Lists of common names exist, and may be used to recognize as a proper noun a word not in the lexicon of the sublanguage. Proper nouns involving more than one word may be written with underscores, as in “Rio_de_la_Plata”. This example shows it may be difficult to spare the author from writing the underscores; proper nouns in several words may be lower or uppercase, and are less predictable than phrasal verbs, for instance.

- **Initial capitals.** Sentences usually start with a capitalized word, even if it is a common noun. If this first word is found in the lexicon as a common noun, it must be considered as such; there may not be a proper noun with the same spelling as a common noun, unless explicitly indicated and recorded. Once it is determined that the first word of the sentence is a common noun, it may be transformed into lowercase before processing.

- **Numbers.** The small cardinal numbers may be included in the lexicon written as text, and recognized as determiners (“three dogs”) or adjectives (“the three musketeers”). The first ordinals may be included as adjectives (“the second world war”). Numbers as digits may be accepted with no limitations. There must be a definition of how to proceed with numbers when coming to build a knowledge representation instance, i.e. the semantic interpretation of these numbers in the knowledge representation scheme which this sublanguage is supporting.

- **Dates.** Dates may be accepted in a predetermined format, such as mm/dd/yy, mm/dd/yyyy, yyyy-mm-dd, or others. Their significance and treatment must be established beforehand, as usual. The same happens with time.
5. Construction of a Sublanguage

- *Measurements and other quantities.* As long as their format, units included, is clearly defined, including measurements or number plus units offers no great difficulty for the sublanguage. Once again, their interpretation must be clearly established for knowledge representation.

5.10. Enhancements

Once a basic sublanguage has been defined and agreed upon by the community of users, and tested to satisfaction, a number of enhancements are possible, both to help the authors write according to the rules and lexicon of the sublanguage, and to make the text more natural and readable. Some possible enhancements are:

- **Word completion:** to complete a partially written word with the word in the lexicon as soon as it is determined, i.e. when sufficient characters have been introduced to select only one word, or showing possible words in a list. This feature will be more useful in languages with longer average word length.

- **Partial parses.** Visualization of incomplete syntactic trees for a sentence, e.g. the longest parse obtained, is helpful to determine why a sentence has not been accepted by the sublanguage.

- **Part of speech anticipation.** Showing the author which parts of speech are allowed for the next word may help a new writer in the sublanguage, and gives an insight of the grammar. Once a writer has become familiar with the sublanguage, part of speech anticipation will be of little use or even become cumbersome. However, it may still be useful to determine why a sentence is not licensed by the grammar. In any case, it can be given as an option.

- **Lexical help.** Several types of lexical information, examples and hints can be given to an author: a word not in the lexicon can be mapped to a word in the lexicon with the same meaning; definitions and examples of use may be shown for a word as in the Wordnet; a wrongly spelled word will not be accepted, but words of similar spelling may be presented as hints.

- **Word sense disambiguation.** A developed sublanguage may have identified all the senses in which words are used in the domain. In an ideal case there may be only one word for each sense, or a canonical word and accepted synonyms. A less restrictive, equally effective requirement is a unique sense for each (headword, part of speech) pair. Even this may be difficult to achieve. If several senses are accepted for a pair (headword, part of speech), the senses may be displayed when writing, for the author to choose.

An idealistic but not impossible enterprise would be a sublanguage with a lexicon mapped to senses, in a Wordnet like manner, or even using or adapting from the Wordnet. Such a lexicon might have one or several words for a meaning, with a canonical word to identify the sense. In such a sublanguage, an author would not be writing in words, but ultimately in concepts, producing a semantically determined text; a knowledge representation could be a representation of synsets. The proposal is as difficult as appealing: exact translation into a sublanguage of another natural language would be possible; the knowledge representation obtained could be read in different languages, its inner content of synsets acting as a kind of
interlingua; authors would not be required to learn a new language, but only a subset of their own.

5.11. Conclusions

The former sections identified some requirements for the development of a practicable sublanguage for knowledge representation, pointed out some tools and procedures offered by Natural Language Processing to deal with them, and gave some hints on the design of such a sublanguage. Though a challenging endeavour, tools available, restriction to a specific domain, and the possibility of gradual development make the task possible. A sublanguage for the purpose of this thesis is inherently limited, not the same work as building a lexicon and generative grammar for natural language. A successful sublanguage should be transparent to readers, and easy to use by writers. Documentation, an intelligent editor, and some training are crucial for a sublanguage to become successful for the purpose of this thesis.

A sublanguage for knowledge representation must be a subset of correct natural language, apt for use in an educational environment, and semantically deterministic to allow for knowledge extraction. The contributions of this chapter addressed the construction of such a sublanguage, by proposing:

- a characterization of a sublanguage for Education and correct writing. The sublanguage proposed lies in between of strict controlled languages and recommendations of style, enforcing some mild limitations on natural language, keeping the natural look of written texts, and readable as correct natural language.

- a first approach to a methodology for the construction of a sublanguage, including design considerations, identification of general writing rules, facilities an application must provide, documentation and tutorials, detected difficulties, and ways to solve them.

- a procedure to convert text written in a sublanguage into a knowledge structure, including the identification of tasks required, the use of ideas from dependency grammars, the definition of a simple tagset, and the use of substitution tags to decouple lexicon and rules.

In the coming chapters of this part, several lexical resources are evaluated towards the compilation of a lexicon for general use, and a syntax oriented generative grammar is proposed, towards a knowledge representation much in the way of a concept map.
6. Lexical Resources

Abstract. A lexicon for a domain specific sublanguage can be compiled from a basic lexicon adding domain specific terms. Several word lists exist and are publicly available. Short word lists, of about 2000 to 3000 headwords, try to capture the most commonly used words, or a set of common words enough for general use. Long lists include a high number of words, and attempt to reflect their use in the language. Short lists are useful to select a core vocabulary, while long lists are useful for verification and auxiliary tasks such as lemmatization. The compiled lexicon is expected to provide support for performing the following tasks on a text: lemmatization, part of speech tagging, and sense determination. A single wordform may correspond to several headwords, several parts of speech and several senses. The compiled lexicon must reduce these ambiguities as much as possible, selecting the most common and the most adequate for the domain. Several short and long lists were selected and brought to a common data structure for self checking, comparison, and evaluation. The short lists were analyzed and compared among themselves in the information provided, number of words, format, and usability as a core lexicon. The long lists were also analyzed and compared among themselves, and evaluated as a reference for checking the short lists or as a resource to widen the coverage of the short lists adding new words. The short lists analyzed were several versions of the General Service List of English words, Simple Wikipedia Basic English, Voice of America Special English and Longman Communication 3000. Long lists analyzed were from the British National Corpus (BNC), the Corpus of Contemporary American English (COCA), SCOWL (Spell Checker Oriented Word Lists), AGID (Automatically Generated Inflection Database), PoS Database (the Part Of Speech Database). The Wordnet, considered here a long list, provides far more facilities than a simple list, and offers many opportunities for compiling and supporting a base lexicon. The short lists were compared in the information provided and in their superposition on each other. The long lists were compared in the information they provided and their size. Leah Gilner version of GSL was selected as a model short list, and tested for inclusion in the long lists. Tests of coverage of the short lists in BNC and COCA long lists showed a coverage of 87% for GSL Gilner and 94% of Longman 3000. Though Longman 3000 shows somewhat better coverage, it is a purely statistical list, while GSL Gilner attempts at a usable set of common words for everyday use; both can be used as core word lists for a lexicon. AGID list can be used for lemmatization, and PoS DB for part of speech tagging.

A sublanguage apt for knowledge representation must comprise a lexicon of accepted words, and a grammar to specify the syntactic structures of its sentences. The compilation of a lexicon need not start from scratch; several lexical resources may be used, from short lists of commonly used words, to very large lists usable in language processing tasks such as lemmatization or part of speech tagging. Elaborate resources such as the Wordnet may be called for to detect
6. Lexical Resources

synonyms, provide senses, show definitions, and examples of use. In this chapter, a number of lexical resources are identified and compared. To this purpose, a common data structure is proposed, and some tests of self consistency and comparison among lexicons is performed. The most suitable resources are selected for the compilation of a small lexicon of general use, a task carried out in the next chapter.

6.1. Introduction

In the definition of a Domain Specific Sublanguage (DSS), a core lexicon of manageable size and wide coverage can be taken as a first approximation towards a domain specific lexicon; experts in the domain can then add to this core lexicon the terms usually employed in the domain. The number of added terms is expected to be relatively small and very well known to the experts in the domain, which makes the task manageable. The most used words in the English language have long been studied, and several lists of frequently used words do exist, in a range of 1500 to 3000 words. A modern study of the General Service List of English Words (GSL) shows a coverage of 70% to 90% in texts coming from any source or domain [Gilner, 2011]; the Longman Communication 3000 list claims a coverage of 86% of spoken and written language [Longman, 2008]. Several word lists are publicly available, and can be exploited in different ways: to select the most frequent words, to determine headword for a wordform (lemmatization), to determine part of speech from a wordform or headword (PoS tagging), to provide definitions of the meaning assigned to each headform in the target domain.

Two types of lists are available: short lists and long lists. Short lists attempt to capture the most frequently used words, the smallest set of words necessary to understand the language, or the minimal vocabulary for successful study at a certain level. Long lists try to capture all the words used in the language, that is, all the words which appear with some frequency in a large corpus of written or spoken English. Short lists are helpful to define a core lexicon, long lists are useful to test the core lexicon and to accomplish several other tasks such as lemmatization and PoS tagging.

In this chapter, several publicly available lists were brought to a common data structure, evaluated and compared to assess their potential usefulness towards the compilation of a general purpose basic lexicon.

6.2. Terms

As is usual in language related disciplines, many terms are defined differently, and arguments are held on their meaning. This section explains the sense in which some terms are used in this work. No attempt at a formal definition is made; the purpose of this section is only to state what is meant by each term.

A lexicon may be the collection of terms in a language, or the collection of terms used in a profession or subject; also, vocabulary. A glossary is a list of terms peculiar to a field of knowledge with their definitions. A wordbook is a reference book containing a list of words with their definitions.

A word is usually defined as a unit of language that native speakers can identify. Thus a word may designate a word that appears as an entry in a dictionary as well as all its variations. The word used as an entry in a dictionary is called a headword, lemma or citation form of the word. Variations of a headword, such as "dogs" from "dog", or "makes, made" from "make",
6.3. Lexical Resources

are called *inflected forms* or *inflections*. The headword and any of its variations is called a word form, usually written as *wordform*. *Lemmatization* is the operation of determining the headword corresponding to a wordform [Farlex Inc., 2010].

*Part of speech* (PoS) is the class of all the words that may occur in the same places in a sentence, or accomplish similar functions in context. The parts of speech recorded in the Wordnet are nouns, verbs, adjectives and adverbs. Other parts of speech include determiners, prepositions, pronouns, conjunctions and interjections. Parts of speech are also called *syntactic categories* [Farlex Inc., 2010]. A part of speech is frequently associated with a label or *tag* as a shorthand for the whole part of speech name, such as V for verb, N for noun or Adj for adjective. A *tagset* is the collection of labels or tags used to refer to parts of speech. Except for the main categories, there is no uniform recognition of syntactical categories, which may go to very minute details, resulting in very complex schemes of classification. Hence, a number of tagsets exist. A tagset is usually adopted or adapted from existing ones according to the purpose pursued. The operation of recognizing the part of speech of each word in a sentence is called *Part of Speech tagging*, or *PoS tagging*.

Drawing from Computer Science, the *expressiveness* or *expressive power* of a language may be defined as the scope of ideas that can be expressed in that language. A language will be more expressive if it can express more ideas, or if it can express the same ideas without recurring to unwieldy, far fetched constructions.

6.3. Lexical Resources

Word lists contain different types of information: wordforms, headwords, frequencies of occurrence in some corpus, part of speech, definitions. Frequencies are useful to determine which words should be prioritized in the compilation of the lexicon. Part of speech and correspondence between wordforms and headwords are useful for knowledge extraction, in the operations of lemmatization and PoS tagging, required to perform syntactic validation and syntactic tree construction. Though definitions are not required for syntactical analysis nor knowledge extraction, availability of word definitions as used in the domain is valuable in itself: whatever the area of knowledge, agreement on the meaning of words is a must to ensure accurate communication among practitioners.

Word lists may be classified in two main categories: those obtained by language and statistical processing of corpora or “large lists”, and those compiled for a purpose, or “small lists”.

Statistically compiled word lists are specially useful for frequency of occurrence; this helps determine the most common words, either in general use or specific genres. These lists are also useful for lemmatization and PoS tagging. Statistically obtained lists tend to be large in size, since they attempt to capture as much variety as possible. For this reason, they are normally based on very large corpora, of a million words or more.

Word lists compiled for a purpose, such as second language teaching, try to determine the relatively few words which allow the highest amount of communication within a domain. Their typical size goes from about 1000 to 3000 words or little more. A student mastering this basic lexicon is expected to understand most of normal written or spoken communication in the language.

As stated in the Guiding Principles of this thesis, a domain specific sublanguage aims at simplicity both in structure and use of words. Hence, the compilation of a lexicon for a domain specific sublanguage should be small number, sufficiently expressive, and agreed upon
6. Lexical Resources

by practitioners. The number and selection of words should allow the natural, straightforward writing of readable texts easy to understand, with little or no ambiguity, as far as it is possible in a human language. Existing resources may be engaged to compile a core lexicon, to which domain specific terms may be later added. For a final, production quality lexicon, a manual adjustment by both linguists and experts in the domain will most probably be required.

This section reviews some selected lexical resources. A comparative evaluation follows in the next section.

6.3.1. Word Lists Content and Uses

Word lists provide a variety of information, sometimes on different arrangements difficult to consolidate. Usual types of lists are:

- **lists of words**, with no other information, ordered by frequency of use, alphabetically, or in no particular order. Lists may be of headwords or wordforms. For short lists, inclusion is already informative, since it means accomplishment of some condition; long lists attempt to capture all words meeting some condition of wide range, or just try to capture as many words as possible.

- **frequency lists**: entries are word and frequency count; word may be headword or wordform. An indication of dispersion may be given: a word with high dispersion appears in more texts or genres, and hence is considered of more general use; a word with low dispersion may be used only in some areas.

- **inflection lists**: entries are headword with their corresponding wordforms or inflected forms.

- **PoS lists** give the lexical categories for words. Again, word may be a headword or wordform. A word may belong to more than one lexical category. Sometimes an indication of frequency for a pair (word, PoS) may be given.

- **definition lists**: definitions are given only in the short lists, corresponding to the most frequent meanings. Lexical category may be present, since many words can be used as different parts of speech, with a difference in meaning.

In the context of this project, word lists are useful for the following tasks:

- **selection of words**: which are the most frequent, or the more adequate words to use in an area, which is the minimum set for the natural expression of ideas in the area. This may and will usually include determining the lexical categories under which each word will be accepted, and definitions of each pair (word, PoS). A single definition per pair (word, PoS) is a desirable goal, since this tends to avoid ambiguity.

- **lemmatization**: determining the headform for a wordform. A wordform may be a headword or an inflected form of a headword. Definitions and lexical categories are usually attached to headwords, not wordforms. Texts are written using wordforms, not only headwords, hence the need for lemmatization.

- **PoS tagging**: determining the lexical category under which a wordform is being used in a text. This is required to build the syntactic tree of a sentence, a step previous to knowledge extraction as used in this project.
6.3. Lexical Resources

- definitions, though not directly used in the processing of texts written in the compiled lexicon, are of great help to writers and to practitioners, since it acts as a written agreement on the meaning of words as used in the area.

Lists may provide several of the former pieces of information in the same list, or even combine items, such as frequency by (headword, PoS). Owing to these differences, lists have to be processed on a one by one basis. To be used in text processing applications, lists had to be brought into a common format.

The short lists, compiled for a purpose, offer a good start point to compile a lexicon: adoption of a short list can save a lot of work and provide for basic communication in almost any area. Domain specific vocabularies can then be added, taking care not to introduce ambiguity when incorporating new meanings to existing words.

The long lists offer a good testing platform for coverage: they are usually compiled from large corpora of known origin and characteristics. The long lists are also useful to assimilate new words into the written text, such as new spellings of wordforms for headwords in the lexicon, or the PoS tagging of words not included in the lexicon but so frequent as to warrant inclusion in the domain specific lexicon.

### 6.3.2. The General Service List of English Words

The General Service List of English Words (GSL) was compiled in 1953 by Michal West. The purpose of the list was to gather the most useful words for learners of English. The list comprises about 2000 headwords, each corresponding to a “family” of wordforms, a concept only loosely defined in the original work. Though these words occur very frequently in written or spoken English, they are not necessarily the most frequent ones: they were chosen to be of maximum “expressiveness”, that is, to allow for the widest communication of ideas as possible in as straightforward a way as possible. GSL attempted to gather the most useful words for everyday situations or “general service” in foreign or second language learning [Bauman, 2002]. It can be said that the targeted words were those 2000 words with which most ordinary texts could be written or read.

The original work is a list of headwords alphabetically ordered. For each headword, the information given includes frequency of occurrence, inflections with their frequencies, frequency in a sense, and a short definition of each sense. Frequency in a sense is very uncommon in frequency word lists, and significantly hard to compile.

In 1995 John Bauman and Brent Culligan created a version of the GSL ranked by frequency, at the same time addressing some of the problems posed by the original list. A revision was carried out to group derived forms into group families, and frequency numbers from the Brown corpus were used [Bauman, 2002]. The Brown corpus is a now classic compilation of texts in a variety of genres by Francis and Kucera [Francis, W. N. and Kucera, H., 1979]. The Bauman version of the GSL list ended up with 2284 words [Bauman and Culligan, 2002].

Lextutor, a language resource site in Canada for learning, teaching and research in English and French, provides several word lists in its download page [Lextutor, 2011]. In the Lextutor site, GSL lists are available as English 1000 and English 2000; GSL 1000 provides the 1000 most common words, and GSL 2000 provides the next common 1000 words. Both lists are presented in two plain text files, one for families and one for heads; heads shows only headwords, and families include wordforms indented under each headword. AWL, the Academic Word List, is also available in text files heads and families. AWL is a set of 570 words commonly used
6. Lexical Resources

in a variety of academic texts. AWL was compiled by Averil Coxhead as a complement to
the 2000 words in the GSL; these additional words were selected for the use of teachers and
students in tertiary institutions, or learners wishing to improve their knowledge or English to
face tertiary studies \cite{Coxhead, 2000, Lextutor, 2011}.

Professor James Dickens of Salford University, U.K. \cite{Dickens, 2007a}, compiled an Extended
Version of A General Service List of English Words in electronic sheet format (MS Excel)
\cite{Dickens, 2007a}, with semantic field categories taken from the Longman Lexicon of Contemporary
English (LLCE) by Tom McArthur (Longman, 1981) \cite{McArthur, 2006}. Unfortunately,
neither the list nor the categories are available on the University of Salford site; copies of both
resources and Professor Dickens web page were downloaded 2007-07-26.

Leah Gilner and Franc Morales published several machine readable files for GSL and AWL
lists, one with a wordform per line, and another with a headword and its wordforms on each
line. Also available are files of word by part of speech for both GSL and AWL: adjectives,
adverbs, conjunctions, determiners, nouns, prepositions, pronouns and verbs. The lists were
compiled in 2005; minor corrections were added in 2006 and 2007 \cite{Gilner and Morales, 2007}.
The same authors produced lists for the English function words: auxiliary verbs, conjunctions,
determiners, prepositions, pronouns, quantifiers. Function words are considered closed
categories, hence this lists are expected to contain them all.

6.3.3. Simple English Wikipedia

The recommended vocabulary of Simple English Wikipedia is based on Ogden’s Basic English,
but is not so strict, and is not based on a single word list.

Wikipedia Basic English alphabetical word list, also called BE 850, is a list of the 850 words
used in standard Basic English \cite{Wikipedia, 2011a, Ogden’s Basic English, 2002}; this is the
primary list to be used when writing Simple English Wikipedia articles \cite{Wikipedia, 2011c}. If
a less common word is required, either for precision of meaning, or to avoid awkward language
expressions, the Simple English Wikipedia author is directed towards BE 1500 or VOA Special
English. BE 1500 is a short name for the Basic English Combined Wordlist. BE 1500 adds
to BE 850 five subordinate word lists, ending in more than 2600 words. A student mastering
BE 1500 is considerably more advanced towards standard English than a student mastering
only BE 850 \cite{Wikipedia, 2011b, Ogden’s Basic English, 2006}.

6.3.4. VOA Special English

Voice of America (VoA) Special English is a simplified version of English addressed to English
learners of intermediate to advanced level. Broadcast is done at slower speed, sentences are
short, and vocabulary is limited to about 1500 words. Some more specific words may be used as
subject demands, but the main vocabulary is fixed. The list was first published in 1962, but has
been updated following the evolution of the language \cite{Voice of America, 2011}. VOA Special
English word list is publicly available as a wordbook \cite{Voice of America, 2009}. The format
of the book is that of a simplified dictionary: a reduced part of speech tagset is used, words
are alphabetically ordered and each entry consists of headform, part of speech, definition and
optionally an example of use. For machine processing, this word list can be downloaded from
Manythings.org in HTML format \cite{Voice of America, 2010}. Manythings.org provides resources
for learners of English as a second or foreign language \cite{Kelly and Kelly, 2010}; several other
word lists are also available in the site.
6.3. Lexical Resources

6.3.5. Longman Communication 3000

The Longman Communication 3000 is a purely statistical selection of the 3000 most frequent words of the English language, compiled from the 390 million words contained in the Longman Corpus Network. This corpus gathers a number of “authentic English language” written texts and transcripts from spoken English. The list is considered useful to students of English: it claims to have a coverage of 86% of common use written or spoken English. Hence a student may concentrate on mastering this vocabulary to eventually be able to understand 86% of normal communication in the language. Each entry in the list comprises a headword, part of speech and one or two indicators of frequency. These frequency indicators may be W1, W2, W3 if the word is counted in the top 1000, 2000, or 3000 most common words in written English, and S1, S2, S3 if they are in the top 1000, 2000, or 3000 most common words in spoken English. When a headword can be related to more than one part of speech, the headword has as many entries in the list [Longman, 2008].

6.3.6. BNC Word Frequencies

The British National Corpus (BNC) is a 100 million word collection managed by the BNC Consortium, an industrial academic consortium lead by Oxford University Press and including several major dictionary publishers. Words are taken from both written (90%) and spoken (10%) samples from a wide variety of sources [BNC Consortium, 2007].

The Word Frequencies in Written and Spoken English is a word frequency compilation based on the British National Corpus, published as a book. The website companion for the book provides a number of word lists under a Creative Commons Attribution-Share Alike license: frequencies for the whole corpus, separate frequencies for spoken and written English, comparison in spoken English between conversational and task oriented speech, comparison in written English between imaginative and informative writing, and frequency lists by lexical (part of speech) categories [Leech et al., 2001].

6.3.7. COCA

The Corpus of Contemporary American English (COCA) is a 425 million word corpus of American English created by the Brigham Young University in 2008. It comprises both written and spoken English from a balanced variety of genres, and it is regularly updated [Brigham Young University, 2011a]. Word lists of 60,000 headwords are arranged in several formats. Two lists are free: top 5,000 headwords and top 500,000 wordforms with part of speech tagging [Brigham Young University, 2011b].

6.3.8. SCOWL, AGID and PoS DB

SCOWL, AGID and PoS Database are very long lists gathered together and packed by Kevin Atkinson in his Kevin’s Wordlist Page [Atkinson, 2011].

SCOWL (Spell Checker Oriented Word Lists, 2011) is a collection of word lists to be used in spell checkers. There are 10 word lists corresponding to different frequency levels, with a total of 652,475 words. Size 10 contains the most common words, and is 4,427 words large. Size 35 is recommended as small size, with 50,039 words; size 50 amounts to 97,304 words and is considered medium; size 70 is the large size, with 161,521 words, including all words in dictionaries. The rest of the lists include very uncommon words, or words not considered
6. Lexical Resources

words at all, but seen in texts. As indicated by the numbers, these are very large lists. Several spellings are available: American, British, Canadian, along with some variants.

AGID (Automatically Generated Inflection Database, 2003). Each line contains a headword, part of speech, and a list of inflected forms (wordforms for the headword). The author attempted to reach 100% accuracy in “non-questionable entries”.

The PoS Database (the Part Of Speech Database, 2000) gives a list of part of speech tags on each word. It combines information from Grady Ward’s Moby Project [Ward, 2000] and Wordnet [Princeton University, 2011].

6.3.9. Wordnet

Wordnet is a lexical database which organizes words around concepts or “synsets”. A synset is a data structure used to describe a single concept. A synset is associated with several words usually recognized as referring to this concept; these words are considered synonyms, since they can be exchanged in a context with no alteration of meaning. The name synset suggests “set of synonyms”, which is the form a concept is characterized in the Wordnet. The same word may refer to different concepts (polysemy), so this same word will be present in different synsets. A part of speech is included in each synset, describing the grammatical function played by the words in the synset. Each of the words in a synset is called a lemma.

Table 6.1 shows synsets and lemmas for word ‘salt’, taken from the NLTK Wordnet corpus [Bird et al., 2011]. A synset is identified by a string such as ‘salt.n.02’, formed by the first lemma in the synset, a part of speech tag, and an ordinal sense number; synset identifiers are thus unique. Lemmas for this synset are identified by a string such as ‘salt.n.02.salt’ or ‘salt.n.02.common_salt’, formed with the synset identifier to which this lemma belongs, plus a lemma name. As can be seen, the same word ‘salt’ leads to several synsets, corresponding to different concepts evoked by the word ‘salt’, including an acronym. Furthermore, the same synset comprises several lemmas. Lemmas in the same synset differ in the lemma name; lemmas in different synsets differ in their synset identifier part; in this way, lemma identifiers are also unique. Three different syntactic roles may be played by the word ‘salt’: noun (‘n’), verb (‘v’) and adjective (‘s’ or ‘a’). Also shown in table 6.1 are frequency counts, the number of times the word appeared in this sense in a tagged corpus.

Each synset may also contain a definition or gloss, and optionally examples:

```plaintext
Synset('salt.v.04')
  definition: preserve with salt
  examples: ['people used to salt meats on ships']
  lemma('salt.v.04.salt'); frequency count: 0
```

Wordnet identifies a number of different relationships among synsets, resulting in a hierarchy of concepts, containment relations (is part of), and others. Most of these relations offer very interesting possibilities to approximate a semantic interpretation of a text. For example, a student’s answer may not succeed in remembering “pemmican” as “lean dried meat pounded fine and mixed with melted fat; used especially by North American Indians”, but may say “meat” instead; “meat” is an hypernym of “pemmican”, which means “pemmican” is “a kind of meat”. The answer may be not fully correct, but indicates some knowledge on the part of the student.

Wordnet may be of help for the compilation of a lexicon at least in these ways:
Synset('salt.n.01')
  Lemma('salt.n.01.salt'); count: 9
Synset('salt.n.02')
  Lemma('salt.n.02.salt'); count: 5
  Lemma('salt.n.02.table_salt'); frequency count: 0
  Lemma('salt.n.02.common_salt'); frequency count: 0
Synset('strategic_arms_limitation_talks.n.01')
  Lemma('strategic_arms_limitation_talks.n.01.Strategic_Arms_Limitation_Talks'); count: 0
  Lemma('strategic_arms_limitation_talks.n.01.SALT'); count: 0
Synset('salt.n.04')
  Lemma('salt.n.04.salt'); count: 0
  Lemma('salt.n.04.saltiness'); count: 0
  Lemma('salt.n.04.salinity'); count: 0
Synset('salt.v.01')
  Lemma('salt.v.01.salt'); count: 1
Synset('salt.v.02')
  Lemma('salt.v.02.salt'); count: 0
Synset('salt.v.03')
  Lemma('salt.v.03.salt'); count: 0
Synset('salt.v.04')
  Lemma('salt.v.04.salt'); count: 0
Synset('salt.s.01')
  Lemma('salt.s.01.salt'); count: 1

Table 6.1.: Synsets and lemmas for word 'salt'

- by providing online help to writers, who may want to see the definitions of the synsets to which a word belongs, or other lemmas (synonyms) which may better suit their needs.
- restrict the meanings of the words in a lexicon to only those needed in the area. In an ideal case, each word would evoke only one meaning. Though this may be difficult to achieve, cutting down the number of senses associated to a word is an effective way to fight lexical ambiguity.
- compile a lexicon “from meaning to words”, first identifying all the concepts (synsets) needed in an area and one of their lemmas as a canonical word for the synset. A further step may identify synonyms for the canonical word.
- help reach the ideal lexicon: only and all the synsets needed for the clear expression of ideas in an area of knowledge, only one canonical word for each synset, not a word pointing to more than one synset. This does not exclude synonyms: a synset may have other lemmas besides the canonical lemma, if each included lemma belongs only in this synset.

Except for the first one, namely providing writers with definitions and synonyms at their will, the rest of this wish list is no easy task. Though this direction will not be further pursued in this work, some testing was done, as summarized at the end of the following section.

6.4. Evaluation and Comparison

Almost all the word lists examined showed a certain amount of anomalies, either on inspection or on machine processing. Lists were read into a common format for comparison, with only a
6. Lexical Resources

minimal amount of correction done, trying to preserve the original as much as possible.

In the analysis of these lists, only information useful for this project was taken into account, and only when the lists could be read without major changes in format. Lists were read from text files, PDF files, HTML pages, and electronic sheet files. Word counts as presented in the comparison tables may differ from counts in the original lists, owing to some minor corrections, mostly suppression of repeated, invalid or incomplete entries. A log of tests performed on these lists and detailed notes on the anomalies detected and corrections made, may be seen in the site of this thesis [González-Barbone, 2012].

6.4.1. The GSL versions

The General Service List of English Words (GSL) was originally compiled for students of English as a second language, with the purpose of identifying the minimum set of words with maximum coverage and expressiveness in the common use of the English language.

The GSL is a long standing work, dating back to 1953. In time, revisions, updates and conversion to electronic formats have led to “versions” of the original GSL, including different information, with differences even on the words included. Moreover, these differences are not always easily traceable to the original list.

The GSL version most informative and closest to the original work is Professor Dickins electronic worksheet: it includes frequencies, inflections, senses, frequency in a sense, definitions and examples. Though in electronic format, the list seems to have been thought for human look up rather than machine processing: there are commentaries in some entries, interrogation marks, alternatives (e.g. “holiday/s”, “though/although”), or reference to other entries (e.g. “other ... see ONE”, “one... see ONE”). Some cells contain abnormal values (e.g. “Err:5082, “5000000”, “36e”, “=fillin!the!form”). Part of speech tags are recorded for human reading, with no clear criterion, and some are probably misspelled (e.g. ’v & n’, ’n& v’, ’noun substittue’, ’adjher’) [Dickins, 2007a] [Dickins, 2007b]. While many of these errors can be easily corrected, manual revision is required. It is not so easy to clarify part of speech tagging, and it may be difficult to correct frequency counts and percents. With all these drawbacks, the resource remains a valuable one, the only one to include frequency in a sense. Correction of this list with the help of linguists would be a valuable effort.

The Bauman and Culligan revision of GSL is a text list of three columns: an ordinal number, a frequency count, and the word. The list is ordered on frequency count, from the most frequent (“1 69975 the”) to the last in the list (“2284 0 plural”) [Bauman and Culligan, 2002]. Only two entries were detected as abnormal, “14 8516 I” and “1740 29 FALSE”; turning the words to lowercase were the obvious corrections.

The GSL lists in Lextutor are organized in three pairs of text files: the 1000 most frequent words presented as families and heads (files 1000_families.txt and 1000_heads.txt), the next frequent 1000 words (files 2000_families.txt and 2000_heads.txt), and the AWL words (files awl_families.txt and awl_heads.txt). The lists are in text format lines with tab separated fields; words are in uppercase [Lextutor, 2011]. The GSL lists found in Lextutor have some misplaced tabs, but were successfully read into our data structures. In the GSL comparison table, the Lextutor lists include the AWL list.

The GSL lists compiled by Gilner and Morales are the best GSL resource found, both for content and consistency. Moreover, the files are all machine readable with no format errors [Gilner and Morales, 2007]. In the GSL comparison table, the Gilner list includes AWL words, which partly explains its greater size. The lists are usable for lemmatization and PoS tagging.
6.4. Evaluation and Comparison

Some small errors were detected and corrected: ‘I’ was in uppercase in one list when all words are in lowercase in every list, and 5 words were repeated in GSL and AWL lists (‘framework’, ‘medical’, ‘network’, ‘mechanism’, ‘partner’).

Table 6.2 summarizes the information provided by each version of the GSL lists. The most informative is GSL Dickins, only lacking in wordforms. Though not recorded in the table, it is the only GSL list to provide frequency in a sense information. Next comes GSL Gilner, which does not inform on frequencies nor provides definitions, but may be used to lemmatize and PoS tag. GSL Lextutor allows for lemmatization only, and GSL Bauman informs on frequencies only.

<table>
<thead>
<tr>
<th>Feature</th>
<th>GSL Dickeins</th>
<th>GSL Gilner</th>
<th>GSL Lextutor</th>
<th>GSL Bauman</th>
</tr>
</thead>
<tbody>
<tr>
<td>wordform</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>headword</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>PoS</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>frequency by headword</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>frequency by wordform</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>definition</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>size (headwords)</td>
<td>2555</td>
<td>2284</td>
<td>1702</td>
<td>2849</td>
</tr>
<tr>
<td>size (wordforms)</td>
<td>10932</td>
<td>-</td>
<td>-</td>
<td>10984</td>
</tr>
</tbody>
</table>

Table 6.2.: Word lists information: the GSL versions.

Though the most informative, the formerly noted flaws in GSL Dickins count against its adoption: a lot of correction is needed. GSL Gilner comes next in information provided, besides its being the most consistent and machine readable. It is also the longest list for more than 300 words over the next competitor. A comparison of GSL Gilner against the other versions shows that:

- there are 2 headwords in GSL Bauman, 30 headwords in GSL Dickins and 88 headwords in GSL Lextutor which are not present in GSL Gilner.
- there are 115 headwords in any of the other GSL version and not in GSL Gilner.
- notably, GSL Gilner lacks days of the week and months of the year, 19 of the 115 headwords lacking.
- when excluding days of the week, months of the year, written numbers, letters of the alphabet, words written together and found in GSL Gilner when separated, foreign words, and misspelled words, the list of 115 words not included in GSL Gilner boils down to 30 words.

Days of the week and months of the year seems a sensible inclusion: no major alteration of the original spirit of the list is expected by this addition. Therefore, adoption of GSL Gilner might be complemented by some decision on the inclusion of the following 30 words:

- absolutely, accountable, apologise, apologize, barely, billion, burial, centimetre, cheque, gram, in-law, informal, kilometre, litre, metre, milligram, millilitre, millimetre, miner, mister, mrs, nonsense, numerical, offence, provision, rat, really, saws, steal, typical
6. Lexical Resources

In spite of its age and several questionings [Bauman, 2002] [Gilner, 2011], the GSL list remains of value to date, mainly because it was compiled considering sense and attempting at expressiveness. Modern studies led by Gilner and Morales showed an estimated coverage of 70% to 90%, whatever the texts [Gilner, 2011].

Though revisions and time have led the original GSL list to diverge a bit, careful consideration should be given to modifications of a list compiled with semantics in mind, as is GSL. In some domains, inclusion of the 8 words naming units of measure might be left aside in favour of the inclusion of a set of units of measure adequate to the domain. A word like 'mister' might be considered old fashioned and 'mr' proposed instead. Other words might have synonyms of current use and render unnecessary their inclusion. Again, the GSL is a list compiled with a purpose of general usefulness, and modifications should be left to linguists.

For practical use, GSL Gilner is recommended over the other GSL versions, with the modifications formerly stated.

6.4.2. The other short lists

**BE1500.** Simple English Wikipedia Basic English Combined Wordlist, also called BE1500 [Wikipedia, 2011l] contains about 2,000 headwords classified in several categories. BE1500 extends Ogden’s Basic English original word lists. As GSL, Basic English is an attempt to identify the most necessary words for learners of English. Also like GSL, it is a long standing enterprise. BE1500 contains only headwords, with no additional information, but is considered a valuable resource owing to its purpose and origin. There are 28 words with non alphabetic characters, which include 15 with a hyphen: 'centi-', 'dancing(to)', 'deci-', 'dressing(up)', 'ear-ring', 'fire-engine', 'first-rate', 'good-morning', 'gun-carriage', 'kilo-', 'laughing(at)', 'looking(up)', 'looking-glass', 'micro-', 'milli-', 'pleased(with)', 'pointing(at)', 'talking(of)', 'touching(up)', 'turning(over)', 'twenty-one', 'used(to)', 'well-being', 'well-off', 'working(on)', 'working(out)', 'working(up)', 'x-ray'. The number is small, and corrections simple, once a criterion is defined.

**VoA SE.** Voice of America (VoA) Special English (SE) word list contains about 1,500 headwords with part of speech tag and definition [Voice of America, 2010]. This list is somewhat smaller than GSL and BE1500, but its purpose and intended audience are similar, and has been updated regularly. There are 5 words capitalized: 'Congress', 'I', 'Internet', 'Senate', 'Web site'; these words may be kept as such or converted to lowercase, depending on the intended use. The following 8 words contain non alphabetic characters, and require some decision: 'Web site', 'a (an)', 'air force', 'case (court)', 'case (medical)', 'civil rights', 'seek(ing)', 'swear in'.

**Longman 3000** is a short list, but unlike the preceding ones, it is based only on statistics. It is compiled from a very large corpus of carefully selected items, provides part of speech tagging and a level of frequency (top 1000, 2000 or 3000 more used words) in both written and spoken English.

There are 5 words in capitals: 'CD', 'DVD', 'I', 'OK', 'TV'; there are 17 words with non alphabetic characters: 'according_to', 'all_right', 'each_other', 'good_morning', 'good_night', 'ice_cream', 'long-term', 'mobile_phone', 'no_one', 'no_way', 'of_course', 'one_another', 'ought_to', 'o'clock', 'post_office', 'so-called', 'used_to'. Words with '_' result from replacing a space, e.g. 'all_right' was transformed into 'all_right', to preserve the compound word when tokenizing.

102
6.4. Evaluation and Comparison

6.4.3. Comparison of the short lists.

Even though all the short lists require some manual correction to be used in a machine processable lexicon, the number of words involved is manageable, and in most cases corrections are simple to decide.

Table 6.3 compares the short lists: GSL Gilner version, BE 1500, VoA Special English and Longman 3000.

<table>
<thead>
<tr>
<th>Feature</th>
<th>GSL Gilner</th>
<th>BE 1500</th>
<th>VoA SE</th>
<th>Longman 3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>wordform</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>headword</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>PoS</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>frequency by headword</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>level</td>
</tr>
<tr>
<td>frequency by wordform</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>definition</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>size (headwords)</td>
<td>2849</td>
<td>2021</td>
<td>1510</td>
<td>3159</td>
</tr>
<tr>
<td>size (wordforms)</td>
<td>10984</td>
<td>-</td>
<td>12081</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.3: Word lists information: the short lists.

Table 6.4 shows inclusion of words in the short lists into one another. When read by row, figures on cells show which percent of list in this row is contained in the other lists. When read by column, figures on cells show which percent of words in the other lists are included in list in this column. Examining row GSL Gilner, 42% of GSL Gilner words are in BE1500, 43% are in VoA SE, and 73% are in Longman 3000. Examining column GSL Gilner, this list includes 59% of BE1500, 81% of VoA SE and 66% of Longman 3000.

Though the different sizes of the compared lists distort the appreciation, the low percents of superposition stand out: if all lists claim to gather the most frequent words, a closer coincidence was naturally expected. VoA SE is the shortest list; a high coincidence is expected, and indeed found: 81% of VoA SE is included in GSL Gilner, and 84% in Longman 3000, but only 52% of its words are included in BE1500. Next in size is BE1500, with 60% included in GSL Gilner and 61% in Longman. These figures seem low, since BE1500 size is 70% of GSL Gilner, and 64% of Longman; a significant number of words in BE1500 are not in the other lists. GSL Gilner is only 10% shorter than Longman, and percent included amounts to 73%; this means 2098 of its 2849 words are included in Longman and 751 are left out.

These numbers suggest BE1500 diverges from both GSL Gilner and Longman more than GSL Gilner diverges from Longman.

A word of caution is in order when considering these numbers: as already stated, the comparisons, and the lists themselves, have been machine processed and machine read in a variety of formats from sources prepared mainly for human reading and interpretation; only minimal, obvious corrections were made on the original lists. For example, in BE1500 there are words such as ‘purr’, ‘bloodvessel’, ‘ear-ring’, ‘well-being’, ‘touching(up)’, ‘dancing(to)’, ‘centi-’, ‘postoffice’, ‘undercooked’, ‘gun-carriage’, ‘stoppingup’, which may not be easily found in other lists.
6. Lexical Resources

<table>
<thead>
<tr>
<th>Word List (size)</th>
<th>GSL Gilner</th>
<th>BE1500</th>
<th>VoA SE</th>
<th>Longman 3000</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSL Gilner (2849 words)</td>
<td>100%</td>
<td>42%</td>
<td>43%</td>
<td>73%</td>
</tr>
<tr>
<td>BE1500 (2921 words)</td>
<td>60%</td>
<td>100%</td>
<td>39%</td>
<td>61%</td>
</tr>
<tr>
<td>VoA SE (1510 words)</td>
<td>81%</td>
<td>52%</td>
<td>100%</td>
<td>84%</td>
</tr>
<tr>
<td>Longman 3000 (3159 words)</td>
<td>66%</td>
<td>39%</td>
<td>40%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Table 6.4.: Short lists inclusion: percent of words on list on row contained in list on column

6.4.4. The long lists

BNC. The companion site for the book *Word Frequencies in Written and Spoken English* compiled from the British National Corpus offers lists of words separated by part of speech. The lists for nouns, verbs and adjectives are lemmatized, the lists for adverbs and pronouns are not. Since adverbs in English do not vary much, this may be only a slight limitation. A comparison of words taken from frequency lists against words coming from the PoS lists show some differences, with a number of words in one list and not in the other, in both directions. From the PoS lists, there are 3012 nouns, 1110 verbs, 1022 adjectives, 380 adverbs and 208 words in closed categories (pronouns, determiners, prepositions, conjunctions and interjections). Since a number of words count in different PoS categories, a total of 5050 headwords results.

COCA (Corpus of Contemporary American English) free lists are 5,000 lemmas and 500,000 wordforms with PoS. The 5,000 lemmas lists contains a few lines with repeated rank and no frequency value. Since lemmatization is a required step for PoS tagging, the list of PoS by wordform is not very useful. The other lists in COCA are commercial. There are 65 words not in lowercase, 19 words with non alphabetic characters, of which 16 contain a hyphen.

The SCOWL, AGID and PoS lists found in Kevin’s Word Lists Page are all three very long lists [Atkinson, 2011].

SCOWL arranges words in categories or levels of frequency: level 10, the most common words, is already quite large, with 4427 words, more than any of the short lists. This list was not used; frequency information in categories is too coarse grained for word selection.

AGID lists contain the inflected forms of its words, which provides a useful resource for lemmatization, particularly powerful because of its large size. The PoS information given in AGID is limited to only three categories: verb, noun and adverb or adjective. There are 257,907 words in this list; 13,000 words are not in lowercase, 9 include non alphabetic characters, all of which contain a hyphen. As an example, both ‘caterpillar’ and ‘Caterpillar’ are in the list, as well as their corresponding inflections ‘caterpillars’, ‘Caterpillars’; preserving case may give a hint of sense, indicating whether a larva, a propelling system, a vehicle or a trademark is meant.

The PoS Database from Moby and Wordnet may be somewhat difficult to manage owing to the duality of the sources: in many entries, tags from the Moby database are given first, then a | separator and tags from the Wordnet; some criterion must be adopted for its use. There are 295,172 words in this list; 53,682 are not in lowercase, 89,017 contain non alphabetic characters, of which 14059 are hyphenated.
6.4. Evaluation and Comparison

6.4.5. Comparison of the long lists

Table 6.5 summarizes information provided by the long lists; GSL Gilner is also shown for comparison. AGID and PoS DB are really long lists; the difference in headword count is significant, though. Some words that are in PoS DB and not in AGID are: 'unsupportable', 'unattackable', 'nonreceivable', 'chloroguanide', 'Glackens', 'utnapishtim', 'alible', 'woods', 'clotted', 'Tillford', 'fabianism', 'Nampa', 'Culosio', 'Fabron', 'cardiospermum', 'mantatological', 'localized', 'houyhnhnm', 'beadsmen', 'taraktagenos'. Though far fewer, there are also a few words in AGID not in PoS DB, such as: 'clotter', 'Elopidae', 'slepe', 'Cosmoline', 'Zairese', 'bioethic', 'unessence', 'pawnor', 'phenomenologist', 'yikker', 'bromelain', 'trawling', 'overissuance', 'Haliotis', 'optophone', 'hypate', 'deducement', 'microdensitometer', 'egotize', 'Euro'. As for these samples, the lists differ in rather uncommon words.

<table>
<thead>
<tr>
<th>Feature</th>
<th>GSL Gilner</th>
<th>BNC</th>
<th>COCA</th>
<th>AGID</th>
<th>PoS DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>wordform</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>headword</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>PoS</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>n.v.a</td>
<td>yes</td>
</tr>
<tr>
<td>frequency by headword</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>frequency by wordform</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>definition</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>size (headwords)</td>
<td>2849</td>
<td>5050</td>
<td>4353</td>
<td>103938</td>
<td>295172</td>
</tr>
<tr>
<td>size (wordforms)</td>
<td>10984</td>
<td>-</td>
<td>-</td>
<td>257907</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.5.: Word lists information: the long lists.

Table 6.6 shows inclusion of the short lists in the long lists. BNC and COCA, though relatively short, contain more than 80% of the words in GSL Gilner, VoA SE and Longman; the exception is, again, BE1500, with 70% and 67% included in BNC and COCA. Both AGID and PoS DB contain more than 90% for all lists, BE1500 included.

There are 101 words in GSL Gilner not in AGID, but only 4 not in PoS DB: 'so-called', 'old-fashioned', 'maximise', 'criteria'. However, ‘maximize’ and ‘criterion’ can be found in PoS DB.

The discussion held so far provides some guidance on the adoption of a short list for a core lexicon, and on the use of other lists to validate the lexicon or add to it.

<table>
<thead>
<tr>
<th>Short lists / Long lists</th>
<th>BNC</th>
<th>COCA</th>
<th>AGID</th>
<th>PoS DB</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSL Gilner (2849 words)</td>
<td>87%</td>
<td>81%</td>
<td>96%</td>
<td>99%</td>
</tr>
<tr>
<td>VoA SE (1510 words)</td>
<td>91%</td>
<td>91%</td>
<td>96%</td>
<td>99%</td>
</tr>
<tr>
<td>BE1500 (2021 words)</td>
<td>71%</td>
<td>68%</td>
<td>92%</td>
<td>98%</td>
</tr>
<tr>
<td>Longman 3000 (3159 words)</td>
<td>94%</td>
<td>90%</td>
<td>92%</td>
<td>99%</td>
</tr>
</tbody>
</table>

Table 6.6.: Short word lists inclusion in long lists.
6. Lexical Resources

<table>
<thead>
<tr>
<th>Short lists / Long lists</th>
<th>BNC</th>
<th>COCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSL Gilner (2849 words)</td>
<td>87%</td>
<td>86%</td>
</tr>
<tr>
<td>VoA SE (1510 words)</td>
<td>74.6%</td>
<td>77.0%</td>
</tr>
<tr>
<td>BE1500 (2021 words)</td>
<td>71.2%</td>
<td>71.3%</td>
</tr>
<tr>
<td>Longman 3000 (3159 words)</td>
<td>93.6%</td>
<td>94.4%</td>
</tr>
</tbody>
</table>

Table 6.7.: Coverage of the short lists referred to the long lists

6.4.6. Coverage of the short lists in the long lists

Coverage is a measure of the capacity of a word list to account for the occurrences of words in a text. This means that if a word in the list occurs 5 times in the text, it is counted as 5. Coverage is determined by summing all the times a word in the list appears in the text, dividing by the length of the text in words. This can be expressed as a percentage. Conceptually, coverage is a measure of how much of the text the list is able to “understand”. This is the reason why a word in the list is counted as many times as it occurs: if the word is in the list, i.e. it is “understood”, then it is “understood” as many times as it appears in the text.

Coverage can also be calculated against a list of word frequencies, summing the frequencies for all the words in the reference list. The list of word frequencies has been compiled from a usually very large corpus. The coverage calculated against the list of word frequencies gives a measure of how much of the original corpus might be “understood” by the list in question. More precisely, a person mastering the vocabulary in a word list is expected to understand a fraction of the text or corpus equal to the coverage.

Table 6.7 shows coverage of the short lists against the long lists which provide frequency counts, namely BNC and COCA. As can be seen, for each of the short lists coverage in BNC or COCA is almost equal. Longman 3000 is the list with higher coverage, 94%. A 7% below is GSL Gilner with 87%. VoA SE and BE1500 follow, with 76% and 71%, a 10 to 13% below GSL Gilner.

These numbers confirm the claim of Longman 3000 [Longman, 2008]: a person knowing its vocabulary can be said to understand the language, a condition recognized when 95% of the words are known (assuming grammatical constructions are also understood) [Nation and Waring, 1997]. GSL Gilner, a shorter list, reaches 87%, as claimed by Gilner and Morales [Gilner, 2011].

6.4.7. Wordnet in numbers

The following data were obtained by testing the nltk.corpus.reader.wordnet module from NLTK version 2.0b9 [Bird et al., 2011].

There are 117659 different synsets in the Wordnet, leading to 148730 different lemmas, which accounts for 1.26 lemmas per synset.

Senses are ordered by frequency of use, with the most frequent senses coming first. Hence, a low ordering number indicates more frequent use of this sense. Not all senses are semantically tagged, though; those senses not tagged come last. Each lemma is also provided with a frequency count, determined by the number of times this lemma is tagged in a small corpus of concordance tests.
6.4. Evaluation and Comparison

<table>
<thead>
<tr>
<th>Synset</th>
<th>Lemma name</th>
<th>Count</th>
<th>Order</th>
<th>COCA</th>
<th>BNC</th>
</tr>
</thead>
<tbody>
<tr>
<td>person.n.01</td>
<td>person</td>
<td>6833</td>
<td>1</td>
<td>361</td>
<td>335</td>
</tr>
<tr>
<td>not.r.01</td>
<td>not</td>
<td>1837</td>
<td>2</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>location.n.01</td>
<td>location</td>
<td>992</td>
<td>3</td>
<td>1409</td>
<td>1687</td>
</tr>
<tr>
<td>look.v.02</td>
<td>seem</td>
<td>607</td>
<td>4</td>
<td>175</td>
<td>142</td>
</tr>
<tr>
<td>two.s.01</td>
<td>two</td>
<td>508</td>
<td>5</td>
<td>84</td>
<td>0</td>
</tr>
<tr>
<td>besides.r.02</td>
<td>also</td>
<td>411</td>
<td>6</td>
<td>92</td>
<td>81</td>
</tr>
<tr>
<td>get_down.v.07</td>
<td>begin</td>
<td>375</td>
<td>7</td>
<td>178</td>
<td>210</td>
</tr>
<tr>
<td>many.a.01</td>
<td>many</td>
<td>347</td>
<td>8</td>
<td>105</td>
<td>105</td>
</tr>
<tr>
<td>become.v.01</td>
<td>become</td>
<td>327</td>
<td>9</td>
<td>150</td>
<td>129</td>
</tr>
<tr>
<td>own.s.01</td>
<td>own</td>
<td>259</td>
<td>10most</td>
<td>probably</td>
<td>138</td>
</tr>
<tr>
<td>never.r.01</td>
<td>never</td>
<td>255</td>
<td>11</td>
<td>148</td>
<td>171</td>
</tr>
<tr>
<td>again.r.01</td>
<td>again</td>
<td>249</td>
<td>12</td>
<td>184</td>
<td>169</td>
</tr>
<tr>
<td>three.s.01</td>
<td>three</td>
<td>221</td>
<td>13</td>
<td>141</td>
<td>0</td>
</tr>
<tr>
<td>always.r.01</td>
<td>always</td>
<td>196</td>
<td>14</td>
<td>213</td>
<td>203</td>
</tr>
<tr>
<td>nowadays.r.01</td>
<td>now</td>
<td>182</td>
<td>15</td>
<td>70</td>
<td>72</td>
</tr>
<tr>
<td>there.r.01</td>
<td>there</td>
<td>181</td>
<td>16</td>
<td>35</td>
<td>116</td>
</tr>
<tr>
<td>excessively.r.01</td>
<td>too</td>
<td>180</td>
<td>17</td>
<td>137</td>
<td>122</td>
</tr>
<tr>
<td>however.r.01</td>
<td>however</td>
<td>180</td>
<td>18</td>
<td>276</td>
<td>155</td>
</tr>
<tr>
<td>about.r.07</td>
<td>almost</td>
<td>175</td>
<td>19</td>
<td>309</td>
<td>300</td>
</tr>
<tr>
<td>ask.v.02</td>
<td>ask</td>
<td>165</td>
<td>20</td>
<td>134</td>
<td>150</td>
</tr>
</tbody>
</table>

Table 6.8.: Wordnet 20 first words. Columns: Wordnet Synset and Lemma name, count for lemma, order in Wordnet, order in COCA, order in BNC.

Of the 117659 synsets in Wordnet, only 27190 have a nonzero frequency count, 23.1% of total synsets.

Counting on lemmas, of the 148730, 16361 have a nonzero frequency count, an 11.0% of total lemmas. This is as expected, since a synset may be described by several lemmas.

Since a count of 0 may indicate lack of information, or a very low frequency of use, further tests left aside synsets and lemmas with zero count.

Table 6.8 shows the first 20 synsets obtained from the first 20 lemmas with higher frequency count in the Wordnet. The columns show the synset name, the lemma name which lead to the synset, the frequency count of the lemma, the ordering number of the word in Wordnet by frequency, and the ordering by frequency in COCA and BNC lists of the corresponding wordforms. A 0 in any of these last two columns indicates the word is absent from the list. The frequency ordering in COCA and BNC differs widely from the order in Wordnet; the set of “most frequent” words in the Wordnet cannot be taken as the most frequently used words in the language.

To verify the former statement, let us compare the first 1000 most frequent lemmas according to this criterion from the Wordnet against the 1000 most frequent headwords in COCA and BNC. Of these 1000 Wordnet lemmas only 288 and 292 are contained in the 1000 most frequent headwords of BNC and COCA, respectively; this is less than 30%. However, BNC and COCA share 827 headwords of the 1000 most frequent words in each list, which is more than 80%.

From the former results, no inference may be made as to frequency of use from Wordnet.
6. Lexical Resources

counts in lemmas. The value of counts is probably more significant within synsets, to identify the most frequent lemmas associated with this synset. In the next chapter, the Wordnet is explored as a resource to complement a list of commonly used words, namely GSL Gilner, our preferred choice of general service words.

6.5. Conclusions

This chapter made the following contributions:

- An identification of resources freely available, an evaluation of their usability, and tasks they can do. Checks for internal consistency, detection of errors, correction of some simple errors without altering the essentials of the original lists (chapter 6, chapter 14, software tools and tests available at [González-Barbome, 2012]).

- A comparison among available lexical resources. Comparison of the short lists of common words for superposition, coverage of short lists in the long lists, what to do to improve or make them usable.

The compilation of a word list is subject to a number of non trivial difficulties, starting with purely linguistic ones such as what is considered a word, what word is a headword or a word-form, when a multi token word should be considered a single word [Nation and Waring, 1997]. These difficulties lead to some degree of divergence, even among the most carefully compiled lists. Formatting problems, though seemingly trivial, may be difficult to avoid completely. Most of the examined lists showed some anomalies on examination or processing; only the most trivial ones were corrected, either manually or mechanically, to avoid altering the lists as they were originally conceived. GSL Gilner was selected as the most reliable version of GSL; the Academic Word List was included as a rule.

Any of the short lists examined successfully identifies a set of words with wide coverage of the the English language. A learner of English will do better if she is led to concentrate in first mastering the vocabulary of any of these lists, since she will achieve a reasonably wide understanding with as few words as possible. Though Longman 3000 exhibits the best coverage, it is a purely statistically compiled list: coverage alone may not be enough for a “general service” purpose, leaving aside words needed for ordinary communication not reflected in the highest frequencies. GSL, BE1500 and VoA SE were compiled with a “general purpose” in mind, and specifically oriented towards language learning or plain communication. The Gilner version of GSL showed to be the list with better coverage of these three. Though purpose or the targeted area of knowledge may justify a preference for BE1500 or VoA SE, GSL Gilner emerges as the most commendable short list.

Concerning the long lists, AGID and PoS DB are indeed very long lists, and may be of use in the processing of any texts, with no previous selection of source or genre. All the four short lists are almost completely contained in these long lists. The AGID list may be used for lemmatization; the PoS DB list may be used for PoS tagging. For frequency studies, BNC and COCA provide frequency counts for their words. The four short lists submitted to coverage analysis based on these two lists showed an outstanding agreement, which contributes some confidence to the procedure and results.
7. Compilation of a Lexicon

Abstract. Though a lexicon may be compiled from an existing, domain specific corpus, wordlists of English can be used as a core lexicon to which domain specific vocabulary can be added. To support the compilation of a lexicon, a framework of tools and data structures was developed, supporting operations such as test for inclusion, lemmatization, PoS (part of speech) tagging, frequency of use, definitions and synset mapping. The GSL Gilner wordlist was selected as a base lexicon. The sublanguage this lexicon is intended for will be limited to simple declarative sentences in the active voice, with no relative clauses, and no pronouns to avoid anaphora. This determined the set of function words to be used. Content words were taken from the GSL Gilner wordlist, excluding function words to avoid overlapping. Several self consistency checks were performed on the list, some commonly used words not in GSL Gilner were added, some words with several PoS tags were selected according to frequency or left out. Issues like several PoS per word, inflections, proper nouns, dates and numbers, were studied, and some ways to deal with them were proposed. Sense can be added by mapping content headwords from the lexicon to synsets in the Wordnet. Some tests on dealing with several synsets per headword were done, based on the order labels of the synsets. The lexicon was finally tested for coverage against the Brown corpus, reaching from 86% to 91% coverage according to genres. An experimental lexicon based on Longman Communication 3000 produced similar results. These were general lexicons, submitted to a wide spectrum of sources and genres; a domain specific lexicon against a corpus of the same domain is naturally expected to perform much better. The resources used, the data structures developed, and the methodology employed showed a reasonable base lexicon could be assembled with a moderate effort.

The purpose of this chapter is to compile a lexicon of general purpose to be used in a sublanguage apt for knowledge representation. Such a lexicon must be short, but exhibit wide coverage of common written texts, and allow for natural writing. Once built, this lexicon can be adapted to different domains by the addition of domain specific vocabulary. In this chapter, a base lexicon of less than 3000 words is compiled from the General Service List of English Words, and validated by testing coverage against the Brown Corpus of texts. From the tasks performed, a methodology for the compilation and testing of restricted lexicons is inferred.

7.1. Introduction

A domain specific lexicon can be compiled by adopting an existing word list and complementing it with domain specific vocabulary. A carefully compiled lexicon provides a solid foundation for any area of knowledge, giving for each word the precise meaning as it is understood in the field. Subject areas usually have vocabularies defining specifically the terms
7. Compilation of a Lexicon

employed in their practice, taking for granted the meaning of “general use” words. A domain specific sublanguage must include both kinds of words: general purpose and subject specific.

This chapter describes the compilation of a lexicon of limited size and general purpose use, to serve as a basis for a domain specific sublanguage. Existing word lists provide a core lexicon, to which domain specific terms can then be added. If this core lexicon is well chosen, it may be used as a common start point for the compilation of lexicons specific to different domains; the differences among these lexicons will be essentially domain specific terms.

7.2. Approach

A domain specific lexicon may be compiled from a corpus of texts of the target domain, carefully selected and large enough to be considered “representative”. The texts can be tokenized into a list of wordforms using existing tokenizers such as those provided in the NLTK toolkit [Bird et al., 2011]. Wordforms can then be mapped to headwords using an inflections database such as AGID [Atkinson, 2011]. PoS (part of speech) tagging can be inferred directly from the texts using PoS taggers, with a variable margin of error requiring manual correction, or else include all possible PoS for each headword as recorded in lexicons such as COCA [Brigham Young University, 2011a], BNC [BNC Consortium, 2007] or PoS DB [Atkinson, 2011]. A list of all wordforms for each of the headwords selected may be obtained as a convenience, again using AGID. Processing of texts in the corpus also allows for frequency counts; based on frequency, the lexicon can be compiled to include as many words as necessary to cover a predefined percent of the texts, such as 95%, a value generally accepted as enough to understand a language.

This approach has the obvious advantage of being strictly domain specific, drastically reducing the needs for domain specific terms addition. Several difficulties arise to blur such a promising picture: it may be difficult to obtain or gather a large enough, representative corpus; tokenization, PoS tagging and other NLP operations on texts are far from foolproof, and a number of anomalies is to be expected. Manual correction becomes a must, and may be time consuming.

There are other reasons not to follow this analytical approach in this project. If the corpus is based on texts from experts, the vocabulary obtained may be deprived of correct, everyday common words perfectly acceptable in a student’s writing. On the other hand, including students’ writings in the corpus will lead to accept a number of terms whose use may not be the best to recommend. Since one of the guiding principles of this project is to promote the correct use of language, we prefer to ground a compiled lexicon on a generally accepted set of general purpose words, such as GSL, VoA or BE1500; these short lists were analyzed in chapter [6]

7.3. A data structure for compiled lexicons

Comparison of word lists and compilation of a lexicon call for a common data structure. Word lists come in different formats, most usually several fields on lines of text, but also HTML, CSV, XML or other. Fields, separation characters, tag sets are all different from one word list to another. The analysis of word lists reported in chapter [9] was carried out by first bringing all the lists analyzed into a common data structure. Now this same structure is used in the compilation of a lexicon, advancing the purpose of performing several tasks required to license sentences written in the lexicon against a generative grammar.
The data structure for a lexicon is expected to provide support for the following operations:

- **test for inclusion**: verify if a word is included in the lexicon. Since the word may be a wordform or a headword, either two lists must exist, or efficient lemmatization provided, to convert from wordform to headword.

- **lemmatization**: convert from wordform to headword. Information such as part of speech, frequency of use, part of speech, senses, are usually referred to headwords. However, text is written in wordforms, so a reliable lemmatization facility is necessary. A wordform may lead to more than one headword; this leads to an ambiguity difficult to deal with.

- **PoS tagging**: determine the parts of speech (lexical categories) of a word, which accounts for the different grammatical roles a word may be playing in a sentence. This is usually recorded for headwords. A headword may belong to several different lexical categories, and play different roles in a sentence; a headword may be associated with several part of speech tags.

- **frequency of use**: several lists provide some indication of frequency of use, such as number of occurrences in a corpus. This information may be recorded separately for wordforms and headwords, or assigned to tuples (headword, PoS), which adds precision.

- **definitions**: determine the definitions (meanings, or senses) corresponding to a headword.

- **synsets**: determine WordNet synsets corresponding to a headword.

Headwords and wordforms may be implemented as lists, to show number of words, content of the lexicon, inclusion of a word, coverage of a corpus, or to compare with other lexicons. A list of distinct items allows fast testing of inclusion; size of lexicon is the length of the list. To compare lexicons, a list may be easily converted to a set, and set operations of difference and intersection applied.

When a wordform points to more than one headword, it is not possible to formally know the intended meaning. This is a lexical semantic ambiguity very difficult to deal with without “knowledge of the world” or by asking the author.

> In the design of a lexicon, a wordform must correspond to a unique headword, or be ready to ask the writer for the headword (meaning) intended.

The reverse is not required; a headword may lead to several wordforms with no practical consequence. This is because text is written in wordforms, not in headwords. Understanding the text goes from wordform to headword, and then from headword to sense.

A headword, or its corresponding wordforms, will frequently be used as different parts of speech. A lexicon where a headword corresponds to a single part of speech may be overly restrictive, leading to difficult writing and awkwardness of expression.

> In the design of a lexicon, a headform may correspond to more than one lexical category (PoS); disambiguation must be left to the engine applying the rules of production, or the writer asked for it.

Frequencies of wordforms and headwords are of interest not only to define inclusion of terms in the lexicon, but also to help writers choose the most frequent word, to reach the widest
possible audience. When comparing different word lists, size of the corpus from which the frequencies were determined must be considered.

Though few word lists provide a mapping of headword to definition or sense, this is one of the main purposes of a compiled lexicon: to map each word to a single meaning. Less restrictive, and more precise, is to map a pair (word, part of speech) to a unique sense. This is rarely the case for any word in any language, but in a specific domain experts may succeed in assigning a unique a meaning to each word. Considering part of speech helps focus meaning: a headword used as a verb differs in meaning from the same headword used as a noun, if only because one means an action or state and the other an animal, person or thing abstract or concrete.

From the preceding discussion, the following design decisions were made:

- **wordforms and headwords: implemented as lists.**
- **correspondence (wordform, headword), single valued: a dictionary makes search efficient.**
- **correspondence (headword, PoS), multi valued: a dictionary of lists or of text lines with delimited fields allows for efficient search.**
- **frequencies of wordforms or headwords: lists of tuples (frequency, word), ordered by frequency.**
- **senses of headwords, multi valued for selection of meaning, single valued in final lexicon; search for meaning is an usual operation; a dictionary of lists or of text lines with delimited fields; the dictionary key may be (headword, PoS).**

The former design considerations are implemented in the class Wordlist of the Klear Project. Documentation for The Klear Project describes this class in detail: Appendix B gives a summary of this documentation; full documentation can be found at [González-Barbome, 2012].

### 7.4. A GSL based experimental lexicon

This section describes the compilation of an experimental lexicon. This lexicon is mainly compiled with the purpose of experimenting on a methodology of construction and testing. It is based on the General Service List of English Words, as compiled by Leah Gilner and Franc Morales [Gilner and Morales, 2007]. As its source, this lexicon is expected to address “general use” of English, usable as a core lexicon to which domain specific terms may be added as needed.

#### 7.4.1. Aims and constraints

Following the main trend of this project, our experimental lexicon was designed for the writing of texts intended to be licensed by a relatively simple generative grammar. Such an experimental grammar is described in chapter 8. This imposes some constraints on the lexicon. The texts to be written with this experimental lexicon are expected to have the following limitations:

- Only simple sentences: the meaning implied in a compound sentence may be expressed in several simple sentences.
7.4. A GSL based experimental lexicon

- No complex sentences: the dependent clause in a complex sentence is expected to be expressed as another sentence, relating its meaning to the main clause by sequence or reference. E.g. “Sally visited her mother before she left for Europe” can be expressed as “Sally visited her mother. Sally left for Europe”.

- No relative clauses: the additional information provided by a relative clause is expected to be expressed as another sentence. E.g. “The house (that) my father bought is close to the University campus” can be expressed as “My father bought a house. The house is close to the University campus”. A sequential scheme for representation of sequential events is explored in chapter [12].

- Only declarative sentences: the texts in mind only give information; no interrogative, imperative or exclamatory sentences are needed. A language with imperative sentences may be very useful for manuals and instruction; this possibility is explored in chapter [12].

- No pronouns. Use of pronouns require anaphora resolution, to determine which noun each pronoun refers to, within one sentence or along several sentences. The referred noun can be repeated as needed, as in the following example: “My father bought a house. He had to sell his car to pay for it”; this can be expressed as “My father bought a house. My father had to sell his car to pay for the house”.

- Function words not overlapping with content words, to help PoS tagging and grammar licensing. Function words not overlapping among themselves in different categories, as far as possible.

The question of readability arises: these constraints may lead to unnatural or utterly awkward expressions. Though a certain amount of repetition is to be expected, in particular by leaving out pronouns, inconvenience to the reader will hopefully go no further.

7.4.2. Function words

*Function words* are commonly defined as words that indicate a grammatical relationship to other words. Function words are considered to have little semantic content. There are a finite, identified set of function words, classified in categories according to the grammatical relationship they establish. These categories are considered *closed categories*, since function words vary very little in time. The closed categories under which function words are commonly classified are: determiners, prepositions, conjunctions, and pronouns. The closed categories contrast with the *open categories*, where new words are constantly added and other words become obsolete and seldom used. The open categories are nouns, verbs, adjectives and adverbs. Words in the open categories are also called *content words* [Altenberg and Vago, 2010, Farlex Inc., 2010]. The most common English words in the closed categories are shown in table [7.1]. The list of function words is taken from [Altenberg and Vago, 2010].

The task of detecting the syntactic structure of a sentence becomes easier if each word is related to only one part of speech. There are some function words which may act as two or more different parts of speech in a sentence. A study of this superposition of categories in function words shows:

- determiners: no superposition, there are no two words in different subcategories.
### 7. Compilation of a Lexicon

<table>
<thead>
<tr>
<th>Categories</th>
<th>Wordforms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Determiners</strong></td>
<td></td>
</tr>
<tr>
<td>- articles</td>
<td>a, an, the</td>
</tr>
<tr>
<td>- demonstratives</td>
<td>this, that, these those</td>
</tr>
<tr>
<td>- possessive pronouns</td>
<td>my, your, his, her, its, our, their</td>
</tr>
<tr>
<td>- possessives</td>
<td>'s, s'</td>
</tr>
<tr>
<td>- quantifiers</td>
<td>all, any, both, each, either, enough, every, few, little, most, much, neither, no, several, some</td>
</tr>
<tr>
<td><strong>Prepositions</strong></td>
<td></td>
</tr>
<tr>
<td>- prepositions</td>
<td>about, above, across, after, against, along, among, around, at, before, behind, below, beneath, beside, between, beyond, by, despite, down, during, for, from, in, into, like, near, of, off, on, onto, out, over, since, through, throughout, till, to, toward, towards, under, until, up, upon, with, within, without</td>
</tr>
<tr>
<td>- prepositional phrases</td>
<td>across_from, ahead_of, along_with, because_of, by_means_of, due_to, for_the_sake_of, in_addition_to, in_front_of, inside_of, in_spite_of, instead_of, on_account_of, on_top_of, out_of, over_to, together_with, up_to</td>
</tr>
<tr>
<td><strong>Conjunctions</strong></td>
<td></td>
</tr>
<tr>
<td>- coordinating</td>
<td>and, or, but, for, so, yet, nor</td>
</tr>
<tr>
<td>- subordinating</td>
<td>after, although, as, as if, as though, because, before, even if, even though, how, if, in order that, once, rather than, since, so, than, that, though, till, unless, until, what, when, whenever, where, wherever, whether, which, while, who, why</td>
</tr>
<tr>
<td>- correlative</td>
<td>both...and, either...or, if...then, neither...nor</td>
</tr>
<tr>
<td><strong>Pronouns</strong></td>
<td></td>
</tr>
<tr>
<td>- subject</td>
<td>I, you, he, she, it, we, they</td>
</tr>
<tr>
<td>- object</td>
<td>me, you, her, him, it, us, them</td>
</tr>
<tr>
<td>- reflexive</td>
<td>myself, yourself, himself, hersel, itself, ourselves, yourselves, themselves</td>
</tr>
<tr>
<td>- demonstrative</td>
<td>this, that, these, those</td>
</tr>
<tr>
<td>- nominal possessive</td>
<td>mine, yours, his, hers, its, ours, theirs</td>
</tr>
<tr>
<td>- interrogative</td>
<td>how, what, when, where, which, who, whom, whose, why</td>
</tr>
<tr>
<td>- relative</td>
<td>that, which, who, whom, whose, whatever, whoever, whomever, whoever</td>
</tr>
</tbody>
</table>

Table 7.1.: Function words
7.4. A GSL based experimental lexicon

- prepositions: no superposition, there are no two words in different subcategories.
- conjunctions: word so is both a coordinating and subordinating conjunction.
- pronouns: words it, that, which, who, whom, whose, you, belong to more than one subcategories of pronouns.
- determiners and prepositions are disjoint.
- determiners and conjunctions: words so, that are found in both categories.
- prepositions and conjunctions: words after, before, for, since, so, till, until are found in both categories.
- pronouns show superposition with all the other categories:
  her, his, it, its, that, that, these, this, those, which, who, whom, whose, you, are also determiners;
  it, that, which, who, whom, whose, you, are also prepositions;
  how, it, so, that, that, what, when, where, which, which, who, who, whom, whose, why, you, are also conjunctions.

Considering the constrains stated for the design of this lexicon, inclusion of function words is done as follows:

1. Determiners: all determiners are included, no superposition among them.

2. Conjunctions, coordinating: though compound sentences will not be allowed, coordinating conjunctions can be used to join nouns and adjectives. Conjunctions and, or, but are the most common; for, so, yet, nor are less common. All coordinating conjunctions may be included, with no superposition among them nor with the other function words previously included.

3. Conjunctions, subordinating: since no complex sentences are allowed, subordinating conjunctions need not be included. So far, if they were included, no superposition will happen.

4. Conjunctions, correlative: conjunctions such as both...and, either...or, if...then, neither...nor, may be difficult to recognize in a generative grammar; special provisions must be made. Correlative conjunctions were not included in this experimental lexicon.

5. Prepositions: all prepositions can be added with the sole superposition of for, which is both a preposition and a coordinating conjunction. Since the use of for as a coordinating conjunction is not common, for can be added as a preposition only; the other less common coordinating conjunctions so, yet, nor may be included or left out according to their need in the domain of application.

6. Pronouns: wordforms it, that, which, who, whom, whose, you, may be used as two different types of pronouns; that, which, who can also be subordinating conjunctions; that is also a demonstrative determiner. Besides superposition, pronouns require anaphora resolution. Including pronouns requires caution and the anticipated resolution of superposition with other categories and anaphora. Use of pronouns may be dispensed by explicitly stating the noun which they stand for.
Table 7.2 show the function words chosen for this experimental lexicon. Please note for is included only as a preposition, not as a conjunction.

<table>
<thead>
<tr>
<th>Categories</th>
<th>Wordforms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Determiners</strong></td>
<td></td>
</tr>
<tr>
<td>- articles</td>
<td>a, an, the</td>
</tr>
<tr>
<td>- demonstratives</td>
<td>this, that, these those</td>
</tr>
<tr>
<td>- possessive pronouns</td>
<td>my, your, his, her, its, our, their</td>
</tr>
<tr>
<td>- possessives</td>
<td>'s, s'</td>
</tr>
<tr>
<td>- quantifiers</td>
<td>all, any, both, each, either, enough, every, few, little, most, much, neither, no, several, some</td>
</tr>
<tr>
<td><strong>Prepositions</strong></td>
<td></td>
</tr>
<tr>
<td>- prepositions</td>
<td>about, above, across, after, against, along, among, around, at, before, behind, below, beneath, beside, between, beyond, by, despite, down, during, for, from, in, into, like, near, of, off, on, onto, out, over, since, through, throughout, till, to, toward, towards, under, until, up, upon, with, within, without</td>
</tr>
<tr>
<td>- prepositional phrases</td>
<td>across_from, ahead_of, along_with, because_of, by_means_of, due_to, for_the_sake_of, in_addition_to, in_front_of, inside_of, in_spite_of, instead_of, on_account_of, on_top_of, out_of, over_to, together_with, up_to</td>
</tr>
<tr>
<td><strong>Conjunctions</strong></td>
<td></td>
</tr>
<tr>
<td>- coordinating</td>
<td>and, or, but, so, yet, nor</td>
</tr>
<tr>
<td><strong>Pronouns</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.2: Function words for experimental lexicon

### 7.4.3. Content words

Function words are predictable: they form a small, closed set quite stable in time; their semantic content is limited and generally agreed upon. Content words are numerous, full of meaning, and varying in time. A domain specific lexicon may be very limited, restricted to only a few words besides the terms used in the domain, but it may also be quite large, including a number of general purpose words besides specific terms. As an extreme example, a domain specific lexicon for History will include a large number of general purpose words, while a domain specific lexicon for the Systematics of Sharks may succeed with only a small number of general use words. Whatever the case, a domain specific lexicon may generally be conceived as general purpose words plus domain specific terms.

The experimental lexicon proposed in this section assumes the need for general purpose words as equivalent to those of everyday use in normal communication. This is the aim of the short word lists analyzed in chapter [6]. Therefore, one of GSL, BE1500 of VoA Special English word lists is considered a good start point. To choose one, the primary consideration should be purpose: for a corpus obtained from the Simple English Wikipedia, for instance, the immediate selection would be BE1500, or at least the first one to try. In this experimental lexicon, GSL Gilner was be adopted as the core lexicon for content words.

The GSL Gilner lists contain information to perform the tasks of lemmatization and part of
speech tagging. In the normalized data structure defined, an object of class Wordlist contains a dictionary \{wordform: headword\} to support lemmatization, and a dictionary \{headword: part of speech\} for part of speech tagging.

### 7.4.4. Adjustments and additions to the lexicon

Treatment of function words as different from content words requires some tests to ensure self-consistency of the lexicon, in particular to verify function words and content words are disjoint sets. Function words introduce grammatical constructions which require a semantic interpretation for knowledge extraction.

These elaborations allow some function words to be recovered as content words, for words which may play both roles: if these words were not accepted as function words, they may be accepted as content words.

Some other very common words may be added to the lexicon, such as days of the week, months of the year, and the lower written numbers. These adjustments and additions are described next.

**Self consistency tests.** To ensure self consistency in the supporting data structure of this example lexicons, at least the following conditions were verified:

- there are not repeated headwords; there are not repeated wordforms.
- there are not repeated PoS tags in the PoS tags assigned to each headword.
- there are not repeated headwords in the headwords that correspond to a wordform.
- there are not content words including a PoS corresponding to a function word. When this situation aroused, preference was given to the function word, and the word was not admitted as a content word. As a consequence, function words and content words are disjoint sets.

**Addition of some frequently used words.** The only GSL list to include days of the week and months of the year is GSL Lextutor, probably due to considering them proper names. These are frequently used words, though, and may be included in the lexicon as common nouns, which allows for phrases like *on Monday or in January* to be recognized as a prepositional phrases. Months of year *May* and *March* are also verbs, *may* and *march*; *may* is suggested to be kept in both roles, noun and verb; *march* as verb is less used, though, so it is suggested to be kept only as a noun.

Written numbers, specially lower numbers, may also be included, at least from 1 to 10, or 1 to 20, or other ranges according to purpose. Their role must be defined; they may be considered as quantifiers, which can act as determiners, and in this way allow phrases such as *three monkeys, or ten students*.

A function in the Klear toolkit for lexicons allows to easily include a list of words indicating their associated PoS tags [González-Barbome, 2012].

**Function words that are also content words in GSL Gilner.** Even in the rather restrictive selection of function words adopted in this example lexicon, there are 5 words which appear both as function words and content words. The following lines indicate the PoS assigned to
these five words in GSL Gilner, their PoS as function words, and the PoS under which they were accepted for the experimental lexicon:

- **most**: GSL pos: Adj;Adv; FW pos: Det; KL1 PoS: Det
- **enough**: GSL pos: Adj;Adv;Nc; FW pos: Det; KL1 PoS: Det
- **much**: GSL pos: Adj;Adv;Nc; FW pos: Det; KL1 PoS: Det
- **little**: GSL pos: Adj;Adv;Nc; FW pos: Det; KL1 PoS: Det
- **out**: GSL pos: Adj;Adv;Nc;V; FW pos: Prep; KL1 PoS: Prep

These words appear in GSL Gilner with part of speech tags corresponding to content words: adjectives, adverbs, and even nouns. These words were accepted in the lexicon only as function words, with the tags indicated.

**Content words left out from lexicon.** There are words in GSL Gilner with a part of speech tag corresponding to a function word which are not included in the list of accepted function words. These words are left out of the lexicon to preserve the highly desirable condition of function words not superposed with content words. Some of these words may act also as content words, since they have some other part of speech besides those of function words. These words may be accepted with no harm in any of the content words categories. In this example lexicon, there are 70 words which may be added to the lexicon in non function word categories. To help decide their inclusion, some frequency data may be called for. The BNC lists provide frequency counts for pairs (word, PoS). The following words appear in the BNC lists in roles other than function words:

- **besides**: 18 Adv;
- **less**: 243 Adv;
- **thus**: 205 Adv;
- **one**: 953 Pron; 118 Nc;
- **past**: 89 Adj; 86 Nc; 67 Prep; 21 Adv;
- **outside**: 116 Prep; 53 Adv; 37 Adj;
- **other**: 1336 Adj;
- **therefore**: 232 Adv;
- **save**: 118 V;
- **opposite**: 32 Adj; 12 Prep; 11 Nc;
- **however**: 605 Adv;
- **bar**: 101 Nc;
- **ahead**: 63 Adv;
- **inside**: 74 Prep; 50 Adv; 13 Nc;
- **nevertheless**: 72 Adv;
- **hence**: 48 Adv;
- **round**: 138 Adv; 115 Prep; 47 Nc; 28 Adj; 15 V;

The numbers are frequency counts in the BNC corpus. Some words did not produce any output: *another, many*, are classified as determiner pronouns in BNC; *notwithstanding, whichever*, do not have an entry; these words were not included in the lexicon. *Besides, less, thus, however, ahead, nevertheless, hence*, can be included as adverbs; *save* can be included as a verb; *bar* can be included as a noun. However, words like *one, outside, inside, round*, are used as both prepositions and adverbs, in some cases with comparable frequencies. If these words are included, users of the lexicon must be made aware of the syntactic role under which these words were accepted.
Words with several PoS. Many words can play different roles in a sentence, i.e. a word may be associated with more than one part of speech. These words are a source of syntactic ambiguity: a sentence may be parsed into two or more different syntactic trees, with possibly different meanings. Limiting the number of roles a word may play in a sentence reduces the risk of ambiguity, though it may also trouble writers, compelling them to know in which syntactic roles these words are accepted. Not all the syntactic roles a word may play are equally frequent, though; dispensing with syntactic roles seldom used will hardly be perceived by writers.

A word list including frequency of use for each pair (headword, PoS) may be called for help, to keep only those (headword, PoS) pairs exhibiting a minimum frequency of use. This was done with the available word lists from BNC.

The lists in BNC can be used to determine which are the most frequent parts of speech under which a word is used. The 2849 headwords in GSL Gilner lead to 4739 pairs (headword, PoS); the 5050 headwords in the available BNC lists lead to 5730 pairs (headword, PoS); there are 1725 pairs (headword, PoS) from GSL Gilner not in the BNC lists. Assuming these pairs are not in BNC because they are not frequently used, they may be excluded from GSL Gilner, bringing the number of pairs down to 3014, a reduction of 37%. However, this exclusion leaves 411 headwords in GSL Gilner with no PoS assigned, too high a number for a short list.

There are pairs (headword, PoS) in BNC with frequency count 0. It is tempting to exclude them, but if all parts of speech for a headword have frequency count 0, the headword will be altogether excluded. For a certain headword, only the most frequent parts of speech might be kept, say only those parts of speech which account for the 70% percent of the total frequencies for this headword in all its lexical categories (PoS). Again, this leads to completely exclude headwords with frequency 0 in all parts of speech.

The BNC lists available count frequencies per million words; the total number of headwords, 5050, is not so much bigger than the 2849 headwords in GSL Gilner. Unless more extensive and fine grained information on frequencies by (headword, PoS) becomes available, reducing the number of (headword, PoS) pairs is not advisable.

The COCA word list contains 4353 headwords, and 5000 pairs (headword, PoS). Being of about the same size as BNC, it was not tested: it uses a more complex PoS tagset, and it did not seem to offer ground for better results.

For this example lexicon, no pairs (headword, PoS) from GSL Gilner were excluded. The method employed may be applied, though, if a more reliable frequency by (headword, PoS) list becomes available.

Addition of domain specific words. Any addition must be done specifying the syntactic roles (part of speech) the words can perform in a sentence. For domain specific vocabulary, most added words will be nouns, followed by verbs; adjectives and adverbs are expected to be less frequently needed.

Proper nouns, dates and numbers. These categories may be detected outside the lexicon, by preprocessing the text.

Inflections. It is not strictly necessary to include inflections in a compiled lexicon: lemmatization can be achieved outside the lexicon using the large word lists. It is not much work,
7. Compilation of a Lexicon

though, to use those large lists when compiling the lexicon, and include a dictionary of word-form: headword for lemmatization within the compiled lexicon data structures.

7.5. Adding sense

A possible approach for a lexicon of commonly used words might proceed from sense to words, by first identifying the senses or concepts most frequently used, and then taking the most frequently used word associated to each of these senses of concepts. The Wordnet provides the raw material for this development. This would probably be the best approach towards a domain specific sublanguage lexicon, but its own nature calls for heavy human intervention: little can be expected from machines when it comes to select the most common senses managed in an area of knowledge, which is the previous step to determine the most common word to use for that sense.

To advance in the more classical approach we have been following, a list of frequently used words may be enriched by associating to each word the sense or senses most commonly meant when the word is used.

7.5.1. GSL Gilner in Wordnet

Each of the headwords in the GSL Gilner word list leads to one or several synsets. The 2849 headwords in GSL Gilner point to 16108 synsets, about 5.65 synsets per headword. Excluding synsets with frequency count 0, GSL Gilner headwords point to 10237 synsets, about 3.6 synsets per headword. A synset is considered to have 0 frequency count in Wordnet if all lemmas associated with this synset have a frequency count equal to 0.

Considering all synsets pointed to by headwords goes probably beyond the intentions of the speaker: a headword may point to several synsets, including synsets for which the most common lemma is another word different from the headword in GSL Gilner, and hence not intended as a sense by the typical speaker. For example, considering all the lemmas associated with all the synsets obtained from the word *salt* produces lemmas of names *salt*, *table_salt*, *common_salt*, *Strategic_Arms_Limitation_Talks*, *SALT*, *saltiness*, *salinity*.

A different approach is to first get all lemmas referred to by a headword, and select the synsets pointed to by these lemmas only. This leads to 15820 synsets, about 5.55 synsets per headword. These numbers are almost the same as those obtained directly from headword to synsets. The intersection of both sets shows that the set of synsets obtained from headword to lemma to synset is included in the set of the synsets obtained directly from headword to synset. Though the difference is not significant (less than 2%), the procedure of determining first the lemmas for a headword, and then the synsets for those lemmas, is considered closer to the intended meaning of the speaker. For this reason, the next tests are performed on the set of synsets determined through the lemmas.

The resulting number of synsets probably exceeds the “general service” needs. An immediate way to limit this number is by considering only the synsets with frequency count greater than 0. Since synsets are ordered by frequency of use, the order label can be used to reduce the number of synsets to the most frequent senses, for example considering only those labeled 01, 02, 03. Table 7.3 shows in columns the number of synsets considering all synsets, only synsets with order label 01, with order labels 01 or 02, and with order labels 01, 02, or 03. The first row shows the numbers for the list of all the GSL synsets, the second row restricts the synsets to those with frequency count greater than 0.
7.5. Adding sense

GSL synsets | all orders | order 1 | orders 1, 2 | orders 1, 2, 3
--- | --- | --- | --- | ---
all | 15820 | 5459 | 8542 | 10496
count > 0 | 10175 | 4453 | 6723 | 7979

Table 7.3.: GSL Gilner Synsets, from headword to lemma to synset.

The most restrictive option gives 4453 / 2849 = 1.56 synsets per headword; the least restrictive 15820 / 2849 = 5.55 synsets per headword. On the conservative side, if a limitation is chosen, it would be one of the least restrictives: 10496 / 2849 = 3.68 or 7979 / 2849 = 2.80 synsets per headword. Adoption of one of these sets requires further study, and will depend on the intended purpose.

A different approach to reduce the number of synsets evoked by a headwords is to first limit the number of pairs (headword, PoS) to only those above a frequency threshold, as described before.

Once a set of synsets is selected, a correspondence between headwords and synsets may be established. GSL Gilner includes PoS tags, hence a correspondence (headword, PoS) → [synset, synset, ...] may be established. As stated before, this correspondence is actually built as (headword, PoS) → [lemma, lemma, ...] → [synset, synset, ...], but it is finally used as a dictionary (headword, PoS) → [synset, synset, ...].

7.5.2. From words to meaning

The compilation of a dictionary (headword, PoS) → [synset, synset, ...] in our example lexicon was done from the list of (headword, PoS) included in the lexicon, determining first the lemmas recalled from a (headword, PoS) pair, and then the synsets recalled for each lemma. No attempt was made to limit the number of synsets by frequency count; this approach may lead to completely exclude some synsets of interest, because of the limited significance of the frequency counts in the Wordnet as formerly analyzed. The order of synsets recalled from a word or a lemma provides a more reliable hint on which sense is meant more frequently than the other senses recalled by the headword. As was seen, this is recorded in the order label of the synset, a two digit string number with '01' as the most frequent.

Table 7.4 shows the quantity of synsets recalled by headwords for different ranges of order labels from the Wordnet. The last column shows the number of synsets for content words only, excluding synsets recalled from pairs (headword, PoS) where PoS is one of the function word tags.

<table>
<thead>
<tr>
<th></th>
<th>headwords</th>
<th>(headword,PoS)</th>
<th>synsets, all</th>
<th>synsets, Content Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>all order labels</td>
<td>2826</td>
<td>4540</td>
<td>18953</td>
<td>18695</td>
</tr>
<tr>
<td>order labels 01 to 03</td>
<td>2826</td>
<td>4540</td>
<td>12723</td>
<td>12529</td>
</tr>
<tr>
<td>order labels 01 to 05</td>
<td>2826</td>
<td>4540</td>
<td>15449</td>
<td>15216</td>
</tr>
<tr>
<td>order labels 01 to 10</td>
<td>2826</td>
<td>4540</td>
<td>17830</td>
<td>17572</td>
</tr>
</tbody>
</table>

Table 7.4.: Synset mapping. Quantity of synsets recalled by headwords in lexicon, for different order labels in the Wordnet.

What help a user can expect from all this when consulting for a word? Suppose the user is
7. Compilation of a Lexicon

in doubt about the word *salt*. Our example lexicon was compiled with a mapping to synsets which include only the three most often used synsets, Wordnet labels 01, 02 and 03. The synsets recalled by salt provide the following information:

```plaintext
=== Word 'salt' as Adj
Synset('salt.s.01'), total count: 1 (100%)
definition: (of speech) painful or bitter
total count, total count: 1 (100%)
examples: ['salt scorn'- Shakespeare', 'a salt apology']
Lemma('salt.s.01.salt'); count: 1.0 (100.0%)
--- Word 'salt' as N
Synset('salt.n.01'), total count: 9 (100%)
definition: a compound formed by replacing hydrogen in an acid by a metal (or a radical that acts like a metal)
total count, total count: 9 (100%)
examples: []
Lemma('salt.n.01.salt'); count: 9.0 (100.0%)
Synset('salt.n.02'), total count: 5 (100%)
definition: white crystalline form of especially sodium chloride used to season and preserve food
total count, total count: 5 (100%)
examples: []
Lemma('salt.n.02.salt'); count: 5.0 (100.0%)
Lemma('salt.n.02.table_salt'); count: 0.0 (0.0%)
Lemma('salt.n.02.common_salt'); count: 0.0 (0.0%)
--- Word 'salt' as V
Synset('salt.v.01'), total count: 1 (100%)
definition: add salt to
total count, total count: 1 (100%)
examples: []
Lemma('salt.v.01.salt'); count: 1.0 (100.0%)
Synset('salt.v.02'), total count: 0 (100%)
definition: sprinkle as if with salt
total count, total count: 0 (100%)
examples: ['the rebels had salted the fields with mines and traps']
Lemma('salt.v.02.salt'); count: 0.0 (0.0%)
Synset('salt.v.03'), total count: 0 (100%)
definition: add zest or liveliness to
total count, total count: 0 (0.0%)
examples: ['She salts her lectures with jokes']
Lemma('salt.v.03.salt'); count: 0.0 (0.0%)
```

For a user application, this crude output should be presented in a prettier format, but so far the user knows, among other things: that the word *salt* is included in the lexicon; that it may act as an adjective, a common noun or a verb; that as a common noun it may refer to the chemical compound or the ingredient used in the kitchen to season food; that in this latter sense it may also be referred to as *table salt* or *common salt* with exactly the same meaning, but more specific.

7.5.3. Further exploitation of the Wordnet

The Wordnet offers a number of possibilities, both for the design of a lexicon and for its use. Though this course will not be pursued further, some of the more immediate applications are listed here:

- canonical lemma: a preferred lemma may be chosen for each synset, both to reduce ambiguity and to become as precise as possible as to meaning. For example, if both salt as a chemical substance (any salt) and salt used for cooking (sodium chloride) are to coexist in the lexicon, one might choose to use lemma *salt* for the chemical substance, and lemma *table salt* for salt used in cooking. This is an approach to the goal “one word, one synset”.

122
7.6. A note on Methodology

- synonym addition: a constrained lexicon of very few terms carefully mapped to synsets may be enriched using some of the other lemmas attached to a certain synset as synonyms for this synset, verifying or imposing that these newly added lemmas do not lead to other synsets than the one desired. This may help both writer and reader with a less strict vocabulary without losing the relation from words to meaning.

- hypernym, holonym and the other relations defined in the Wordnet may be used to capture less specific, more elusive meanings that may appear in a student’s answers. Following a previous example, pemmican has hyperyms pemmican → meat → food. If a question required producing pemmican as the name for ‘lean dried meat pounded fine and mixed with melted fat; used especially by North American Indians’ (Wordnet definition), getting meat will not be the right answer, but may be accepted with less marks; getting food as an answer at least shows the student has an idea of what it is all about.

For the purposes pursued in this work, the Wordnet is a very valuable resource, and deserves a careful exploration of the possibilities it offers.

7.6. A note on Methodology

The example lexicon compiled so far was based on a general purpose word list, and had no target domain in mind. The main purpose of the task was to evaluate the effort demanded, assess the usability of the lexical resources studied, develop the necessary tools, and verify the feasibility of the task. Along the work, the main ideas of a possible methodology emerged. The steps followed during research are commented in the next paragraphs.

1. Select a suitable set of function words. Function words do not have much meaning in themselves, but act as links between words defining different ways of structuring clauses and sentences. The accepted set of function words will result in the different grammatical constructions allowed in the language, and these constructions determine some of the relations among the concepts evoked by content words. Hence, it must be clear from the very beginning how these constructions will be recognized and processed. In other words, whenever a function word category is accepted in the lexicon, there must be a routine to process the grammatical constructions these function words allow in the language. The same wordform may be found in different function word categories, acting as different function words; there must be a provision for recognizing the wordform as the function word it is representing, which means knowing the lexical category under which it is playing, indicated by its part of speech tag. In some cases this may be achieved by allowing a wordform to play only as one function word with only one part of speech tag; in other cases, there must be a way to recognize which part of speech tag the wordform is playing.

2. Select a short list as a core lexicon. Depending on the target domain, and the complexities of the concept relations allowed a small, very specific vocabulary may suffice, or a wide general purpose lexicon called for. The general purpose short lists will probably be the best start point for most situations; additions and subtractions will happen later on as purpose and domain require. Which of the short lists to use will depend on the domain, purpose and to a certain extent on personal preference. If the target domain
were news of the world, the VoA short list would probably be tried first, since it has a long run in the field. If GSL is chosen, the Gilner version is recommended as the most reliable one, with the addition of the days of the week, the months of the year, and some or all of the 30 words included in the other versions of GSL.

3. **Coordinate function words and content words.** The short lists include function words, which must be limited to the set of function words previewed in the design of the lexicon. Some function words are also content words; some decision must be made on this point, either to suppress them as content words, or make provision to differentiate the same word as a function word or as a content word. Frequency of use, availability of synonyms, and the capabilities of the parser may help towards a decision.

4. **Lemmatization.** Some lists allow to determine the headword(s) for a wordform, others do not include the data. Lemmatization can be done with the help of the long lists; the AGID list is probably the best choice among the long lists analyzed. If the short list contains the data, it may be more efficient to solve lemmatization within the short list itself.

5. **PoS tagging.** Categorizing words in a text is a difficult problem. The easiest way is to allow only one part of speech for each headword, but this may prove too restrictive. The price to pay is potential syntactic ambiguity: more than one syntactic tree for each sentence, according to the different roles a word can play. The indetermination increases exponentially with the number of PoS per headword. Even if processing is not a concern, it may be altogether impossible to opt for one syntactic tree among others without the explicit PoS tagging of some words. This finally comes to asking the writer, the ultimate authority on what is meant in a text. Even if this were an option, it is far from desirable, and should be kept to a minimum. There are several measures to reduce the impact of this problem. A set of rules of production where the different roles of the same headword never end in two or more syntactic trees will vanish ambiguity, but it is very difficult to assemble, even if it were possible. A powerful, reliable, fast parser, capable of trying a number of alternatives for each word will not guarantee a single syntactic tree, but effectively discard alternatives not ending in a complete parse. A lexicon with as few PoS tags per headword is a more practicable way; there may be synonyms for some of the categories which this headword may play, thus helping to prevent ambiguity. Considering frequency of use, and identifying the most common word and PoS used in a particular field provide some hint on what role (or roles) each word must be accepted in.

6. **Add specific vocabulary.** Words from glossaries specific for the area of knowledge may be added to the lexicon. These will be mostly nouns, then verbs; some adjectives and adverbs may also be required as usual in the field, but in much smaller numbers. These additions to the lexicon must not compromise its consistency, and follow the design rules applied so far. A good glossary should contain words, synonyms if there exist, and definitions, to state clearly the meaning of the words in the area of knowledge or target domain.

7. **Add definitions (optional).** For domain specific words, definitions can be added from a domain specific glossary, which should ensure a unique sense for each headword. Al-
7.7. Validating a compiled lexicon

ternatively, definitions may be obtained by mapping headwords to Wordnet synsets, as described in the next step.

8. Map to synsets (optional). This is by no means a trivial task, but may be undertaken in a gradual way. A mapping of words to synsets ensures semantic determinism, and makes definitions and examples available.

The lexicon should be tested against a corpus “representative” of the domain. This may be a small collections of texts in the area, or perhaps a standard or agreed upon corpus may be available.

A routine to deal with words not included in the corpus must be defined. Most of them will be nouns. Some guidelines follow:

1. lemmatize and verify synonyms for the intended sense; in some cases a synonym for each of the headwords obtained may be found in the base lexicon.

2. for headwords with no suitable synonym in the base lexicon, follow one of these steps:
   a) add this headword to the lexicon, including PoS and wordforms. PoS will be determined by the sense of interest; wordforms will most likely be found in one of the long lists.
   b) instead of the headword as such a synonym may be added to the lexicon, if it seems more appropriate for the domain.
   c) if the headword is considered inadequate, not essential or unusual, a phrase explaining the meaning and appropriate substitutions may be of help to users.

3. register actions in (b) and (c) in the complementary glossary accompanying the lexicon. This glossary should contain words, synonyms, equivalent explanatory phrases and definitions, as needed.

Once the lexicon is ready, an adjustment stage is to be expected when put in production; some words not found in corpora may have been forgotten, and will turn up on use. As a representation of an area of knowledge, the lexicon is expected to vary along with the practice and research in the area.

7.7. Validating a compiled lexicon

The purpose of compiling lexicons in the scope of this work is more to “write” than to “read”: the lexicon will primarily be used by practitioners in the field to write documentation, reports or answers to questions. The use of the lexicon assumes awareness on the part of the writers of the deliberate limitations imposed right from the design phase, and the benefits expected from texts written within these limitations, essentially the production of unambiguous texts and the ability to extract knowledge from them by machine processing. Validation should proceed, then, by the use of the lexicon in the writing of texts. There is some certainty of success, though, since the lexicon constructed was based on a well known and widely tested list of frequently used general purpose words.

A more demanding, but also more definite test, is to determine the amount of coverage the lexicon can achieve on an existing corpus. Let us recall that coverage is determined by
7. Compilation of a Lexicon

counting all occurrences of a lexicon word in the corpus, for all the words in the lexicon that effectively appear in the corpus; coverage is presented as a percentage, which is the percent occurrences the lexicon could “cover” of all the occurrences in the corpus (which is the number of words in the corpus).

The example lexicon was tested against the now classical Brown Corpus. This is a general purpose, experimental lexicon; to be of most value, a lexicon for a specific domain should be tested against a representative corpus texts from the target domain.

7.7.1. Coverage in the Brown Corpus

A coverage test on the Brown Corpus was carried on on the same lines, using the example lexicon based on GSL Gilner plus complements, and the list of proper names from BE (Basic English). Though the Brown Corpus is considered a bit dated and rather small (a million words), it was chosen because of its availability and wide use. Results of the coverage tests are shown in Table 7.7. The Brown Corpus version used was the one included in the NLTK toolkit. Occurrences were obtained using the functions provided in the NLTK toolkit. Occurrences were counted after filtering out non alphabetical tokens. Total occurrences in the Brown Corpus, measured like this, amounts to 981716 occurrences, about a million words, which is the number of words usually cited for the Brown Corpus.

The mean percent coverage is 88.68%. Though it does not reach the excellence mark of 95% for language understanding, this must be considered a very good result, since the Brown Corpus is a collection of “real world” texts from a variety of genres, with no filtering nor adequacy of any kind.

<table>
<thead>
<tr>
<th>Brown category</th>
<th>total occurrences</th>
<th>covered occurrences</th>
<th>% covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>adventure</td>
<td>56658</td>
<td>50108</td>
<td>88.64</td>
</tr>
<tr>
<td>belles_lettres</td>
<td>149046</td>
<td>132614</td>
<td>88.98</td>
</tr>
<tr>
<td>editorial</td>
<td>52765</td>
<td>47121</td>
<td>89.30</td>
</tr>
<tr>
<td>fiction</td>
<td>57086</td>
<td>50819</td>
<td>89.02</td>
</tr>
<tr>
<td>government</td>
<td>59959</td>
<td>54295</td>
<td>90.55</td>
</tr>
<tr>
<td>hobbies</td>
<td>60796</td>
<td>61152</td>
<td>87.62</td>
</tr>
<tr>
<td>humor</td>
<td>17776</td>
<td>15448</td>
<td>86.90</td>
</tr>
<tr>
<td>learned</td>
<td>157035</td>
<td>138165</td>
<td>87.98</td>
</tr>
<tr>
<td>lore</td>
<td>94741</td>
<td>83712</td>
<td>88.36</td>
</tr>
<tr>
<td>mystery</td>
<td>46007</td>
<td>42024</td>
<td>90.17</td>
</tr>
<tr>
<td>news</td>
<td>83562</td>
<td>73642</td>
<td>88.13</td>
</tr>
<tr>
<td>religion</td>
<td>33958</td>
<td>30561</td>
<td>90.00</td>
</tr>
<tr>
<td>reviews</td>
<td>34108</td>
<td>29299</td>
<td>85.90</td>
</tr>
<tr>
<td>romance</td>
<td>56857</td>
<td>51318</td>
<td>90.26</td>
</tr>
<tr>
<td>science_fiction</td>
<td>11762</td>
<td>10417</td>
<td>88.56</td>
</tr>
</tbody>
</table>

Table 7.5.: KL-Eng1 coverage of the Brown Corpus

The Brown category most poorly covered is Reviews, with an 85.90%; this category comprises newspaper articles on theatre, books, music, and dance. Humor follows, with 86.90%; next come Skills and Hobbies (87.62%), and Learned (87.98%). The Learned category collects texts from the Natural Sciences, Medicine, Mathematics, Social and Behavioral Sciences, Po-
7.8. Discussion

Political Science, Law, Education, Humanities, and Technology and Engineering [Francis, W. N. and Kucera, H., 1979]. Government, Mystery, Religion and Romance are the best covered categories, all them reaching 90% coverage.

In the tests performed a list of 16K proper nouns was used, obtained from the Basic English Institute. When this list is not used, the percent coverage decreases to a mean of 84.69%, an almost exactly 4%. Detection of all proper nouns will most probably bring a slight improvement in coverage.

For a small lexicon of less than 3000 words to reach a coverage from 86% to 91% (85.90% to 91.55%) in a variety of genres must be considered a promising result. This is a coverage test, not usable as a measure of how many words are left out; there may be many, but the more they may be the less frequent their use will be. A domain specific lexicon is expected to be more systematically and rigorously developed than texts from the media; words to be added for a coverage of domain specific texts will either be found in specific glossaries, which means addition is straightforward, or will be of rather seldom use, which may allow for their substitution for synonyms more frequently used, and already included in the list.

7.7.2. A Longman based experimental lexicon

EnLong3k is an experimental lexicon based on Longman Communication 3000 built along the same lines as the GSL based lexicon previously described. When compiled in the same way as the experimental lexicon bases on GSL, this experimental lexicon based on the Longman list contains 3183 headwords, increasing to 3199 if some prepositions and modals are added (via, according_to, unlike, would, shall, could, concerning, opposite, per, should, plus, used_to, including, might, ought_to, must). Tests of coverage on the Brown corpus produces almost exactly the same numbers obtained with the GSL lexicon: 84.25% if proper nouns are not detected, and 88.27% if proper nouns are detected based on the BE proper noun list.

The Brown corpus is recognized to be somewhat dated for a selection of common use words as recent as Longman Communication 3000. This tends to explain the lack of improvement verified when using a list a bit longer than GSL; words like computer, email, cellphone are found in Longman 3000 but not in GSL.

7.8. Discussion

Lists of basic words for learners provide a solid start point for a lexicon of general use: they attempt at wide coverage and expressiveness with as few words as possible, selecting from the most commonly used words. Some differences do exist, however, among different word lists, specially in the less frequently used words. An immediate approach is to adopt those words present in several lists, and consider inclusion of less frequent words according to the target domain.

For a domain specific lexicon to be used in a sublanguage, inclusion of function words subcategories must be done only if the resulting grammar constructions they enable are acceptable for the purpose of the lexicon. Function words usually allow for new syntactic constructions. If some form of knowledge extraction is to be performed on the text, the handling of these syntactic constructions must be previously defined. This means the inclusion of function words depends on the rules of production allowed for the sublanguage and the subsequent processing. For instance, to include pronouns, a strategy to deal with the resulting anaphora
must be defined. The framework developed allows to optionally include different subcategories of function words. Ensuring only one PoS per function word simplifies subsequent text processing.

Content words belong to basically two groups: words of general use and words specific to the domain. The latter are easier to deal with: they are well known, their senses are clear, and they tend to be used in only one sense, and even one PoS. General use words are necessary for expressiveness and readability, but it may be very difficult to define a single meaning for each headword, or even a few. How many senses to admit will ultimately depend on the purpose of the lexicon. Ensuring a single PoS for each word may also be trying. In some cases, a carefully designed grammar and processing can admit several PoS with no confusion. A carefully designed lexicon and related set of rules which ensures a single meaning for each pair (headword, PoS) is as good as can be wished.

The experimental lexicons compiled in this chapter were not expected to define syntactic structures at this stage: facilities to optionally include required subcategories of function words confer the necessary flexibility to adapt the lexicon to support different sets of syntactic rules.

7.9. Conclusions

Contributions of this chapter include:

- two experimental general purpose lexicons, one based on the General Service List of English Words (GSL), and another based on Longman Communication 3000. Both are lists of commonly used words.

- a first approach to a methodology to compile general purpose or domain specific lexicons; function words were analyzed separately, since they have little semantic content but articulate syntactic structures which must be recognized.

The framework developed and the methodology described allowed to compile a base lexicon with moderate effort. Refinement against a specific domain is further required, but the specific lexicon is known to the practitioners, and a glossary may be available. The tools developed allow to use other short word lists as a start point to compile a lexicon. The analysis of function words is expected to be almost universally valid. The analysis of function words to include in a lexicon must always be done, but the guidelines here given will hopefully simplify the task.

Tests of the example lexicon against a well known corpus were satisfactory. This was a very general case, with texts from a wide spectrum of sources; a lexicon for a specific domain built along the lines suggested in this chapter should perform equally well against a corpus of texts specific to the domain.
8. Syntax based sublanguages

Abstract. A first approach to a sublanguage for knowledge representation may be based on the syntax of natural language. An experimental sublanguage of Spanish is built for simple declarative affirmative sentences, with no conjunctions nor pronouns, for testing purposes and evaluation of effort. The rules must license a unique syntactic tree for each sentence. A simple tagset is proposed, including some special tags for terminals which stand in place of wordforms, to decouple the lexicon from the rules. A small vocabulary, the set of rules, a set of test sentences and results of these tests comprise each example sublanguage. The tools for testing can produce summary results for a set of sentences or the complete syntactic tree for a particular sentence. For sentences which cannot be licensed, its longest partial parse helps detect the reason of failure. Spanish01 is a straightforward transcription of common sentence structures in Spanish; traditional constituents are recognized, such as noun complement, copulative verbs and attributes, transitive and intransitive verbs, direct and indirect object. To avoid some cases of syntactic ambiguity, square brackets are used to delimit constituents, and also to allow for the nesting of structures. This version assumes some grammar consciousness when writing, which in some contexts may be a desirable feature; successive versions try to overcome this requirement by suppressing the identification of some constituents, while keeping the expressiveness of the language and the condition of a unique syntactic tree per sentence. Spanish05 distinguishes only copulative from predicative verbs, and only one kind of verb complement. English05 is a set of rules for English built along the same lines as Spanish05. Spanish05 contains 17 rules, English05 contains 19 rules; both should be within reach of any post secondary student. Both Spanish01 or Spanish05 generate naturally readable sentences, and both can be used as scaffolding grammars to be enhanced into more concise writing, such as by the addition of conjunctions to avoid repetition, or the addition of quantifiers. The examples show a syntax based sublanguage with a reasonable expressiveness can be compiled with moderate effort, producing only one syntactic tree, and allowing the writing of texts apt for knowledge representation.

Besides a lexicon, a sublanguage needs a grammar to determine valid syntactic constructs in sentences. The syntax of a language may be described as a generative grammar, which is a set of rules for the construction of a sentence. A generative grammar may be implemented as a software application, which can determine if a sentence is valid, i.e. if it has been built according to the syntactic rules of the language. This chapter describes the compilation of some generative grammars based on the syntax of natural language. A series of versions lead to a simplified grammar for Spanish; an equivalent grammar is proposed for English. For knowledge representation, the main requirement on grammars is to enable only one syntactic tree per sentence.
8. Syntax based sublanguages

8.1. Introduction

The following sections describe some example sublanguages. Each sublanguage is presented as a set of rules of production, a set of test sentences, and the results of building the syntactic trees for these sentences. The lexicon used in each case is a small one built for the tests; a lexicon for field testing or production can be compiled as described in preceding chapters.

These example sublanguages have two essential requirements: they must use a known vocabulary, and they must license only one syntactic tree. A known vocabulary means a collection of words with meanings known and agreed upon by the users. A single syntactic tree for each sentence written in the sublanguage ensures a deterministic syntactic relation among the words in the sentences, which allows for the construction of a knowledge representation instance.

The main purpose of these example sublanguages is to test feasibility, to show that a reasonably expressive sublanguage can be built, and obtain an idea of the effort required. Many details and enhancements are possible, and likely to be required for use in the classroom; quantification and dates are immediate examples. The relative small quantity and simplicity of the rules in the proposed examples allow for the gradual addition of support for new features of natural language. These example sublanguages should not be considered finished products, but as the groundwork on which to build a more complete set of rules for general use. Notwithstanding their limitations, the complexity of the test sentences show the capability of these languages to tell a short story or write a small description, provided some care is given to the allowed forms of expression, and text is written accordingly.

Testing sentences are nothing more than that, syntactically correct sentences to test the grammar; though readable, they are not expected to be meaningful, specially the long ones. In some cases, unlicensed sentences are shown; they come from limitations deliberately imposed to the generative grammar, such as “not more than two verb complements”.

8.2. Lexicon

Compilation of a lexicon was discussed in chapter 7. The lexicon used for this experimental sublanguage is very limited, and devised just for the purpose of testing the rules of production. Testing of the generative grammar is ultimately done against a set of terminals which map parts of speech, to decouple the lexicon from the rules. Thus, any lexicon with abilities for lemmatization and part of speech resolution can be used with these rules. Only one part of speech per wordform is assumed, though; more advance Natural Language Processing techniques, such as bigrams, may help admit more than one wordform per part of speech (PoS), but this is considered an enhancement, as was discussed in chapter 5.

8.3. Design considerations

Design considerations for the syntax based sublanguage proposed in this chapter follow the considerations explained in chapter 3 on construction of a sublanguage. Since this is an experimental grammar for testing purposes, many possible enhancements and features have been left out for simplicity. The proposed grammar has the following limitations:

- active voice sentences only.
8.4. Tagset

- declarative sentences only.

- affirmative sentences only; negation, though simple in the grammar, requires a non trivial semantic definition for consistent knowledge representation.

- simple sentences only, no compound or complex sentences; use several simple sentences instead.

- no conjunctions; several sentences can be used instead, at the cost of some repetition in the text.

- no pronouns, to avoid anaphora; though this adds repetition it also avoids potential confusion, besides the requirement of anaphora resolution.

- lexicon decoupled from rules, parsing on “pos” or “fake PoS”, lowercase terminals of the same name as PoS which act instead of wordforms for parsing.

Some of these requirements may be relaxed with not much difficulty in further versions of the sets of rules proposed here, e.g. use of conjunctions for multiple subjects, “Rufo, Toppy and Wanda are dogs”, instead of a sentence for each. However, this experimental version was intentionally kept clear of even relatively simple extensions, to help testing and proof of concept.

8.4. Tagset

The tagging system is very simple, but enough for our purposes. The Spanish and the English tagsets are very similar; tags are named differently for mnemonic reasons, to make them easily recognized by speakers of each language. Non terminal symbols include tags for syntactic constructs and also tags for functional constructs: a noun phrase is a syntactic construct which may play different functions as a component of a sentence; a direct object is one of these functions, the attribute of a linking verb is another. This combination of syntactic and functional groupings provides a way to keep control on the number of syntactic trees licensed by the grammar for a single sentence. Another device to the same purpose is the use of square brackets [ ] and ]. Enclosing constituents between square brackets help the parser identify only one possible syntactic function for this constituent in the sentence.

The tagset for terminals and non terminals is shown in the following box. Terminals include the square brackets. In non terminals, an ad-hoc grouping is defined within phrases, such as “AdjG” for adjective group or GN for “noun group”; this grouping provides a more fine grained control on the syntax of sentences, at the price of adding a level to the syntactic tree.

Tags for terminals, Spanish tagset:
8. Syntax based sublanguages

**Terminales / Terminals**

Np : nombre propio / proper noun  
Nc : nombre común / common noun  
Adj : adjetivo / adjective  
Adv : adverbio / adverb  
Vcop : verbo copulativo / linking verb  
Vpred : verbo predicativo / predicative verb  
Vintr : verbo intransitivo / intransitive verb  
Vtran : verbo transitivo / transitive verb  
PrepCI : preposición complemento indirecto ('a', 'para') / indirect object prepositions ('to', 'for')  
Prep : preposiciones / prepositions  
Det : determinante / determiner

**Terminales requeridos por la sintaxis / terminals required by syntax**

P0 -> ']'  
P1 -> ']'  

Tags for non terminals, Spanish tagset:

**No terminales / non terminals**

O : oración / sentence  
SN : sintagma nominal, GN grupo nominal / noun phrase, noun group  
SAdj : sintagma adjetival, GAdj grupo adjetival / adjective phrase, adjective group  
SAdv : sintagma adverbial, GAdv grupo adverbial / adverbial phrase, adverbial group  
SPrep : sintagma preposicional / prepositional phrase  
SV : sintagma verbal / verb phrase  
SVtr : sintagma verbal transitivo / transitive verb phrase  
SVintr sintagma verbal intransitivo / intransitive verb phrase  
SVcop : sintagma verbal copulativo / linking verb phrase  
SVpred : sintagma verbal predicativo / predicative verb phrase  
CN : complemento del nombre / noun complement  
CD : complemento directo / direct object  
CI : complemento indirecto / indirect object  
CC : complemento circunstancial / circumstantial complement  
Attr : atributo / attribute

Management of lexicons becomes more flexible if parsing is done against the parts of speech corresponding to wordforms in a sentence instead of the wordforms themselves. To this purpose, a substitution tagset is used. Before submitting a sentence to parsing, its wordforms are substituted for the corresponding part of speech tags, in lowercase to differentiate the tags substituting the wordforms for the parts of speech used for parsing. In other words, the lowercase part of speech tags act as if they were the wordforms. Hence, the parser builds a
8.5. Procedure

From the former design considerations, a study of the grammar of the language helped select the required grammar constructions. The sources selected for the grammar of the language were mainly secondary school level \[\text{CNICE, 2005}\], with an occasional reference to deeper works, such as \[\text{Alarcos Llorach, 1994}\]. A set of tests were put together to test each of the different functional groupings. Each sentence is expected to be licensed by the grammar, and produce only one syntactic tree.

Summary results of parsing the test sentences are provided in the following form: each line gives the number of wordforms in the sentence, the number of licensed syntactic trees, and the sentence itself. Sentences with no licensed tree, or with more than 1 licensed tree, are marked with *. Test sentences are presented in small groups, each one targeted at testing a different type of non terminal. At the end, a couple of examples show the grammar can deal with relatively long sentences of about 50 words. The maximum length recommended for clear writing is 25, though smaller numbers of about 15 words have also been proposed.

Example grammars were compiled for Spanish; a parallel procedure was later carried out to compile an English grammar of similar capabilities, described later in this chapter. An excerpt of the tests performed on transitive verb phrases follows:

### SVtran, sintagma verbal transitivo

```
## con CD
4 1 Juan pintó [la casa]
6 1 Juan pintó casi totalmente [la casa]
6 1 Juan pintó [la casa][casi totalmente]

## con CD y CI
```

Etiquetas de sustitución / substitution tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Substitution Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Np</td>
<td>'np'</td>
</tr>
<tr>
<td>Nc</td>
<td>'nc'</td>
</tr>
<tr>
<td>Adj</td>
<td>'adj'</td>
</tr>
<tr>
<td>Adv</td>
<td>'adv'</td>
</tr>
<tr>
<td>Vcop</td>
<td>'vcop'</td>
</tr>
<tr>
<td>Vintr</td>
<td>'vintr'</td>
</tr>
<tr>
<td>Vtran</td>
<td>'vtran'</td>
</tr>
<tr>
<td>Vpred</td>
<td>'vpred'</td>
</tr>
<tr>
<td>PrepCI</td>
<td>'prepci'</td>
</tr>
<tr>
<td>Prep</td>
<td>'prep'</td>
</tr>
<tr>
<td>Det</td>
<td>'det'</td>
</tr>
</tbody>
</table>

syntactic tree where leaves are these lowercase part of speech tags. The reverse substitution of lowercase tags for the original wordforms produces the syntactic tree for the sentence. This decouples the lexicon from the parser; the lexicon can be modified at will, at any time, with no consequences on the parsing tools, which only perceive the lexicon as a dictionary \{wordform : subst_PoS_tag\}, a mapping of each wordform into a substitution part of speech tag which will act in place of the wordform in parsing. The substitution tagset is shown in the following box, for the Spanish tagset.
8. Syntax based sublanguages

6 1 Juan pintó [la casa] [para Pedro]
9 1 Juan pintó [la casa [de la esquina]] [para Pedro]
16 1 Juan pintó casi totalmente [la casa [de la esquina]] [para la vieja madre china [de Pedro]]

## con CD, CI y CC
13 1 Juan pintó [la casa] [para la madre [de Pedro]] [el domingo [de tarde]]
23 1 Juan pintó totalmente [la vieja casa [de madera lustrada] [de la esquina]] [para la vieja madre [de Pedro]] [el domingo [en la tarde]]
17 1 Juan pintó totalmente [la casa] [para la madre] [con la estudiante [de arquitectura]] [el domingo [de tarde]]

## prueba con CD y CC
8 1 Juan pintó [la casa] [el domingo [de tarde]]
12 1 Juan pintó [la casa [de madera [de roble]]] [el domingo [de tarde]]
15 0 *Juan pintó totalmente [la casa] [con la estudiante [de arquitectura]] [en una tarde] [el domingo]

A detailed description of the process can also be produced, showing the syntactic tree built, or the longest edge recognized. The sublanguage Spanish03 sentence

Juan pintó [la casa] [para la madre [de Pedro]] [el domingo [de tarde]]

is parsed into the following syntactic tree:

--- Complete parses: 1

(0
  (SN (Np Juan))
  (SVtran
    (GVtran (Vtran pintó))
    (CD (P0 []) (SN (Det la) (GN (Nc casa))) (P1 []))
    (CI
      (P0 [])
      (PrepCI para)
      (SN
        (Det la)
        (GN (Nc madre))
        (SPrep (P0 []) (GPrep (Prep de) (GN (Np Pedro))) (P1 [])))
    (P1 [])))
  (CC
    (P0 [])
    (SN
      (Det el)
      (GN (Nc domingo))
      (SPrep (P0 []) (GPrep (Prep de) (GN (Nc tarde))) (P1 [])))))

In the following example, for sentence

Juan pintó totalmente [la casa] [con la estudiante [de arquitectura]] [en una tarde] [el domingo]
no valid syntactic tree results, but a partial parsing is available. Spanish03 grammar does not admit more than two CC in the sentence, while this one contains three. The longest partial parse found is:

(Sub)Tree from longest edge
(0
  (SN (Np Juan))
  (SVtran
    (GVtran (Vtran pintó) (GAdv (Adv totalmente)))
    (CD (P0 [) (SN (Det la) (GN (Nc casa))) (P1 ]))
    (CC
      (SPrep
        (P0 [)
        (GPrep (Prep con)
          (Det la)
          (GN (Nc estudiante))
          (P0 [)
          (GPrep (Prep de) (GN (Nc arquitectura)))
          (P1 ]))
        (P1 ])))))
  (CC
    (SPrep
      (P0 [)
      (GPrep (Prep en) (Det una) (GN (Nc tarde)))
      (P1 ]))))
)

No complete parses.

Appendix A contains a full description of the example sublanguages compiled, with their rules, testing sentences, and summarized results. Tools for compiling and testing the sublanguage are described in chapter 14. Documentation of the software modules is included in summary format in Appendix B. Tool and structures related to sublanguages are in the klear.sublang modules; full documentation can be found online [González-Barbome, 2012].

8.6. Syntax-based Spanish grammar versions

This section describes an evolution in the design of a Spanish sublanguage: the first version follows rather closely traditional grammar structures; subsequent versions gradually relax some grouping restrictions keeping only the forms required for the licensing of syntactic trees apt for knowledge representation. The first version, closely tied to traditional syntactic structures, and the last version, with only the most necessary formal structures, are both considered useful for further development. The traditional grammar based version is to be preferred on knowledge areas where grammar is a subject taught or required to be known. The last version imposes less restrictions, giving more freedom to the writer, which also means less restrictions towards correct use of the language.

The main sources for the study of Spanish grammar were [CNICE, 2005] and [Alarcos Llorach, 1994]. Some tutorials and resources for secondary school usage were also consulted, such as [Liroz, 2008], mainly to verify the proposal was within reach of post secondary school students.
8. Syntax based sublanguages

8.6.1. Spanish01

Spanish01 is a quite straightforward transcription of the most common sentence structures in Spanish grammar.

Verbs are differentiated in subcategories: linking verbs, transitive verbs and intransitive verbs. In a linking verb phrase, an attribute follows the linking verb; in a transitive verb phrase, a direct object follows the transitive verb; in an intransitive verb phrase the intransitive verb may stand alone. Transitive and intransitive verb phrases admit an optional indirect object. All three verb phrases admit other verb complements in the form of prepositional phrases, called "complemento circumstantial" in Spanish, abbreviated CC in the tagset.

The recognition of indirect objects requires the listing of Spanish prepositions 'a', 'para' ("to", "for" in English) as different from the other prepositions. As other verb complements can also be prepositional phrases, if the distinction is not made there is no way to distinguish an indirect object from other prepositional phrase complements. Since indirect objects are optional, a pair of empty square brackets is considered an empty indirect object, which allows for its inclusion, which may be useful for application in grammar concerned domains.

The other verb complements that can be added to all three verb phrases were limited in number to only two, to keep the sentence from becoming cluttered. For knowledge representation, further complementing can be achieved adding a new sentence with the same subject and verb, incorporating the new complement. The sequence of sentences "Rufo is a dog", "Rufo is a big dog", "Rufo is a big black dog", "Rufo is a dog with long ears", "Rufo is a dog with very long ears", "Rufo is bad tempered", "Rufo is a dog with hairy tail" are all equivalent to the single sentence "Rufo is a bad tempered big black dog with very long ears and hairy tail"; whatever the way of expression, in one sentence or in various potentially overlapping sentences as the former ones, representation of knowledge is the same in both cases.

Nouns in noun phrases admit different types of complement, designated as noun complements ("complementos del nombre", tag CN). Adjective phrases can be added before and after the noun, as is usual in Spanish. Adverb phrases can act as complements of adjectives and verbs.

Square brackets are used in prepositional phrases, direct objects, indirect objects and verb complements, as a way to correctly identify each, a requirement to license a unique syntactic tree for each sentence.

8.6.2. Spanish02

Spanish02 was an experimental version for testing, discarded in the course of development.

8.6.3. Spanish03

Spanish03 dispenses with the grouping of noun complement, using a prepositional phrase instead. As Spanish01, these rules admit nesting of prepositional phrases, using square brackets for correct resolution. Spanish03 also dispenses with square brackets for linking verb attributes, except for prepositional phrase attributes, but this is because prepositional phrases are required to be enclosed between square brackets themselves, for their correct recognition, as previously stated. Spanish03 is also relieved of the empty indirect object, a feature considered too technical and only called for in grammar concerned knowledge areas.
8.6.4. Spanish04

Spanish04 attempts to suppress the distinction between transitive and intransitive verbs, an inconvenience at the time of building the lexicon. A predicative verb phrase is introduced to account for both transitive and intransitive verb phrases. With no further modifications, this simplification leads to the licensing of more than one syntactic tree both in transitive and intransitive sentences, since the rules allow for the recognition of a prepositional phrase in more than one role in the sentence, e.g. as an indirect object and as another verb complement.

A relatively simple sentence like

Juan pintó [ la casa ] [ el domingo [ de tarde ] ]

produces 2 syntactic trees:

--- Complete parses: 2

(0
 (SN (Np Juan))
 (SVpred
 (GVpred (Vpred (Vtran pintó)))
 (CC (P0 [) (SN (Det la) (QN (Nc casa))) (P1 ]))
 (CC
 (P0 [)
 (SN
 (Det el)
 (QN (Nc domingo))
 (SPrep (P0 [) (GPrep (Prep de) (QN (Nc tarde))) (P1 ]))))
 (P1 ]))))

(0
 (SN (Np Juan))
 (SVpred
 (GVpred (Vpred (Vtran pintó)))
 (CD (P0 [) (SN (Det la) (QN (Nc casa))) (P1 ]))
 (CC
 (P0 [)
 (SN
 (Det el)
 (QN (Nc domingo))
 (SPrep (P0 [) (GPrep (Prep de) (QN (Nc tarde))) (P1 ]))))
 (P1 ]))))

The ambiguity comes from the inability of the grammar to differentiate a direct object CD from a complement CC: the sentence is licensed both as a transitive and an intransitive one. This and similar ambiguities may be solved suppressing the rules using CD, CI for a unique rule with a generic predicate in the form of CC:

SVpred → GVpred | GVpred CC | GVpred CC CC | GVpred CC CC CC | GVpred CC CC CC CC

This was done in the following version.
8. Syntax based sublanguages

8.6.5. Spanish05

Spanish05 attempts to correct the licensing of more than one syntactic tree when suppressing the distinction between transitive and intransitive verbs, by suppressing also the identification of the functions of direct object and indirect object. Only one type of complement phrase is then kept, arbitrarily designated with non-terminal tag CC. Verb phrases admit up to four of these complements, which in transitive verb phrases may be equivalent to a direct object, an indirect object, and two other verb complements.

Spanish05 succeeds in licensing the same testing sentences as Spanish03, but expects a lexicon with predicative verbs tagged 'Vpred' instead of 'Vtran' and 'Vintr' for transitive and intransitive verbs.

8.7. A syntax-based English grammar

English05 is an English based sublanguage made along the same lines as Spanish05. The following example sentences are licensed by this grammar. Their main purpose is to test the constituents, but they also give an idea of the expressiveness of the language.

```plaintext
### PrepP, prepositional phrase
Jack walked [to the house]
Jack walked [to the house [in the corner]]
Jack walked [to the house [in the corner [of the street [of the city [of London]]]]]

### AdjP, adjective phrase
the tattered man kissed [the maiden]
the tattered torn man kissed [the maiden]
the very tattered very torn man kissed [the very forlorn very shy maiden]

### AdvP, adverb phrase
Jack walked slowly
Jack walked very slowly
Jack walked very slowly [to the house]

### NP, noun phrase
the man kissed [the maiden]
the man [from the old white house] kissed [the maiden]
the very tattered very torn man [from the very old totally white house] kissed [the very forlorn very shy maiden] [yesterday [in the morning] [at the corner [of the street]]]

### VlnkP, linking verb phrase
## with a noun attribute
Jack is a physician
Jack is a very old very nice physician [from the hospital [in the city [of London]]]

## with an adjective attribute
the man is tattered
the very tattered
* the man is tattered and torn # no conjunctions!
the man is tattered torn # licensed, though unnatural
the man is very tattered very torn

## with a prepositional attribute
Jack is [in the house]
Jack is [in the house [at the corner [of the street [of the city [of London]]]]]
Jack is [in the house [at the corner [of the street [of the city [of London]]]]] [with the tattered man] [near the very nice totally white house [of the forlorn maiden]]
Jack was [in the house] [with the forlorn maiden] [yesterday]
```
Jack was [in the city] [in the house] [with the forlorn maiden] [yesterday]
### VactP, active verb phrases, with transitive verbs
## with a direct object
Jack built [a house]
Jack built [a totally white very nice house]
## with a direct object and indirect object
Jack built [a house] [for the maiden]
Jack built very swiftly [a house] [for the maiden]
## with direct object, indirect object and verb complements
Jack built [a house] [for the forlorn maiden] [yesterday]
Jack built [a house [in the corner [of the street [of the city [of London]]]]] [for the forlorn maiden] [yesterday [in the morning]] [with the tattered man]
## with a direct object and verb complement
Jack built [a house] [yesterday]
Jack built very swiftly [a house] [yesterday [in the morning]] [with the tattered man]
### VactP, active verb phrases, with intransitive verbs
## without complements
Jack walked
Jack walked slowly
Jack walked very slowly
## with verb complements
Jack walked [towards the house]
Jack walked [towards the house] [of wood] [at the corner] [yesterday [in the morning]]
## with advverb phrase and complements
Jack walked very slowly [with the tattered man] [towards the house [of wood] [at the corner]] [yesterday [in the morning]]
All these example sentences produce only one syntactic tree. Except for the square brackets (and the rather nonsensical meaning) they should sound natural to any speaker of English, even to learners.

The rules, lexicon and test sentences for this version are given in Appendix A. The rules and tagset are reproduced here to give a more complete view of this version. The main sources for the compilation of these rules were [Altenberg and Vago, 2010] and [Jurafsky and Martin, 2008].

### En05Rules.txt: rules of production for English05
### Tags
## terminals
# Np : noun, proper
# Nc : noun, common
# Adj : adjective
# Adv : adverb
# Vlk : verb, linking
# Vintr : verb, intransitive
# Vtran : verb, transitive
# PrepIO : indirect object prepositions ('to', 'for')
# Prep : preposition (except 'to' and 'for')
# Det : determiner
## non terminals
# S : sentence, start symbol
# NP : nominal phrase, NG nominal group
# AdjP : adjective phrase, AdjG adjective group
# AdvP : adverb phrase, Advg adverb group
8. Syntax based sublanguages

# PrepP : prepositional phrase
# VP : verb phrase
# VactP: active verb phrase
# VtranP : transitive verb phrase
# VlnkP : linking verb phrase
# Attr : attribute of linking verb
### Sentence, declarative
S -> NP VlnkP | NP VactP
### Sentence, imperative
S -> VlnkP | VactP
### PrepP, prepositional phrase:
## these rules admit nesting of prep phrases:
PrepP -> P0 PrepG P1
PrepG -> Prep NG | Prep Det NG
PrepG -> Prep NG P0 PrepG P1 | Prep Det NG P0 PrepG P1
PrepG -> Prep Det NG P0 PrepG P1 P0 PrepG P1
### AdvP, adverb phrase:
AdvP -> AdvG | AdvG PrepP
AdvG -> Adv | Adv Adv
### AdjP, adjective phrase:
AdjP -> AdjG | AdjG PrepP
AdjG -> Adj | AdvG Adj
AdjG -> Adj Adj | AdvG Adj Adj | Adj AdvG Adj | AdvG Adj Adj
### NP, noun phrase:
NP -> Np
NP -> Det NG | Det NG PrepP | Det NG PrepG PrepP
NG -> Np | Nc | AdjG Nc
## Attribute, for linking verb phrase
Attr -> AdjG | Det NG | PrepP
### verb complements
CC -> PrepP | P0 NP P1 | P0 AdvP P1
### VlnkP, linking verb phrase:
VlnkP -> Vlnk Attr | Vlnk Attr CC | Vlnk Attr CC CC | Vlnk Attr CC CC CC
### VactP, active verb phrase
VactG -> Vact | Vact AdvG
VactP -> VactG | VactG CC | VactG CC CC | VactG CC CC CC | VactG CC CC CC CC
### P0 -> ’[’
P1 -> ’]’

Many grammar structures of very common use in English are excluded from this simple grammar. This was a design decision: it is possible to include more elaborate structures, in some cases with not much complication of the grammar, but with potential difficulties when coming to build a knowledge representation. As a rule, all grammar structures accepted must be associated to a semantic interpretation in a knowledge representation scheme. In a semantic graph representation, for instance, it must be clear which nodes and which arcs will these structures add, if it could not be said otherwise in a natural way, or if different ways of saying the same thing will not result in different, not equally recognizable nodes and arcs. In sublanguage proposals, the complexities of the language are avoided by limiting the language. A brief discussion of some structures of the English language not included in this version follows, with some hints on the points to solve before their inclusion.
8.7. A syntax-based English grammar

- **Conjunctions.** Though compound sentences are discouraged in favor of several sentences with some repetition, conjunctions may be included to form compound noun phrases or verb phrases, as in *Rufio, Hercules and Wanda are dogs*, or *Jack came into the room with his violin, greeted the presents with a bow, and played the solo by heart.* These sentences offer no particular difficulty for knowledge representation, since they can be rewritten in several sentences accepted by a grammar with no conjunctions, and help towards more concise, equally readable text.

- **Quantifiers** can be included as determiners. The inclusion of quantifiers such as *all, most, several, few, no, either, neither*, requires a careful evaluation of meaning when accepting them for knowledge representation: they may be indeterminate, have a negative content, or be applied in rather loosely ways.

- **Numbers,** cardinal or ordinal, may be included in the language in different forms: *two dogs* (determiner), *the second night* (adjective), *the soldiers were thousands* (noun). Again, their handling for knowledge representation must be carefully considered and defined; this conditions the roles in which they will be accepted by the language.

- **Dates.** Representing time in a knowledge representation scheme is no trivial task, which may go from an occasional reference to indicate strict points in time for a sequence, as in a historical chronology. Dates can be included as prepositional phrases like *in 1948, on Saturday, or in February, or on 2012-05-03.* A format for dates can be defined and accepted as in these examples; the difficult point is again their treatment for knowledge representation, which will depend on the meaning the knowledge representation scheme will give to dates.

- **Possessive 's.** A noun with a possessive 's can be treated as a determiner [Jurafsky and Martin, 2008], or as an adjective phrase before a noun, *(AdjP (NP the carpenter) 's) (Nc tools).* Knowledge representation of 's may be made equivalent to a link *belongs-to* or similar to convey the meaning.

- **Noun as adjective.** Constructions like *a mountain bike* or *a morning flight* are common in English, specially for well-known things. They may be included in the grammar, but a proper treatment for knowledge representation must be defined. Alternatively, it can be said *a bike for the mountain* or *a flight in the morning* for an equivalent meaning expressed as a prepositional phrase.

- **Clauses like the last train arriving at the station (gerundive), pottery made in China (-ed clause), the last to deliver his homework (infinitive).** These clauses add to the meaning of different objects: *the train, pottery, a student.* For a factual, declarative sublanguage, some other ways of saying the same thing in an equally natural way may be possible; this is a safe way to go. This type of clauses are seen as difficult to include without adding to the risk of alternate ways of saying the same thing in different forms not easily recognizable as having the same meaning.

- **Subordinate clauses** present similar difficulties as clauses; they also add meaning to existing concepts, but include a verb, as in a stand alone sentence.

The former analysis shows the difficulties of adding some of these grammar structures will not be so much in the grammar but in the way these structures will be converted into a knowledge
8. Syntax based sublanguages

representation in a predictable way. Though the experimental sublanguages developed in this work will not go into these enhancements, knowledge representation schemes may provide mechanisms to assimilate them, such as the definition of clusters, or levels in the knowledge representation: a net of concepts and relations may be built around a central concept or idea, and the whole set managed as a single node. The cluster node may expand or contract in the graph, showing or hiding its contents, or it may be referred to as a separate graph. Though this is a subject of Knowledge Representation, the sublanguage must provide the ways to define these aggregations, for example by recognizing a title and footing, to determine the name of the cluster and the piece of text which defines it.

8.8. Discussion

The Spanish based experimental sublanguages described were a series of steps towards a simple, usable generative grammar for the Spanish language. Spanish05 implies a departure from grammar orientation into a more pragmatic recognition of word grouping in the language, at the price of losing some information of the functions constituent phrases play in verb phrases. Though this simplification may reasonably be considered a loss for teaching the correct use language, it is not expected to have an effect on the construction of a knowledge representation instance from the resulting syntactic tree. Though these simplifies the rules of the generative grammar, and produces flatter syntactic trees, some domains of application may be concerned with the correct identification of syntactic structures, in which case a set of rules based on Spanish01 may be more adequate.

The final version, Spanish05, may be considered a scaffolding grammar on which to add other enhancements and features. As it is presented, Spanish05 includes just the following 17 rules:

```
O -> SN SVcop | SN SVpred
SPrep -> P0 GPrep P1
GPrep -> Prep GN | Prep Det GN
GPrep -> Prep GN P0 GPrep P1 | Prep Det GN P0 GPrep P1
GPrep -> Prep Det GN P0 GPrep P1 P0 GPrep P1
SAdv -> GAdv | GAdv SPrep
GAdv -> Adv | Adv Adv
SAdj -> GAdj | GAdj SPrep
GAdj -> Adj | Adj Adj
SN -> Np
SN -> Det GN | Det GN SPrep | Det GN SPrep SPrep
GN -> Np | Nc | GAdj Nc | Nc GAdj | GAdj Nc GAdj
Atr -> GAdj | GN | Det GN SPrep | SPrep
CC -> SPrep | P0 SN P1 | P0 SAdv P1
SVcop -> Vcop Atr | Vcop Atr CC | Vcop Atr CC CC
GVpred -> Vpred | Vpred GAdv
SVpred -> GVpred | GVpred CC | GVpred CC CC | GVpred CC CC CC | GVpred CC CC CC CC
```

English05 was built along the same lines as Spanish05, but for the English language. Both sublanguages have similar capabilities of expression. Excluding comments and symbols for square brackets, English05 boils down to the following 19 rules:

```
S -> NP VlnkP | NP VactP
```
8.9. Conclusions

Contributions of this chapter include:

- **two sets of rules for a Spanish sublanguage for declarative sentences**: Spanish01 keeps close to the syntactic structures as in traditional grammar, Spanish05 is less grammar conscious, with more coarse grained constituency. Each of these sublanguages comprises rules and example sentences (appendix A). They were tested with toy lexicons, but wider lexicons can be fed into the application.

- **a set of rules for an English sublanguage for declarative sentences, named English05**, which parallels Spanish05 in its minimum requirements of grammar knowledge (appendix A). This set of rules was tested with EnLong3K, a general purpose lexicon compiled from the Longman Communication 3000 word list (chapter 7).

- **a first approach to a methodology for compiling sets of rules for sublanguages**, based on a list of function words already identified, and content words taken from existing word lists, corpora, or manually added.

A syntax based grammar can be put together with only a moderate effort. Enhancements are possible and desirable, provided each is carefully tested and evaluated in its consequences, i.e. license a single syntactic tree, and there is a clear way to translate the structure into a knowledge representation. A grammar conscious sublanguage will follow syntax structures closely, making a difference between linking, transitive and intransitive verb phrases, differentiating attributes, direct and indirect objects, and other complements. This grammar offers support for correct writing, compelling students to be aware of the different constructs in the sentence. A simplified, more pragmatic grammar is also possible, with a minimum of requirements for
8. Syntax based sublanguages

correct writing. This simplification results in easier writing and flatter syntactic trees, and may be used more mechanically, which may not be desirable when teaching languages, for instance. Both designs can equally be transformed into a knowledge representation instance; this will be done in chapter 11.
Part IV.

Knowledge
This part deals with Knowledge and Knowledge Representation, studied from the perspective of Education and Assessment.

Chapter 9 reviews state of the art in Knowledge Representation. The main techniques and languages are examined, with a view to their potential application to learning and assessment.

Chapter 10 deals with issues related to the application of Knowledge Representation to Education. The five traditional roles of Knowledge Representation are discussed in their relative importance for learning and assessment, trying to determine what to look for when choosing a Knowledge Representation language to support educational tasks.

Chapter 11 proposes a syntax based sublanguage and knowledge representation scheme. The example sublanguages are taken as a start point to arrive into a knowledge representation instance in the form of a graph. Several potential difficulties and ideas on how to overcome them are considered. A distance measure between knowledge representations instances is proposed, which accounts for a mark given to the free text answer.

Chapter 12 starts with an analysis of types of knowledge and knowledge topologies to represent them. Requirements for a sublanguage and knowledge representation scheme are identified. Some possible schemes for different types of knowledge are considered, such as object oriented for category and individual representation, imperative models for instructions, a sequential model for time related knowledge such as historical events, together with widely known concept maps, mind maps, and the more formal and promising Topic Maps standard.
9. Knowledge Representation, state of the art

Abstract. Knowledge Representation is usually attached to Reasoning. The capabilities of each proposal for representation and reasoning may differ, ones being more apt for communication, others more apt for inference and question answering. Object-Attribute-Value, uncertain facts, fuzzy facts, rules, semantic networks and frames are recognized techniques for Knowledge Representation. Several languages have been created to represent knowledge in those different techniques. Main Logic based languages are Propositional Logic, First Order Logic and Description Logic. Rule based systems implement if-then structures in different ways. Frames are a structured way of attaching properties to objects, similar to form filling. Visual languages are different forms of semantic networks, with more or less formalism. Concept maps may be rather loosely defined, but are a form of semantic network commonly used. A taxonomy of visual languages shows that most of them are implemented as a form of concept map. The Semantic Web brought ontologies into Knowledge Representation, with a concern for agreement, sharing, merging and reuse. Among the many available, graphical UML notations for ontologies, commonsense knowledge bases, Topic Maps, some refinements of semantic networks, and a controlled language offer a sample of interesting proposals potentially useful for Education. A comparison of these proposals unsurprisingly shows that simplicity and visualization come at the price of some resignation in formality and reasoning capabilities, leaving more to the criteria of the users. The variety of proposals, none of them completely satisfactory, calls for an evaluation in the perspective of their potential use for educational purposes.

Knowledge conveyed in a free-text answer must be extracted and recorded in a data structure, so as to be compared against a reference structure and give a mark to the answer. At this stage, a sublanguage has been developed or adopted, which ensures a single syntactic tree per sentence. Before defining the transformation from text into some kind of data structure, a review and analysis of existing knowledge representation techniques and languages must be done, so as to choose the most appropriate ones. In this chapter, different techniques and languages for knowledge representation are described and analyzed.

9.1. Terms

Different disciplines and fields of interest have been interested in representing knowledge, from the cognitive sciences to Artificial Intelligence and the Semantic Web. Many strategies and models have been devised to represent knowledge, both for machine processing and human communication. Very frequently some common words were combined to designate a specific technique, sometimes strictly specified, such as Sowa’s conceptual graphs or ISO13250 Topic
Maps, but some other times rather loosely defined, as are semantic graphs, concept maps or mind maps. Activity in the field has led to an extensive terminology, where the same words are employed in different or overlapping senses, in different levels of abstraction, or in combinations on which sometimes defy common sense understanding. This section explains the sense in which some terms are used in this work. No attempt at a formal definition is made; the purpose of this section is only to state what is meant by each term. Consulted sources are duly credited, but the wording is mostly ours.

**Data** designates the qualitative or quantitative values of attributes or variables, expressed as symbols. These symbols may be characters, numbers, images, sound, or any output which can be perceived. A **symbol** is something that by association or convention stands for or represents something else.

Data becomes **information** when some sense is given to the symbols of data [Wikipedia, 2012b][Bellinger et al., 2004][Princeton University, 2011][Farlex Inc., 2010]. A person’s name, age, sex and photograph may be collected as characters, numbers and an image; these characters, numbers and image become information when they are identified as name, age, sex and photograph, and associated into a personal record.

**Knowledge** is the collection of information intended to be useful for some purpose [Bellinger et al., 2004]. When a teacher looks at the personal record of a new student and memorizes the information contained therein, she has acquired some knowledge of the student. When this teacher finally meets the student, and appropriately addresses her by name an title (Mr or Ms), the teacher is using her knowledge of the student.

**Understanding** adds to knowledge the capacity to synthesize new knowledge, as when a child applies her knowledge of arithmetic operations and time to determine that a day is $12 \times 60 \times 60 = 43200$ seconds long [Bellinger et al., 2004].

**Reasoning** is the process of drawing conclusions, inferences, or judgements, based on previous knowledge. Reasoning is a way to go from some ideas to other ideas through thinking, in a process considered valid or legitimate to establish such relations. One of these valid processes of thinking is the use of Logic.

**Inference** is the act or process of deriving logical conclusions or making logical judgements from premises or propositions accepted to be true. Inferred conclusions emerge from incidental evidence or from previous conclusions rather than from direct observation; it is the result of thinking.

In a Knowledge Representation context, a **domain** is an area of concern or interest. A domain may also be called a **field**, in the sense of a topic or subject of academic or educational interest.

Some of these terms will be further elaborated on. Other terms will be introduced as needed. Though an attempt has been made to capture the most common senses in which the words are used, all these explanations are to be understood in the context of this thesis.

### 9.2. Knowledge Representation and Reasoning

In the literature, knowledge representation is generally referred to as **Knowledge Representation and Reasoning**, treating representation as closely related to reasoning. This means that whatever the way knowledge is represented, it must allow for inference, i.e. producing “new” knowledge not apparent in the representation, but logically derived from what is apparent (hence, not strictly “new”). However, the balance of representation and reasoning
may abruptly differ in the different proposals. For the purpose of this thesis, representation is critical, and reasoning incidental: our intended audience will make much more from a semantic graph with little or no inference capabilities than from a bunch of statements in first order logic with all its reasoning power. In the following sections, the main characteristics of the most common models and languages for Knowledge Representation are brought forward and evaluated, to the purpose of making a sound selection of the most adequate for the requirements of this thesis.

A formal, essentially logic based approach to Knowledge Representation can be found in [Brachman and Levesque, 2004]. Knowledge is the relation between an agent (person or machine) and the idea expressed in a proposition. A proposition is a declarative sentence that can be true or false. A representation is an arrangement of symbols which stand in place of some domain. Knowledge representation is the use of formal symbols to represent the propositions considered true by some agent. Logical inference produces a new proposition which comes out as a logical conclusion of some previous propositions. A knowledge base is the symbolic representation of a collection of propositions related to some purpose (the intentional stance of the agent). A knowledge based system can assimilate new information and adjust its behavior accordingly.

The former definitions pave the way for several logic based methodologies of knowledge representation with a strong commitment to reasoning.

An ontological oriented approach is proposed by [Russell and Norvig, 2010]. Knowledge representations in non-trivial domains calls for the representation of general concepts common to many different domains. A general framework of such abstract objects is called an upper ontology. An upper ontology of the world is a hierarchical organization of abstract concepts such as Thing, Numbers, Places, Moments, Animals, Humans. Objects are organized in categories, and reasoning can take place at the categories level or at the objects level. A general purpose ontology can be specialized towards the entities of interest in a specific domain.

The former conception presents Knowledge Representation as the building of an ontology: an upper, general purpose, universally agreed upon ontology is enhanced to include the concepts of interest in a domain. Ontologies will be further discussed later in this chapter.

Another ontology oriented approach can be seen in [Gašević et al., 2006]. It starts off from the cognitive sciences, assuming the human mind has a mental representation of the world analogous to computer data structures. There are different types of human knowledge: procedural, declarative, heuristic, uncertain, commonsense, and some others. Each or these types is organized in the mind as different structures, and is used differently. The different mental representations proposed by the cognitive scientists, such as logical propositions, rules, concepts, images and analogies, are the basis for knowledge representation techniques such as logic, rules, semantic networks and frames. These representation techniques are supported by artificial knowledge representation languages in which precise grammar allows for easy parsing and machine processing, though sometimes at the cost of readability. Knowledge representation languages are classified in logic based, frame based, rule based, visual languages, and natural languages.

In this approach an effort is consciously made to bridge the cognitive perspective and the Artificial Intelligence perspective. The distinction between knowledge representation techniques and languages, and the classification of knowledge representation languages will be followed here for the evaluation of the pros and cons of these languages.
9. Knowledge Representation, state of the art

9.3. Knowledge Representation techniques

The variety of cognitive theories makes apparent the difficulty of understanding how the human mind organizes knowledge. Consequently, no single technique can be expected to do so either. The best technique will be the one which mostly fits the purpose of the application desired. The most frequently used knowledge representation techniques are [Gašević et al., 2006]:

- **Object-Attribute-Value (O-A-V):** gives a value to the attribute of an object. The expression Book-color-yellow means “the color of the book is yellow”. Multi valued attributes can be expressed as Book-color-yellow,white.

- **Uncertain facts:** a certainty factor \( cf, 0 \leq cf \leq 1 \), allows for a sentence like “the night will be probably wet” to be expressed as “Night-weather-wet (CF=0.5)”.

- **Fuzzy facts** originate in the imprecision of natural language. A sentence like “the man is old” is uncertain as to age. Fuzzy sets, in which a degree of membership is evaluated by a membership function in \([0,1]\) allow treatment of imprecision [Wikipedia, 2012b] fuzzy sets.

- **Rules** are the typical IF-condition-THEN-conclusion. Rules relate conditions to conclusion, or a situation to an action. Rules applied to fuzzy sets are called fuzzy rules.

- **Semantic networks** are graphs in which nodes represent objects, concepts or situations of a certain domain, and the edges represent relationships among these concepts. There is no standard notation for semantic networks; it should be defined. Semantic networks are clear and easy to understand.

- **Frames** can be described as a form with fields to fill. The form is the frame, and the fields are called slots. Class frames record the properties common to a set of objects; an instance frame is a certain object of a class, slots record the values of properties. Frames are closely related to semantic networks, but a frame may contain executable code, called a facet, associated to a slot; code in a facet executes automatically under certain circumstances, e.g. when a value is changed.

Different languages have been devised for each of these techniques. Some of them are reviewed in the next section.

9.4. Languages for Knowledge Representation

The classification of knowledge representation languages is not strict, but it is useful to group languages in categories when it comes to select the most convenient one for a certain purpose. Ontologies and ontology languages are examined in a later section.

9.4.1. Logic based

There is considerable literature with describes the different types of logic mentioned here. [Magnus, 2010] is an excellent introductory textbook on formal logic. Artificial Intelligence and Knowledge Representation textbooks also describe these systems in detail [Gašević et al., 2006, Russell and Norvig, 2010, Brachman and Levesque, 2004].
9.4. Languages for Knowledge Representation

*Propositional logic,* also called *sentential calculus.* A *proposition* is a logical statement or sentence that may be true or false. A symbol $A$ may be assigned to a sentence like “Agatha lives in Dreadsbury Mansion”, and $A$ will be true or false. Propositions identified by their corresponding symbols can be linked by the logical connectives conjunction (AND), disjunction (OR), negation (NOT), conditional (IMPLIES) and biconditional (EQUIVALENCE) to form more complex expressions that can be evaluated as true or false by a set of inference rules. Propositional logic is a simple logic, generally considered insufficient for most knowledge representation applications where reasoning is required.

*First order logic,* also called *predicate logic* or *quantified logic* goes inside propositions and distinguishes terms, variables and predicates. If “$H$” symbolizes “is hungry” (a predicate) and “$r$” symbolizes “Rufus” (a term), “$Hr$” may be read as “Rufus is hungry”. A variable “$x$” can stand for any term. If an expression such as “$Ax$” is known to be true, it may be concluded that “$Ar$” is also true. First order logic adds two quantifiers: the universal quantifier “for all”, and the existential quantifier “there is at least one”.

All sentences in Logic are assertions; this leaves aside all human reasoning that involves beliefs, assumptions, doubts, desires and the like [Gainvici et al., 2006]. These are not the most difficult shortcomings, though: First Order Logic is expressive enough for many applications. Besides, it exhibits a powerful deductive system. But in many cases the computational cost of deduction may result impractical. A set of propositions $S$ *entails* proposition $p$ if $p$ is true when propositions $S$ are true. What reasoning does is determine all entailments. For a deduction procedure to do so in affordable time, either some incorrect answers will be given, or some correct answers will be missed. This has led to a number of refinements [Brachman and Levesque, 2004].

Quantifiers in First Order Logic pose another challenge: defining the universe of discourse. In the sentence “everyone cheered”, who will be included in “everyone”? All humans, all living creatures, people in this room? Even the use of the existential quantifier may lead to strange results if the universe of discourse is not carefully defined [Magnus, 2010].

Translating from natural language into First Order Logic sentences also requires considerations not usually done. Definition of the universe of discourse, scope of quantifiers, ambiguous predicates, and multiple quantifiers are some of the potential pitfalls [Magnus, 2010].

*Description Logic* is a family of knowledge representation languages for formal reasoning on the concepts of an application domain, what is called its *terminological knowledge.* They restrict and improve on First Order Logic by being decidable fragments of it, which means there are effective procedures for solving inference problems in finite computational time. Several different techniques can be used for inference, which is up to the implementation to decide. Description Logic is based on concept descriptions, which are expressions built from *concepts* (unary predicates) and *roles* (binary predicates), with the constructors provided by each particular implementation. Logical statements relating concepts or roles are called *axioms*. A Description Logic knowledge base has a terminological part called a TBox and an assertional part called ABox. In the TBox statements introduce names for complex descriptions. The ABox statements establish properties of individuals. Description Logic was the basis for the development of several ontology languages, including OWL [Baader et al., 2008].

9.4.2. Rule based

All rule based representation languages are implementations of if-then structures, which is a well known and easy to understand paradigm. On the other hand, there is considerable
9. Knowledge Representation, state of the art

Figure 9.4.1.: RuleML: visualization of sentence “The discount for a customer buying a product is 7.5 percent if the customer is premium and the product is luxury” as an OrdLab tree (taken from [RuleML, 2011]).

difference in the way each language represents if-then rules. Probably their most popular application is in expert systems, for instance to assist physicians in diagnosis. Rule based languages have been combined with frames to achieve object orientation, hierarchical structuring and inference [Gašević et al., 2006].

The Rule Markup Initiative has developed RuleML, a family of XML based rule languages, attempting to promote it as the canonical Web language for rules. RuleML uses XML markup, provides a formal semantics, and looks for efficient implementations. RuleML covers the entire rule spectrum, and can specify queries and inferences in Web ontologies, mappings between Web ontologies, and dynamic Web behaviors.

The RuleML Initiative has also developed a sublanguage called Datalog which can formalize facts expressed as English sentences into markup atoms among which rules can be established. The XML rule serialization can be expressed visually as an “OrdLab tree” (Ordered Labeled Tree) similar to a linguistic parse tree [RuleML, 2011].

Datalog allows to formalize rules expressed in natural language into XML documents apt for sharing and inference, at the same time offering the possibility of a visual representation.

The example sentence “The discount for a customer buying a product is 7.5 percent if the customer is premium and the product is luxury”, represented as an OrdLab tree, is shown in figure 9.4.1.

9.4.3. Frame based

In the representation languages considered so far properties of an entity may be scattered among a collection of sentences. It was natural to think of the properties of an entity and
the entity itself as a whole, which may be called an “object”. The data structure to support information was called a frame. A frame is conceived as a group of individual data items, each identified by a name. These data items are called slots, and the items that they contain are called fillers. The frame is itself identified by a name. A frame structure looks like this [Russell and Norvig, 2010]:

\[(\text{Frame-name}
\quad<\text{slot-name1 filler1}>
\quad<\text{slot-name2 filler2}>
\quad\ldots)\]

There are individual frames, to represent single objects, and generic frames, to represent categories, analogous to classes and objects in Object Oriented Programming. Several special names of slots indicate relationship among slots:

- :IS-A indicates a more general category.
- :INSTANCE-OF indicates the category in which the object belongs.

Slots can contain attached procedures, to be executed in different situations:

- :IF-NEEDED the procedure will only be executed when necessary, e.g. a calculation to determine a value on this slot will only be done if the slot value is interrogated.
- :IF-ADDED for a procedure to be executed on addition of the frame.
- :IF-REMOVED for a procedure to be executed on removal of the frame.

The original frame-based languages lacked a precise semantics. This led to different systems exhibiting different behaviors even when using virtually the same components and relationship names. Such systems could not possibly interoperate nor share knowledge. Later languages introduced formal semantics while keeping ease of representation. KL-ONE (1985) introduced Description Logics into frames, which not only added rigor to frame-based languages but started a new generation of frame-based languages, such as CLASSIC (1991). An example of CLASSIC encoding looks like this [Gašević et al., 2006]:

\[
\text{define-concept[JAPANESE-CAR-MAKER,}
\quad(\text{ONE-OF Mazda Toyota Honda});
\quad\text{define-role[thing-driven]};
\quad\text{define-concept[RICH-KID,}
\quad(\text{AND STUDENT}
\quad(\text{ALL thing-driven SPORTS-CAR})
\quad(\text{AT-LEAST 2 thing-driven})))]
\]

The syntax traces to the underlying Description Logics.

Most frame-based languages end into being just a new syntax for parts of First Order Logic [Gašević et al., 2006]. Though this means nothing new can be expected as to expressiveness, frame languages are a bit easier to interpret in terms of ordinary human communication.

9.4.4. Visual Languages

The use of visual elements for knowledge representation is appealing, first of all because it is usually easier to understand and more intuitive than other media of expression. Besides, knowledge in some domains may be impossible to express otherwise. Visual representations are often used in preliminary studies, even when the target is a formal language.

A Graphical User Interface (GUI) is frequently used to capture information for the construction of a formal representation such as First Order Logic. In this case, the visual elements
simplify interaction with the user, but they do not properly constitute a visual language [Gašević et al., 2006].

Though visual languages may be based on other elements, it is most usually associated with some kind of graph with nodes and arcs.

A semantic network is a graphic notation for representing knowledge in the form of nodes and arcs [Sowa, 2007]. A concept map is a graphical tool for representing knowledge by drawing concepts as labeled circles or boxes, and relationships among concepts as labeled links connecting the corresponding circles or boxes [Novak and Cañas, 2006]. The similarity of both definitions makes them hard to differentiate, and in the literature they are often taken as synonyms. Semantic networks suggest a more abstract concept, though, and the term is preferred in Artificial Intelligence, whereas concept map tends to be more used in Education and the cognitive sciences.

A now classical and frequently cited taxonomy of all visual languages was proposed by [Myers, 1990]. The taxonomy includes languages for knowledge representation, and all of them are shown to be some kind of concept map, in the general, unrestricted sense formerly defined. In practice, all visual knowledge representation languages can be considered a form of concept map or may be implemented as a concept map [Gašević et al., 2006]. Anyway, for concept maps to be used effectively as a knowledge representation language, their syntax and semantics must be formally specified. This has been done in KRS, a visual counterpart of CLASSIC, a frame based language. A more powerful proposal are Conceptual Graphs, which can represent different forms of knowledge, including First Order Logic and even natural language [Gašević et al., 2006] [Kremer, 1998]. Knowledge expressed in Conceptual Graphs may be serialized, and unambiguously read and interpreted by a machine. However, the example sentence, “Tom believes that Mary wants to marry a sailor”, originally given by John Sowa in 1984, shows to be surprisingly complex to represent, as can be seen in figure 9.4.2.

The visual languages for knowledge representation used in practice are almost all a form of concept map, or can be implemented by some form of concept maps. As formerly stated, concept maps themselves are a form of semantic network; the difference between both terms may be more linked to the application domain (Education, Artificial Intelligence) than to essential meaning. Though concept maps are too loose and free to be considered for knowledge representation, their syntax and semantics can be formalized. Several proposals exist, with different degrees of formalization and capabilities. As a rule, more formalism tends to offer better reasoning capabilities, but at the same time removes the language from direct understanding by a non expert.

9.4.5. Natural Languages

Natural language is our immediate way of expression, and most human affairs happen within natural language communication; they must consequently be considered as knowledge representation languages. Natural languages are of very limited use in Artificial Intelligent applications because they are extremely difficult for machine processing [Gašević et al., 2006]. Controlled languages have been used for knowledge representation, though; this was analyzed in chapter 4.
9.5. Ontologies

Though usually not primarily defined as a knowledge representation technique, ontologies can accomplish most of their functions. Ontologies became popular in the context of the Semantic Web. The Semantic Web intends to represent web content in a machine processable form, where meaning is well defined by standards, to overcome some of the limitations of the actual web, specifically searching for and extracting knowledge. This claims for an interoperable infrastructure based on standard protocols universally accepted [Antoniou and Harmelen, 2008] [Fensel et al., 2003]. Though the main purpose of ontologies is to provide support for the semantic web, and this conditioned its conception and design, a successful ontology provides an invaluable resource for any domain of knowledge.

In this section we explore the essentials of ontologies and their potential value as a form of knowledge representation for the purpose of this thesis.

9.5.1. What is an ontology

The word “ontology” made its way from Philosophy into Computer Science; there are a number of definitions in both fields, in many colours. For the purpose of this thesis, an ontology is a formal, exhaustive description of the terms used in a domain of knowledge, and the relationships held among these terms [Princeton University, 2011] [Antoniou and Harmelen, 2008].

An ontology provides a vocabulary, the names used for referring to the terms in the domain of knowledge on which the ontology originates. There are many kinds of vocabularies; the
9. Knowledge Representation, state of the art

simplest one is just a list of the terms used in a domain. A glossary is a list of terms and their definitions (as used in the domain). A thesaurus is a list of terms and their synonyms. A controlled vocabulary provides a list of terms with an unambiguous interpretation of each. Ambiguity is dealt with by not accepting more than one meaning for each term, or distinguishing different meaning for a term in some clear way.

The vocabulary of an ontology is organized into categories of objects in a hierarchical structure or taxonomy, where more general concepts subsume specialized concepts into subcategories. A category may be decomposed into overlapping or disjoint subcategories. An individual is identified as an object of a category. Properties may be assigned to objects. Physical composition may be indicated by part-of relationships, and composite objects may be defined, imposing adequate restrictions. In this way, terms and their relations come to be described by logical statements, and rules may be established for their combination and relations. This relatively complex scheme reflects the complexity of language. An ontology limits the vocabulary to be used in the domain, but also makes it determinate. Besides, it makes clear the relationships held among terms. This enables consistency checking among terms, and helps towards interoperability and knowledge sharing [Gašević et al., 2006] [Russell and Norvig, 2010]. Reasoning can be performed on an ontology, provided it is expressed in a language that allows it.

Ontology editors do exist to assist users in the creation of ontologies, expressing them in several ontology languages. One of the best known is Protégé, a free, open source, Java based application from Stanford University [Stanford, 2012].

9.5.2. OWL

OWL, the Web Ontology Language, is a language for ontology development published as a Recommendation of W3C, the World Wide Web Consortium, the main standards organization for the World Wide Web. Last version is OWL 2, dated 2009. OWL 2 allows to define classes, properties, individuals, and data values. OWL 2 ontologies can be stored as Semantic Web documents, which means it can be expressed in the common formats promoted by W3C for the Semantic Web. In particular, OWL can be serialized into RDF/XML syntax. RDF, Resource Description Framework, is another W3C specification for conceptual modeling which provides vocabulary and syntax for defining classes and properties.

OWL can be expressed in several different forms of syntax: OWL2 Functional syntax closely follows the structure of an OWL ontology; OWL2 XML defines an XML format for serialization also closely following the OWL ontology; Manchester syntax is human readable, in a style similar to frame languages; RDF/XML uses RDF an XML for serialization; RDF Turtle, also based on RDF and apt for serialization, is preferred over RDF/XML for its readability. For exchange, OWL can be expressed in RDF, OWL2 XML, or Manchester.

Meaning can be assigned to OWL2 ontologies in two ways: direct model-theoretic semantics, called OWL2 DL, based on Descriptions Logics, and RDF-Based Semantics, an extension for RDF-Semantics which treats OWL2 ontologies as RDF graphs. In RDF-Based Semantics some additional inferences may be obtained from annotations, which have no formal meaning in OWL2 DL.

OWL 2 is a very expressive language, both from the computational point of view and for the users’ needs. This makes it difficult to implement well. Several profiles have been defined by imposing some limitations on expressiveness, in particular disallowing disjunction and negation, to simplify reasoning: OWL2 EL ensures polynomial time complexity in rea-
9.5. Ontologies

9.5.3. Ontologies and UML

As can be seen in the former descriptions, Ontologies and OWL have much in common with Object Oriented Modeling and UML, the Unified Modeling Language. Though UML is targeted at modeling a domain following an object oriented paradigm, and includes constructs to describe behavior, there is significant overlap in the expressiveness of object oriented models and ontologies. A comparison of their respective capabilities and shortcomings can be seen in [Mika et al., 2007].

The Object Management Group (OMG) has produced a specification entitled Ontology Definition Metamodel for the application of Model Driven Architecture to ontology development. The OMG adopts a definition of ontology which recognizes different levels of expressivity:

"An ontology defines the common terms and concepts (meaning) used to describe and represent an area of knowledge. An ontology can range in expressivity from a Taxonomy (knowledge with minimal hierarchy or a parent/child structure), to a Thesaurus (words and synonyms), to a Conceptual Model (with more complex knowledge), to a Logical Theory (with very rich, complex, consistent, and meaningful knowledge)" [OMG, 2009].

This definition and the analysis of usage scenarios and goals led to six metamodels grouped in three categories:

- formal logic languages:
  - DL, Description Logics, informative.
  - CL, Common Logic, a declarative first-order predicate language.

- structural and subsumption descriptive representations, less expressive than logic based languages, but of common use in the Semantic Web:
  - abstract syntax for RDF Schema (RDFS), for vocabularies. RDF Schema is a set of classes in RDF which provide the basic elements for the building of ontologies.
  - OWL, for describing ontologies.
  - TM, Topic Maps, for describing topics. Topic maps is a standard for the representation and interchange of information with provisions for the findability of resources (ISO/IEC 13250:2003).

- traditional software engineering approaches to concept modeling:
  - UML2, UML models.
  - ER, Entity Relationship modeling.

UML supports the definition of profiles, an extension mechanism to adapt standard UML to different situations in a natural, seamless way. An UML profile is a customization of standard UML for a particular domain or platform defined in accordance with UML extension mechanisms. The Ontology Definition Metamodel (ODM) includes three UML profiles: RDF, OWL
and Topic Maps. These profiles enable the use of UML notation and tools in these technologies, achieving one of the ODM goals: provide the knowledge representation community with a sound semantic based notation, with the further purpose of bringing closer the software and logical visions of representing information [OMG, 2009].

9.5.4. The case for ontologies

Ontologies exhibit a number of convenient characteristics which explain their popularity in the Semantic Web and also in Knowledge Representation. These features turn up from some of the many definitions of an ontology, as reviewed in [Gašević et al., 2006].

An ontology deals with concepts, an abstract view of things in “the world”, which must be understood as a subject area or domain of knowledge. The construction of an ontology explicitly declares the concepts of interest for the subject area which the ontology is representing, as well as their relationships. This is often called a conceptualization.

In an ontology, the concepts and relations are declared in a formal way, using a formal language. In a formal language, the symbols and formulas which combine them stand in explicitly stated syntactic an semantic relations to one another. This is called a specification.

The above terms account for the most cited definition of an ontology, stated by Tom Gruber in 1993: an ontology is the specification of a conceptualization. The definition is as concise as can be, but rather cryptic to non initiates.

The fact that an ontology is expressed in a formal language leads to another important feature: an ontology is machine readable, which means a computer program can interpret the formal language and convert it into a data structure which the program can handle. If the formal language used to represent the ontology allows it, the ontology may also be machine processable. This means the concepts and their relations may be handled by a computer program to answer questions on the subject area, based on the concepts and relations included in the ontology (what the ontology “knows”). In any case, an ontology is not a computer program, it cannot execute by itself, it is only declarative; a program can read the declarations and execute with the data obtained.

An ontology specifies the semantic connections between the concepts, the meaning of the relations held among concepts. This opens the way to establish connections among ontologies, such as relating ontologies for Car and Truck to the more general Vehicle. This allows for an evolutionary development of ontologies.

An ontology allows some forms of reasoning: relations among concepts are not all immediate, and complex queries may be posed as to the way several concepts are related, such as if there is a path from concept A to concept B, and which concepts are in the middle, or if concepts A and B have a common generalising concept.

An ontology is a shared understanding of some area of knowledge: a group of individuals or a community, the users of the ontology, agree on the meanings attributed to the concepts and relations expressed in the ontology. This solves many difficult problems of communication, but requires some kind of consensus, an agreement reached by a group as a whole. However desirable, this agreement may be very difficult to achieve, the more so when trying to integrate different communities or to coordinate related but different subject areas.

The fact that knowledge included in an ontology is agreed upon and shared, plus the possibility of connecting ontologies, give some ground to the dreamy vision of an all encompassing family of ontologies covering all human knowledge. The idea is so appealing that a general textbook on Artificial Intelligence such as [Russell and Norvig, 2010] deals with Knowledge
9.6. Other proposals and resources

The former sections reviewed the most general, commonly accepted techniques and languages for Knowledge Representation. In this section some other works and proposals are reviewed, among the many available, selected for their being considered promising for the needs of this thesis.

As an example of use in a particular domain, a comparison between ontology and semantic networks for knowledge representation in a medical context is realized by [Salem and Alfonse, 2008]. The authors build and ontology in OWL and a semantic network in Prolog to represent knowledge in lung cancer. Their conclusion is that the OWL ontology is better suited for their purpose, but the comparison shows some aspects of interest for the use of semantic networks: they are easy to visualize, and it is easy to cluster related knowledge. The lack of semantics and standards for semantic networks are pointed out as the main shortcomings.

A graphical, UML based notation for ontology modelling is proposed by [Dillon et al., 2008]. The notation includes ontology classes, instances and properties. Classes are differentiated in disjoint, decomposition and partition. Properties may be data type, annotation or object. Properties may be functional, inverse functional, symmetric or transitive. Restrictions categories are quantifier, cardinality, and hasValue. The authors point out that OWL does not properly differentiate aggregation from generalisation, a distinction easy to understand by most audiences. Since they devised their notation with OWL in mind, they implement aggregation as partition. They provide UML notations for classes, properties, and instances in the former variants; relationships are generalisation, partition (to account for aggregation), decomposition, disjoint, and association. The illustrative example is quite readable provided a notion of the former entities is adequately mastered.

Commonsense knowledge bases. Commonsense knowledge refers to the facts and information that any person may be expected to know. A commonsense knowledge base intends to capture all the general knowledge people is expected to know, in a way usable by a machine. A commonsense knowledge base is then a form of ontology. The Cyc project is one such ontology, comprising an extensive knowledge base and inference engine. Cyc uses a logic-based knowledge representation, which provides expressiveness without the ambiguities of general language, precise meaning, reasoning, and a usage neutral representation which renders knowledge more usable. An online browser allows to query and interact with the knowledge base, but requires some familiarity with the terms and schema under which Cyc stores knowledge. The Cyc project started in 1984 and has done considerable progress to date. Cyc is developed by Cycorp, but the Cyc Foundation manages and grows the OpenCyc ontology under a Creative Commons license [Cycorp, 2012].

Topic maps can be seen as a standardized form of semantic network. Concepts are expressed as topics, their relations as associations; the role of a topic in an association may be indicated in the association. Occurrences are resources relevant to the topic, identified by an URI (Uniform Resource Locator). Topics, associations and occurrences can all be typed. Types are themselves topics. The set of types allowed is called the ontology of the
topic map. Topic maps, including their ontologies, can be merged automatically into a wider topic map. Scoping on names allows for the assignment of different names to the same topic [Pepper, 2010].

**Controlled English to Logic Translation (CELT)** is a system which allows the user to write queries and statements in a restricted form of English, translates this text into logic and extracts or adds information to a knowledge base. CELT uses the terms in an upper ontology called SUMO (Suggested Upper Merged Ontology), provided in logic and translated into OWL, extensively tested, extended with domain specific ontologies, and mapped to the WordNet lexicon; this work was manually done. Terms are thus related to definitions in much the same way a term evokes concepts in the human mind [Pease and Li, 2010]. PhraseBank is a corpus of English phrases classified into patterns according to their semantics, which allows the mapping of the component words to entries in the WordNet. Other strategies help to overcome some limitations of CELT syntax, in particular verb tenses, anaphoric references, conjunctions and implications [Pease and Fellbaum, 2003].

An improved version of Concept Maps includes all concepts in frames, allows for long labels, wants all links to have arrowheads in the direction of the connection, calls for verb expressions in links to make propositions readable, adds media, each concept appears only once, organization needs not be hierarchical [Ahlberg, 2004]. These rules limit the rather loose conception of concept maps, rendering them more predictable for representation and comparison. Many proposals exist to introduce different degrees of formalism into Concept Maps.

**Multilayered Extended Semantic Networks (MultiNet)** is a knowledge representation system based on Semantic Networks where nodes are embedded into layers with different attributes. Each node represents a single concept, and relations among them must be expressed in a predefined set of semantic primitive relations and functions, described in second order predicate calculus. The classification of nodes into sorts results in a conceptual ontology which defines the domain and the value restrictions of relations and functions. The system is supported by several software tools. The proposal intends to overcome the limitations of logic based and semantic network approaches by first identifying the criteria a useful knowledge representation system must comply with, and then attempting a design to satisfy them. Aspects such as universality, cognitive adequacy, homogeneity, interoperability, completeness and consistency are claimed to be better met by MultiNet than by other systems, were they logic, frame or network based. Descriptions of acceptable relations comprise name, formula, definition, mnemonics, question patterns, and commentary. Layers where nodes are embedded are characterized by several attributes, such as the "facticity" of an entity, degree of generality, quantification, determination of reference, cardinality, type of extensionality, and variability. A distinction is made between immanent knowledge, which is the semantic content of a concept, and situational knowledge, related to the embedding and use of the concept in a special situation which does not change the meaning of the concept in itself. Three software tools are provided, considered essential for the system to be effective: a knowledge engineering workbench with graphical representation, a translator from natural language, and tools for the lexicographer. The system is based on philosopher Ludwig Wittgenstein's idea of a language game, the shared conceptual parameters that enable the identification and production of signs [Helbig, 2002]. A complete description of MultiNet and its language related foundations can be read in [Helbig, 2006]. Its roots, goals and development make the proposal a very attractive one, conferring semantic networks a formalism they naturally lack, based on the interpretation of signs, with a sound theoretical foundation. The price paid is, as usual, a more complex sys-
tem, requiring consensus, acceptance by a large community, and continuity in development. The last publications registered at the site of MultiNet at the FernUniversität in Hagen is dated 2006, which casts a doubt as to the acceptance received by the system.

A **Semantic Graph** represents semes as nodes and semantic cases as links. *Semes* are the smallest unit of meaning recognized in semantics, and *semantic case* is the role implied in a relation between semes, such as agent, patient, perspective, characteristic, and the like. Nodes are labeled by semes, links are labeled by case, arrows indicate the direction of the relation. Labels for nodes are an open category, but labels for cases are limited in different ways according to discourse, genre, author, or other criteria. The list provided by Hébert contains 15 primary semantic cases, claimed to account for most textual semantic structures. Several relations between graphs are considered [Hébert, 2011]. The proposal is closely related to text. The authors provide examples of small pieces of text converted into semantic graphs, which show a level of complexity similar to MultiNet.

### 9.7. Discussion

First Order Logic (FOL) is a powerful, recognized language for Knowledge Representation (KR); a wide spectrum of knowledge may be expressed in FOL, with proved, ample reasoning capabilities. Notwithstanding some criticism and limitations as to becoming a universal knowledge representation language, FOL support is by all means desirable. For our prospective users, and probably for many others, FOL is very difficult to deal with as such; some kind of user interface may be needed, or FOL may be thought of as the substratum of another language or schema. Use of quantifiers, and careful definition of the domain of discourse may be challenging for many real world applications. The reasoning capabilities of FOL, though desirable, are not strictly needed for the purposes of this thesis.

Description Logic (DL) merits similar comments, though it may be closer to languages more apt for human communication, such as frames. DL is already at the basis of other KR languages and schemes.

Rules are a simple paradigm apt for several purposes and domains, in particular those of diagnosis. Rules are individually easy to understand, though they may become confusing when combined. Rule based applications are generally implemented through a user interface. Conversion from natural language into rules, and the complement of visual representation, bring the language closer to the non expert user. Rule based applications may be considered for some domains and purposes, e.g. diagnostic knowledge bases.

Frames, in its simplest form, are intuitively accepted as form filling. If precise semantics are added, or Description Logics is introduced, the model becomes harder to use and understand.

As a primary conclusion, some type of visual language seems the most adequate for the purpose of this thesis.

Concept maps are well known and widely accepted in Education. There are proposed applications to create or extract concept maps from texts [Villalón and Calvo, 2011]. The main shortcoming of concept maps, and perhaps for some purposes one of its main virtues too, is their lack of a formal, or at least, a generally accepted, syntax and semantics. Each community, tool or author may thus end into the relative isolation of their own options and definitions. Sowa’s Conceptual Graphs is an outstanding example of formalization, which comes at the price of a complexity slightly over what a non expert might tolerate, even with some instruction.
9. Knowledge Representation, state of the art

The virtues of ontologies are desirable for any field of study; their main difficulty lays in their construction, and earning wide enough acceptance. Where an ontology exists, and it may be complemented as needed with a moderate effort, a knowledge base may be implemented and used as a reference. The difficulty of the language persists, though. Some implementations of OWL may bring knowledge representation instances within reach of a non expert user, in particular some of those related to the OMG Ontology Definition Metamodel, UML and Topic Maps.

Commonsense knowledge bases or ontologies, such as Cyc, could be exploited by Education in a wide range of general subjects. However, the Cyc structure is rather complex to use and even to understand. Concepts and individuals are called constants, and it may not be easy to pinpoint the adequate constant. Looking for “kitten”, for instance, shows “kitten” is not a constant, and “cat” is not a constant either; “pet” produces constants “DomesticPet” and “NonPersonAnimal”, where “DomesticPet” is a specialization of “DomesticatedAnimal”, generalized by "TameAnimal", generalized in turn by "NonPersonAnimal" (Cyc tutorial). Although all this hierarchy makes sense for an upper ontology, it becomes cumbersome for general Educational needs.

The Controlled English to Logic Translation (CELT) project attempts to bridge the gap between language and knowledge representation, to help ontology creation. CELT is a general language that must be specialized and extended into a specific domain, including also a domain specific lexicon [Pease and Fellbaum, 2003]. This is by no means a minor work; use of CELT as the basis for a domain specific application is considered an interesting alternative. No clear evidence of activity could be found for this project: no homesite or institution could be located, a search in Google with terms 'CELT "Controlled English Translation to Logic"' showed 14 results in the last year, most of them containing citations to CELT, but no one of them centered on CELT.

MultiNet and Semantic Graph are formal semantic network approaches where typical relations among concepts are identified, which results in predictable ways of relationship among concepts. Both systems exhibit a non trivial level of complexity, as could be expected from any system with their expectations of generality.

9.8. Conclusions

Our ideal system might be a universal ontology of all human knowledge supported by a formal system capable of inference, where portions could be selected by different criteria, visualized with variable degrees of detail, and compared to other KR instances of the same portion for a measure of distance. Conversion to and from a somewhat restricted but readable natural language would complete the picture.

Needless to say, the present state of the art is far from such a system, not only for the size and complexity of the enterprise, but also for human limitations: it is very difficult to reach even the minimal consensus on significant subject areas, reality is differently perceived, in most areas knowledge is not complete or certain, and also not beyond subjective opinions, even in some areas of the hard sciences. The ideal picture, however dreamy, serves as a distant goal to mark the way along which some progress can be done.

Several of the formerly analyzed proposals may be successfully used as a basis for a system adequate to the purposes of this thesis. Some formalism beyond the basic concept maps paradigm is needed, at least to support some simple inferences, and to compare different KR
instances. Graph like visualization is required. The challenge seems to be how to achieve all this and at the same time keep the system manageable by teachers and usable as a learning resource.

The variety of potentially useful proposals, their soundness, formalism, and expressive power, illustrate the far reaching complexities of the problem of knowledge representation: maybe these systems reflect in some way how the mind works, or the mind may work in a totally different, hard to conceive, way. This makes an option difficult. Carefully steering clear of such a deep pit, but keeping aware of its existence, several knowledge representation schemes may be considered for our purposes, selecting one or the other according to the type of knowledge to be represented, and the operations to be performed on it. This will be the subject of the next chapter.
10. Knowledge Representation for Education

Abstract. The less formal knowledge representation languages may be variously used. A knowledge representation scheme will designate a knowledge representation language used in a certain, specifically declared, way. In Knowledge Representation for Education, visualization of knowledge is a primary concern, the “reality” to be represented is the subject matter of a certain field of knowledge, supported by a weak ontology, with limited reasoning capabilities and modest storage and processing requirements. Logic based Knowledge Representation languages require a graphical user interface and the integration of several commonly accepted assumptions. Rule based systems may be applied to some domains, but other kinds of structuring may be more adequate. Frames may be tried for stereotyped knowledge with no formal semantics, at the risk of showing the domain as too strictly categorized. Concept maps are the most adequate for Education, but some agreement on syntax and semantics must be achieved. An existing ontology may be advantageously used, specially if it can be easily visualized. Existing proposals of restricted natural languages are too cumbersome for general educational purposes. Though concept maps are the expected choice for Education, some formalization is required. An ontology expressed as a topic map following OMG’s UML profile for Topic Maps would be our first selection, yet very difficult to obtain. Some simpler, reachable schemes, though limited, may provide usable results and pave the way to more ambitious structures.

The representation of knowledge for educational purposes has its own needs and restrictions, which the knowledge representation techniques and languages analyzed in the previous chapter may meet better or worse. In this chapter, knowledge representation is analyzed from an educational perspective, evaluating the adequacy of different proposals. This results in the selection of a small number of knowledge representation proposals as most adequate for use in Education; the final selection will depend on the educational objectives and subject matter of each learning unit.

10.1. Terms

The term “knowledge representation” may refer to a field of study, a category of entities conceived to express or record knowledge, one of those individual entities (an instance of the class of knowledge representations), or even the task of representing knowledge by making a drawing, writing statements, or any other way. Whether Knowledge Representation refers to a field of study can be inferred from the context, but anyway the term in this sense will be capitalized. A knowledge representation instance is expressed in a knowledge representation language, as formerly discussed. However, some knowledge representation languages admit
to be used in different ways, the more so when the language is loose in its specification (e.g. concept maps). Without attempting to introduce new terms in an already crowded field, we will use the term knowledge representation scheme to mean the use of a knowledge representation language in a particular way. In this way, we will be defining different knowledge representation schemes, for which corresponding sublanguages may be compiled. As previously stated, the correlation between a sublanguage, in particular its rules of production, and way its sentences will be transformed into some form of knowledge representation (the knowledge representation scheme to be used) must be simultaneously, or at least, coordinately defined.

10.2. The five roles of Knowledge Representation

The now classic work of [Davis and Szolovits, 1993] recognizes five different roles in a Knowledge Representation (KR):

1. A surrogate: the representation stands in place of something else, which we may call “the world”, be it tangible, abstract or imaginary. A representation is built for practical reasons, as a help to understand or act in the original domain (“the world”). As a surrogate, a representation will always be incomplete and inexact.

2. A set of ontological commitments: the selection of the things represented and their relations determines what is important for some purpose, and leaves aside the rest. The usefulness of a representation will depend on how well the selection is made, if it includes all the essential aspects its purpose calls for.

3. A fragmentary theory of intelligent reasoning: there is a limit on the conclusions that a particular representation will permit. Again the usefulness of a representation will depend on the reasoning capabilities included, if they are enough for the purpose desired.

4. A medium for efficient computation: knowledge representations are usually built for some kind of machine processing. The complexities associated with human knowledge, in particular when elaborate reasoning is required, very easily lead to proposals of considerable computational complexity, hence very costly, impractical or even impossible with present day means.

5. A medium for human expression: humans must understand the knowledge representation in some way; they may only be interested in the conclusions, or they may be interested in all the details involved in the process.

This partition in roles may be used as a guide to evaluate and compare different knowledge representation proposals. For each purpose and for each domain the requirements of these roles may vary wildly. In the following section we discuss the relative importance of each role towards a Knowledge Representation for Education and Assessment.

10.3. The roles in KR for Education

According to purpose, the five roles of Davis will have different significance. In this section, we discuss the five roles in the perspective of a knowledge representation scheme to be used in Education and Assessment: to teach and learn, and to capture what a student knows on
10.3. The roles in KR for Education

A certain subject. The characterization of each of these roles according to our purpose may help to determine the most promising KR schemes, and eventually to define our own.

A knowledge representation instance is a surrogate of the subject matter we intend to teach, learn or assess; this is an unavoidable fact. The importance of the recognition of a knowledge representation instance as a surrogate resides in its inherent limitations: only partial aspects of “the real thing” are modeled, this model may not be accurate for all purposes, all models are, to a certain extent, deceitful. The subject matter of any area of knowledge is itself a model of the real world, with its own limitations. For our purposes, “the real world” will be the subject matter to be taught, learnt or assessed in the different educational acts of everyday instruction. In other words, we will be taking the subject matter of any field of knowledge, as conceived by the instructors using the system, as unquestionably true in their own view. The purpose of our knowledge representation scheme will be to reflect this view as completely and accurately as possible.

The extent of the ontological commitment of a knowledge representation scheme will depend on a number of factors, such as the field of knowledge, the course level and aims, the state of the art of the field in question, what the audience already knows or agrees upon, the vision of the instructors. A complete ontology from Thing downward to the tiniest concepts and relations in the domain of interest sounds as the ideal of perfection, but besides the non trivial fact that complete ontologies are very hard to build, they easily become very complex even for relatively simple domains. If there exists an accepted ontology for the domain, it will probably be usable, and the best option, but this situation will be an exception more often than not. In most educational situations the audience is expected to have some mastering of the terms used in the domain, which are part of what is to be assessed. The concept of a weak ontology may offer a practicable solution. A weak ontology is rather loosely defined as an ontology not so rigorous as to allow machine processing and inference [Wikipedia, 2012b].

The reasoning capabilities expected from our knowledge representation scheme are basically two: determining if a certain concept is present, and determining if two concepts hold a certain relationship. A rather simple graph can cope with these requirements. Other capabilities of interest are: determining if a concept is a descendant of some other concept (subsumption), if a concept may be considered a part of a larger entity (aggregation, or composition, or both if distinguished), and determining if there is a path from one concept to another (connectivity). For these requirements, the type of relations must be considered, and some processing such as path finding called for. Even for these more advanced features, the reasoning capabilities required are modest.

Though our knowledge representation scheme requires digital support, the storage and processing needs are modest by current standards. Any modern personal notebook has extensive storage, and processing required is simple enough, as stated previously. Path finding in a large graph may be demanding, but we will not be interested in very long paths (concepts too far from each other, only slightly related). Efficient computation is not expected to be a hindrance.

Human communication is the primary concern of our proposal. For assessment, the KR scheme must not only be able to compare two KR instances, but also show where the differences lie. A major goal of this project is to propose a solution to eAssessment of use to learning and teaching. Hence, the KR scheme must also provide knowledge visualization in a clear, intuitive way. The structure of concepts and relations must be apparent: if concepts are organized in a taxonomy, the hierarchy must be evident; if concepts have a precedence relation, as in a road, a story, or a set of instructions, the sequence must be traceable; if there is a central idea, the
10. Knowledge Representation for Education

main concept must stand out at first glance.

Summing up, the five roles of the KR we are looking for may be characterized as follows:

- visualization of knowledge is a primary concern: the structure of concepts and relations must be apparent.
- the “reality” to be represented will be the subject matter of a certain field of knowledge, as conceived by the instructors running the system.
- if there is not an accepted ontology for the field in question, a weak ontology should suffice for most educational needs.
- reasoning capabilities are limited to determining concept and relation occurrence as essential; subsumption, aggregation and connectivity as desirable.
- storage and processing requirements are modest.

In the following section, the studied KR proposals are discussed as to their adequacy for the fulfilment of tasks described by the former analysis of the KR roles.

10.4. Discussion: selecting a KR language

The characterization of the five roles of a KR for Education provides a perspective from which to evaluate the different KR languages and proposals analyzed in the preceding chapter. Almost any proposal may succeed in the proper domain, with the adequate audience. A general educational environment of factual knowledge for high secondary or tertiary studies is our reference for scoring the different proposals.

Logic based languages and proposals provide no visualization capabilities. A Graphical User Interface which merely transforms the logical statements into some form of graphics is not enough: the complexities of Logic based proposals will be present, in graphical form, but with all their subtleties and implications. Some form of elaborate user interface is required, which means that, for human communication, it is this user interface that must be evaluated, rather than the underlying language. Even in natural language interfaces to Logic based proposals, considerations such as “Agatha is a person” or “no one else lives in Dreadsbury Mansion” are required to complete a set of logic statements. Most likely, these statements will be taken for granted by most people. Though “Agatha is a person” may be required, most people will assume Agatha is a person if not specified otherwise, while “Agatha is a cat” will normally be expected to be established. Concerning the people who live in Dreadsbury Mansion, most people will assume no one else lives there, or equivalently that the only people who live in the house are the ones mentioned. If someone else lived in the house and it were not explicitly said, most people would feel some form of joke or cheat is present, both of which are assumed to be absolutely excluded in an educational environment. Besides these commonly assumed statements, Logic based proposals require the user not to forget anything, not even things most people will normally take for granted. In a rough way, it can be said that Logic based proposals require some kind of “over specification” on the part of the user, which is unsuitable for our purposes. All this does not mean that a sound, powerful languages such as FOL or DL must be excluded; rather it must be complemented with a number of accepted assumptions which account for the said “over specification” required to complete a set of statements. These assumptions must be explicit, accepted, and modifiable by the community of users. Agatha
10.4. Discussion: selecting a KR language

is a priori assumed to be a person, but it must be possible to state that she is a cat; this behavior would be similar to our own assumptions in everyday life. This sort of “everyday life assumptions” is by no means an easy task: they are difficult to capture, and difficult to reflect in logic in a consistent manner. Costs and benefits will depend on audience, purpose, and subject matter, among other aspects.

Logic based proposals inherently lack visualization, are hard to read, require minute specification; they offer great reasoning power, which is desirable but not required. Anyway,

FOL an Logic based KR languages may be used provided 1) a graphical user
interface is available, and 2) subtleties required by Logic statements are included
as generally assumed and not required to be specified by the user.

A rule based language such as RuleML Datalog \[\text{RuleML, 2011}\] includes a visualization scheme where rules are seen as a sort of syntactic tree. Modeling of a domain where rules play a substantive role may profit from this proposal. A possible drawback is that the visualization provided may be not be much clearer than the expression of the rules in natural language, or even worse than in some form of structured natural language sentences. Some subject areas may require students to produce the rules relating some concepts of the domain, or even apply the rules to some given set of concepts and values. In these cases, the reasoning capabilities of a rule based language may be called for. Following the former example, a question may be posed as to the amount of discount corresponding to a premium customer buying a luxury product. The variability of questions such as this make the reasoning capabilities of a rule based language desirable: the student must produce an answer, which will be contrasted against the one inferred by the system. Considering Education in general, domains where rules are the core of knowledge do not seem prevalent. Even so, the assessment of knowledge by purely applying rules is probably not the best way to go.

Rule based systems may be applied to domains where rules are the core of knowledge, but probably other kinds of structuring may be better for knowledge assessment.

Frame based languages admit a reasonable visual representation, and may be successfully tried in domains where knowledge is stereotyped enough as to clearly differentiate concepts (classes) and properties (attributes). Adding restrictions on properties (relationships among slots) may require some over consciousness of the underlying frame structure. Introducing precise semantics will require a more elaborate user interface, with its own difficulties. The rather rigid structure of frames may tend to give a view of the domain as pigeonholed or too strictly categorized. In classification oriented domains, such as systematics in Biology, frames may be most adequate.

Frame based languages may be tried in domains where knowledge is stereotyped, and no formal semantics is required. Showing the domain as too strictly categorized when it is not is an educational risk.

Visual languages seem the most adequate and widely usable knowledge representation languages for the purpose of this thesis, and for Education in general. Concept maps are almost universally recognized as adequate for use in Education, and there are myriads of works and proposals. Their versatility comes at the price of a lack of definite syntax and semantics, which forces the community of users to adapt a predefined scheme to their purposes or agree upon
one of their own. A number of proposals to provide concept maps with formal support exist, some rather loose and relatively easy to apply, others strict, logic based and consequently more complex for the user. Loose, easy to use proposals may sound arbitrary to other communities, which will prefer to use concept maps to their own taste, thus loosing in sharing, reuse and communication.

Concept maps are the most adequate for Education, but the lack of universally accepted syntax and semantics compels the users to adopt or define one.

Availability of an ontology for a domain implies a superior degree of consensus and organization of knowledge in the domain, at least concerning terms, which is no minor achievement. A knowledge representation for such a domain should be elaborated around this existing ontology. The task may be challenging depending on how this ontology is expressed, and if there is a usable interface to it. Perhaps the most promising proposals for the purpose of this thesis are OMG’s ontology definition UML profiles, in particular the one concerning Topic Maps. UML was conceived for visual representation and formal specification, with a profile definition mechanism which makes it apt to bridge the gap between visual and more obscure representations.

If there is an accepted ontology for the domain, it should be used. An interface or an UML profile is required for visualization. OMG’s Topic Maps UML profile seems the most promising choice.

Use of sublanguages, controlled natural languages, or some restricted form of natural language for knowledge representation falls within the scope of this thesis. There are several very promising proposals, most of them too complex for ordinary educational use, and at the time of writing they lack enough public support to bet on them. Some of these restricted languages are readable and also formal and powerful for knowledge representation and reasoning. Learning to write in any of these restricted languages requires some training, more than can be expected in an educational environment not specifically oriented to language and writing. On the positive side, they allow for a variety of complex forms of knowledge representation, may map to some form of Logics and consequently support inference. Once more, these complexities place the proposals at a distance from our target audience. Evolution of these restricted languages may change the view, and bring their use more general, thus making the effort of mastering them worthwhile. Pease and Li make an interesting argument towards some restriction in writing: use of language is easy for people and difficult for machines, then let people write, with some guide to remove the ambiguities of language; this may be a small price to pay in view of the benefits it brings [Pease and Li, 2010]. For the purpose of this thesis, these proposals are still too complex to be acceptable.

Existing proposals of restricted natural languages for knowledge representation are too cumbersome for general educational purposes.

The former discussion attempted to assess the potential usefulness of the different knowledge representation languages for the purposes of this thesis. Study and comparison of knowledge representation proposals led to the conviction that no unique scheme can do well for all situations, but on the contrary, different domains, purposes and audiences call for different solutions. This variety may, to a certain extent, be contemplated in a sublanguage relatively simple to use, and a visual, intuitively clear knowledge representation scheme. Different sublanguages and knowledge representation schemes, simple and intuitive, can be conceived for different instruction levels, subjects, and purposes.
10.5. Conclusions

The main contribution of this chapter was:

- *a comparative study and evaluation of different Knowledge Representation techniques and languages* in their usefulness for Education and Assessment, based on the traditionally recognized roles of Knowledge Representation first stated by Davis [Davis and Szolovits, 1993].

The variety of proposals for knowledge representation is an indirect sign of the complexity of the phenomenon of human knowledge. No single scheme may account for all the ways the human mind is able to grasp knowledge. Domain, audience, and purpose are determinants towards the selection of a knowledge representation scheme. In all cases, simplicity is a must: teachers and students should be able to understand a knowledge representation instance almost at a glance, with little or no previous guidance, perhaps with just a simple quick reference card.

Concepts maps are the expected choice in Education, and there are good reasons for it. Some form of minimum formalization is required, as to syntax and semantics. Specific ontologies are more complex, but provide a sound terminological support for a field of knowledge; they should be used, if available and a suitable interface makes them accessible. Topic Maps are an option which provides many useful features, besides being a standard and apt for reuse and merging; it may be the option of choice for a long standing project.

An ontology expressed as a topic map following OMG’s UML profile for Topic Maps would be our first selection, but conditions for development may be difficult to meet. Though a small team may advance several steps towards such a project, the required developments for practical use lay still too far to reach. This said, some simpler, reachable schemes should be proposed. They will be limited in scope and capabilities, but may be conceived as steps to the ideal of ontologies expressed as topic maps, or some other equivalent scheme.

In the following chapters, several schemes for different situations are proposed. Humans are able to understand broad concepts in just one word. In everyday situations it is not necessary to trace the concept of “kitten” to “domestic pet”, “domesticated animal” and so on; people know what a kitten is. In most situations, “kitten” may be taken as a concept which requires no further analysis as to its nature. But if students are expected to know the domestic cat is called *Felis catus*, of genus *Felis*, family *Felidae*, order *Carnivora*, class *Mammalia*, phylum *Chordata* and kingdom *Animalia*, some other kind of knowledge representation is required. This kind of differentiation in audience and purpose leads to experiment different schemes of sublanguages and corresponding knowledge representation schemes.
11. Syntax based Knowledge Representation

Abstract. The simple syntax based sublanguage formerly proposed was based on the semantic interpretation of syntactic constructs. The purpose of such sublanguage was to achieve determinism for knowledge representation while keeping close to the everyday use of the language. Determinism was achieved by limiting the kinds of sentences used in natural language to only a few with unique semantic interpretations. For knowledge representation the syntactic components must be given a semantics: nouns are considered unique entities; action verbs indicate what the nouns do or receive, linking verb 'be' attaches a property to a noun or declares its state, and is not shown in the graph; adjectives and adverbs are linked to nouns, adjectives or verbs as complements to their meaning. Several specific situations arise from this interpretation which must be clarified, transformed or avoided for a consistent graphic representation. A prototype application which validates the sublanguage and builds a semantic graph from it helps illustrate these situations. As an example, prepositional phrases can be brought to the semantic graph in different forms, each with its own advantages and inconveniences. Redundancy arises when a sentence repeats information already stated in a former sentence; different possible solutions are analyzed. Concept clustering, not implemented in the prototype, offers a way to aggregate information related to a concept so that it can be referred to as a unit. Weights can be assigned to nodes and edges in a reference graph. Sentences of a text answer can be validated in the sublanguage and compared against a reference graph. The items from the sentence successfully recognized in the graph add their weights and contribute to the mark given to the answer. A complete example is built from a piece of news, rewriting its content in the sublanguage; a reference graph is built from this transcript, and some sentences are submitted to it for marking. This offers a vision of the use of the sublanguage and knowledge representation scheme in a miniature case study. The experimental testing of this scheme of sublanguage and knowledge representation showed a number of issues to be aware, and which require more elaborate solutions. On the simple examples tested, these issues were circumvented with some crude decisions and ad-hoc conventions, but a working example could be built quite easily with the help of the prototype application developed. Though an application apt for field testing requires more work, it is considered to be within reach of a small team in a half year project.

This chapter deals with the extraction of a knowledge representation instance from text written in a sublanguage, and its comparison against a reference knowledge representation instance of the same type. A complete example shows the assessment of free text answers by the use of a sublanguage and knowledge representation, as is proposed in this work.
11. Syntax based Knowledge Representation

The most immediate form of transition from text to knowledge representation is based on syntax: the structure of the sentences defines the relations among the concepts represented by the words. This chapter deals with the transformation of a text written in a syntax based sublanguage into a form of knowledge representation which follows closely the syntactic structure of the sentences. The transformation is not as straightforward as might be supposed; problems such as redundancy in the text must be solved in the graph, some transformations are required to keep the representation coherent, and dependency on the grammar puts some limits on the generalization of the solutions.

11.1. From text to graphs, a first approach

The following analysis is based on the rules of English05, a sublanguage with only simple sentences, and no use of conjunctions, developed in chapter 8. This often results in sentences with some redundancy. While simple sentences are a usual recommendation towards simple writing, a cautious use of conjunctions can be safely allowed.

The lexicon loaded into the prototype application was EnLong3k; this lexicon is based on Longman Communication 3000, as described in chapter 7. Use of Longman 3000 instead of GSL was preferred because Longman 3000 selects only the most common parts of speech for each word, instead of recording all known parts of speech for each word, as GSL does. In this way, Longman 3000 greatly simplifies part of speech tagging.

Knowledge representation is done in the form of a semantic graph, with lexical entities as nodes, and their syntactic relations as arcs.

Figure 11.1.1. shows a syntax based semantic graph for the sentence the brown fox jumps over the lazy dog. This is a simplified version of the sentence the quick brown fox jumps over the lazy dog, a phrase which contains all the letters in the English alphabet, used in typography to test fonts. Lexical categories are differentiated in shape and colour; direction of the arrows indicate dependency. The active verb jump articulates the sentence into fox and dog; these three words give the essential meaning of the sentence. Adjectives brown and lazy add meaning to nouns fox and dog, respectively. Though function words are considered structural rather than semantically charged words, preposition over adds some meaning, indicating the relative positions of the dog and the fox during the jump. Determinant the has been suppressed in both instances with no loss of meaning.

![Figure 11.1.1.: A simple semantic graph based on syntax](image)

The node in the graph with numerical information is reserved to show the weight of nodes and edges recognized in the graph when assessing a text answer.

A more elaborate example results from the child’s song *This is the house that Jack built*:
This is the horse and the hound and the horn
That belonged to the farmer sowing his corn
That kept the cock that crowed in the morn
That waked the priest all shaven and shorn
That married the man all tattered and torn
That kissed the maiden all forlorn
That milked the cow with the crumpled horn
That tossed the dog that worried the cat
That killed the rat that ate the malt
That lay in the house that Jack built.

(From Wikipedia, This is the house that Jack built, retrieved on 2012-06-08)

The writing of the song in sublanguage English does away with the poetry; this is part of the price we pay for deterministic, processable writing. However, the meaning is preserved:

Jack built [a house]
the malt lay [in the house]
the rat ate [the malt]
the cat killed [the rat]
the dog worried [the cat]
the cow [with the crumpled horn] tossed [the dog]
the forlorn maiden milked [the cow]
the tattered man kissed [the maiden]
the shaven priest married [the man] [to the maiden]
the cock crowed [in the morn]
the cock waked [the priest]
the farmer sowed [the corn]
the horse belonged [to the farmer]
the hound belonged [to the farmer]
the horn belonged [to the farmer]

These are all simple sentences; the structure this is...that was avoided, leaving only the bare facts, as expected from the use of our sublanguage. These simple sentences are easy to represent alone. When taken together, some assumptions are explicitly or implicitly made: in particular, each noun is assumed to refer to one and the same item in all sentences. When speaking of the man, it is the same man in all the sentences. A graph representing this bunch of sentences is shown in figure 11.1.2.

The following sections describe the strategy and the assumptions employed in the construction of these graphical representations, analyze each of the main syntactic structures, point out their difficulties for knowledge representation, and propose some possible solutions. The purpose is twofold: obtain a clear, easy to understand knowledge representation, and preserve the natural use of the language, even if limited by the rules of a sublanguage.

11.2. Syntactic constructs and their representation

This section describes the semantics of the sublanguage: once a lexicon and set of rules have been chosen, the conversion of text written within these constraints into a form of knowledge representation must be defined. The present analysis is for a syntax oriented knowledge representation, in which a semantic graph is built based on the syntactic constructs of the language.
Figure 11.1.2: The house that Jack built
Our example sublanguage is English05, the experimental sublanguage defined in chapter 8. For each lexical category and syntactic construct, their transformation into a semantic graph is described, discussing the potential ambiguities, redundancy and other problems that arise in the conversion.

In what follows we assume a piece of text written in a sublanguage is to be represented in a semantic graph. All the sentences in the text refer to the same domain of discourse, and indeed constitute the domain of discourse: no other texts or inputs are known or required for this representation, there are not texts which refer to the same concepts with other words, or use the same words for other concepts.

### 11.2.1. Nouns

Nouns, proper or common, are considered to refer to unique entities. Different sentences containing the same nouns will result in the addition of information related to these nouns. These sentences all add to the same *dog* entity:

- the brown dog chased [the cat]
- the cow tossed [the dog]
- the big dog belonged [to the farmer]
- the dog lay [in the house]

Figure 11.2.1 shows a semantic graph for these sentences. Different references to *dog*, such as the brown *dog*, or the big *dog*, all result in adding to the unique *dog* entity in the domain of discourse.

Situations where there are more than one dog requires a different name. Natural language expressions like this *dog*, that *dog*, the other *dog*, or the second *dog*, are not considered adequate identification of the entity in this scheme. Identification may be done by naming:

- Rufo is a dog
- Wanda is a dog
- Rufo chased [the cat]
- Wanda chased [the cat]

Figure 11.2.2 shows the graph for these sentences.

Considering the brown *dog* as an identification of a particular dog is not possible in this scheme; there is not a property assignment to the entity as to recognize it by its properties. This can be done by introducing concept clustering, an idea which is discussed later in this chapter.

Proper nouns designate an "object", a particular instance of a category. In the sentence *Rufo is a dog*, the proper noun *Rufo* is an identifier of a particular dog, and *dog* designates the category of all dogs. Though in natural language usage no difference is made between the name of a category and the name of an individual in this category, the distinction is understood by the speakers. The apparently tautological "this dog is a dog" is actually saying "this thing I perceive is an specimen of a dog"; the calling of an object by the name of the class to which it belongs is accepted and understood by the speakers in everyday usage of the language.

The former scheme of identifying individuals of a category by the use of the linking verb be is not free from trouble, though. The sentences
11. Syntax based Knowledge Representation

Figure 11.2.1.: Nouns are unique

Figure 11.2.2.: Rufo and Wanda are different dogs
11.2. Syntactic constructs and their representation

Rufo is a black dog
Wanda is a brown dog
Rufo chased [the cat]
Wanda chased [the cat]

are no longer valid descriptions, unless two categories of dogs are recognized, the brown dogs, and the black dogs. The following sentences offer a better approach to knowledge representation

Rufo is a dog
Wanda is a dog
Rufo is black
Wanda is brown
Rufo chased [the cat]
Wanda chased [the cat]

Figure [11.2.3] shows graphs for both the former groups of sentences. The first group results in a category dog with contradictory properties black and brown; the second correctly assigns properties to individuals. Though this assignment of properties by adjectives seems overly complicated, its causes are not in the sublanguage or the knowledge representation, but in the indetermination of the everyday use of the language, which does not distinguish when a word means a category and when the same word means an individual. A sublanguage and knowledge representation scheme based on individuals and categories is described in chapter 12 as a follow up from classes and objects in Object Oriented Programming.

11.2.2. Adjectives

Properties may be attached to nouns indirectly by preceding the noun with an adjective, or by explicitly assigning the property to the noun using linking verb be. Both forms lead to the same result, the assignment of a property to an noun. In this experimental scheme, the resulting graphs are not exactly the same; the syntactic origin has been kept for study, but the relations are actually the same. Though the former discussion on categories and individuals affect the assignment of properties, it is not a difficulty with adjective assignment, but with the adequate distinction of categories and individuals.

English05 accepts one or two adverbs before an adjective, as in

Wanda is very comfortably seated [on the chair]

Consecutive adjectives require special treatment: one of them should be chosen as “head” in a dependency structure, which forces the second to be subordinate to the first, while both adjectives should be attached to the noun. The situation is depicted in figure [11.2.4]. The upper graph represents the sentence

the quick brown fox jumps over the lazy dog

where adjective brown is incorrectly drawn as depending on adjective quick. The middle graph comes from the sentences

the quick fox jumps over the lazy dog
the brown fox jumps

Here the adjectives are correctly dependent on the noun, but action verb jumps is reiterated. This is correct, since the declaration in the second sentence is not semantically equivalent to the first.
Figure 11.2.3.: Wanda and Rufo are dogs of different color: wrong and right assignment of properties to individuals.
The graph at the bottom comes from
the quick fox jumps over the lazy dog
the brown fox jumps over the lazy dog

Again adjectives are correctly assigned, but a redundancy appears: jumps over is unnecessarily and incorrectly repeated. While treatment of this kind of redundancy ultimately leads to the non trivial problem of graph matching, some common situations can be dealt with as individual cases. The control of redundancy in text to graph transformation is discussed in its own section in this chapter.

11.2.3. Verbs

Traditional grammar classifies verbs into linking verbs and action verbs. Linking verbs are used to relate a subject to a complement; they do not express an action. Seem, resemble, become, are some common linking verbs. Recognition of linking verbs may be done by thinking of them as an equal sign '='; or by substituting them in a sentence by some form of verbs seem or be; if the meaning remains essentially the same, it is indeed a linking verb [Altenberg and Vago, 2010].

The most used linking verb is be. Linking verb be is used to attach properties to a subject in various ways. English05 accepts linking verb phrases composed of the linking verb plus an adjective phrase, a noun phrase or a prepositional phrase, as in the following examples.
the book is yellow
Jack is a doctor
the car is in the garage

Adjectives and prepositional phrases may be attached to nouns in a noun phrase, without a linking verb:
the yellow book is on the table
Jack cleaned [the car in the garage]

Both forms should lead to the same assignment of properties, i.e. the same establishment of associations in the semantic graph. Figure 11.2.5 shows the representation of both cases: though the placement is different, the relations are the same. To achieve this coherency of behavior linking verbs are omitted in the graph. This is a design decision, in which the difference of expression in the sublanguage (or in the natural language, as it may be) leads to the same semantic interpretation in the knowledge representation.

English05 does not accept adverbial complements on linking verbs. Linking verbs are weak in meaning; English05 keeps them limited to their role of nexus. Active verbs accept adverbs as complements, attached to the verb, up to the number of two:
the car climbed very fast [the huge hill]

Main complement of linking verbs is called attribute in English05; in English grammar it is usually called a predicative expression or simply a predicative. Tag “Pred” strongly suggests “predicate”, so “attribute” was chosen instead, as it may be called in Spanish grammar. As stated, English05 attributes can be a predicative nominal, a predicative adjective, or a predicative prepositional phrase.

Active verbs are usually distinguished into transitive and intransitive verbs. This distinction was done in English 01, a more “grammar conscious” sublanguage. English05 resulted from doing away with the distinction for the sake of simplicity, both to users and compilers of the lexicon. English05 can adopt word lists with no need to differentiate verbs into transitive and intransitive, a difficult enterprise since many verbs can act as both. The joint treatment of transitive and intransitive verbs as action verbs also does away with the concepts of direct and indirect object. In English05 direct and indirect objects merge into a wider category of verb
Figure 11.2.4.: The quick brown fox: consecutive adjectives.
11.2. Syntactic constructs and their representation

Figure 11.2.5.: Linking verbs: attachment of properties
complements tagged 'CC'. An action verb phrase can accept up to 4 of these generic complements. Besides attributes, linking verbs also accept up to 3 of these generic complements. CC complements can be prepositional phrases, noun phrases or adjective phrases.

The distinction of direct and indirect object is not devoid of meaning: something is lost with this simplification, which may not be admitted for some purposes. The knowledge representation will not distinguish the receiver of the action (the direct object) from the receiver of the direct object (the indirect object), as in

the carpenter repaired [the small chair] [for the baby]

where the small chair and for the baby will be both tagged CC, as well as any other complement, such as yesterday and in the morning in the following sentence:

the carpenter repaired [the small chair] [for the baby] [yesterday] [in the morning]

Even so, the semantic graph produced contains enough information to clear up the meaning with a little more on the part of the user, whose understanding of verb repair enables her to trace the direct object as the receiver of the action (the small chair), and the indirect object as the receiver of the direct object (the baby).

A similar sentence with different meaning shows nesting of prepositional phrases, and illustrates the role of the square brackets, in this case to distinguish nesting from adjacent prepositional phrases:

the carpenter repaired [the small chair] [for the baby] [yesterday [in the morning]]

Both sentences are illustrated in figure 11.2.6

Phrasal verbs, so common in English, may be accepted in the lexicon by the use of an underscore, like this: look_up, look_after, look_for. This transforms the phrasal verb in a single word verb with no loss of meaning, and a bit more effort in writing. There is no simple way to detect phrasal verbs as such. An arguably convenient way would be to preprocess the text, recognize phrasal verbs, and automatically transform them into underscore equivalents.

11.2.4. Adverbs

English accepts adverb groups of one or two adverbs as complements of an adjective or a verb, as in

Jack is very comfortably seated [on a chair]
the cat ran very quickly upstairs

In figure 11.2.7 the order of adverbs, as by the arrows in the arcs, is unnatural: in very quickly, adverb very is a complement of adverb quickly, hence the head of the verb group is in effect quickly; since the arrows point to the head of each constituent, the diagram suggests otherwise. This is a limitation of the dependency tree builder, which in its present version is not able to adequately select the correct head in a group with several words with the same tag (in this case Adv Adv).

11.2.5. Prepositions

Prepositions are used before a noun to indicate direction, location or time. Prepositions are also used before a noun to relate it to a verb, an adjective or another noun [Altenberg and Vago, 2010]. Some of the former example sentences showed the use of prepositions in prepositional phrases. In dependency grammars, the head of a prepositional phrase is
11.2. Syntactic constructs and their representation

11.2.6: Active verbs, objects and complements

11.2.7: Adverbs
11. Syntax based Knowledge Representation

the preposition itself. When nesting prepositional phrases, this leads to a chain of prepositions, as in sentence

Jack walked [to the house [in the corner [of the street]]]

depicted in the upper diagram of figure 11.2.8. Changing the rules for dependency, indicating the noun as the head in a prepositional phrase, establishes the links among the nouns, with prepositions as accessory.

This option fails when different prepositional phrases refer to the same noun, as in

the carpenter lives [in the house]
the cat ran [into the house]

As can be seen in figure 11.2.9, prepositions in and into are both linked to house; it is not clear to which of the prepositional phrases the preposition belongs. This situation is correctly solved if prepositions are designated as the head of prepositional phrases. Though not as intuitive, taking the preposition as the head of the prepositional phrase leads to a more accurate representation.

11.2.6. Determiners

The different types of determiners compel to decide on each type according to the purpose of the sublanguage, and decide which treatment to give them:

- Articles (a, an, the) are not semantically charged, and can be dispensed with in the knowledge representation. They must be included in the sublanguage, though, for the sake of readability.
11.2. Syntactic constructs and their representation

- **Demonstratives** (this, that, these, those) point to something or someone. English05 does not accept demonstratives. To support demonstratives, their meaning and importance must be defined beforehand; it must be stated what it means to refer to something as *this* or *that*, if it is worthwhile to represent them in a semantic graph, and if they will have some value when coming to mark concepts and associations recognized in a graph. For declarative or descriptive writing, demonstratives may well be left out.

- **Possessives** are of little value if pronouns are not accepted, except for possessive 's, which is of interest. Though English05 does not provide support for possessive 's, it should probably be included in a production sublanguage. A construction with belong to can be used in a sublanguage with no possessive possessive 's. Though less direct, this makes the relation explicit.

- **Quantifiers** do have a meaning, and inclusion of them requires a definition of their exact significance and value; they may call for special production rules to support them. Quantifiers like *enough, few, little, most, much, several, some,* may lack precision for adequate treatment, and may even call for the differentiation of countable and uncountable nouns (things and substances). Besides excluding them completely, the safest way may be to include some carefully selected quantifiers, with a clear definition of meaning and usage.

11.2.7. Conjunctions

Though none of the experimental sublanguages developed in this work support support conjunctions, a measured introduction of them is considered desirable, to avoid unnecessary repetitions in the text. As an example, the use of the comma and coordinating conjunction *and* to list several nouns, as in *Rufio, Wanda and Toppy are dogs,* is relatively safe and easy to implement.

Coordinating conjunctions are the less dangerous, but even so their use requires some care; connecting nouns may be relatively easy to deal with, but connecting verbs or sentences may bring unexpected complexities. Subordinating conjunctions are difficult to deal with; the idea of concept clustering in a later section of this chapter may provide a way to achieve similar expressiveness more accurately.
11.3. Control of redundancy

Redundancy occurs when a sentence repeats all or some of the constituents already written in a former sentence. Control of redundancy means to recognize those repeated parts, which have been already included in the knowledge structure, add to the graph only the new nodes and edges, and correctly attach them to the nodes already in the graph. As a simple example, a bunch of sentences like

- the man kissed the maiden
- the tattered torn man kissed the maiden
- the man kissed the forlorn maiden
- the tattered torn man kissed the forlorn maiden

must produce the same result as a single sentence

- the tattered torn man kissed the forlorn maiden

Control of redundancy is dependent on the semantic interpretation of the sublanguage for knowledge representation. For each new sentence, we must detect which parts are already represented in the graph, and only add new knowledge. The following types of redundancy are recognized:

- Nouns represent concepts, hence a noun, whether common or proper, must be unique in the graph. Mention of a noun already in the graph is a form of redundancy, and it must be understood that this new mention refers to the same noun already in the graph.

- Adjectives attached to nouns add to the meaning of those nouns; adjectives linked to nouns may appear several times in the graph, each time linked to a different noun. Redundancy appears when a constituent (adjective, noun) in a new sentence is already in the graph, which means the graph already shows the same adjective linked to the same noun. Nouns are unique, adjectives are unique only when linked to the same noun.

- Adverbs may be attached to verbs, adjectives, or another adverb; adverbs may appear several times in the graph, each time linked to a different verb, adjective, or adverb. Adverbs are unique only when linked to the same verb, the same adjective or the same adverb.

- Action verbs express the action of a subject (a noun phrase); since the action may be differently modified by adverbs, a verb may appear several times even for the same subject.

- Prepositions may be repeated only when the element complemented by the prepositional phrase (e.g., a noun or verb) is new in the graph, or when the noun phrase object of the preposition is new in the graph.

The former conventions for representation, and the analysis of the production rules in the grammar, dictate the rules to detect and control redundancy for each new sentence that is to be added to the graph. The following points explain some of the situations where redundancy control is required, both to avoid repetition and for coherency of the semantic content of the graph. In a new sentence,

- A noun already in the graph is the same noun as in the sentence; it will not be added.

- An edge adj1/Adj -> noun1/Nc will be added to the graph only if there is not an edge adj1/Adj -> noun1/Nc already in the graph. If noun1 exists, but not adj1, the adjective
### 11.3. Control of redundancy

Adj1 and the edge Adj1/Adj -> noun1/Nc are added to the existing node noun1/Nc. Sentences the brown dog chased the cat, the big dog ate the food, both add adjectives brown and big, but next sentence the carpenter owns the big brown dog does not, while new sentence the brown cat sleeps on the roof adds brown to cat.

- if the edge Adj1/Adj -> noun1/Nc has been added by qualifying a noun, as in the brown dog chased the cat, or in a sentence with a linking verb, as in the dog is brown, treatment should be the same; if the information is already there, it should not be repeated, no matter the syntactic structure from which it was added to the graph.

- edge Adv1/Adv -> verb1/Vact will be added if it is not there for the same subject, the same verb, and one complement of the verb. Sentences Jack walked slowly to the house, Jack walked slowly with the girl, add all elements as different instances except Jack. The semantic content of these sentences is different, they have only the subject in common. Adding sentence Jack walked slowly to the house with the girl should do away with the structures created with the former two sentences, since the semantic content of this new sentence is not the same as the semantic content of the first two.

- addition of edges Adv2/Adv -> Adj1/Adj, Adv3/Adv -> Adv4/Adv should not happen if the sentence reiterates the complemented adjective or adverb as complements themselves of existing nouns or verbs. After sentences the quick fox ran to the woods, sentence the dogs chased the very quick fox should add very to quick, but the very quick fox is brown would only add brown. Sentence the fox is very quick should not add anything else.

- the same action verb is included in different instances only if the new sentence does not contain the same subject and one complement already in the graph. Sentences the man built the house, the man built a ship will produce two instances of verb built, but adding the man built the house yesterday should only add complement yesterday. Adding sentence the man carefully built a house should add carefully to instance of verb built related to house only, and not to ship.

The former examples show control of redundancy is a delicate matter. In the general case, concepts related in a dependency tree are to be matched to a semantic graph reflecting the same kinds of dependency in its nodes and edges. This is an instance of the Graph Isomorphism Problem, a recognized difficult problem, for which a number of algorithms exist [Skiena, 2008]. In our case, the problem is restricted to detect if the graph produced by the dependency tree built from the sentence can be paired to a subgraph of the semantic graph, even partially. Addition of a new sentence to the graph would proceed as follows:

1. build the dependency tree for the sentence;
2. build a graph S from the dependency tree;
3. compare graph S to graph G, the target semantic graph.
4. add to graph G those parts of graph S which are not in G.

This purely algorithmic solution to redundancy may prove difficult to implement, since treatment of nodes is not uniform (e.g. nouns are unique, other types no). Development of a strategy to control redundancy based on syntax and the semantics of the conversion from text
11. **Syntax based Knowledge Representation**

to graph may be the best approach. Contradictions introduced by successive sentences, as the ones formerly analyzed, may be difficult to deal with whatever the approach. Ultimately, the writer must be aware of the knowledge she is conveying into the graph.

In the scope of the project, control of redundancy is a plus to the writer, a facility to enable a less strict way of writing. A carefully composed text will smoothly lead to a semantic graph without the need of elaborated procedures to do away with redundancy. The prototype application developed is able to read a text and draw the semantic graph in no time; adjustment of a text can be done with little effort in several trials; an incremental development of the text, sentence by sentence, is also a practicable approach.

For the recognition of a text in an existing graph, redundancy can be made irrelevant by only accepting sentences with all its elements present in the graph. This allows for the recognition of a sentence in the graph even if not all complements are given. However, it does not recognize a sentence if just one element is not present in the graph. This may sound too hard, but may be argued in favour for an assessment situation, i.e. “do not write what you are not sure of”.

### 11.4. Concept Clustering

Tools for representing graphs of medium to large size usually allow to expand or collapse nodes: a bunch of nodes and edges are aggregated into a container node identified by a name; when the node is collapsed, only the container node is shown; when the node is expanded, its contained nodes and edges are shown. This results in a layered organization of the graph: an upper layer with all nodes collapsed shows less detail, which is good for a general view; a lower layer shows the details around a node; general relations can be established among groups of nodes by linking the collapsible nodes. The idea is the essence of mind mapping, where a central concept is related to a number of others in an essentially radial topology.

Gathering a bunch of concepts and relations around one central concept, in what we might call a concept cluster, is useful both for organizing and escalating the representation of knowledge in a subject.

Support for concept clustering in the sublanguage requires the possibility of declaring a concept as the main concept in a text or, what is the same, by stating that the concepts and relations in the text relate to or deal with a certain main concept. In practice this can be done by introducing a notation for titles: the title indicates the main concept, the text that follows contains the concepts and relations associated with the main concept in the title. Though the use of titles is usual in all kinds of literature, their use in a sublanguage for knowledge representation must adhere to rules: besides employing the accepted notation, a title must always be a concept around which relations to other concepts are established, and the following text must establish these relations. The visualization of the graph representing the concepts and relations declared in the written text can be done at the time of writing, in an interactive way; the author is not wading through her text all in the dark, but seeing the visual representation of her writing just as it goes.

Once the idea of using titles for concept clustering has been adopted, a hierarchy of titles for several levels of clustering is an immediate extension.

The idea of concept clustering is relatively simple to implement. It is also relatively easy to work with, if some care is given to the organization of knowledge in the target field. However, as simple and tempting as this proposal may be, it calls for very careful planning to be put to work. In an educational environment, a taxonomy can be presented to students or
11.5. Other patterns and conventions

collaboratively built, as a sort of regulated table of contents. Texts can then be written for each title.

Representing knowledge for each title can be done in an straightforward way, as with the prototypes in this work. Gathering together all these instances poses its own challenges, though.

In a sentence like

\textit{Cinderella went [to a party [in the palace [of the Prince]]]}

\textit{Cinderella, party, palace, Prince} may all be concept clusters, and a piece of text written for each, such as a description of the palace, with its rooms, towers and cellars. Of course other relations may exist between these concepts, different from the one stated in the sentence, or even between one of the main concepts and a concept which is subordinate in another cluster: Cinderella lives in a house with her stepsisters, this house also has rooms, like the palace; the stepsisters go to the party of the Prince, hence they are related to him and the palace... the list is endless. The handling of all these complexities will be more on the side of the users than on the facilities that can be given to them. All the systems and proposals discussed in this work assume the extension of an educational unit; size of the knowledge structures is determinant to define clustering levels, or if concept clustering is required at all.

11.5. Other patterns and conventions

A doubt that persistently emerges when dealing with any kind of restriction in writing, were it a mere style guide or a severely cut controlled natural language in all rigour, is this:

\textit{Will anybody accept to write in this way?}

The answer depends strongly on two main points: how ‘unnatural’ the rules may result to a user, and the engagement of the users with the proposal. Though many usual writing patterns can be detected in natural language texts, and eventually transformed into their equivalents in meaning, this is not a way to follow in all cases, not only for its difficulty, which is considerable, but for one of the main purposes of using a sublanguage, which is to keep writing simple, clear and direct. As a trivial example, we might say “the purpose of a gun is to shoot a bullet”, “a gun is made to shoot a bullet”, “a gun was conceived to shoot a bullet”, and other endless variations. Our sublanguage would perhaps accept only “a gun shoots a bullet”, and this may be the only one form required for successful declarative writing in many fields and for many purposes.

An effort must be made to keep writing in a sublanguage as simple and clear as possible; both virtues go hand in hand. The sublanguage shall be simple to write, and simple to read. The sublanguage can be enhanced to accept different patterns of expression, even though they mean the same thing, and effectively detect their equivalent meaning to represent it in a knowledge structure. But a strong engagement in simple, clear writing, excludes elaborate patterns and keeps only a very small number. The accepted patterns will be those most people will use and understand when freely expressing factual knowledge. Though the language is primarily a means of communication, many social, conventional and educational habits drive speakers and writers away from the plain, factual expression sublanguages try to recover.
11. Syntax based Knowledge Representation

11.6. Distance measure

Marking of a free text answer is done by the recognition of sentences in an existing semantic graph. The existing semantic graph acts as a reference for all correct answers, including all the knowledge the student is expected to have in an answer to the proposed question. The graph resulting from each sentence in the answer is compared against the reference graph. Each node and each edge in the reference graph has a weight, given by the instructors according to the importance of the concept or relation in the unit of knowledge which the reference graph represents. If the sentence in the answer can be matched against the reference graph, the weights of its nodes and edges, recorded in the reference graph, are added together; the resulting number is the mark contributed by this sentence to the whole answer. The sum of the marks obtained in all the sentences of the answer is the mark given to the text answer.

In our present scheme, a sentence must match entirely into the reference graph to get the marks. Giving marks for parts of a sentence is an alternative, but is considered less effective for an assessment of knowledge.

The use of graph distance measure for marking a text answer is best seen in use, as in the example described in the following section.

11.7. A complete example

The following example was taken from the first link in section “In the news” from the main page of the Wikipedia; it is not a piece of text selected with any particular criteria. The original text ran as follows:

The 2012 transit of Venus, when the planet Venus appeared as a small, dark disk moving across the face of the Sun, began at 22:09 UTC on 5 June 2012, and finished at 04:49 UTC on 6 June.[1] Depending on the position of the observer, the exact times varied by up to ±7 minutes. Transits of Venus are among the rarest of predictable celestial phenomena and occur in pairs separated by eight years:[2] the previous transit was in June 2004, and the next pair of transits will not occur until December 2117 and December 2125. (From Wikipedia, Transit of Venus, 2012, retrieved on 2012-06-06)

The next sentences contain essentially the same information rewritten in English05:

the TransitOfVenus2012 began [at UTC2209] [on Day20120605]
the TransitOfVenus2012 finished [at UTC0449] [on Day20120606]
Venus appeared [as a small dark disk]
Venus moved [across the face [of the Sun]]
the exact times varied [with the position [of the observer]] [by a maximum [of seven minutes]]
the transits [of Venus] are very rare
a transit [of Venus] is a predictable celestial phenomenon
the transits [of Venus] occur [in pairs [with an interval [of eight years]]]
the PreviousTransitOfVenus occurred [in Year2004]
the NextTransitOfVenus occurs [in Year2117]
11.7. A complete example

The former transcription does not provide a solution for any of the difficulties pointed out in the former sections of this chapter, such as the handling of dates, control of redundancy, identification of instances. The transcription was carried on as a test of the capabilities of the primary approach of an experimental sublanguage in recording the knowledge of a small piece of text casually chosen, i.e. the first piece of news on the Wikipedia on the day of writing. The deficiencies were not corrected by adequate provision and processing, but circumvented by transformations which may eventually provide a basis for more formal solutions. The information is there, though, and a semantic graph can be drawn from these sentences, such as the one in figure 11.7.1.

An analysis of the different transformations used in the transcription follows:

- the transit of Venus in year 2012 is a particular event, indeed the subject of the text. An aggregation of Venus, transition and year 2012 might have been defined as a unit; this is an example of concept clustering as formerly described. The alternative employed in the example is the assignation of a name to the phenomenon, TransitOfVenus2012. These solutions can be combined: TransitOfVenus2012 may well designate an arrangement of planet, Venus, transit and year 2012.

- dates were handled as common nouns, which is accepted by the grammar and does not require any particular handling, though a particular handling of dates may be desirable. In this example, a day is associated to a date in ISO format, as in Day201120605; when only a year is given, a notation like Year2012 is used.

- besides additions of words not included in the base lexicon, other modifications were required, e.g. maximum is recorded in Longman 3000 as an adjective, and here it is used as a noun. In this instance, the lexicon was modified accordingly. An alternative would be to use a construction like the exact times varied [with the position [of the observer]] [by maximum eight minutes].

Word maximum is included in the dictionary as both an adjective and a noun; it is up to the users of the sublanguage to prefer one or the other, or expect to be asked for a disambiguation if both forms are included. Some advanced tool such as bigram analysis may be included to determine the part of speech best suitable for a sentence.

- the graph layout of subsequent adjectives was formerly pointed out. In this example no attempt was made to correct the anomaly, since it does not seriously compromise the interpretation.

The text chosen is a piece of news; no formal management of knowledge is done. In an Astronomy course we would expect to see that Venus is a planet, that a transit has a beginning and an end, astronomical would be used instead of celestial, and hence a transit would be an astronomical phenomenon. An adjustment of the lexicon, the arrangement of the learning material in related units, and the collaborative writing of the original texts will contribute towards a more strict description of knowledge in Astronomy. More formal schemes, also based in the use of a sublanguage, but addressed towards different knowledge representation techniques, is analyzed in chapter 12.

So far, a reference knowledge structure has been created from text, which will act as a reference for assessment. A possible question for this piece of knowledge might be “what do you know about the transit of Venus this year?”, or more formally “briefly describe the transit
of Venus in 2012”. Students will write their answers trying that their text was recognized in the graph. They know the graph, because they have been actively present during the creation process, since the agreement on the lexicon, through the writing of the text, to the building of the semantic graph. They have acquired the knowledge represented in the reference graph during a learning activity, they took an active part in their building; in the assessment instance it is the same old graph which will give them their marks.

During the same process, the students have got in touch with the grammar, studied the rules, examined a number of examples, and practised the writing in the sublanguage. In a collaborative environment, the students should be responsible for most of the texts which lead to the building of the reference graph. They must know enough of the sublanguage to write a piece by themselves, with little failure in validation by the grammar.

Assignment of weights to the nodes and edges in the graph is a task generally reserved to the instructors, though it is desirable that the students know the criteria employed to assign those weights. They may also have taken a part in determining the relative importance of the different elements, which is in fact the selection of the essential concepts of the subject. In this example, assignment of weights was done on a very simple manner, assuming nouns are more important than verbs, and then come adjectives and adverbs. In this way, weights were assigned as follows:

- Noun, proper: 6; nouns, common: 5
- Verbs, active: 4
- Adjectives: 2; adverbs: 1
- Other categories: 0
- Edges: 0

In our tests, the reference graph is shown blurred, which may be good for training; in a real assessment instance the reference graph would be invisible. Each new sentence, if valid and successfully mapped into the reference graph, illuminates in color the recognized nodes and edges. A text box shows the sum of weights for recognized entities, the total weight in the graph, and the percent weight recognized, which is the mark given to the answer.

Figure 11.7.2 shows the recognition in the graph of a first sentence in the text answer:

\textit{the times varied \{with the position \{of the observer\}\}}

The weight obtained by this sentence is 19 in a total of 53, or 33.03%. For a clearer diagram, only part of the whole reference graph is shown in the figure.

Figure 11.7.3 shows the recognition of the following two sentences, second and third in the answer:

\textit{the transits \{of Venus\} are rare}

\textit{the transits \{of Venus\} are very rare}

The third sentence repeats the second except for the adverb. The repeated parts did not originate any new weight, but the adverb, new in the third sentence, added to the recognized weight, which is now 33 in 56, or 58.93%. Again, only part of the whole reference graph is shown in the figure.

As these examples show, a complete sentence with partial knowledge is accepted, and recognized if correct. Further sentences can add new knowledge, repeating the main structure of the sentence if necessary; only new knowledge in the sentence will add to the recognized weight. In all cases, a complete sentence validated by the grammar is required, and it is not recognized in any of its parts if one single element cannot be recognized.

The percent weight recognized in the graph from the sentences in the answers can be taken as a mark. Weights can be assigned to each node and edge; generally a subset of nodes and
11.7. A complete example

Figure 11.7.1.: The transit of Venus in 2012
11. Syntax based Knowledge Representation

Figure 11.7.2: The transit of Venus assessment: situation after the first sentence of the answer
11.8. Conclusions

The main contribution of this chapter was:

- **a syntax based knowledge representation scheme**, defining the transformation of syntactic structures into nodes and edges in a semantic graph. The syntactic structures were those of English05, the sublanguage formerly developed (chapter 8), but the transformation is based on ideas of dependency grammar which can be applied to different sublanguages. A distance measure allows for the recognition of knowledge coming from new text in the existing reference graph.

We started with a semantic interpretation of the syntax based sublanguage previously designed. The interpretation of the different lexical categories towards their representation in edges will have more value, when they are the essential concepts and relations in the unit of knowledge represented in the reference graph.

Once again, this is an experimental prototype for proof of concept; its many drawbacks have been formerly pointed out, as well as alternatives to solve them. A more elaborate prototype is required for field testing in a classroom. The prototype used to produce these examples is described in chapter 14.

Figure 11.7.3: The transit of Venus assessment: situation after the second and third sentences of the answer
a semantic graph showed several conventions, transformations or design decisions must be implemented or adopted to arrive at a consistent, intuitive, widely acceptable representation of text written in the sublanguage.

As formerly stated, the design of the sublanguage, its semantic interpretation for knowledge representation, and the knowledge representation model must be developed in coordination. The experiments described in this chapter defined a knowledge representation scheme for text written in English05, the experimental sublanguage formerly developed, with no modifications; the lexicon loaded was based on Longman Communication 3000. No attempt was made to introduce in the prototype application any solution to the problems formerly analyzed in this chapter, but some rather crude conventions employed instead, such as the ones for identifying the 2012 transit of Venus as an event, or the handling of dates, days and years. All these detected problems call for more elaborate solutions, but the convenience solutions adopted here may be refined into more formal proposals. Within these constraints, a working example could be built quite easily, just with the help of the prototype application developed. Though an application apt for field testing requires more work, it is considered to be within reach of a small team in a half year project.
12. Knowledge Representation Schemes

Abstract. The experimental syntax based sublanguage developed was based on the semantic interpretation of syntax constructs into a knowledge representation in a straightforward way. However, knowledge representation need not be linked to the syntactic structures of a language. Human knowledge can be classified as declarative, procedural, structural, heuristic, commonsense, approximate or uncertain, and meta knowledge. Each of these categories call for different, possibly overlapping, sets of accepted relations among concepts. In analogy to network topologies, different “knowledge topologies” may be roughly recognized: mind maps recall a star topology, concept maps a mesh, taxonomies a tree, procedures and sequences of facts a line, stories a bus. Drawing on these sources, different sublanguage and knowledge representation schemes are possible. UML and the classes and objects paradigm mark the difference between instances and objects, properties and values, and allow to state the value of a property in the general case, but accept a different value for this property in a certain individual. Representation of time is no trivial task, but for the purpose of a story of sequential development a simple timeline based scheme may suffice. A set of instructions or the description of a procedure call for the use of imperative sentences in sequence, complemented by if-then constructs to include alternatives. Knowledge evolving around a central idea suggests a mind map represented as a tree with the central idea at the root. If there are several outstanding concepts, a mesh topology with a clearly stated but simple semantics may confer a light formality to the usual concept maps. Topic Maps can be used simultaneously to represent knowledge and locate resources; they follow a standard, and are suitable for the Semantic Web. The complexity of Topic Maps call for professional development in a full project, but their potential benefits make the try worthwhile. In most cases, the analysis of these examples show they are within reach of a small team of instructors and developers, and can be approached incrementally.

This chapter explores the possibilities offered by different schemes of sublanguage and knowledge representation models, each apt for different types of knowledge and educational objectives. The syntax based sublanguage and knowledge representation scheme developed in chapters 8 and 11 is just one possible scheme. Different types of knowledge call for different sublanguages and knowledge representation models. A simplified description of the different types of knowledge and their typical forms of expression is of help to define the syntactic constructs and patterns to be recognized in the sublanguage, as well as the knowledge representation language best suited for the domain. Knowledge can be arranged in different structures or “topologies”, such as a radial network for a mind map, or a tree for a classification in categories. Drawing from other disciplines, such as network theory, graphs and Object Oriented Analysis, different sublanguage and knowledge representation schemes may be conceived. The nature of different subject areas may prove some of these alternatives schemes to
be better suited for learning and assessment than a syntax based scheme.

12.1. Types of knowledge

Cognitive psychology states humans manage different kinds of knowledge, and identified different categories of knowledge, such as procedural, declarative, heuristic, structural, uncertain, commonsense, ontological, meta knowledge [Gašević et al., 2006]. A different, more formal classification, can be seen in [Russell and Norvig, 2010]. These different kinds of knowledge may call for different approaches for their representation. Categories in a common classification are given below, in a very simplified form [Gašević et al., 2006] [Wikipedia, 2012b], together with some comments on their expressive requirements:

- **procedural knowledge**, also called **imperative knowledge**, is about how to do something, how to accomplish a task. Though this kind of knowledge is sometimes not expressible, being just the knowledge a person has about how to do something, when expressed in language it uses imperative sentences, as in requests or commands.

- **declarative knowledge**, also called propositional knowledge or descriptive knowledge, state what is known about a topic or problem. In language, it is expressed in declarative sentences, such as a statement.

- **heuristic knowledge** includes rules that apply to the solution of a problem, but give no guarantee of arriving at the best solution. They are based on a deep knowledge of the subject and experience in handling its problems, but are expressed as simple rules. This kind of knowledge is similar to procedural knowledge, and can be expressed in imperative sentences; the difference lays in their quality and certainty.

- **structural knowledge** describes the organization of knowledge in a topic, by stating relationships between different pieces of knowledge, such as part_of, kind_of, or inclusion in a set based on some common property. This kind of knowledge calls for the recognition of some syntactic constructs to have a special meaning, and map them to specific relationships in a knowledge representation.

- **ontological knowledge** describes the categories of things in that domain (classes), and the terms with which they are referred to. Requirements to express this kind of knowledge are similar to those of structural knowledge; some language patterns must be recognized and represented accordingly.

Other categories, such as inexact or uncertain knowledge, commonsense knowledge, or even the heuristic knowledge described before, are qualitatively different, but their expressive requirements are similar to those in the previously described categories.

Bringing into play the different types of knowledge representation languages described in chapter 9, a number of possible schemes for sublanguages and knowledge representation emerge. As an example, a frame based knowledge representation can be chosen for subjects best described as procedural or structural knowledge; a rule based knowledge representation may be chosen for a topic comprising heuristic knowledge or procedural knowledge as in a manual of instructions. Once a knowledge representation language has been chosen, a sublanguage to support it can be designed, so as to write text which can be transformed into rules or frames, for instance. The design of the sublanguage can be based on the tools and methods
used for the syntax driven sublanguages developed in chapter \[8\], recognition of patterns, such as is_part_of, and their corresponding semantics for the transformation into the knowledge representation instance, are the necessary additions. The recognition of specific patterns can be easily achieved through preprocessing, as discussed in chapter \[8\] and the parsing of the sentence into a dependency tree can be achieved with the same software application tools.

This project has concentrated on syntax based knowledge representation using semantic networks, but other schemes such as the ones outlined well deserve a try.

12.2. Knowledge topologies

Some knowledge representation languages, such as frames or rules, are relatively formal in their conception; using them in knowledge representation is possible in a rather straightforward way, and this helps towards the definition of a sublanguage to support their construction from text. On the other hand, semantic networks and all devices based on them, such as concept maps, are very loose in their rules, almost limited to establish the existence of nodes for concepts and edges for relationships among those concepts. Many different arrangements are possible in a semantic network. A set of instructions can be expressed as a sequence of imperative sentences, possibly with some branches such as “if level exceeds 5 open relief valve”. This leads to an essentially sequential or linear development when a semantic graph is drawn. The same happens in the telling of a story, though here the sentences will be statements. A categorization of species in a taxonomy results in a tree. Mind maps produced when brainstorming around a central idea will result in a radial topology.

The former examples are reminiscent of the topologies recognized in network theory. In networks the emphasis is on connectivity, not in orientation, i.e. left to right or up to down do not mean anything. However, looking at network topology diagrams and thinking in representing knowledge, a sequence from left to right may easily suggest a sequence of events, in analogy with left to right writing, or chronological and graphic presentations. Other topologies suggest similar things:

- **line**: concepts connected in sequence, as in a set of instructions, or a story.

- **bus**: a sequence with derivations, such as clusters of concepts connected in sequence. A series of events occurring in a historical period may be represented as a bus topology, where each event comprises a bunch of concepts and relations (a “concept cluster”), all linked to a chronological axis represented by the bus.

- **star topology**: mind maps are built around a central concept; though relations among elements other than the central may exist, the resulting diagram is reminiscent of a star topology.

- **tree topologies** are typical of taxonomies and hierarchies. Though geometrically a star topology is a tree, the disposition on the plane of a star tends to erase the perception of levels given by the tree.

- **mesh topology**: nodes are in no particular arrangement, and partially connected; this is similar to the diagrams representing concept maps.

All the former “interpretations” are of course a matter of convention, but some of these conventions are easy to adopt, since they are commonly accepted in different fields. Many types
of diagram have been used to represent information, in almost all disciplines of knowledge. Drawing on network theory offers a more abstract model, not related to a particular discipline or type of knowledge. Awareness of the arrangements which better represent the subject matter of interest may be of help to select the best way of representing knowledge in a particular situation.

12.3. A Representation Scheme

In the scope of this thesis, a sublanguage and knowledge representation scheme, or representation scheme for short, includes:

- a sublanguage, with its lexicon and rules.
- a semantics of the sublanguage, stating how the constructs of the sublanguage are interpreted towards the mapping into a some form of knowledge representation.
- a knowledge representation language.
- the conventions of use of the knowledge representation language, stating the forms in which the interpreted constructs of the language are recorded or visualized.

In the previously developed syntax driven sublanguage, the former elements are characterized as follows:

- the sublanguage was English05 with its lexicon based on Longman Communication 3000; the rules stated the parts of speech, their grouping in constituents, and how to combine these constituents into sentences;
- the semantics of the sublanguage stated which parts of speech were considered the heads of each constituent, and how the syntactic trees were transformed into dependency trees;
- the knowledge representation language chosen was a semantic network, visually identical to a concept map;
- the conventions of use of the knowledge representation language established the mapping of words into nodes, stated that nouns were unique for all mentions, defined the different shapes for each part of speech, their labels, the edges to be drawn from the dependency tree, their labels, and which parts to omit, such as the linking verbs, which were not drawn.

Though it is too early to propose a design pattern for a representation scheme, and advance of its contents may be attempted. A design pattern for a representation model should include:

- a name.
- purpose: intended use, area or field of knowledge, the types of knowledge to represent.
- the former four elements required to define a representation scheme.
- a set of examples, covering as best as possible the different typical constructs.
- documentation, specially tutorial material, for its successful use.
12.4. An object oriented representation scheme

- limitations, what the model cannot do or cannot do effectively.

The success of a representation scheme like the ones proposed in this thesis is heavily dependent on the adequacy of the documentation. The scheme will be used both by teachers and students; both must know the necessary detail to use the system, but must not be overburdened with grammatical or technical stuff. In the syntax driven scheme, the grammar should be familiar for any mid secondary school student, but support material for revision may be useful. Examples of typical constructs are also a great help. As with any new proposal, great care must be given to the first experiments. Though a defective proposal is difficult to save from failure even with the best efforts, a good proposal may easily fail on careless application.

12.4. An object oriented representation scheme

The discussion of the syntax based sublanguage example in chapter 8 almost immediately brought the question of categories and individuals. A simple analysis of the sentences

- dogs have four legs
- the dog ate the food

shows the word dog in the first sentence refers to all dogs, while the dog in the second refers to a particular dog; the legs in the first sentence are individual legs; the food in the second sentence most probably refers to a portion of some food, and not to food as a general, uncountable concept. The contrast between the extreme simplicity of these sentences, and the subtleties of the questions aroused, gives an insight of the complex process the human mind can go through to make out the meaning of a sentence.

In some subject areas it may be essential to make the distinction between category and individual explicit. This section discusses a representation scheme to support categories and individuals.

12.4.1. Categories and individuals

The human ability for abstraction allows to recognize common attributes in a collection of things and give a common name to all the members in the collection, without referring to one in particular, and even to do so in different levels.

Both terms class and category are defined in Wordnet as "a collection of things sharing a common attribute". In this context, these "things" may be material or immaterial. Also in Wordnet, concept is defined as "an abstract or general idea inferred or derived from specific instances". For our purposes, a class or category will designate a set of common attributes usually gathered under a name. This name thus stands for a concept inferred from all the individuals having the set of common attributes. This distinction allows to say "dog" without referring to any dog in particular, even in a situation where not any dog exists (is included in) our domain of discourse.

An individual or an object will designate a particular instance of a class or category. "Rufo is a dog" means something called Rufo has all the required attributes to be considered a member of the category Dog. Capitalization of the category is usual in Object Oriented Programming to designate classes; the same name in lowercase would be interpreted as an instance of the class with the same name as the class. Hence, "this dog is a Dog" has a meaning.

Relations of inclusion among categories define hierarchies of concepts or taxonomies. A taxonomy stratifies the set of properties into different levels through the concept of inheritance, by which a subclass is recognized to share all the properties of the class in which it is included.
12. Knowledge Representation Schemes

An object may be part of another; this is usually called the part_of relation, or composition. There is no inheritance in composition; the parts are just associated into a compound unit. Categories of composite objects can be defined, in hierarchies analogous to taxonomies.

The main facts to state on categories and individuals are summarized below, together with some specific situations the modeler must be aware of. All these facts can be expressed formally in FOL [Russell and Norvig, 2010]. Instead, they are described here using particular natural language constructs; this leads the way towards a particular sublanguage to state facts about categories and individuals.

• category: declare a category, by stating its name. E.g. “Dog is a class”.
  - definition of categories may result in different decompositions: a *disjoint decomposition* means categories have no members in common; an *exhaustive decomposition* means that any individual in the domain of discourse must necessarily belong into at least one category; a *partition* is a disjoint exhaustive decomposition.
  - it may not be possible to arrive at strict definitions for categories; in many cases, it will also not be necessary, the audience will know what is meant.

• property: declare a property of a category, by stating the name of the property. Stating a property for a category assumes all members of the category share the property. E.g. “Dogs are friendly” states all dogs are friendly.
  - A property may be a value such as a number or label, or a complex concept better expressed as a category.
  - Exceptions: a property may be absolute with no exceptions, or general with some exceptions. E.g. dogs may be usually friendly, or friendly unless provoked, or an individual dog may not be friendly at all, or even the Doberman Pinscher breed, a subclass of Dog, may not be friendly.
  - Properties may be loose in their meaning, and not have clear or real antonyms: “not friendly” does not have the same meaning as “unfriendly”, “not beautiful” does not necessarily mean “ugly”.

• membership: declare an individual is a member of a category: state the name of the individual, and the name of the category.
  - an individual may be a member of several different categories: a man may be at the same time a Driver, a Father, and a Son (will always be!).

• subclassing: declare a category is included into another category, “category A is a kind of category B”; all individuals of category A will also belong into category B, which means all individuals of category A will have the same set of attributes as those of category B, possibly including some other new ones.
  - as with individuals, a category may be a subclass of several categories. A new category may be defined just indicating several classes from which it is a subclass: a “Dog is a kind of DomesticAnimal and a kind of Canis”, jackals and wolves are also of genus Canis but not domestic.
12.4. An object oriented representation scheme

- composition: We can say a car comprises a body, a motor and four wheels by stating “c is a Car”, “b is a CarBody”, “m is a Motor”, “w1, w2, w3, w4 are Wheel(s)”, “b, m, w1, w2, w3, w4 are part of c”.

  - a category Car can be defined as a composite, indicating the parts it comprises stating which type of objects they are. An instance of a car will then require the existence of instances of all the parts, or at least one of them, e.g. the body; we can think of a car with no wheels or motor, but something must exist!

  - in UML two types of composition are possible: aggregation involves a light coupling with the parts, and the object may exist even if none of its parts exist; composition involves stronger coupling, and the object does not exist if their parts do not exist. The distinction is not easy to make in every case, and it may ignored by the modeler.

  - stating firmly that a car has exactly four wheels, not more and not less, requires some kind of convention. In UML, the cardinality of an association declares this.

  - some type of aggregation is required for properties concerning several objects as a group: the number of passengers a dozen of taxicabs can transport requires the concept of Fleet or similar, defined as an aggregation of objects of type TaxiCab.

- measurements are the values assigned to some properties, which usually comprise a number and a unit of measurement. Several different units may be used for the same kind of measure; this usually requires the statement of a conversion formula. Some properties do not admit numerical values, but anyway admit an ordering, which allows for comparison: “the taste of fish is stronger than the taste of bread” may be enough for most purposes.

- things and stuff: things stand for countable nouns such as beans and stars; stuff for mass nouns such as water or freedom. In some ontologies, Thing designates the upper category for discreet objects, Stuff designates the upper category for substances. For practical purposes, an object consisting in one litre of water is a Water object, but may easily become two Water objects of half a litre water each; this does not happen with an orange, half an orange is no longer an Orange object.

Most everyday things we recognize do not have a clear cut definition, and their properties may vary wildly, yet we are able to say “this is a pear” or “it smells of smoke” with no doubt. These natural kinds are very difficult to define formally. This is not a major inconvenience for our purposes, since our agents will mostly be humans rather than machines.

12.4.2. Sublanguage constructs

Exiting proposals for the use of a sublanguage or a controlled natural language for class and object definitions in Object Oriented Programming generally provide support for the construction of class and object diagrams in UML, as part of the software development process. In this section we explore a simple scheme for a sublanguage and knowledge representation model to represent categories and individuals in an educational environment. Departing from the usual phrasing for class and object definition, some syntax constructs closer to natural language are attempted. In everyday use of the language, references to categories are present, as well as to individuals, in syntactic constructs which speakers interpret correctly. An analysis of these constructs and a formal interpretation of their meaning allow to define a sublanguage.
12. Knowledge Representation Schemes

<table>
<thead>
<tr>
<th>Language construct</th>
<th>Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A is a category</td>
<td>creates class A</td>
<td>Dog is a category</td>
</tr>
<tr>
<td>A is a class</td>
<td></td>
<td>Dog is a class</td>
</tr>
<tr>
<td>a is a kind of A</td>
<td>creates class A (if necessary), creates object a of class A</td>
<td>rufo is a kind of Dog</td>
</tr>
<tr>
<td>a is a(n) A</td>
<td></td>
<td>rufo is a Dog</td>
</tr>
<tr>
<td>B is a subclass of A</td>
<td>creates A,B (if necessary); creates subsumption relation</td>
<td>Doberman is a subclass of Dog</td>
</tr>
<tr>
<td>B is a subcategory of A</td>
<td></td>
<td>Doberman is a subcategory of Dog</td>
</tr>
<tr>
<td>class B is a kind of A</td>
<td></td>
<td>class Doberman is a kind of Dog</td>
</tr>
<tr>
<td>category B is a kind of A</td>
<td></td>
<td>category Doberman is a kind of Dog</td>
</tr>
</tbody>
</table>

Table 12.1.: Object Oriented sublanguage constructs as may be used in a programming environment.

to write texts not much different from everyday use of the language, yet defining which concepts are categories and which are individuals.

The syntactic constructs proposed in this section are just a first approach. For test in the classroom, a more rigorous treatment will be required, considering the issues detailed in the former section, and defining appropriate criteria to handle them.

Table 12.1 shows some language constructs much in the way of a declaration in a programming language. Though these constructs make the distinction between category and individual very explicit, and can be understood by any speaker, they do not sound as normal use of the language.

Table 12.2 shows some equivalent language constructs in a more natural use of the language. These constructs are usually found in common speech and writing, and their meaning is understood by speakers. Though not conscious for most speakers, this understanding goes as far as distinguishing if references are made to an individual or a category. The design principle of this proposed sublanguage is to capture the distinction between category and individual as it is made in everyday use of the language, and make their meaning explicit.

In the constructs of table 12.2, the distinction between categories and individuals is implicit: plurals with no determiner are assumed to have general value, and refer to categories, as in “Dobermans are Dogs”; a determiner indicates an individual, as in “rufo is a Dog”. Here a category “Dog” will be created if it does not exist, and an individual identified as “rufo”. The former sentence, “Dobermans are Dogs”, will create categories “Doberman” and “Dog” if they do not exist, and consider “Doberman” a subcategory of “Dog”, stating that all members of category “Doberman” are also members of category “Dog”. Sentence “rufo is a Dog” must be recognized as a construct with a special meaning, the declaration that “rufo” is an individual of category “Dog”.

In the examples, capitals were reserved for categories, and individuals, even if identified by proper nouns, are in lowercase. To implement the sublanguage, a possible way is to separate nouns into nouns for categories and nouns for individuals, and rearrange the rules accordingly.

The concept of inheritance is considered part of the category - individual distinction: all the properties and relations established in a category are of value for all the individuals in this category. The language constructs do not hinder multiple inheritance, i.e. a category being a subcategory of several other categories, or an individual belonging to one of such categories.
12.4. An object oriented representation scheme

Table 12.2.: Category and class sublanguage constructs in a more natural use of the language.

<table>
<thead>
<tr>
<th>Language construct</th>
<th>Explanation</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>a is a(n) A</td>
<td>creates class A (if necessary), creates object a of class A</td>
<td>rufo is a Dog</td>
</tr>
<tr>
<td>Bs are As</td>
<td>creates A,B (if necessary); creates subsumption relation</td>
<td>Dobermans are Dogs</td>
</tr>
</tbody>
</table>

12.4.3. Properties

This section discusses properties assigned to categories or individuals.

Properties may be named, or the name of a property be assumed:

- tomatoes are red
- tomatoes are of color red

Though it seems better to have properties specifically mentioned by their name, such as “color” in the former sentences, in some situations it may not be necessary or convenient; the audience will understand.

- this tomato is green
- this tomato is of color green

A category may have a property with a value, and an individual have the same property with a different value. This happens in reality, so the model is appropriate. Properties may be distinguished in mandatory, when membership to the category implies the property and the value, or optional, when the individual may possess the property with a different value from the usual one established in the category. Depending on the situation, this distinction may be useful or unnecessary.

- the color of tomatoes is red

This sentence is equivalent to the former ones, may be included in the sublanguage as a different way of setting a property.

- the green tomato is in the basket

Here an individual is assigned a property as part of a sentence with other content, i.e. it does not use linking verb “be”. The assignment of a property may be enough to identify an individual, as this sentence seems to indicate. In the present scheme, an individual will be represented by a node, and properties may be recorded in the node; hence, nodes representing individuals of the same category may be distinguished by their different properties, or different values for the same properties.

- cars have a registration number
- registration number of this car is SBL-8591
- this car has registration number SBL-8591

The first sentence declares a category to have a certain property; the next two sentences are two alternative forms to set the value of this property for a determined individual. These last sentences may be accepted even in the absence of the first, in which case the property would be assigned as a property of the individual, but not of the category. This allows individuals to have properties of their own, not common to the whole category, as is possible in the real world.
12. Knowledge Representation Schemes

12.4.4. Part and whole

Though UML defines two forms of modelling part and whole, namely composition and aggregation, the frontier among both forms is blurred [Fowler, 2004]. Composition indicates a closer relation between the whole and its parts, and the parts not conceived to exist without the whole. An invoice may be considered a composition of invoice lines; an isolated invoice line does not make much sense. In aggregation the parts may exist independently of the whole. The wheels of a car may exist as spare parts, independent of the car. Whether to use composition of aggregation is a modelling decision, sometimes difficult to make.

For knowledge representation an alternative view of different levels of aggregation may be better suited. A car may be modelled as an abstract entity consisting of an aggregation of body, wheels, engine and so on, all gathered under the concept of Car. A node Car will then represent the whole when it is not necessary to distinguish the parts. At the option of the user, the entity Car may be shown in detail with its parts and relations. This different degree of detail is common in the graphic presentation of complex networks, such as those in Topic Maps.

12.4.5. Associations

Associations are the simplest relations among concepts. In common language, these type of relations are established through verbs; the meaning of the verb characterizes the relation. Use of verbs in the sublanguage require some definitions. In sentences

merions sleep 12 hours
sleeping time of merions is 12 hours

'sleep' is an intransitive verb which may be considered to assign a value to a property. The second sentence uses the formerly proposed notation to assign values to properties. Transitive verbs, on the other hand, may be better modeled by associations:

merions build tunnels
dogs chase cats

These sentences would create the categories and establish the corresponding relations.

12.4.6. Representation in a semantic graph

The most common form of representation of categories and individuals is to draw two separate diagrams, one for the categories, another for the individuals; this is the usual practice in Object Oriented Programming. The UML notation provides a type of diagram for each: the class diagram for categories, the object diagram for individuals. There must be a connection among these diagrams, a form of saying that an individual belongs to a certain category, or to several categories. In UML, nodes corresponding to objects are labeled in a notation 'id: class', where 'id' is an identifier of the particular object, and 'class' is the name of the class to which it belongs. Following this model, a notation 'rufo: Dog' will indicate 'rufo' is an individual of category 'Dog', and 'jack: Doctor, Father, Driver' will indicate 'jack' is an individual which belongs simultaneously to the categories 'Doctor', 'Father' and 'Driver', i.e. individual 'jack' complies with all requisites to belong in each of those categories.

Perhaps the main objection to this form of presentation is the use of two diagrams, and the "weakness" of the connection between the two; the category to which an individual belongs is recorded in the labels but not visualized in a link. Though a special type of edge, such as a dashed or dotted line, may be reserved to indicate that an individual belongs to a certain
category, this single diagram soon becomes too crowded to be clear. An intermediate solution
is to preserve the two diagrams but use shape and colors to reinforce the idea of membership
and relation between the diagrams: a node for category Dog may be colored brown, and a
particular dog also colored brown. This only works for single inheritance, though, and does not
help in showing how the properties of general categories are present in specialized categories.
Notwithstanding these limitations, use of shape and color may help, even though it may have
to be applied differently in each case.

12.4.7. Evaluation

Application of this scheme to support categories and individuals is appealing, first of all for
its modelling capabilities, solidly established by Object Oriented Programming. Though there
may be little harm in leading students to differentiate category from individual, reflecting all
the complexities described here in a scheme to be used in the classroom may prove unaffordable
in time, effort and usefulness for the subject being taught. On the other hand, there are
domains where it may even be required, such as the Natural Sciences, where taxonomies are
essential.

To convert the hints formerly given into a complete system with sublanguage and semantics
is no easy task; many details require careful consideration. However, use in the classroom may
not require a complete system; in many cases the distinction of category and individual can
be taught and profitably used with only a simple, approximate scheme.

Though strong as a modelling tool, and even if readable text can be written to support this
scheme, these texts will be much more restricted than those resulting from the syntax based
scheme of chapter 8. Domains where exact modelling is a primary concern, or a lightweight
application of these concepts, make this scheme interesting enough to give it a try.

12.5. A sequential model, representing time

A modern formalism for the representation of time is Event Calculus, introduced by Kowalski
and Sergot in 1986; a simplified version was introduced by Kowalski in 1992. Event Calculus
builds on first order logic, introducing suitable predicates and functions to infer the truth
value of a conclusion from a narrative of events and a description of the effects of actions.
Event Calculus works with events or actions, fluents and time points. A fluent is a quantity,
a proposition or anything that changes in time. Time points or moments have 0 duration,
and are used as a point in a scale; intervals have a duration, and last between two moments.
Predicates express relations such as "fluent f is true at time t", "event e happens over time
interval i", "event e causes fluent f to start to hold at time t", "event e causes fluent f to
stop to hold at time t". Axioms relate predicates in relations such as "fluent f holds at a
time t if it held at time 0 and has not been stopped between 0 and t" [Shanahan, 1999]
[Russell and Norvig, 2010]. Being logic based, Event Calculus shares in its power and in its
complexities. Though such a formalism goes far beyond our needs, a more simple handling of
time will be well founded if inspired in its ideas.

A formal analysis of how time is referenced in language can be seen in chapter 22 of
Jurafsky and Martin, 2008, which discusses time information extraction from unrestricted
natural language texts. Though the problem is much harder than the one at hand, the dif-
f erent forms time is referred to in language, the different approaches for its recognition, and
patterns proposed to standardize representation of times an duration are of interest as a guide.
Timelines are a simple, traditional tool to represent time, frequently used in Education. A timeline represents a sequence of events in chronological order, displayed along a line, usually drawn from left to right. Moments can be indicated on the line, and additional material linked to them. Intervals can be defined between moments, such as historical periods.

A simple use of a timeline in assessment consists in presenting to students a bare line with identified initial and final moments, and ask students to place the events they remember on the line. In a History lesson, the timeline may be just the initial and final years of a period, a regularly marked chronological line, sub intervals or periods already marked, or whatever referential information is considered useful. The answers will show which events the student knows to have happened in the whole period, if she can place the event in a sub period, if the order of precedence of significant events is correct, or if the year indicated is exact or close to the real date.

Some possible requirements for a sublanguage to support a simple timeline scheme like this are:

- state dates for an event: “in 1949, Columbus discovered America”; “Shakespeare lived in the 16th century”, and also “Shakespeare lived in the 17th century”, both correct. A beginning and end date can be equally established, such as “World War Two started in 1939 and ended in 1945”.

- state dates for a period: a beginning and end date established as for events, provided some phrase construct is to distinguish a period is defined, such as “the Roman Republic is a period from ___ to ___”, “the Roman Republic extends from ___ to ___”, where underscore lines may be given in years, or centuries, as has been accorded.

- place an event in a period: “the Crusades happened in the Middle Ages”, “the Punic Wars happened in the Roman Republic”.

- phrase structures, according to the expressivity desired for the sublanguage, can be drawn from the rules of the syntax oriented sublanguage in chapter 8.

An event may be attached to a single date, or given a start date and an end date, if the event lasted for some years. A period can also be defined by two dates, but a period will admit events to be attached to it, as if to a specific date.

A graphic presentation for this scheme implies support for a sequence in which specific points and precedence can be determined. A simple scheme is to define a sequence of time points as a reference upon which events and periods can be attached. In figure [2.3.1] the central line shows time points labeled d1, d2, ..., which may stand for dates. This reference timeline is assumed to exist before text entries add events and periods. Events can be assigned to a date, as event 1 in the figure, or to two dates, as event 2, to indicate the beginning and end of a lasting event such as a war or the building of a cathedral. Periods can be defined by a start date and an end date, as the Medieval Ages are divided into Early Middle Ages (476 to 1000), High Middle Ages (1000 to 1300), and Late Middle Ages (1300 to 1453). In the figure, the period of study is the whole timeline, divided in two periods 1 and 2. Events can be attached to periods instead of to strict dates, as events 3 and 4 are attached to period 2 in the figure.
12.6. An imperative model, for instructions

Instructions are step by step directions on how to do something, such as how to assemble, operate, or repair an equipment. Instructions are to be performed in sequence, and are frequently presented as numbered lists. Instructions are recommended to be written in simple language, addressing only one specific task. A relatively complex instructions manual, such as the user guide for a microwave oven, may contain a number of these small tasks. Putting the equipment to work may require the performance of several of these small tasks [McMurrey, 2011].

Manuals require descriptive text besides instructions; handling of descriptive text can be done separately from instructions, using one of the schemes proposed. This section is concerned only with the bare instructions for the performance of a specific task.

Instructions are usually expressed in imperative sentences like “turn the dial clockwise until noise disappears”, “right click on the network icon, select edit connections”, “put on the brakes”. The normal sequence of instructions may be modified by conditions found along their application, in the usual “if ... then” structure. A sublanguage for instructions has the following requirements:

- imperative sentences; subject in imperative sentences is always assumed to be the reader, so syntactic structure is simpler than in statements.
- phrase structure; some or all of the same phrase structures defined for syntax based sublanguages.

Figure 12.5.1.: Representation of a sequence, based on a timeline reference for events and periods

Though attachment to reference points and precedence relations are the essence of this scheme, it is not difficult to build the graph from text if the timeline is assumed to exist. The recognition of the corresponding language constructs create the nodes for events or periods, and establish the links to the date nodes, or from an event to a period, as the case may be. Once the whole reference graph is built, matching of a text answer written in the sublanguage against the reference graph can be done in the same way as in the syntax based scheme.

12.6. An imperative model, for instructions
12. Knowledge Representation Schemes

- support for conditional constructs of the form “if <condition> then <instruction>”; the condition is a statement, affirmative or negative, and the instruction an imperative sentence.

- for statements, a subset of the constructions accepted in syntax based sublanguages may suffice.

Knowledge representation for instructions has long been done through flowcharts; the most usual shapes are a box for instructions, and a diamond for conditionals; several types of box may be used for different types of instruction, and other symbols stand for start, end or other state. The UML activity diagram provides similar support but allow to handle parallel behavior [Fowler, 2004]. Though this may be not required, the fact that UML is a well established standard calls for choosing activity diagrams for the description of the sequence and conditions of a flow of instructions. Though the capabilities of Activity Diagrams greatly exceed the simple scheme here proposed, it is the recommended model to follow: it is a standard, can be used partially, and offers an excellent guide for further development of the scheme.

12.7. Concept maps, mind maps

The syntax based sublanguage proposed in chapter 8 results in a semantic network of a particular type, where the concepts and relations come from the syntax of the original sentences. Formally, a concept map is a semantic network. Though concept maps are very free in their conception and use, there may be a restriction on the types of relations that can be established among concepts, for example by specifying the labels on relations. Some common labels for relations are “includes”, “is comprised of”, “necessary for”, “begins with” [Novak and Cañas, 2006].

When the labels have been defined, a sublanguage to recognize them can be compiled with little effort; concepts will be recognized as nouns, perhaps in phrase form, such as “units of meaning”, “hierarchically structured” or “cognitive structure” [Novak and Canas, 2006]. If labels to relate concepts can be freely chosen, the sublanguage must provide rules to detect them as such. Not using a fixed set of labels known beforehand by the users will most surely result in many different forms of representing the same meaning, which prevents any reliable matching, and consequently cannot be used for assessment.

Mind maps are similar to concept maps, but there is a central idea, and other concepts relate to this central idea directly or indirectly through intermediate concepts. Though imposing a strict control to force the hierarchical tree structure of mind maps may be difficult to achieve, this restriction can be left to be taken care by the users. In this way, the sublanguage for concept maps is also usable for mind maps, the only additional requirement being to declare the central idea. A simple solution is to assume the first line of text contains just this main concept, as if it were a title.

Texts produced by sublanguages oriented towards concept maps or mind maps will be more constrained and less natural than syntax based sublanguages. A carefully chosen set of labels for relations can produce more readable texts. The wording used to refer to the same concepts may also vary widely with different people, or the concepts must be very well known to everyone as in a glossary.
12.8. Topic Maps network

Topic Maps is a technology for describing knowledge structures and improve the findability of information. Topic Maps are described by standard ISO/IEC 13250. Topic Maps stress the importance of the subject in a search, i.e. the subject to which the search belongs, or is most closely related. A *subject* is anything about which information is desired, such as a concept, thing, person or place. A *topic* is a symbol used to refer to a subject, so that statements can be made about the subject. Associations establish relations among topics; an *association* is an n-ary combination of topics. Subjects are identified by an URI (Universal Resource Identifier); this URI is a *subject locator* when it points to a network accessible resource which acts as the topic identifier, or is a *subject identifier* if its purpose is to identify the subject to computers, which need to determine if two topics are the same. An *occurrence* relates a topic to information resources; an *occurrences* is normally described by an URL (Universal Resource Locator) [Pepper, 2010].

Topic Maps put no restrictions on the domain to represent (have no predefined ontology); they may be used to model temporal relations, abstract concepts, forms of first order logic, or more traditional information such as thesauri or glossaries. Topic Maps clearly separate the domain, expressed as the topics and their associations, from the resources, which allows them to be useful both to locate resources and to represent knowledge. The mechanism of Topic Maps can remain hidden to users, which can concentrate on the subject, yet have an effective way to access resources. Topic Maps can be layered for different degrees of detail, and are easily merged, owing to the strict mechanism of identification of topics [Ahmed and Moore, 2005].

The essential concepts underlying Topic Maps are easy to understand, and the complexities of handling them can be made transparent to the user through a suitable application. Construction of the knowledge representation of a subject, and the gathering of study or reference material can be carried along as normal learning activities. A carefully done set of Topic Maps instances may well act as a framework around which a whole course can be built or a subject described, from the ontology to the resources.

As a knowledge representation language, Topic Maps are not much different from semantic networks, or from concept maps, by the way. However, Topic Maps are formally defined in a standard, which not only uniform their creation but allows them to be transferred in standard formats, merged with other topic maps instances, and shared among the community of users.

A sublanguage for Topic Maps creation from text is an endeavor for information technology professionals. There exist several projects in this direction.

The Linear Topic Map notation (LTM) is a simple textual format for topic maps, compact and simple enough to be understood by humans. LTM allows to create a topic map with a text editor. Application software can convert the text into XML for interchange. LTM also intends to simplify the development of software for human oriented applications, steering away from the needs of specialized topic map editors which spare users from the syntactic details of the notation. LTM intends to make it possible for users to create topic maps in a text editor for direct exchange with other users via email, forums and similar situations. LTM was also conceived to easily create small topic maps, which brings it closer to the purposes of this
12. Knowledge Representation Schemes

thesis [Garshol, 2007].

Though relatively easy to understand, LTM is still very reminiscent of the textual structure of a programming language. The Pidgin English for Topic Maps Knowledge Engineering is an initiative of the AsTMa Topic Map Processing project to create topic maps from language construct closer to natural language, as in the following example [Barta and Heuer, 2007]:

paul-mccartney
  plays-for The-Beatles,
  which is-a music-group and which
  is-located-in London,
  playing piano and has shoesize 42

Unfortunately, the development group last activity date is 2007 [AsTMa, 2007], and no complete proposal could be found.

The formerly indicated advantages offered by Topic Maps signal them as the way to go. However, their use for the purposes of this thesis exceeds the capabilities of a small group of teachers with moderate information technologies skills which we set as our target users.

12.9. Conclusions

Contributions of this chapter were:

- a guide to develop a sublanguage and knowledge representation scheme for category and individual distinction, in the way of class and objects in Object Oriented Programming, but using everyday language expressions and extracting from them the category and individual distinction implied in those expressions. A proposal which adapts Object Oriented methodology to learning.

- a guide to develop other sublanguage and knowledge representation schemes for sequential development and instruction manuals.

The different types of human knowledge recognized, and even argued about, by cognitive psychologists, show that no single knowledge representation technique or language can successfully cover all aspects and peculiarities of human knowledge. In this chapter different alternatives for sublanguage and knowledge representation schemes were analyzed, trying to assess the difficulties of each and giving some hints on how to develop them.

In a Semantic Web perspective, Topic Maps offer the best opportunities, but development of a scheme based on them calls for a lasting project. The imperative and sequential models are within reach of a small team of developers to come out with a specification and a usable application; a group of teachers with some information technology support or knowledge can also attempt an approach to their implementation. The object oriented approach requires more effort, there are a number of details to attend to; the fact that it is a well known paradigm in Software Engineering provides a solid guide to development. For learning and assessment, any of these schemes may be approached by a partial development. The most important point to make is that there are a number of choices for sublanguages and knowledge representation schemes, ones better suited than others for specific situations, and the learning and assessment proposal made in this project need not be tied to only one of them.
Part V.

Solution Proposal
This part describes the solution proposed in this work, the prototypes built for proof of concept, and deployment considerations for trial in the classroom.

Chapter 13 describes the proposed system, discusses its feasibility, gives a view of the system, and provides an example of its use. An analysis of the learning and assessment process the proposed solution implies is also given, describing the necessary steps to take when teaching a learning unit following the principles proposed in this thesis.

Chapter 14 describes the KLEAR project; KLEAR stands for Knowledge and Language for Education, Assessment and Research, and is the name under which testing and prototype software tools were developed. A Lexicon module defines a common data structure for lexicon testing and support; a Semantic Graph module handles graphs and their visualization; Dependency Tree modules bridge the gap between the sublanguage and the graph representing the knowledge inferred from a text. A demonstrative application allows to write some text, eventually add to the lexicon, represent concepts and relations in a graph, and recognize the knowledge contained in a sentence as a part of the graph.

Chapter 15 goes into the requirements, preparation and general conception of teaching and learning involved in bringing the proposed solution into the classroom. Besides a suitable, friendly application, the proposal requires adequate planning, division of learning content into learning units, acquisition of some skills on the part of the students and teachers, and a collaborative approach engaging both students and teachers into the learning and assessment process.
13. Overview of the Proposed Solution

Abstract. The feasibility of this project is based on the possibility of compiling a suitable sublanguage, converting it into a knowledge representation instance, and comparing this instance to a reference. A sublanguage called English 05 and a syntax-based knowledge representation scheme which produces a semantic graph were developed as proof of concept, together with a prototype application into which the sublanguage and knowledge representation scheme could be loaded and demonstrated. A general view of the components of the system, followed by an example of use, give a feeling of its work and look. The application reads from files the lexicon, the production rules, and the conversion rules to transform text into a semantic graph. This allows for the application to be used with different sublanguages and knowledge representation schemes. The prototype code includes routines to compile a lexicon, an editor to license sentences against the sublanguage, functions to transform text into a semantic graph, and a distance measure to compare two semantic graphs. Different sublanguage and knowledge representation schemes may be conceived for different purposes, each with its own pros and cons. For test in the classroom, the teachers decide on a sublanguage and knowledge representation scheme, define learning units of one single learning objective as in learning objects, train the students in the use of the system, promote the collaborative creation of the knowledge structure instances for the learning units, prepare the questions based on the knowledge structures created, and submit them to the students for assessment. The students answer the questions on the learning units they have studied, recorded in the knowledge representation instances they have created, writing texts in the same way and in the same application they have used during study. The proposal is thus conceived not merely as an assessment system, but as a whole learning activity of which assessment is just a culminating step.

This chapter describes the proposed solution, both as a system and as a methodology (how it works). After going through the assumptions under which the idea was conceived, the feasibility of the project is assessed. The proof of concept prototype application developed is briefly described, together with some hints on how this ideas can be brought to the classroom. Potential difficulties and expected benefits are then discussed. Most aspects mentioned here are further developed in the following chapters.

13.1. Thesis contents

This thesis includes:

- the development of some examples of domain specific sublanguages and the selection of knowledge representation types. Better results are obtained and less effort is required if the generation of the sublanguage and the selection of the knowledge representation
13. Overview of the Proposed Solution

are thought out right from the beginning and adequately correlated. Both sublanguage and knowledge representation depend on the kind of assessment in mind and the informational characteristics of the domain.

- tutorial information and recommendations on how to generate a domain specific sublanguage, language processing tools available and resources that can be exploited. The sublanguage comprises a deterministic vocabulary (one sense per wordform) and deterministic parsing (one syntactic tree per sentence). Text written in the sublanguage can be parsed into a collection of syntactic trees, one for each sentence.

- tutorial information and recommendations on the selection of a knowledge representation type among a small collection of types considered the most adequate for education and the purpose of this thesis. The knowledge representation type must be able to record the concepts and relations as they are found in the parsed text (the collection of syntactic trees).

- selection or development of prototype tools to:
  - check compliance of text with the vocabulary and rules of a sublanguage,
  - create a knowledge representation instance by extracting concepts and relations from text,
  - define a distance measure between two instances of the same knowledge representation type. This implies the assignment of values to concepts and relations, and a test to determine if concepts and relations in one instance are present or absent in the other.

- the assembly of a prototype application to assist in the edition of text in a certain sublanguage. This application is able to:
  - assist the writer in producing text compliant with the vocabulary and rules of a sublanguage.
  - enforce compliance with the sublanguage specification, i.e. its vocabulary and rules.
  - parse the text into a collection of syntactic trees.
  - assimilate different sublanguage specifications as structured text files that the application can read or can be loaded into it.

- the assembly of a prototype application to generate a knowledge representation instance from parsed text (syntactic trees), according to a predefined semantic interpretation. These applications are able to:
  - generate a data structure supportive of the knowledge representation type selected.
  - visualize in a graph the knowledge representation instance.
  - visualize in a graph the difference between two instances of the same knowledge representation type. A simple scheme is to superimpose the testing instance in colors over the reference instance in gray.

- the assembly of a prototype application to assign weights to nodes and links in a knowledge representation instance, compare a test KR instance to a reference KR instance.
13.2. Feasibility

and calculate the distance measure among the two. The simplest calculation adds the weights of nodes present in both instances, and determines its percent value against the sum of weights of all nodes in the reference instance.

- an analysis and evaluation of techniques, tools and resources which could be employed in future developments of the ideas conceived in this thesis. These enhancements include, but are not limited to:
  - providing more assistance in the writing of sublanguage compliant text, such as indicating the next allowed word category (part of speech), accepting synonyms, predictive writing, dynamic construction and visualization of the partial parse trees for a sentence.
  - providing more assistance in the generation of a sublanguage, such as generate dictionaries from a collection of texts, add stemming to detect the root of wordforms and reduce vocabulary size, use Wordnets to gather synonyms accepted in the domain.
  - exploiting existing resources, such as word lists, Wordnets, common sense databases, domain specific vocabularies, publicly available ontologies.
  - using existing applications to generate or exploit existing knowledge bases, such as the Protégé editor [Stanford, 2012] to create ontologies or perform knowledge acquisition.

13.2. Feasibility

This project proposes a system for the computer assisted assessment of free text answers based on the use of a restricted language to write the answers, predictable knowledge extraction from the answers into a knowledge structure, and the comparison of knowledge structures to obtain a mark. It was conceived under the following assumptions:

- it is possible to compile a sublanguage expressive enough for the learning needs of a specific domain, similar enough to common natural language as to be easily mastered, and as readable as natural language texts.

- it is possible to adopt or adapt a knowledge representation language to build a knowledge structure intuitive enough to be almost immediately understood, such as a concept map or semantic network.

- it is possible to define a transformation from text written in the sublanguage into a knowledge representation instance of the type chosen, in a unambiguous way. The knowledge representation instance must be the same or equivalent if the sentences of the text are given in different order, or the sentences have a different syntax, provided the knowledge conveyed by these sentences is the same.

- it is possible to compare and determine a distance measure between two different instances of knowledge representation. When measured between an instance corresponding to an answer, and an instance taken as a reference of the correct answer, this distance stands as a mark that can be given to the answer.
Along the project, a sublanguage called English 05 was developed, a semantic network was adopted as a knowledge representation language, a conversion from text to graph was inferred from the syntactic structures of the sublanguage, weights were assigned to nodes and edges in a graph, and a distance measure was defined by adding the weights of nodes and edges from the answer graph which were also found in the reference graph.

The syntax-based sublanguage proposed is only one possible development of a sublanguage and knowledge representation scheme. Other combinations of sublanguage and knowledge representation were analyzed: category and individual (object-oriented) models, sequential models as in narrative or history, an imperative model for instructions or manuals, and less formal representations such as concept maps or mind maps. All these models can be implemented with moderate effort by a team of teachers, were they provided with the adequate tools. Use of or export to Topic Maps, an ISO/IEC standard for knowledge representation, was also analyzed. Though more complex, Topic Maps offer great potential for lasting educational projects.

Some tools to help in the compilation of a sublanguage, and a prototype application to license text and build the knowledge representation graphs were developed along the project. These tools and prototypes provide enough facilities to demonstrate the proposal, from the writing of text to the construction of the knowledge representation, the comparison of answer to reference and the calculation of a distance. A complete application apt for classroom testing can be based on this prototype code. A small team of two or three developers working half time for six months should be enough to build a usable application, user manuals and all.

13.3. View of the System

The tools and prototype application developed were collected in a hierarchy of Python modules under the name KLEAR, which stands for “Knowledge and Language for Education, Assessment and Research”.

Figure 13.3.1 shows a schematic view of the system. Upper blocks are end user documents, those mainly read, created and seen by the students, shown in white color. The KLEAR project prototype modules are rounded boxes in yellow. Definition documents, which regulate the functioning of the system, and are created according to formal rules, are shown in light blue. Objects in light green are Python objects of several classes defined in the KLEAR project code.

A basic lexicon obtained from word lists, and a specific lexicon of terms used in the subject area are compiled by the KLEAR lexicon routines into a KLEAR lexicon object, which essentially contains dictionaries of wordforms, headwords, parts of speech, and possibly definitions. The lexicon object can, at any time, be complemented by additional words coming from textbooks and documents used by the students. The production rules document defines the syntactic structures accepted by the sublanguage.

The production rules document holds a strong relation to another document, labeled “Text to Sm Objects”, which defines how the text will be converted into the KLEAR objects (Sm Objects) which support the knowledge structure. The production rules say how the text should be written; the Text to Sm Objects document determines how the text is to be transformed into the objects which constitute the semantic network, as the KLEAR project implements it. The production rules and the lexicon object define the sublanguage; the KLEAR editor will accept text according to the lexicon and rules it has been fed with.
Reference texts to create the semantic networks, and free text answers to questions, are both written with the help of the editor. The outcome of the editor is a bunch of syntactic trees corresponding to the input texts. The syntactic trees are elaborated by the KLEAR graph module into SmGraph objects, the data structure which the KLEAR project uses to support semantic networks. The objects representing the semantic networks are built according to the rules stated in the Text to Sm Objects document.

An additional formal document, also fed into the KLEAR graph module, labeled “Sm Objects to graphics”, define how the objects will be presented in graphic format. The graph module can show the graph structure in graphic format as a PNG image.

Objects labeled “ref” and “ans” stand for the SmGraph objects corresponding to the reference graph and to the answer graph. The KLEAR distance module calculates the distance between the reference and answer graphs, producing the total marks in the reference object, the marks gained by the answer object, and the percentage of the answer in the reference graph.

Objects “ref” and “ans” can be represented in graphic format, the reference graph blurred and the answer graph coloured over it.

Figure 13.3.2 shows a reference graph created by these sentences:

Columbus was an explorer. Columbus was a navigator. Columbus was a colonizer.
Columbus was born in Genoa. Columbus made [several voyages [across the ocean]].

The following answer sentences were recognized in the graph:

Columbus made [several voyages [across the ocean]]. Columbus was a colonizer. Columbus was a navigator.
13. Overview of the Proposed Solution

Total marks in the reference graph were 45, the answer got 30 marks; the answer made 66.67% of the total.

Figure 13.3.2.: Columbus: an answer recognized in a reference graph.

13.4. How it works

The KLEAR project is conceived as a learning proposal, not limited to assessment. The assessment instance comes as a final step in a learning process in which the system has been used as one of the activities the students carry on along the course, namely the construction of the knowledge representation instances of the different learning units they have been taught. The following steps briefly describe how the system was designed to work. Chapter 15 on deployment of the system gives more detailed suggestions on the requirements and procedures to bring the activity into the classroom.

1. **Preparation.** The teachers agree on a sublanguage and knowledge representation scheme. The sublanguage lexicon may be based on a word list plus specific vocabulary. The production rules depend on the knowledge representation model of choice. English 05 and the syntax based knowledge representation scheme described in chapters 8 and 11 are an example.

2. **Course plan.** The course is divided into learning units. A learning unit implies knowledge content satisfying only one learning objective, much in the way a learning object is defined. More details are given in chapter 15.

3. **Training in the sublanguage.** The students are introduced to the sublanguage: its motivation, purpose, structures, lexicon, the editor, and a number of examples. The students
should be told of the complexity of natural language, and made conscious that by writing in a sublanguage they will also be writing correct natural language. The students must be proficient in the elementary grammar underlying the production rules, or be guided through remedial work to master them. Chapter 5 gives an account of support material for training in the use of a sublanguage.

4. **Collaborative construction.** The students study all the material for some learning unit, and start writing small pieces of text to describe its contents. Under the guide and supervision of the teachers, the students write in an editor which validates the text in the sublanguage, and draws the corresponding semantic graphs. The outcome is one or more reference graphs which contain all the knowledge the students are expected to have in this unit.

5. **Questions.** The teachers prepare the questions for a learning unit. The questions address the contents recorded in the reference graphs for the unit, possibly less but never more, nor any knowledge corresponding to another learning unit.

6. **Assessment.** The students answer the questions writing in the sublanguage editor, conveying the same knowledge they have contributed to record in the reference graphs for that learning unit.

7. **Feedback.** Students get immediate feedback, they see the reference graph in blurred lines with coloured parts on the items they succeeded to convey in their answers.

8. **Evaluation of teaching.** Errors and differences in the interpretation of the reference graphs may show faults in teaching. Since the construction of the reference graphs was done collaboratively and as a learning activity, there were plenty of opportunities to correct errors and agree on differences. The shared responsibility of students and teachers in the correct recording of knowledge is considered a valuable experience of team work.

### 13.5. Discussion

Compiling a sublanguage is no trivial task, but resources exist to make the task feasible. Word lists of common words can form the basis of a lexicon; many domains of knowledge have their own glossaries, new words can be added on the fly. The design of a set of production rules closely following common structures of the language, with the required expressivity, and simple to manage, brings in some additional difficulties. The scope of use of such a set of rules is wider than that of a lexicon, which allows their use in different domains, thus making the effort less costly. The contribution of a linguist may simplify matters and save time. Teaching the students to write in a sublanguage designed as previously stated should not be seen as a waste, neither in effort nor in usefulness: the students will be learning to write correctly in their language, though in a restricted way, but producing texts which anyone can read and understand with no particular effort; most readers will not even notice the texts are written in a sublanguage. The habits of writing introduced by training in a well designed sublanguage should not stir more suspicions than training in technical or scientific writing. A sublanguage is much of that, a way of inducing a clear, simple, straightforward style required by many
13. Overview of the Proposed Solution

professions. An expected benefit of this proposal is an improvement in the writing capabilities of the students.

Integrating assessment with study as proposed is considered an effective way of reducing the tensions involved in an assessment instance as such. In this proposal, the students come to an assessment instance to do the same work they have been doing in former sessions of learning, with the same tools, and working on the same contents. They write and see the results of their writing instantaneously, in a graphical presentation where they can check and correct. Building a semantic graph from text in an editor will probably be perceived as more fun than writing a summary or drawing a concept map. Having immediate feedback, both when learning and in assessment, is universally recognized as desirable. Last, but not least, no marking session for teachers; marking is done automatically, on content well known to and agreed upon by both teachers and students.

No system will perform well outside of the scope for which it was designed. Sublanguages and knowledge representation schemes are well adapted for factual knowledge, where assertive or imperative sentences are the natural form of expression. Keeping within the intended scope, a solid design of sublanguage and knowledge representation scheme, an adequate software application, and competent management of the class, are all essential to successfully bring this proposal into the classroom.

13.6. Conclusions

This chapter provided an overview of the system, including:

- a list of the resources, software components, and documentation included in this thesis.
- an analysis of the feasibility of the proposed system.
- a description of the system architecture, its modules and their relations.
- an end to end view of the process of its application
- a discussion on some aspects relevant to the use system.

After going through this summary view of the system, perhaps the main point to stress is the conception of the proposal as a learning activity, and not just as an assessment system. The use in the classroom of the ideas proposed in this thesis are expected to improve learning in all. Most of the time, students will be using the system to learn, which is definitely the main goal, of which assessment is just a measure.
14. The KLEAR project

Abstract. Tools and software developed along this project is collected under the name KLEAR, which stands for Knowledge and Language for Education, Assessment and Research. A common data structure was defined for the testing and comparison of existing lexicial resources. Several publicly available word lists, both long and short, were brought into this common format, analyzed, and compared. In module klear.sublang, a set of classes and functions provide support to the lexicon and rules of a sublanguage. The lexicon can be taken from existing word lists, with additions and corrections, or manually compiled. The rules can be written with a simple text editor. The application code reads the lexicon and rules from files, and hence supports different sublanguages according to lexicon and rules fed into it. A function in this module validates the sentence against the sublanguage, and if successful produces a syntactic tree. Module klear.semgraph contains a set of classes and functions to support a semantic graph. A function in this module produces a dependency tree from a syntactic tree, according to head detection rules recorded in a text file. Classes SmNode and SmEdge define nodes and edges collected in an SmGraph object, which represents the graph. A function in this same module matches a dependency tree into the graph, colouring the recognized elements. Finally, module klear.kldemo integrates text functions and graph functions in a text driven application for demonstration purposes. The tools developed allowed the support, use and comparison of lexical resources, and the end to end demonstration of the proposal. Based on the effort demanded by the prototype development, a more elaborate version for testing in the classroom is considered within reach of two developers in a six months project.

This chapter describes the software application and tools developed for the demonstration of the ideas supporting this work. This included the development of a sublanguage, the conversion from text to a semantic network, the comparison between semantic networks, and the packing of these tools in a prototype application to emulate the assessment process from end to end. Considerable software development was required, amounting to roughly ten thousand lines of code. These tools and applications were collected under the name of KLEAR (Knowledge and Language for Education, Assessment and Research). They are mostly reusable, either for further prototyping or for a production quality development apt for use in the classroom. Documentation for the code is summarized in appendix B. The code and complete documentation is available at the site of this thesis [González-Barbón, 2012].

14.1. The KLEAR project

Software tools and prototypes developed during the course of this thesis is collected under the name KLEAR, which stands for Knowledge and Language for Education, Assessment and Research. Further development of these tools and prototypes may become The KLEAR Suite,
14. The KLEAR project

a collection of software applications for the use of sublanguages and knowledge representation in learning and assessment.

The programming language used to write this code was Python [Python community, 2012]. The Natural Language Toolkit (NLTK) provided support for language related tasks [Bird et al., 2011]. The Graphviz package was used for the graphic presentations [Graphviz, 2012]. Documentation was written in the code and HTML pages generated with Epydoc [Epydoc, 2012].

14.2. The Lexicon module

The lexicon module, called klear.lexicon, defines a data structure to support lexicons, and contains functions to compare word lists. These data structures can be used to create lexicons for a sublanguage.

This package provides tools to bring existing word lists into a common format; this allows for the comparison of word lists and provides support for the compilation of a new lexicon for a specific purpose and domain. A lexicon may be built from one or several known word lists, from a user’s own wordlist, or from both. The resulting lists may be (should be) manually adjusted to fit the intended purpose of the lexicon.

Words in closed categories are recorded as lists in this package; words in open categories may be read from known word lists. Custom lists may be obtained from one or several reference lists; some manual correction is deemed necessary before production or even field testing use. Custom lists may be exported as user’s lists in Python’s pickle format or in text format to be fed into a generative grammar.

Functions in this module allow to:

- read from different word lists into Python lists or dictionaries of different content.
- combine known wordlists with no redundancy into this package word lists.
- maintain this package word lists (add or eliminate words).
- check this package word lists for consistency: if a word is in different categories, or if different categories contain the same words.
- deliver a lexicon in a format apt to add to a generative grammar, as a list [(wordform, headword, pos)], or as text in lines wordform:headword:pos, wordform:pos.

Module klear.lexicon.wordlist provides a common data structure for different word lists. Class Wordlist manages lists of words extracted from existing word lists or word lists created for a purpose. Objects of class Wordlist can be used to compare different word lists, or to compile a new word list from existing ones. Frequencies of use offer a criterion to select a set of words for a predetermined coverage in a certain corpus. An object of class Wordlist may be used to create a lexicon apt to be processed by a syntactic parser.

While open word categories (nouns, verbs, adjectives, adverbs) often require machine processing, closed categories (determiners, pronouns, prepositions, conjunctions) may be managed as predefined, fixed sets, with little need for comparison or merge. A Wordlist object can anyway be used, and frequencies tested to determine which words to include in the closed categories of a compiled lexicon.
14.3. Demo application

A uniformity in PoS tags is required. To this purpose, some simple tags are used as a normalized tagset. Tags used in existing word lists are preserved as such, but functions to substitute their own tags for normalized tags are provided.

Only word lists for the English language were tested in this stage, but the data structures are not limited to English. The tested word lists were:

- BE1500, Basic English combined word list, used by Simple Wikipedia.
- BNC, frequency word lists from the British National Corpus.
- COCA, Corpus of Contemporary American English.
- GSL, General Service List of English Words, from several sources.
- AGID, Automatically Generated Inflection Database, by Kevin Atkinson.
- PoS DB, Part of Speech Database wordlist from Moby and Wordnet, by Kevin Atkinson.
- Longman Communication 3000 word list.
- Experimental functions to query Wordnet.

The tests performed on these lists are documented as DocTest strings, which allows them to be run at any moment. This material may be seen in the repository of this project [González-Barbón, 2012].

14.3. Demo application

A prototype demo application is provided for the testing of a sublanguages and its knowledge representation. The application initializes a sublanguage from a lexicon and a rules file, accepts text to validate against the sublanguage, builds a graph from text and recognizes in a reference graph the nodes and edges extracted from text. The demo application was tested with English05, the syntax based sublanguage introduced in chapter 8, and the lexicon based on Longman Communication 3000 compiled as described in chapter 7.

The main classes and functions are summarized below:

- functions to read from files a lexicon and set of production rules, and build a generative grammar against which sentences are to be licensed. This defines the sublanguage.
- functions to build a syntactic tree from a sentence. Unknown wordforms are presented to the user to add to the lexicon; the user must indicate the corresponding lemma and part of speech. Wordforms with more than one lemma or part of speech are presented to the user for disambiguation. If no complete parses are possible, the longest partial parse is shown to the user to help him determine the cause of failure, and resubmit the sentence.
- classes for semantic graph support, SmGraph, SmNode, SmEdge.
14. The KLEAR project

- an SmNode represents a node or vertex in a directed graph. An SmNode object contains a label (wordform or lemma), a syntactic tag representing the tag under which it was recognized during the parsing of the sentence, the part of speech of the label (wordform or lemma), and a unique identifier or 'nid'.

- an SmEdge represents and edge in a directed graph. An SmEdge is defined by a source and a destination SmNode objects (graph is oriented). An SmEdge contains a label for the edge, and optionally labels for head and tail.

- an SmGraph is a data structure for a directed graph, with a a list of SmNode objects and a list of SmEdge objects, in correspondence.

- all three classes have attribute dictionaries which regulate their graphical presentation.

  • a function to convert a syntactic tree into a list of nodes and edges representing a dependency tree.
  • a function to add the list of nodes and edges representing a dependency tree into an SmGraph object.
  • functions to visualize graph, in full colors or “blurred”.
  • a function to recognize constituents of a sentence in an existing reference graph. A sentence whose constituents exist in the graph in the proper relations, i.e. it represents a subgraph of the reference graph, triggers the decoration in color of the subgraph against the blurred reference graph. The colored subgraph on the blurred graph shows the “knowledge” contained in the sentence is correct, since it was recognized in the reference graph.
  • a function to calculate the distance between a reference graph and the graph resulting from several sentences, which represent the student’s answer. Each node and edge has a weight, the marks contributed by this node or edge. A correct sentence decorates in color a subgraph in the blurred reference graph, and adds the weights of the recognized nodes and edges. The addition of all the weights collected by the set of sentences in the answer divided by the total weight of the reference graph gives a mark to the answer.

The prototype is implemented as a command line application with a text menu.

14.4. Semantic graph module

This module contains classes to represent a graph, a node and an edge.

Class SmGraph, defined in module klear.semgraph.SmGraph, is a container class for nodes and edges of the semantic graph. An SmGraph object contains a list of SmNode objects and a list of SmEdge objects, the nodes and edges in the graph. The class contains dictionaries of default attributes for the graphic presentation of the graph, the nodes and the edges.

Class SmNode, defined in module klear.semgraph.SmNode, represents a node in the graph. A node has an internal identifier, a label, a syntactic tag (a part of speech if it is a terminal), an integer which indicates the “weight” of this node to be used as a mark in assessment, and attributes for the graphic presentation of this node.
Class SmEdge, defined in module klear.semgraph.SmEdge, represents an edge in the graph. This class contains a label, a source node, a destination node, arrow head and tail labels, an integer which indicates the weight of this edge, and attributes for the graphic presentation of the edge.

## 14.5. Dependency tree modules

Module klear.semgraph.deptree contains functions to build dependency trees and add them to graphs. A syntactic tree is transformed into a semantic graph based on the syntactic structure of a sentence. The approach is similar to Dependency Grammars: nodes are the heads of phrases in a sentence, edges link nodes according to dependencies inferred from the constituents in the syntactic tree.

The syntactic tree is an nltk.tree.Tree representing a sentence, from the NLTK toolkit [Bird et al., 2011]; the semantic graph is represented by an instance of class smgraph.SmGraph.

This module is specific for a scheme of sublanguage and knowledge structure. Previous requirements are:

- a generative grammar, i.e. a lexicon and a set of production rules. The grammar should allow a unique syntactic tree to be licensed from a sentence.

- a mapping indicating the head (a terminal) for non terminals in the grammar. Heads are terminals, and will be the nodes of the semantic graph. Determining the head of a constituent in the general case is no trivial task; this mapping avoids this problem. This mapping is sometimes called a 'head percolation table' in Dependency Grammars.

- if several types of nodes are to exist for different categories of terminals, a list for each category of terminal, indicating the non terminals which have these terminal as head. E.g., if nodes representing adjectives are a particular shape and color, the list of non terminals which have an adjective as head.

A dependency tree is represented by an object of class SmTree, defined in module klear.semgraph.stree. An SmTree is a tree whose nodes are SmNode objects. This module contains a function to match a dependency tree against an existing graph, recognizing the nodes and edges from the dependency tree into the graph. Nodes and edges recognized in the graph can be painted to distinguish them from yet unrecognized nodes and edges.

## 14.6. The Demo modules

A simple prototype application is implemented in module klear.kldemo; module klear.kleditor validates a given text; klear.klgraph draws the graph.

The demo application can be run in a command interpreter, like this:

```bash
$ python kldemo.py

== KLEAR Demo ==
1. Accept input text, validate, create graph, show.
2. Read sentences from file.
3. Save graph to file.
```
14. The KLEAR project

4. Load graph from file.
5. Accept text, recognize in graph (first create or load a graph).
6. Assign weights (first create or load a graph).
   r, R. Reset graph, clear nodes and edges.
   q, Q. Quit.

Option:

A typical session would create a graph with options 1 or 2, or load a previously created graph from a file with option 4, then use option 5 to write some sentences and see if they are recognized in the graph.

Modules kleditor and klgraph can be run separately, for testing and debug purposes. Invocation and accepted options are shown with their --help option:

$ python kleditor.py --help
kleditor.py: definition and test for the KlEditor class.
Options:
--nosave : do not save session changes for next session
--rebuild : rebuilds sublanguage, otherwise use last session data
--lexicon=<filename> : a pickle file with the lexicon
--rules=<filename> : a text file with the rules
--sent=<sentence> : takes <sentence> as input for parsing
--file=<filename> : reads text from <filename> for parsing
--help: print this help message
To rebuild sublanguage --rebuild must be given; if --lexicon or --rules are not given, rebuilds with default lexicon and rules.

$ python klgraph.py --help
klgraph.py: definition and test for the KlGraph class.
Options:
--file=<filename> : reads lines from <filename>, a sentence per line
--sent=<sentence> : builds a semantic graph for <sentence>
--sents=<sentence> : same, asks for another sentence
--textfile=<filename> : builds graph from text in <filename>, asks for sentences to recognize in graph.
--help: print this help message.

This demo application was fed with the English05 experimental sublanguage described in chapter 8. The use of a particular sublanguage is determined by loading a module with initialization variables; for English05 this module is klear.initeng05. Modules for other sublanguages can be built using initeng05 as a template.

14.7. Conclusions

This chapter described the software components and demo application contributed by this project.

The tools developed for the test of lexical resources were of great help, to run consistency checks on the lexicons, insert some minor corrections, get information and compare the lexicons among themselves. Most of the lexical resources analyzed showed to have errors and inconsistencies, sometimes minor and very obvious; the mechanical tests carried out with the
14.7. Conclusions

Software tools developed helped to detect these flaws, assess their importance, and eventually correct them. The definition of a common data structure to support lexical resources made these tests and comparisons possible.

The demo application is the final outcome of several modules of code written for tests and proof of concept, many of them lately discarded. Each module was written with its own tests, either as doctest code or through a complementary module specific for the module under test. The systematic use of doctests, as well as the systematic documentation of each module, class and function, were a great help to keep under control a development which grew far more than expected. The final version ended by having about 9900 lines of code, documentation included.

Despite the limitations of the application, it succeeded in putting all the pieces together and showing how the whole project might look like in a more elaborate development. Besides feasibility, this experimental code shows an application is within reach of a couple of half time developers in a six months project, or within reach of three students working on their graduation project in three terms.

A second version of this demo application should be implemented as a web application, for easy access with no software installation. An early experimental development of an editor for a controlled language confirms the project is also attainable with moderate effort.
15. Deployment considerations

Abstract. The assessment proposal described in this thesis cannot be seen just as an assessment system. Assessment is conceived as the last stage of the learning process, and integrated into it as a learning activity. Students must be introduced into the sublanguage, learn to use it, and try it in the construction of small semantic networks. Along the course, they must go on recording their knowledge of each learning unit in a set of semantic networks, as if they were writing summaries or creating concept maps. The activity must be carried on regularly along the course. Learning to write in a sublanguage, knowing its design principles and purposes, eventually introducing modifications or enhancements for specific purposes, takes a time, but students will become more skilled in objective, factual writing. The sublanguage is a subset of correct natural language; learning to write in the sublanguage is learning to write in the natural language, producing simple, clear, unambiguous texts. The writing qualities of texts written by students submitted to this training are expected to be far more readable and correct than unrestricted, unguided writing in the same course. Representation of knowledge is a recognized learning aid and activity; doing so from texts written by the students themselves simplifies the construction of knowledge representation structures. Students who have worked their way along the course, writing the texts, seeing them converted to graphs, working together towards a semantic graph where they can see knowledge recorded, will have little difficulties in reproducing the process in an assessment instance. The assessment question is just a target given to them, their answer will be just one more writing task like the ones they did along the course, matching a reference graph they have created themselves.

This chapter provides a first approach to the requirements and methodology to bring the ideas of this thesis into the classroom. The application of the ideas proposed in this thesis must meet some unavoidable requirements to be successful. In the first place, this proposal must be seen as an educational activity for learning, rather than an assessment system. Along the learning process, the material for assessment is created, and the skills for handling this material are developed; assessment comes as the last activity of the learning process, and works in much the same way as the creation of the structures.

15.1. Purpose

Most of the content of this thesis is, as might be expected, research work. Bridging the gap from research results and proposals towards real life application may be as challenging as research itself. This is more so when learning, people and computers are involved. The requirements and hints for application given in this chapter is far from a complete plan for deployment or application. The purpose of this analysis is twofold: to act as a reminder or check list to keep in mind in an attempt to bring the proposal into the classroom, and as an
15. Deployment considerations

estimation of the work, resources and engagement required to make the enterprise successful. A successful test does not mean a success of the system: a successful test is a test carried out in adequate conditions as to fairly assess the performance of the system and the value of its ideas. Many improvements and corrections will emerge in practice; if the adequate conditions are not met, if the engagement of the actors is poor, if no adequate instruction and means are available, it will be impossible to determine if flaws originate in the system under test or in the conditions under which the test was carried out.

This chapter discusses the application of these ideas as a collaborative activity in a course, characterizes a computer application to support the proposal, suggests the base knowledge and skills expected from students and calls for their engagement in the project.

15.2. A learning activity

The use of a sublanguage and knowledge representation scheme must be considered the tools for a learning activity. This learning activity is the creation of a semantic graph for a learning unit in a particular subject. The students listen to the lectures, look for references, read the assigned texts, discuss the subject, clear up their doubts, and go about the usual tasks to familiarize with a new subject. The construction of one or several semantic graphs for the learning unit is given to them as an assignment. The assignment must have a clearly defined scope, for the students to know what to include, what to leave out, and the level of detail. Alone or in small groups, they write small pieces of text describing the main concepts and relations in the subject. An application in their computers shows them the semantic graph they are building, so that they can correct and complete, until they cover the scope established in the assignment. The graphs are built by the students as a collaborative activity, guided and supervised by the teachers. Once the scope of the assignment has been reached, the teachers declare the graphs “frozen”, so that no more changes are accepted. The frozen graphs become the reference graphs for that particular learning unit, and the reference against which the students will be examined. The assessment instance consists of open questions on the knowledge contained in the reference graphs. The answers of the students are small pieces of text similar to the ones written by themselves when the graphs were under construction. Feedback and marking are immediate: the application shows the answer subgraph in color against the grey of the reference graph, and shows the percent gained by the answer.

Familiarizing with the sublanguage, the knowledge representation model, and the application to support them, conform a learning unit by itself. The students must learn the use of these tools as they must learn to use a software application, concept maps drawing, or other tools auxiliary to learning the subject. The cost of learning to use the system is expected to be slightly more than with common learning techniques such as concept mapping, but expected benefits are also greater. Probably the highest barrier will be the students’ lack of skill in the syntax and grammar of natural language. Their expected knowledge is that of a mid secondary school student. A summary revision of secondary school grammar and showing their use in a practical task is a desirable goal for Education in general.

15.3. A suitable application

An application software is an essential tool for the use of a sublanguage and knowledge representation scheme. Requirements for a sublanguage editor were discussed in chapter □ Select-
tion of a knowledge representation apt for the purposes of a course was analyzed in chapter 10. A syntax based sublanguage and knowledge representation scheme was described in chapter 11. Some hints on alternative sublanguages and representation schemes were given in chapter 12.

The application software must be capable of providing support to different sublanguage and knowledge representation schemes. As was stated in chapter 12, the complexity of human knowledge cannot be captured in a single knowledge representation model, nor is it necessary for educational purposes; the subject matter on hand will be better described in some schemes than in others. The purpose of the course, its scope, and the audience, provide further guidance as to the best scheme to use. The design of the application software must be able to assimilate different schemes. In the prototypes described in chapter 14, the sublanguage is loaded from files, and a configuration file initializes the necessary variables to build dependency trees and define the presentation of the graph. Though the prototype confines itself to syntax oriented sublanguages, the same mechanism can be used to support alternative schemes.

In the present state of the art, a strong recommendation is to implement the software application as a web application, even for testing purposes. A web application is available from everywhere, requires no installation on client computers, and centralizes control of the application software and data. Changes in the sublanguage and knowledge representation scheme are immediately reflected on the client computers, a valuable feature in experimental software.

15.4. Learning units

In course planning, once learning objectives for the course have been determined, content is arranged in topical units. Each topical unit typically expands several hours [FSU, 2011, ch 2]. A learning unit, in the scope of this work, is closer to the idea of content assigned to a learning object. Learning objects are small units of learning with durations between 2 and 15 minutes. A learning object addresses only one learning objective, and is self contained, which means it can be taken independently. Learning objects can be aggregated into large collections which conform the learning material of a course [Beck, 2010].

The proposal of this thesis is not based on the learning object paradigm, but fits well into it. Use of a sublanguage and knowledge representation scheme is effective for learning units with the same characteristics of content packed into learning objects. Besides duration, the extension of a learning unit can also be regulated by the size of the semantic graphs built, but most probably both criteria will yield similar extensions.

A learning unit will then address a single learning objective, be self contained, and prepared to be aggregated into a collection of similar units to conform a course.

In planning a course, to test the sublanguage representation scheme, the activity can be proposed to some specific units, without engaging the whole course. It must be noted that the sublanguage representation scheme proposed in this thesis is by no means a learning technique, but just a learning activity, a complement to the other learning activities, which adds the plus of preparing the students and the material for an automatic assessment instance.
15. Deployment considerations

15.5. The representation scheme

The representation scheme includes the sublanguage and the knowledge representation model. A sublanguage and a corresponding knowledge representation model must be chosen or developed from the start. The lexicon can be adjusted as it is being used; some new terms or some changes in existing terms, e.g. their part of speech, are quite harmless. The rules are difficult to bring together into a consistent set, and are difficult to test in all cases. Rules must always lead to correct use of the natural language, even if limited to the sublanguage constraints expressed in these same rules. On the positive side, the rules are less prone to changes and more widely usable; the same rules can be used for very different subjects.

Some hints on the construction of a sublanguage were discussed in chapter 5. The sublanguage is closely related to the knowledge representation model; the transformation of the syntactic structures into the components of the semantic graph must also be carefully defined. English05 is a syntax based sublanguage described in chapter 8; its corresponding knowledge representation model is described in chapter 8. Some hints on alternative sublanguages and knowledge representation models are discussed in chapter 12.

15.6. Documentation and tutorials

Adequate documentation and tutorials are essential for the successful application of a representation scheme. Resources required for writers in a sublanguage were discussed in chapter 5. A list of the essential support material includes, but is not limited to:

- a tutorial of the sublanguage, as described in chapter 5
- a lexicon, easy to look up, with information on wordforms, headwords, parts of speech, and senses for each (headword, part of speech) pair.
- a tutorial and user manual of the application, describing how to use the editor, how to build the graphs, and how to enter text to be recognized in a reference graph.
- additional documentation on the sublanguage, such as a specification for the construction and maintenance of the lexicon and production rules.
- a tutorial of the knowledge representation model, its purpose, graphics conventions, and the rules for the transformation of the syntactic constructs of the sublanguage into the nodes and edges of the semantic graph.

Specially in tutorial material, examples are essential; many people learn better and quicker by example. Since the structures of the sublanguage are structures of the natural language, and probably the most commonly used, a well contrived set of examples may be all it is required to start writing in the sublanguage.

Syntax based representation schemes will most probably be accepted as natural, but other representation models will require a sound explanation of their purpose, foundations and conventions of representation. To understand the object oriented representation proposed in chapter 12, most people will need to be told of categories and individuals, why two separate graphs are needed, and how they relate to each other. A sequential model is not so difficult to grasp, but in all schemes some degree of convention is present, and these conventions must be explicitly told.
15.7. Basic knowledge and skills

A carefully designed application for the support of a sublanguage and knowledge representation scheme will be easy to master. The most difficult skill to develop will most surely be getting familiar with the sublanguage and writing compliant text. Anyway, the sublanguage should not be an obstacle for any student with a reasonable knowledge of the grammar of the natural language. As stated in chapter 11 on syntax based sublanguages, the expected knowledge in grammar is that of a mid secondary school student. Getting to use a restricted lexicon is essentially a matter of habit; all accepted and frequently used words will be in the lexicon, either from the start or because the users have added them. The lexicon is conceived as a flexible tool which can be dynamically adapted. As for the rules, the experimental sublanguages described in 11 required about 20 rules. The rules reflect very common structures, and are easy to grasp. Addition of conjunctions for use in some situations, and other extensions, will most probably keep the rules under 30. This is still a very modest number.

Nowadays, knowledge of grammar in secondary students is not as good as it should be. Though this is a drawback for the adoption of the proposal, in this situation the use of a sublanguage becomes a tool for remedial work on grammar: the grammar reflected in the rules of the sublanguage are still fully valid rules in the grammar of natural language.

The teachers should be reasonably proficient in the grammar of the natural language, to guide the students as necessary. Including a language teacher in the team, or working collaboratively on language subjects, will be most advantageous.

Familiarizing with the knowledge representation model may require some tutorial and training, too. Representing knowledge in semantic graphs is generally accepted as almost natural; other structures may require more tutoring work. Syntax oriented schemes are most immediate, but an object oriented scheme or a sequential scheme like the ones described in chapter 12 will demand specific training.

15.8. Engagement of the actors

The immediate actors in the learning process are the students and the teachers. Course planners, support programmers, document writers and other staff are not so visible, but play a significant role in the process. Though a small scale experiment may be carried on just inside the classroom, almost immediately it will require some additional support: if there is a web application, the system administrators must be aware, storage space must be provided, assessment instances securely managed, and a wider attention called for. Adequate planning is always the best way to go.

As with all new ideas, difficulties may arise to engage people to try a new activity, the more so if it involves assessment. In an experimental situation, the assessment instance may be just informative, with no consequences on the marks given to the students. Several terms may be necessary to come out with a reliable, accepted scheme. What is necessary is that all people involved in the project are convinced it deserves a try, and do their best to make it work. To this purpose, they must be aware of the purposes, the tools, the assignments, the benefits, the limitations. A sound proposal, a practicable plan, and responsible discussion are the ways to earn the engagement of the actors.
15.9. Conclusions

This chapter provided a guide of deployment, gathering the main aspects and cares we consider essential towards a successful test and use of the system in a course.

Though testing learning tools in the classroom is generally a difficult task, the former analysis suggests the endeavour is practicable. A suitable application, a clear understanding of the tools, a collaborative approach, and the engagement of both teachers and students, are considered the key requirements for successful testing.
Part VI.

Final
This part includes evaluation of the proposed solution, contributions of this thesis, future work, and conclusions.

Chapter 16 discusses some issues related to the evaluation of assessment systems, requirements for their validation, how to conduct a classroom evaluation, proof of concept evaluation, and more specific aspects of evaluation related to the sublanguage and knowledge representation scheme adopted.

Chapter 17 describes the contributions made in this thesis, in the areas of Education, Assessment, Language, and Knowledge Representation, as well as the software tools developed, which may be taken as a start point for a production-quality application to support the learning and assessment model proposed in the solution. Several directions of future work are also identified.

Chapter 18 discusses the conclusions inferred along this work. The chapter starts with an analysis of the extent of accomplishment of the objectives originally proposed in this work. Conclusions range through the three main fields involved in this work, namely Education and Assessment, Language and Knowledge Representation. The solution proposed is more like a complete learning activity than just as an assessment system; it is a learning activity that includes assessment as the final step of the learning process.
16. Evaluation

Abstract. The proposal of this thesis is presented as a learning activity. Evaluation of a learning activity must be done in the classroom. A sound classroom testing requires further development to reach a production quality application, course planning in the form of learning units, training of the students on the methodology and use of the system, as previous steps to introduce the students to the learning units, the construction of the knowledge structures, and assessment. A small team of developers and teachers could carry on this plan in two semesters. Validation of the proposal was made as proof of concept, based on the prototypes developed, the tests performed, the examples solved and the impressions of observers. The resources evaluated, the data structures developed, and the tests performed showed a sublanguage can effectively be compiled, based on a lexicon of commonly used words, and rules to support declarative sentences of quite wide application. The prototype application, though limited, was effective in licensing the sentences according to the sublanguage loaded into it, provided enough help to detect errors by showing partial parses, produced a clear, intuitive visualization of the knowledge structure with the parts recognized from the text answer displayed in colour, and a mark indicating the percent matched by the answer in the graph was immediately available. Semantics for the transformation of text into graphs could be established based on ideas from dependency grammars. A syntax based sublanguage and knowledge representation scheme was developed and tested by examples. Different sublanguages and knowledge representation schemes were analyzed, several of them able to be implemented with some additional work from the syntax based scheme developed. Future work includes development of a production quality application, extensive field testing, further research and experimentation on sublanguages and knowledge representation schemes, engagement of resources such as Wordnet, ontologies and common sense databases. The proposal, in its present state of development, has shown to effectively work as a text to graph knowledge representation facility, apt to be used both as a learning tool and as an assessment facility. Equally important, these developments and tests, having been a single man's work, show the real world application of the proposal is within reach of a small team of developers and teachers: required development and a reliable classroom test can most probably be carried out in a year's project.

Evaluation tries to determine if a proposal is worthwhile; validation establishes the soundness of its statements, and if they can hold in the real world. This chapter states some considerations and principles which guided the evaluation step of this thesis, and provides a proof of concept of the proposal, showing that its real world realization is possible and within reach of a small team in about a year's work.
16. Evaluation

16.1. Introduction

Proposals for the automatic assessment of free text answers usually take the form of a system: the students' answers are fed into the system, algorithms are applied, and marks produced. Validation of these proposals essentially consist in determining if marks given agree with marks held to be accurate, such as marks given by human examiners.

The nature of the solution proposed in this thesis is that of a learning activity, not of an assessment system; assessment comes as the last step of a knowledge representation construction, which the students did by themselves as part of a learning process. Owing to the nature of this proposal, the critical points of evaluation will not be on the accuracy of the answers when compared to a reference; this agreement is guaranteed by the matching of the answers against the knowledge representation instance, which is strict.

The following sections discuss the limitations of typical evaluation forms for this kind of proposal, identifies the critical points on which evaluation must concentrate, and describes the realizations and warrants which support the validity of this proposal.

16.2. Evaluation of assessment systems

Most tests of validation of assessment systems consist of single assessment instances: written answers to an exam question are brought to the system for correction, and the results produced by the system are compared against a reference. The reference is usually the same set or a representative subset of the students' answers corrected by human teachers. Except for very simple questions, human marking exhibits dispersion, may differ in criteria, and is not free from bias. However, there is probably no better alternative. Human marking has a long tradition, is universally accepted, and brings in human judgement, a virtue no machine or system may claim to provide. Once the reference has been accepted, the main point of evaluating assessment systems is their accuracy: how close are the marks given by an automatic system to the accepted reference of correct marking?

This question is not relevant for the evaluation of the solution proposed in this thesis. The students' answers are recognized in a previously built knowledge structure; the correct answer exists before the students produce their free text answer, the knowledge extracted from the students' texts either matches or does not match the reference knowledge structure.

The assessment proposal described in this thesis is conceived as a learning activity. The assessment ‘system’ is embedded in the learning process, it cannot be evaluated separately.

16.3. Requirements for validation

Evaluate means to ascertain or fix the value or worth of something; it also means judge carefully, appraise something. Validate means to establish the soundness of something, corroborate or confirm [Farlex Inc., 2010]. Validation can be conceived as a part of an evaluation: if the proposal can be validated, i.e. its soundness can be proved, then the value of the proposal is increased.

It is always challenging to ensure a validation scheme is trustworthy. The validation of an educational proposals offers particular challenges: the changes an educational proposal is expected to bring occur inside the individuals, alter their knowledge, and modify their behavior.
The big risk of validation is doing it the wrong way. Faulty validation may give false evidence both on the positive or the negative side: false evidence may lead to consider a proposal successful when it is not, or conversely, conclude a proposal is a failure when it may not be so. One of the most frequent reasons for defective validation is the lack of adequate conditions, either because they could not be achieved or because they were unknown at the time of performing the validation tests.

Evaluation of a learning activity must necessarily be done in the classroom. Different instances, different populations, different subject matter, different teachers, adequate tools, careful recording of circumstances, comparison of results obtained in the different instances, comparison against other forms of learning the same subjects, are some of the elements to consider.

Requirements for validation of the educational proposal made in this thesis have been pointed out in chapter 15, where deployment considerations were analyzed. A summary list follows:

- a suitable application; it may be an evolution of the prototypes developed.
- a course plan, divided into learning units, with a well defined scope.
- a sublanguage and representation scheme adapted to the subject matter of the course.
- documentation, tutorials, specifications, including a guide for the maintenance or enhancement of the sublanguage and knowledge representation scheme.
- basic knowledge of grammar, the representation scheme and other previous knowledge and skills, or a remedial work plan to account for them.
- engagement of the actors: students, teachers, developers, and institutional support.

In the next sections some hints on how to carry on a classroom test are proposed, followed by the proof of concept validation which backs up the claims of this proposal.

16.4. A classroom test

A classroom test must be done with the help of a production quality application, adequate course planning, and a team of teachers and students engaged in the enterprise. The present state of development does not comply with the minimum requirements for a full, conclusive validation of the proposal in the classroom. Research and development carried along this project produced the basics upon which the tools and methods for a learning activity can be developed.

A classroom test can be planned in two terms of one semester each:

1. Preparation: development of the application, adjustment of the sublanguage and knowledge representation scheme, high level course plan of learning objectives and learning units.
2. Application: detailed planning of each learning unit, remedial work for previous skills, training in the use of the application, study and practice with the sublanguage and knowledge representation scheme, collaborative construction of the knowledge structures, and assessment for each learning unit.
16. Evaluation

The preparation time will be difficult to reduce; it was formerly estimated in a half year project for a small team of developers (chapter 11). The application time might be reduced if only a few learning units were taken. This reduction involves some risks:

- the preparation of the students will take some time, which must be enough to familiarize them with the application, the representation scheme, and probably do some remedial work. Undue reduction of the preparation time will work against the validity of the results.
- the first two or three learning units will not provide sound evidence of performance, since the students, and even the teachers, will be still getting confident in the proposal.

Classroom testing does not require a special course to be put up; it can be integrated into an existing course, just selecting some learning objectives and articulating the learning units accordingly. In any case, adequate preparation and enough application time must be ensured.

16.5. Proof of concept validation

In fields as different as engineering, business, film making or drug development, full testing of new ideas may not be possible owing to time, funding, resources, or other limitation. Some kind of testing is required, though, to prove the validity of the ideas, bring some confidence to investors, and get the funds for development or further testing. This partial testing is often called proof of concept, though it is essentially a feasibility test, a proof or reasonable evidence that the proposal will be effective in its expected results.

Proof of concept is the realization of a certain method or idea to demonstrate its feasibility. A proof of concept is frequently small, not complete, and not apt for real use. A prototype may be constructed as a proof of concept; this prototype need not be a first version of the product, its only purpose is to show its working possible.

The following sections review and comment on the proof of concept realizations of this project.

16.6. Sublanguage

The construction of a sublanguage includes a base lexicon and a set of grammar production rules. A review of sublanguage related realizations follows.

16.6.1. Lexicon

The compilation of a lexicon was based on the collection, testing and transformation into a common data structure of word lists and other lexical resources freely available. Lexical resources characteristics and usefulness were discussed in detail in chapter 6, the compilation of a lexicon was described in chapter 7.

Existing resources, freely available, could be successfully engaged into selecting the most commonly used words with their most common part of speech; these resources could be used for lemmatization and part of speech tagging. Definitions could be brought into the lexicon from the Wordnet. For a production system, definitions must be agreed upon by the experts of the field, since each word usually points to more than one.
The base lexicon used in experimental sublanguage English05 was based on the Longman Communication 3000 word list [Longman, 2008], with very little changes. Function words were selected from an English grammar book [Altenberg and Vago, 2010]; not all function words were included, since function words contribute significantly to the syntactic structures allowed. Function words were handled separately from content words; wordforms accepted as function words were not included as content words. For English05, a set of function words were selected as necessary for the rules; the Longman 3000 word list was only allowed to contribute content words; content words expressed by the same wordforms as function words were excluded from the English05 lexicon.

16.6.2. Production rules

The rules for the generative grammar which licenses the sentences in the sublanguage were brought in from natural language grammar books. Good secondary school textbooks on grammar proved a useful source; secondary school level of natural language grammar knowledge is what is expected from the students, and good enough for the teachers. Both for Spanish and English a set of about 20 rules were enough to allow for a reasonable variety of declarative sentences in the syntax based sublanguage. The resulting rule sets are quite simple to understand, and within reach of any student with a decent secondary level knowledge. Though putting them together in a consistent way is not a simple task, it could be done in a relatively straightforward way following the grammar books.

Rule sets for English05 and Spanish05 described in chapter 8 are very close to rule sets usable in production; the expressiveness allowed is wide enough for use in many subjects. A number of specific situations such as measurements, dates, proper nouns, compound words, can be dealt with by preprocessing the text and pattern recognition, as explained in chapter 5.

The rules were written in a plain text editor. The NLTK framework offered the tools to read the rules from text to implement the generative grammar, and parse the sentences into syntactic trees [Bird et al., 2011].

Details for the construction of a sublanguage were given in chapter 5; the experimental sublanguages developed were described in chapter 8; the complete listing of rules and example sentences for each sublanguage developed can be seen in appendix A.

16.6.3. The KLEAR editor

The prototype sublanguage editor proved not only to be capable of licensing sentences according to the grammar, but also showing partial parses, wordforms and part of speech, which helps the author to rewrite a sentence which could not be licensed by the grammar. Though the enhancements listed in chapter 5 would mean valuable additions to the editor, the prototype version developed was perfectly usable, showing partial parses for sentences not licensed. Presentation of syntactic trees, whether partial or complete, even if in text form, were relatively easy to understand, separated in lines and adequately indented.

Though some practice is unavoidably required, use of the editor proved it to be of effective help when coming to write in the sublanguage.
16. Evaluation

16.7. Knowledge representation

A knowledge representation scheme is the use of a knowledge representation language in a certain way (chapter 10). Examples of knowledge representation schemes were given in chapter 12. A sublanguage and knowledge representation scheme, or representation scheme for short, includes the use of the knowledge representation language in a certain way, plus the sublanguage and conversion rules from text to graph (chapter 12).

16.7.1. The knowledge schemes

A syntax based knowledge scheme was developed for the syntax based sublanguage. The transformation from the syntactic trees produced by the syntax based sublanguage was done through an algorithm based on the ideas of dependency grammars: the head of each constituent was raised up in a tree, to obtain a dependency tree which reflected the concepts and relations extracted from the original sentence. The rules for the transformation of the syntactic tree into the dependency tree were fed into the application in the form of a list which gave the head for each constituent. This not only allowed for experimentation and adjustment, but also for the use with other sublanguage and knowledge representation schemes.

The syntax based sublanguage and knowledge representation scheme provided an end to end proof of concept of the system. Restricting the syntactic structures allowed to build a knowledge representation instance which adequately reflected the semantic content of each sentence, and successive sentences could be added into the knowledge structure to reach the scope of a learning unit. As stated in chapter 11, several improvements are required for a production system, in particular control of redundancy, which was only partially implemented, but can be done through pattern matching. Even with these limitations, the syntax based representation scheme successfully extracted knowledge from the text written in the sublanguage.

Other knowledge representation schemes were analyzed in chapter 12. Though they were not effectively developed, some of them can be implemented by introducing modifications in the syntax based sublanguage developed; this is the case of the sequential and the imperative representation schemes. The object oriented scheme is a well known paradigm, and the syntactic structures proposed in chapter 12 result in a manageable grammar; visual representation will most surely follow the UML diagrams. A representation scheme based on Topic Maps is the most promising, but calls for a lasting project for adequate development; this places this proposal outside the reach of a small team in half a year, which was the estimation for development of the other schemes.

16.7.2. Visual presentation

The data structures for knowledge representation were defined as Python classes; these classes are essentially the data structures of a graph, in which nodes and edges can be assigned a variety of properties. The same Python classes can, consequently, support different knowledge representation schemes. The preferred form for visual presentation was the semantic graph, for its visual effectiveness and intuitive appreciation, which allow a newcomer to almost immediately realize the conveyed meaning.

In several informal sessions, the example representations created through English05 were immediately grasped by the observers, even casual observers. The questionable point of indeterminate semantics of graph representations, such as concept maps or semantic maps, is here
resolved by the interpretation given to the syntactic structures and the rules of transformation of text into graph. Each sublanguage and knowledge representation scheme must define its own semantics, hence an observer which knows the scheme will see the meaning in the graph with little doubt.

16.7.3. Distance measure and marking

The prototype application and the data structures supporting the knowledge representation scheme allow the assignment of marks to each node and each edge. In the prototype application, this was done through part of speech, considering nouns as the most significant, followed by verbs, and finally adjectives and adverbs; edges were not given any marks. Knowledge extracted from the free text answers is immediately recognized and visually presented in the reference graph, originally blurred, then coloured on the recognized parts. At the same time, the count of marks gained through the correct answer are added, and the totals and percent presented in the same graph. Immediately after the introduction of the answer text, the percent obtained is shown.

This way of presenting the recognized knowledge in a graph, and the percent gained by the answer, proved to be very intuitive and immediately understood. Even a limited application as the one developed succeeded in bringing up these qualities.

16.8. Evaluation results

The present proposal for eAssessment is based on a relatively long term collaborative work along which the students create the knowledge structures that will be used to assess their knowledge. Assessment is presented here as the final activity of a series of learning activities carried along in a course. Definite testing of such a solution must necessarily be made in the classroom. Reliable classroom testing was formerly estimated to be a year’s work of a small group of developers and teachers; this will be a small project on its own.

Validation of the present proposal was done as proof of concepts. From the work done, the tests carried out, and the comments of observers, it can be stated, with certainty enough, that:

- **the proposal is feasible**: a sublanguage can be compiled, a knowledge representation defined, conversion from text to structure performed, visual presentation of the answer shown and a mark given.

- **the proposal is within reach of a small team in a year’s work**, classroom validation included. The code, prototypes, examples and tests were all a single man’s work, and most of it is reusable.

The nature of the proposal exhibits some features that, though difficult to quantify, makes it attractive both for students and teachers, a difficult point to make by an assessment proposal:

- the system is fair: all elements and steps in the learning process are in full knowledge and consent of both teachers and students; both teachers and students share in the same sublanguage, the same representation scheme, the same learning material, and the same recording of knowledge in the knowledge structures. Even the weight given to items in the knowledge structure may be discussed with the students (though here the teachers should have the final word).
16. Evaluation

- the students learn with the same tools and application they will be using in assessment; an assessment session is quite similar to any of the previous sessions on which the learning activity was performed.

- the students know exactly what they will be questioned about; the assessment question is just a focalization on this or that part of the knowledge structures.

From an educational point of view,

- the students improve their writing: the sublanguage is correct natural language; the students become conscious of ambiguity and the complexities of the language, and improve their accuracy in communication; the students experiment the benefits of simple writing, and clear out of involved constructions.

- the proposal combines writing with comprehension, the students identify concepts and relations, select the essentials, build a “summary” of the subject by recording what they learn in the knowledge structures built from their texts.

The construction of knowledge structures is an activity similar to concept mapping; in this proposal, it is carried along by writing in a sublanguage. Concept mapping is by this time a well known and accepted learning activity. This leaves the sublanguage as the critical part of the proposal. The limitations and risks of compiling, and using a sublanguage, were discussed in chapters 5 and 8. The simple languages such as Basic English are used to teach the language; many tutorials and recommendations on simple writing exist. A sublanguage for the purposes of this thesis is similar to those simple languages.

The critical point of a sublanguage leads to the critical activity of compiling a sublanguage; much will depend on its simplicity and proximity to natural language, two crucial design considerations insistently remarked. A sublanguage was proposed instead of a controlled language to avoid the strict rules of a specification, which might frighten away both students and teachers. An evolution of this proposal, with adequate tools, might evolve or adopt an existing controlled language, with the advantage of wider use and a carefully engineered specification.

16.9. Conclusions

The nature of the proposed solution and the limits of a thesis work prevent conclusive evaluation of this proposal. However, even keeping on the conservative side, the proposal is seen as feasible, cost effective, and very attractive from the educational point of view. As pointed in chapter 15, a great deal will depend on the actual deployment of the system.

The nature of the proposal, conceived as a learning activity, gives further confidence to invest on it. Adequately conducted collaborative work has proved an excellent educational feature. Involving the students in the assessment process, telling them they will be creating the structures against which their exam answers will be marked, is as strong a motivation as can be.
17. Contributions and future work

Abstract. Original contributions of this thesis lie in the combination of ideas, methods and techniques of Assessment, Natural Language Processing and Knowledge Representation, towards the automatic marking of free text answers. In Assessment, contributions of this proposal comprise: a determination of the educational requirements for an automatic assessment system, acceptable for both teachers and students; the shared responsibility of both teachers and students in the creation of the assessment material; an effective integration of assessment into learning, as one more learning activity. In Language, contributions of this proposal comprise: the characterization of a sublanguage for Education; a methodology for building such sublanguage; a procedure to convert text into knowledge structures. For lexicons, contributions include: the identification, self checking and comparison of several lexical resources; the definition and testing of a common data structure to support them; a selection of English function words to be handled separately from content words. Two experimental lexicons were contributed, based on freely available short lists, namely the General Service List of English Words and Longman Communication 3000; their coverage was tested against the Brown corpus. This thesis contributed several experimental but usable sets of production rules for both Spanish and English, which were tested on small sets of representative sample sentences. Contributions in Knowledge Representation include: an evaluation of techniques and languages in their potential for Education; a syntax based sublanguage and knowledge representation scheme, developed and successfully tested in a prototype application, which proved end to end feasibility; a critical appreciation of other possible knowledge representation schemes, with hints on the types of knowledge relations they can best model. For the practical application of the ideas in this thesis, a guide to deployment, for field testing and for use in the classroom, was put together. Future work includes development of a production quality application, extensive field testing, further research and experimentation on sublanguages and knowledge representation schemes, engagement of resources such as Wordnet, ontologies and common sense databases. The most valuable contributions of this thesis are thought to be a practicable proposal that effectively integrates assessment into the learning process, and a satisfactory end to end testing of the system which proves its feasibility.

The proposal for the eAssessment of free text answers made in this thesis led to contributions in the areas of Assessment, Language, and Knowledge Representation. This chapter summarizes these contributions.
17. Contributions and future work

17.1. Contributions

The automatic assessment of free text answers, as treated in this thesis, is related to three different fields of knowledge: Assessment, Natural Language Processing and Knowledge Representation. Though some original contributions may be assigned to these fields, the originality of this work lies rather in the combination of existing ideas, methods and techniques from these areas towards a new solution for an existing problem, a solution which is considered better than existing ones. This is an application of the methodology proposed by Adrion for software engineering research [Adrion, 1993], adopted in this thesis, as stated in chapter 1.

17.1.1. Assessment

In Assessment, contributions of this proposal comprise:

- a determination of the requirements for an assessment system satisfactory both for students and teachers (chapter 2). An effective solution is not just one that works, but one that is also accepted as valid, useful and reliable. Expectations of teachers and students are no easy to make compatible; knowing their respective expectations is a first step. This was done by an evaluation of importance and the consequent selection of the requirements to be considered for a reliable and acceptable system. First we established what we consider the most widespread expectations of the actors, together with the educative values to preserve in an open answer system. Then, the limits inherent to automatic assessment determined the type of answers to accept and the kind of knowledge to be assessed, i.e. assertive sentences and factual knowledge. Going through the assessment process, the facilities expected to be found in an application emerged, which determined what we consider an ideal system. The former items were analyzed, selected and organized according to our own view. Though they are by no means conclusive, they provided an orientation to our work, and provide a sound base for discussion in future work.

- the shared responsibility of teachers and students in assessment. A computer assisted assessment system was originally thought the best option, instead of a purely automatic system (chapter 2). In the end, the proposed system became purely automatic for assessment, but since the teacher guided the creation of the knowledge structures for assessment, no further intervention is deemed necessary (chapter 13). The students and the teachers share the responsibility for the correct creation of the knowledge structures to be used in assessment (chapter 15).

Though collaborative approaches in learning are almost universally encouraged, it proves difficult to put them into practice in assessment: the student must demonstrate her knowledge or abilities, the teacher must determine if such demonstration is acceptable; a teacher will do her best to show the student her mistakes or deficiencies, but ultimately the teacher is judge. In our proposed system, all disagreements must be cleared up during the learning process, when the knowledge structures are built. The reference knowledge structure against which the student’s knowledge is to be measured has been put together, and agreed upon, by both students and teachers. Any difference of appreciation on a subject item must be dealt with during learning, at a moment when
17.1. Contributions

reference material, other opinions, and constructive discussion can be brought into play. The reference knowledge structures are thus known and agreed upon by both students and teachers; since the marking process is now entirely automatic, there is no place left for doubt. For its educational value, we consider this feature a very strong point in favor of our proposed system.

- the effective incorporation of assessment into learning, as one more learning activity. This proposal implements this concept in its nature, since the students build the structures to be used for the assessment of their knowledge, and work in the assessment instance in the same way they did when they were studying the subject (chapter 15).

The answering of traditional assessment questions, closed or open, bring the student into a situation which is radically different from what she has been doing at learning time: some unknown questions must be answered by choosing alternatives, producing some words or results, or writing a piece of text. In many cases, even the type of question is not known to the student beforehand. There is usually no previous training instance, and the student eventually becomes familiar with assessment only through experience, along her life as a student. In some stages of education this previously unknown way of assessment may even be desirable, since it somehow emulates the unpredictability of life, but in most others it puts a quote of uncertainty and stress on the student which impairs her performance. In our proposed system, the assessment instance consists of producing a text which maps to a knowledge structure exactly in the same way the student did when studying the subject, in the certainty that what she produces will be compared against the same knowledge structure she has known when studying, and has helped develop. Nothing new, no surprises will turn up in the assessment instance, just the same task the student has been doing when studying. This relieves the student of the stress of the unknown, and gives her more confidence in her own performance. For this reason, we consider this feature another strong point in favor of our proposed system.

17.1.2. Language

In the field of Language, contributions of this proposal comprise:

- the characterization of a sublanguage for Education and correct writing (chapter 5). The sublanguage proposed lies in between of strict controlled languages and recommendations of style, enforcing some mild limitations on natural language, keeping the natural look of written texts, and readable as correct natural language.

Except for some proposals for second language teaching, we do not know of sublanguages designed for educational purposes. Our view on the features which make restricted languages desirable for education were given in chapter 4; some of the difficulties and prejudices which weigh against the use of restricted languages in education were analyzed in the same chapter, and an argument made to demonstrate most of them can be successfully coped with. We reviewed existing restricted languages, focusing on human oriented languages for general use, to select the main requirements for a sublanguage suitable for education (chapter 5). We added some additional requirements to ensure determinism (i.e. unambiguous texts), an essential quality to extract knowledge from text. This resulted in a set of writing rules which favor tractability of the texts while
17. Contributions and future work

preserving readability and correct writing (chapter 5). Though not mandatory, and
arguable in their relative importance, this compilation of writing rules provide a basis
for the design of a sublanguage for education. Some knowledge of essential grammar is
required to write in such a sublanguage, and since writing in a sublanguage demands
a bit of discipline on the writer, some resources and tools were detected as desirable,
or even utterly necessary, to render the system friendly to the user (chapter 5). The
analysis performed, and the products obtained, provide a guideline towards the design
of a sublanguage for educational purposes, while keeping the designer free to draw from
existing proposals or devise one of her own.

- a first approach to a methodology for the construction of a sublanguage, including design
  considerations, identification of general writing rules, facilities an application must pro-
  vide, documentation and tutorials, detected difficulties, and ways to solve them (chapter
  5).

Based on our previous characterization of a sublanguage for education, several example
sublanguages were developed (chapter 8), in a design of our own. The experience gath-
ered in the construction of these example sublanguages is recorded in chapter 5. Features
such as units of several words, proper nouns, initial capitals, numbers, dates and mea-
surements, can be handled by preprocessing the text and applying pattern matching.
A discussion of other enhancements included word completion, partial parses, part of
speech anticipation, lexical help, word sense disambiguation. Besides proving the feasi-
bility of the endeavor, the experience of building these example sublanguages allowed to
define a first approach to a methodology for the construction of a sublanguage which, in
our view, provides a helpful basis for future developments. The more specific methodol-
dy inferred from the design of production rules for a generative grammar is discussed
in its own paragraph later in this chapter.

- a procedure to convert text written in a sublanguage into a knowledge structure, including
  the identification of tasks required, the use of ideas from dependency grammars, the
  definition of a simple tagset, and the use of substitution tags to decouple lexicon and
  rules (chapter 5, chapter 11).

The design of the sublanguage to write texts apt for knowledge representation must par-
allel the knowledge representation language, since conversions from all syntactic struc-
tures of the sublanguage into corresponding structures of the knowledge representation
language must be defined. The steps and tasks identified in chapter 5 were later applied
in the conversion of a syntax based sublanguage into a knowledge representation in the
form of a graph, as described in chapter 11, and also in the discussion of the other
sublanguage and knowledge representation schemes proposed in chapter 12.

Concerning lexicon, resources available are far more mature and abundant in English than
in Spanish; studies of lexicon were limited to English for this reason. However, the tasks
developed and the methodologies inferred can be equally applied to lexical resources in Span-
ish, or other occidental languages. The Wordnet, in particular, is available in English as a
free resource, while the EuroWordnet project requires a fee. Concerning lexicon, this thesis
provided the following contributions:
17.1. Contributions

- **identification of resources freely available, evaluation of their usability, and tasks which they can do** (chapter 6). Checks for internal consistency, detection of errors, correction of some simple errors without altering the essentials of the original lists (chapter 6). Software tools and tests available at [González-Barbone, 2012].

Our use of lexical resources was limited only to those freely available on the Internet: GSL (4 versions), Simple English Wikipedia (related to Basic English), VoA Special English, Longman Communication 3000, and the long lists BNC, COCA, SCOWL, AGID, and PoS DB; though strictly not a word list, the Wordnet was identified as offering great potential for future developments (chapter 6). Several checks for internal consistency were run on each list, such as repetitions, non alphabetic characters, and format anomalies. Some errors were found, even in lists very carefully maintained. Some small corrections were made, specially in the short lists used (GSL Gilner and Longman 3000). Work done on these lists allowed to determine their coverage, quality, and tasks they can perform. A description of each, with some examples of the anomalies found and size information are given in chapter 6. Software developed for the handling of lexicons is contained in the lexicon module of the KLEAR project, described in chapter 14. Documentation of the software tools developed, Python modules for testing, and doctest files of the tests performed are available at [González-Barbone, 2012].

- **a comparison among available lexical resources.** Comparison of the short lists of common words for superposition, coverage of short lists in the long lists, what to do to improve or make them usable (chapter 6).

Several lists were tested one against the others, such as words in the short lists contained in the long lists, or comparisons of the short lists among themselves. Tables in chapter 6 summarize results. Python modules for testing, and doctest files of the tests performed are available at [González-Barbone, 2012].

- **a data structure and functions to support lexicons, converting from different original formats** (chapter 14) software tools and tests available at [González-Barbone, 2012]. A common data structure allows for the testing and comparison of lexical resources, and for their use in experimental applications.

The data structure to support lexicons was designed to accomplish the tasks of inclusion (if a word is in the lexicon, either as wordform or headword), lemmatization (eventually more than one headword per wordform), PoS tagging (eventually several PoS per word), frequencies of use (by wordform, by headword and PoS, by headword alone), definitions (meanings of a headword), synsets related to a headword. Efficiency was a concern, so most structures were implemented as Python dictionaries, which are very fast on access by key; some controlled redundancy was introduced as deemed necessary. Tasks for a lexicon are given in chapter 7 the lexicon module is described in chapter 14; software documentation is available at [González-Barbone, 2012].

- **two experimental general purpose lexicons**, one based on the General Service List of English Words (GSL), and another based on Longman Communication 3000 (chapter 7). Both are lists of commonly used words.
17. Contributions and future work

These experimental lexicons are available as Python pickle files at [González-Barbone, 2012]. As they are read they become available as Python objects in the common data structure defined for handling lexicons; documentation of the lexicon module is available at [González-Barbone, 2012].

- **a first approach to a methodology to compile general purpose or domain specific lexicons;** function words were analyzed separately, since they have little semantic content but articulate syntactic structures which must be recognized (chapter 7).

The experience gained in the compilation of the example languages developed provided a first approach to compile general purpose lexicons to serve as a basis for a domain specific sublanguage. The data structures for lexicon support were designed to accomplish tasks such as inclusion (if the word is in the lexicon, as wordform or headword), lemmatization, PoS tagging, frequency of use, definitions. A GSL based general purpose lexicon was compiled, first selecting function words, then adding content words, but keeping both sets disjoint; this required some careful design decisions. Several adjustments were considered, such as words with several PoS, function words which are also content words, content words left out of the lexicon, self consistency tests, handling of proper names, numeric quantities, dates, inflections, and sense (chapter 7). The construction of a GSL Gilner based lexicon was later followed by the construction of a Longman 3000 based lexicon, along the same lines; the approach proved to be equally useful in both trials.

- **tools and procedures to test coverage against text corpora** (chapter 14, software tools and tests available at [González-Barbone, 2012]). The experimental lexicons developed were tested against the Brown corpus.

The tools and software developed in this project are usable as such or easily adapted to test coverage against other corpora; adaptations will be required mainly to access these other corpora, which may be using different tagsets, or be presented in different formats. The Brown corpus is now dated, but is freely available and very well known; tests of coverage against other corpora is equally possible, once the format and tagset variations are made consistent.

On rules of production for generative grammars, the following contributions were made:

- **two sets of rules for a Spanish sublanguage for declarative sentences:** Spanish01 keeps close to the syntactic structures as in traditional grammar, Spanish05 is less grammar conscious, with more coarse grained constituency (chapter 8). Each of these sublanguages comprises rules and example sentences (appendix A). They were tested with toy lexicons, but wider lexicons can be fed into the application.

These example sublanguages are a design of our own, and may be used as such or as a basis for further development.

- **a set of rules for an English sublanguage for declarative sentences, named English05**, which parallels Spanish05 in its minimum requirements of grammar knowledge (chapter 8 appendix A). This set of rules was tested with EnLong3K, a general purpose lexicon compiled from the Longman Communication 3000 word list (chapter 7).
17.1. Contributions

These example sublanguages are a design of our own, and may be used as such or as a basis for further development.

- a first approach to a methodology for compiling sets of rules for sublanguages, based on a list of function words already identified, and content words taken from existing word lists, corpora, or manually added (chapter 8).

In the example sublanguages developed, the different versions of Spanish sublanguages explored several alternative sets of rules, and provided insight into their construction process (appendix A). Example sublanguages for English drew on the experience, an only two versions were preserved, with different levels of “grammar consciousness” on the part of the writer (appendix A). In both cases, the tagsets proposed were made as simple as possible, but enough for our purpose, and decently readable, since the writers must be aware of PoS tags (chapter 8). The construction of the rules for these example sublanguages, together with their corresponding sets of example sentences, provided a first approach to a methodology for compiling sets of rules. The separate treatment of function words and content words, together with the use of test sentences for each rule are perhaps the most important methodological recommendations.

17.1.3. Knowledge Representation

In the field of Knowledge Representation, the following contributions were made:

- a comparative study and evaluation of different Knowledge Representation techniques and languages in their usefulness for Education and Assessment, based on the traditionally recognized roles of Knowledge Representation first stated by Davis [Davis and Szolovits, 1993] (chapter 10).

The traditional five roles attributed to knowledge representation were considered and evaluated in their relative importance to represent knowledge for educational purposes. This evaluation was used as a guide to qualify different knowledge representation languages for use in education, namely First Order Logic, rule based systems, frame based systems, concept maps and semantic networks, ontologies, and restricted languages (chapter 10). Our discussion was led in a wide perspective, based on the recognized fact that a unique knowledge representation language cannot possibly represent all the different kinds of knowledge the human mind is able to manage. Besides this initial qualification, selection of a knowledge representation language will depend on the field of knowledge and purpose addressed. However, some knowledge representation languages seem more appropriate than others, and different degrees of evolution and availability also put some weight on the decision.

- a syntax based knowledge representation scheme, defining the transformation of syntactic structures into nodes and edges in a semantic graph (chapter 11). The syntactic structures were those of English05, the sublanguage formerly developed (chapter 8), but the transformation is based on ideas of dependency grammar which can be applied to different sublanguages. A distance measure allows for the recognition of knowledge coming from new text in the existing reference graph (chapter 11).
17. Contributions and future work

In our approach, ideas from dependency grammars were used to detect in the text the concepts and relations that later become the nodes and arcs in a graph. The sublanguage employed was English 05, described in chapter 8. The conversion of the syntactic constructs of English 05 into a graph is given in chapter 11. Some considerations on how to avoid redundancy, and more advanced ideas such as concept clustering, were also discussed. This elaboration of us is just one possible, and other schemes may be used. The one employed showed to be enough for our purposes, and since it closely follows common grammatical structures, it must be easy to master by anyone familiarized with secondary school grammar. Further evolution of English05 plus the transformation rules is possible, widening the sublanguage syntax, and introducing the appropriate relations in the transformation rules.

- a guide to develop a sublanguage and knowledge representation scheme for category and individual distinction, in the way of class and objects in Object Oriented Programming, but using everyday language expressions and extracting from them the category and individual distinction implied in those expressions (chapter 12). A proposal which adapts Object Oriented methodology to learning.

A discussion of the main aspects of a sublanguage distinguishing categories and individuals, and the conversion of syntactical constructs into an UML diagram or an equivalent graph, are given as a design guideline for a complete development. Several common language syntactical constructs were identified to define categories, define an individual as belonging to a certain category, subclassing, assignment of properties to categories or individuals, integration of parts into a whole, and establishing named associations. The syntactical constructs are given in chapter 12 as can be seen, most of these constructs are common expressions in everyday use of the language, which makes the sublanguage sound very natural. However, it results in a very elaborate knowledge construction exhibiting the main features found in an object oriented model.

- a guide to develop other sublanguage and knowledge representation schemes for sequential development and instruction manuals (chapter 12).

A discussion of the main ideas for two other types of knowledge are given as guidelines for a complete development, hinting on the solution of some specific aspects of each. A chronicle or sequence of facts require a chronological order; different language constructs to fix time references, define periods of time, and insert events in a timeline were proposed, and an example of a timeline graph were given. Instruction manuals or procedure descriptions require the handling of imperative sentences, and the ability to keep them in a certain order. Assertive statements may be drawn from our example proposals; the sequencing may be inspired in the precedent discussion of timelines, though simplified; the addition of conditionals and bifurcations in the course of events is also possible. This completes the main ideas for the design of a sublanguage and knowledge representation scheme to write answers describing how to do something.

The possibilities and drawbacks of knowledge representation in the form of concept maps, mind maps and similar rather informal languages were also discussed. The high potential of Topic Maps, a knowledge representation language regulated by a standard, makes them a promising alternative to represent knowledge for educative purposes, though this
endeavor calls for a long standing project.

17.1.4. Prototypes and deployment

Though strictly not research work, development of code reached a considerable size, is well documented, and mostly reusable, either for testing or for application development. For this reason, it is listed here as a contribution, together with considerations and suggestions on how to deploy the system.

- *data structures and functions for lexicons*, to load existing lexical resources into a common format, for self checking, comparison, and use in sublanguage construction (chapter 6, chapter 14 software tools and tests available at [González-Barbone, 2012]).

- a prototype application for writing in a sublanguage, drawing a semantic network from text, and recognizing text in the semantic network. The lexicon, rules of production, dependency data for conversion of syntactic trees into dependency trees, and graphical attributes for visual presentation are all loaded from files, which allows the application to be used for testing different sublanguage and knowledge representation schemes (chapter 14 software tools and tests available at [González-Barbone, 2012]).

- a guide of deployment, with suggestions on the necessary steps and requirements to successfully bring the proposal into the classroom, for testing in a first stage (chapter 15). The proposal is presented as a learning activity, and careful deployment, even for testing, is required.

17.2. Future work

Along this document, several sections dealt with improvements, enhancements, and possible ways of further development. A summary of these points follows:

- development of a production quality application. A production quality application is required even for reliable field testing. The facilities required by such an application were described in chapter 5.

- extensive field testing. As previously stated, this is a year’s project, including the application development, course preparation and the course itself. Considerations on the practical use of the system were analyzed in chapter 15. Documentation and tutorials are essential, and must not be overlooked.

- further research and experimentation on sublanguages. Even simple proposals as the syntax based sublanguages described in chapter 8 call for a revision, and the implementation of several extensions, some of which were described in chapter 5.

- the Wordnet offers many possibilities, which were only superficially tested in this work; much more profit can be obtained from this resource, a lasting, mature project which is accessible for free. Some tests performed and other possibilities were outlined in chapter 7.
17. Contributions and future work

- development and experimentation of the sublanguage and knowledge representation schemes discussed in chapter 12. Except for the one based on topic maps, these alternative schemes are within reach of a small team of developers and teachers.

- further research, experimentation and development on a sublanguage and knowledge representation scheme based on topic maps, described in chapter 5. This calls for a lasting project, incremental construction and testing, and professional developers. The expected benefits are proportionally greater: a whole subject can be supported on topic maps, with access to resources, an ontology for the subject area, and sharing with other similar projects. This is a long time perspective, both because of the effort demanded and for its reach in years to come.

- the same paradigm of texts written in a sublanguage and their transformation into knowledge representation instances can be carried along with other knowledge representation languages, such as frame or rules, or even logic based languages, as analyzed in chapter 10. This requires further research, but as stated, different kinds of knowledge call for different knowledge representation models.

- existing ontologies and common sense databases are resources which offer an enormous potential for their use in Education. Engagement of these resources would mean a shift of this proposal from the realm of a class or a school into a world wide scope. Ontologies and common sense databases were analyzed in chapters 9 and 10.

17.3. Conclusions

The original idea of this thesis was using some form of restricted language for the writing of texts, so as to build knowledge representation instances among which a distance could be measured. Contributions were made in assessment, sublanguage construction and use, knowledge representation scheme proposals, and proof of concept. Most of the testing code is reusable, the prototype application provided an end to end test of the system, and may be considered a first version of a production system in beta testing. Future work may be envisioned to provide much more friendly and helpful applications to ease the writing and improve visualization.

The most valuable contributions of this thesis are thought to be a practicable proposal that effectively integrates assessment into the learning process, and a satisfactory end to end testing of the system which proves its feasibility.
18. Conclusions

Abstract. Traditional methods of learning die hard. New developments almost universally acknowledged as valuable rarely creep down to the classroom as tools of everyday use. This project attempted a practicable solution, not only through feasibility, but also through acceptance. Creation of the knowledge structures from texts written in a sublanguage is a learning activity equivalent to summarizing or drawing concept maps. The structures created are those used for assessment, which gives a fair assessment system acceptable to the students. Assessment is prepared in a learning activity, marking is automatic, and feedback to students immediate. Teachers can devote assessment time to teaching, and are rid of the chore of marking. A well designed sublanguage, its correct application within its target domain, and a clear knowledge of what a sublanguage is, should avert any suspicion of limitations to the freedom of expression. No sublanguage can nor intends to substitute natural language; every sublanguage has a purpose, and will not be used otherwise. Knowledge is an extremely complex phenomenon, different knowledge representation schemes account for some of the very many modes the human mind captures the real world. A knowledge representation scheme will not be useful for every situation; different sublanguage and knowledge representation schemes may be used to advantage if carefully selected for each situation. The application of this proposal will induce students into correct writing, teach them to organize their knowledge, provide a collaborative experience, and integrate assessment into learning, drastically reducing the exam feeling stress. Based on the work of this project, further research and development in the use of sublanguages and knowledge representation for education and assessment are considered a worthwhile effort.

This chapter records the conclusions which emerged along the work done in this thesis. Since assessment is conceived as a learning activity, conclusions range in the areas of Assessment, Education, Natural Language Processing, and Knowledge Representation.

18.1. Accomplishment

The engineering methodology followed along this work was the one proposed by Adrion in 1993, and implied the following steps:

1. critical evaluation of existing proposals for the automatic marking of free-text answers;
2. the design of a solution to solve or improve on the shortcomings of existing systems;
3. testing, evaluation or validation of the proposed solution;
4. start the process all over again.
18. Conclusions

After some considerations on Assessment and eAssessment in chapter 2, existing proposals for the eAssessment of free text answers were evaluated in chapter 3. The solution proposed involved Natural Language Processing and Knowledge Representation. Sublanguages and their state of the art were discussed in chapter 4 and considerations for the construction of a sublanguage usable in Education were made in chapter 5. Existing resources for lexicons were evaluated in chapter 6, the compilation of a lexicon from commonly used word lists was described in chapter 7, and in chapter 8 some experimental syntax bases sublanguages were developed. State of the art in Knowledge Representation was discussed in chapter 9, and an characterization of the most adequate options for Education was made in chapter 10. A syntax based knowledge representation was developed in chapter 11 following the syntax based sublanguage English05 developed in chapter 8. Considerations for the compilation of alternative sublanguage and knowledge representation schemes were made in chapter 12. A prototype application for testing sublanguages and knowledge representation schemes was developed in chapter 14 and considerations for bringing the proposal to the classroom were made in chapter 15. A proof of concept evaluation of the proposal was given in chapter 16. Chapter 17 listed the contributions of this work, and pointed out future lines of research and development, thus closing the cycle in Adrión's engineering methodology.

In the following sections, the conclusions emerging from the work done are recorded.

18.2. A practicable proposal

As early as 1967, professor L G Alexander complained that “traditional methods of learning a foreign language die hard” [Alexander, 1967], referring to important observations on learning made by Dr Harold Palmer in 1921, the general agreement on the validity of their principles, and the general disagreement on how to put them into practice. Though professor Alexander was referring to foreign languages, the same can be said in many fields. Advent of the digital era has brought a profusion of new techniques, many of them outstanding by their learning value and cost effectiveness, as is the case of simulation, in which a student interacts with a computer as if she were in a cockpit or putting up a communications network. Less impressive, but equally valuable, the tools available today in an ordinary netbook allow for the use of teaching and learning techniques that, if not new, are far more practicable than ever before.

Notwithstanding the value of many learning proposals, an alarmingly small number of them ever tread into a classroom for a real course. Concept mapping may be a paradigmatic example: a huge number of articles, books, tutorials, and web pages describe them, promote their use, and state their advantages. However, a simple conversation with fellow teachers in different places will most probably show most of them know about concept maps, consider them very interesting and valuable, but do not use them in the classroom. Maybe some of those fellow teachers have tested concept maps a couple of times in their classes, but it is hard to find a teacher who systematically uses concept maps in her teaching. Of course, there may be schools which have adopted them, and many teachers in the world regularly use them, but concept maps are very far from being universally used. Curiously enough, it may be said that they have been universally accepted, though rarely used.

A guiding principle of this thesis was to produce a practicable proposal. In a first approach, this means something that could be put into practice by a small team, with limited resources, not requiring computer experts or knowledge engineers for each course, though their support may be invaluable in the development of an application. Later on, a practical proposal pre-
sented itself as something that could be accepted both by teachers and students, that both students and teachers would be ready to try, to work with, eventually to adopt. A look at the many existing proposals sleeping in the shelves or spinning around in the web should suffice to convince anyone this is a very challenging goal.

Two provisions were made towards acceptance: the integration of assessment into learning, and the shared engagement of teachers and students in both activities.

Assessment material is put together by the students while they learn; they learn through creating the structures which will be the reference for assessment. Teachers are relieved from the chore of laborious exam preparation; they only have to include in a question those parts of a reference structure they want the students to reproduce. The teachers “prepare the exam” when they guide the learning of their students, a far better use of time.

Collaborative work is universally acknowledged as a desirable feature in Education. Here the students work collaboratively in the construction of the knowledge structures, together with fellow students, under the guide of the teachers. Teachers and students share the responsibility for the quality of the knowledge structures created. The shared construction process offers the opportunities to clear up doubts, decide on arguable points, agree on scope. Once learning activities are finished, the assessment instance should bring no surprises, no stress, and not much work.

18.3. A learning activity

This project started with the aim of improving on an assessment system, but research work, design and testing of prototypes, and the experience of years spent on teaching, gradually led to the idea that a system would fall short of a practical and educationally valuable solution. What was needed was an educational proposal. Systems working on free text answers written in natural language, even if successful, do not mean any particular contribution to the learning process. Besides the difficulty of dealing with texts written in full natural languages, with all their complexities and ambiguities, these systems have to deal with a different, unexpected problem: the bad use of the language made by students. On systems which work on training sets, bad use of the language may go totally unnoticed; the system will learn whatever is in the training set, and mark new answers accordingly. On systems based on previously loaded structures, or correct answers written by teachers, or textbooks, the system will expect answers written correctly in the language, and mark as wrong what they cannot recognize, not making a difference between bad writing and wrong knowledge.

The former considerations lead to some crucial questions:

- **How bad is the use of language in students answers?** Educational level, social origin of the students, subject areas on which they have been trained, the scale to value good use of the language, are only some factors to consider for an answer. Being very conservative, it can be said that use of language in written texts is, in general, below the expected standards for each educational level.

- **Must an assessment system concern with use of language?** The goal of an assessment system is to determine the level of achievement in some knowledge or skill, not to teach. A system may suffer in its accuracy owing to bad use of the language, specially if working on training sets, but in a strict definition of roles, it is not the responsibility of an assessment system to ensure correct use of the language.
18. Conclusions

- What is “good use of the language”? Criteria may differ wildly among institutions, communities, and individuals; expectations of an agreement are hopeless. Adoption of a standard is a way out of stagnating discussions. Some of the simplified languages discussed in chapter 4 offer a lexicon, grammar rules, reading material and tutorials, which in all warrant good use of the language.

One of the principles under which this thesis was developed was to come out with a proposal actually useful to teaching and learning, not only to assessment (chapter 1). In this view, good use of the language is a concern. The project evolved into an educational proposal, in the form of a learning activity both helpful to learning and to assessment, which effectively integrates both learning and assessment, and in which assessment comes as a natural consequence of the learning activities formerly developed.

18.4. The language issue

The idea of restricting the use of the language is, a priori, very questionable, to the point of being considered unacceptable in some contexts. Two very important aspects may turn black to white: purpose and quality.

No sublanguage, controlled language, or any restricted language can be expected to replace natural language in all circumstances. Sublanguages emerge naturally in a community for the needs of its members; controlled languages are designed for a purpose, in a formal specification; sublanguages in this project are conceived as designed for a purpose but not necessarily adhering to an acknowledged standard, though this may eventually happen, and be advantageous.

Writing in a sublanguage does not mean bad writing. On the contrary, it may effectively contribute towards good writing. A sublanguage is a subset of natural language; sentences written in a sublanguage form a subset of all sentences written in the natural language; ideally, sentences written in a sublanguage would be a subset of all correct sentences written in natural language. In this context, correct does not mean only lexical and syntactic compliance, but also meeting the general speaker’s concept of correct use of the language, for instance as it is taught in schools or as a second language.

A sometimes invoked evil is a limitation in freedom of speech. This is nonsense, at least for two reasons. First, some sublanguages are addressed to human communication, they pride themselves in clearness, simplicity and expressiveness, and only establish formal limitations towards good use of the language. Second, a sublanguage has a purpose, and hence a context of application, within a community which accepts the sublanguage as a convenience, because their members see benefit in its adoption. As said, no sublanguage intends to replace natural language. No one would think of imposing a sublanguage in literature, philosophy or politics, nor even in chat rooms, social networks, emails or cell phone short messages.

Making the students conscious of lexicon, grammar, the ambiguities and complexities of the language, may also be seen as a hindrance in an always overloaded school programme. Here the dividing line may be drawn by the effort requested from the students: a sublanguage proposal for the purpose of assessment as conceived in this thesis should not demand a significant extra effort beyond that required by Language courses in secondary school teaching. The sublanguages proposed in chapter 8 and most of those analyzed in chapter 12 are considered to be within reach of mid secondary school students with only a few hours training and practice.
This project intended to improve on the computer assisted assessment of free text answer by imposing mild restrictions on the use of natural language so that knowledge extraction would be possible with no ambiguities. At first this was seen just as a way to bring a difficult problem into tractable terms, a necessary inconvenience. However, the educational possibilities of a well designed sublanguage supported in a friendly application immediately turned up. The use of sublanguages in secondary school teaching and further, in secondary language teaching, and in the construction of small ontologies, deserve more attention than they have received so far.

18.5. Knowledge acquisition

One of the aspects of learning is knowledge acquisition. The proposal in this thesis concerns with a particular type of knowledge, that which can be expressed in declarative sentences, such as events, facts, abstract concepts or imaginary deeds. This “declarative knowledge” is just a part of all knowledge a student is expected to acquire in most courses. Though limited, declarative knowledge is the most tedious to evaluate; it generally asks the student to recall what she has learnt, in different forms. Closed questions are targeted at this kind of knowledge. The use of free text answers, even short answers consisting of at least a complete sentence, compel the student to produce something more elaborate than what closed questions demand. Though a carefully contrived multiple choice question may leave a good student pondering, her answer will be a tick in a box. A text answer, even if a single sentence, is a more creative type of response, complete in itself. A sentence says something by itself, is able to stand alone, is a little creation, and is more rewarding. A few sentences in a text answer can tell quite a lot of what a student knows, and is able to write.

The former considerations were not made to slight or diminish the value of closed questions, but to remark the value of students producing something more elaborate than choosing or matching. The use of free text answers in assessment, even in a small scale, with short answers, can be seen as something a little more demanding, a little more informative test of what knowledge the student has, and whether she can tell of it.

Texts and graphs are learning devices, they are used much in the same way as students’ notes, summaries or outlines of the main concepts and relations in the unit of study. For teachers encouraging the use of concept maps, this proposal just adds the use of the sublanguage to their creation. The construction of the knowledge structures is very similar to the construction of a concept map, though within the restrictions imposed by the sublanguage and knowledge representation scheme of choice. As discussed in chapter 12, this may be beneficial, contributing some more formal representation rules into the rather loose concept maps conventions.

As discussed in chapter 12, different language and knowledge representation schemes are possible, ones better suited than others for models of knowledge which stress different kinds of relations among concepts. These different ways of structuring knowledge reflect some of the many ways the human mind apprehends the world. No single representation scheme will be best in all cases, no single scheme can account for the many ways the human mind thinks. Subject areas, the objectives of a course, the instruction level, will render some representation schemes more effective than others. An education proposal should consider these different schemes, and choose the best for each circumstance.

Other knowledge representation languages, such as frames or rules, may have a role to
18. Conclusions

play in education and assessment. Adequacy of different knowledge representation languages for use in Education was discussed in chapter 10. For rules and frames, a sublanguage and knowledge representation scheme may be developed along the lines sketched in chapter 5 on the construction of a sublanguage, and in chapter 12 where several example schemes were analyzed.

A lesson learned in knowledge representation is to remain open to alternatives, to study the different languages, consider their possibilities, and evaluate their adequacy for the purpose in hand. It is not possible to advance steadily in knowledge representation, and assessment based on knowledge representation, without being aware of the essential relation between the kind of knowledge and the chosen representation scheme. Neither students nor teachers need become cognitive specialists; a general knowledge will be enough, some consciousness of the language and knowledge representation devices required by the different types of knowledge.

Sublanguages are being developed for ontology creation; they may eventually evolve to be used in education, or a simplified version of them. Creating topic maps from text is another powerful, far reaching alternative. Though a more ambitious project, all the material of a course may be knitted into a topic maps network; resources may be integrated into the knowledge structures, adding findability, sharing and merging, thus ripping the benefits of sticking to a standard.

18.6. Summing up

Along this work, the idea of using a sublanguage and knowledge representation for the marking of free text answers proved to be feasible, at least to the extent a research work and a single man’s effort can reach. The tools and prototype developed give reasons to assert that a production system can be put together, and a test in the classroom carried out, in a year’s project by a small team of teachers and developers.

The proposal is presented as a learning activity which involves students and teachers in collaborative work. The students build the knowledge structures under the guide of the teachers, an activity equivalent to summarizing or drawing concept maps. The assessment instance comes as just another session of this learning activity. The reference knowledge structures are built by the students, they share in the preparation of the assessment instance; this results in a fair system, and exam stress is greatly reduced. There is almost no preparation time for the teachers, marking is automatic, and feedback to students immediate.

The use of a well designed sublanguage contribute to improve the use of natural language and the development of communication skills. Drawing graphs by writing readable text offers a kind of challenge similar to a computer game, which makes it more attractive than other equivalent learning activities.

The combination of sublanguage and knowledge representation emerges as a promising technique for educational purposes, deserving far more attention than it has received so far.

18.7. A last word

The application of this proposal will induce students into correct writing, teach them to organize their knowledge, provide a collaborative experience, and integrate assessment into learning, drastically reducing the exam feeling stress.
Based on the work of this project, further research and development in the use of sublanguages and knowledge representation for education and assessment are considered a worthwhile effort.
Part VII.

EAssessment de respuesta a texto libre basada en sublenguajes de dominio específico y representación de conocimiento
E-Assessment de respuesta a texto libre basada en sublenguajes de dominio específico y representación de conocimiento

E-Assessment of Free-Text Answers Based on Domain Specific Sublanguages and Knowledge Representation

Esta parte incluye la traducción al español de los siguientes extractos:

- Portada y Resumen
- Definición del proyecto
- La solución propuesta
- Contribuciones
- Conclusiones

This part includes the Spanish translation of the following excerpts:

- Cover and Abstract
- Statement of the project
- Overview of the proposed solution
- Contributions
- Conclusions
19. Portada y Resumen

E-Assessment de respuesta a texto libre basada en sublenguajes de dominio específico y representación de conocimiento

Grado de Doctor por la Universidad de Vigo con Mención Internacional

Autor:
Víctor González Barbone

Supervisor de Tesis:
Martín Llamas Nistal

Universidad de Vigo

Universidad de Vigo
Departamento de Ingeniería Telemática
Programa de Doctorado en Ingeniería Telemática

2013
Oigo y olvido.
Veo y recuerdo.
Hago y entiendo.

Confucio (551 AC – 479 AC)

Vita brevis,
ars longa,
occasio praeceps,
experimentum periculosum,
judicium difficile.

Hipócrates (c. 460 AC – c. 370 AC)
Resumen

Las respuestas a texto libre pueden ser evaluadas desde dos amplios puntos de vista: corrección del contenido de información y calidad de escritura. La evaluación de contenido resulta finalmente en la comparación de las respuestas de los estudiantes contra una respuesta o un modelo de referencia. La referencia puede obtenerse de respuestas de los profesores, respuestas de los estudiantes corregidas manualmente, un corpus de textos de referencia, o estructuras tales como una red semántica. Las técnicas usadas para la evaluación de texto libre incluyen el análisis de palabras clave, análisis semántico latente (LSA, Latent Semantic Analysis), rasgos lingüísticos superficiales, categorización de textos, extracción de información y agrupamiento (clustering). Existen muchos sistemas diferentes, aplicados a diferentes áreas, con diferentes métricas de evaluación, lo cual hace muy difícil compararlos. La principal debilidad de la mayoría de estos sistemas es la falta de un corpus suficientemente grande de respuestas corregidas o algún otro tipo de referencia confiable. La dispersión entre los profesores ha probado también ser un factor de distorsión. En este trabajo, el modelo de referencia es una red semántica inferida del texto escrito en un sublenguaje fácil de usar, construido por los propios estudiantes en una actividad de aprendizaje. Esto permite esperar una mejora en las carencias de los sistemas existentes, puesto que no se requiere un conjunto de entrenamiento, la referencia ha sido acordada entre estudiantes y profesores, los textos han sido escritos de manera predecible, los términos se usan uniformemente, y el sistema de evaluación funciona de la misma forma en que la referencia fue construida. Además, la propuesta de esta tesis se concibe principalmente como una actividad de aprendizaje, soportada por una herramienta de aprendizaje que puede ser usada en la evaluación. Se discute la construcción de un sublenguaje utilizable por estudiantes de enseñanza secundaria, y se desarrollan algunos sublenguajes experimentales. Las respuestas escritas en el sublenguaje pueden ser convertidas en una red semántica, la cual puede compararse con una red semántica de referencia. La red semántica de referencia ha sido confeccionada gradualmente por los propios estudiantes, como una actividad del proceso de aprendizaje, en forma similar a la toma de apuntes, la confección de un resumen o la construcción de un mapa conceptual. Un prototipo para prueba de concepto incluye facilidades para guiar la escritura en un sublenguaje, transformar el texto resultante en una red semántica, y colorear la red semántica de referencia para mostrar qué partes de la respuesta han sido reconocidas, junto con una calificación. La evaluación se prepara como una actividad de aprendizaje, la calificación es automática, y la retroalimentación a los estudiantes es inmediata. Se identifican varios recursos léxicos, se prueban y analizan y usan para construir un léxico de propósito general apto para diferentes
dominios. Varias versiones de gramáticas generativas se proponen y analizan. Se propone una metodología para el desarrollo de un sublenguaje. Se estudia la Representación de Conocimiento en su aplicabilidad a la educación y el aprendizaje, identificando las técnicas más promisorias. Se analizan varios esquemas posibles de sublenguaje y representación de conocimiento, cada uno dirigido a un propósito diferente. Se presenta un ejemplo completo de sublenguaje basado en sintaxis y red semántica como modelo de representación de conocimiento, y se lo prueba usando la aplicación prototipo desarrollada. Las contribuciones originales de esta tesis resultan de la combinación de la evaluación, los sublenguajes y los modelos de representación de conocimiento en una propuesta practicable que efectivamente integra la evaluación al proceso de aprendizaje, y un ensayo satisfactorio del sistema extremo a extremo que prueba su factibilidad.

Palabras clave: eLearning, eAssessment, respuestas a texto libre, Representación de Conocimiento, Sublenguajes, Educación, Evaluación.
20. Definición del Proyecto

Resumen. Esta tesis propone un sistema factible para la calificación asistida por ordenador de respuestas a texto libre guiando la escritura de las respuestas dentro de un subconjunto determinístico del lenguaje, y representando el conocimiento en estructuras de datos entre las cuales pueda establecerse una comparación. La solución debe ser suficientemente simple como para estar al alcance de un pequeño equipo de instructores. El sublenguaje debe ser suficientemente natural como para ser considerado escritura correcta. El esquema de representación de conocimiento debe ser suficientemente intuitivo como para ser usado tanto para la evaluación como para el aprendizaje. Los estudiantes deben poder crear las instancias de representación de conocimiento de referencia como una de las actividades de aprendizaje, quedando así involucrados en la preparación de la instancia de evaluación. Las propuestas existentes se basan normalmente en el tratamiento estadístico de las respuestas, lo cual requiere una cierta cantidad de respuestas corregidas manualmente o reservadas para entrenamiento. Estas aproximaciones exhiben una dispersión inherente que amplía el intervalo de aceptación; respuestas correctas no previstas pueden pasar inadvertidas o calificarse como erróneas; la incertidumbre estadística o el diferente tratamiento de las respuestas puede ser impugnado por los estudiantes. Desarrollos existentes en léxicos y wordnets, herramientas disponibles para el Procesamiento de Lenguaje Natural, y varios esquemas factibles para la Representación de Conocimiento dan una idea de la factibilidad del proyecto. La mejora en las habilidades de escritura y el compromiso de los estudiantes en las actividades de representación de conocimiento se agregan al interés principal de la calificación asistida por ordenador. Las características de la solución se formulan como principios guía. La metodología adoptada es el método de ingeniería propuesto por Adrien en 1993; se sigue con cierta flexibilidad un ciclo tradicional de desarrollo de requerimientos, diseño, implementación y prueba para el desarrollo de prototipos. Este documento se organiza en siete partes: la Introducción explica la motivación y propósitos, describe la solución propuesta y da una visión general de la tesis y sus contribuciones; en Evaluación se trata el estado del arte en la evaluación electrónica (eAssessment) de respuestas a texto libre; en Lenguaje se trata el estado del arte, recursos existentes, y la construcción de un sublenguaje; en Conocimiento se revisa el estado del arte en la Representación de Conocimiento, se seleccionan algunos modelos adecuados para Educación, se desarrolla un esquema de lenguaje y representación de conocimiento basado en la sintaxis, y se analizan otros esquemas de representación de conocimiento. En Prototipos se describe el código de pruebas desarrollado y se realizan algunas consideraciones sobre el despliegue; una parte final contiene la evaluación, trabajo futuro y conclusiones; los Apéndices incluyen ejemplos de sublenguajes, documentación y material más técnico.
20. Definición del Proyecto

La calificación automática de respuestas a texto libre o composiciones (“essays”) es un problema de larga data. Se han intentado muchas soluciones, y se han obtenido resultados promisorios, pero a la fecha no se ha encontrado una solución satisfactoria. Es un problema difícil, donde las complejidades del lenguaje natural se agregan a las complejidades de la extracción y representación de conocimiento. Esta tesis propone una solución basada en la guía de la escritura de las respuestas para usar solamente un subconjunto del lenguaje natural, y representar el conocimiento en estructuras de datos entre las cuales puedan establecerse comparaciones. Tan- to el subconjunto del lenguaje natural como las estructuras de datos para la representación de conocimiento deben ser suficientemente intuitivas para ser de valor tanto para la calificación como para el aprendizaje, y por esto igualmente útiles para instructores y estudiantes. Este capítulo discute la motivación del proyecto, los fundamentos para la perspectiva adoptada, y por qué se considera valioso seguir esta línea de investigación.

20.1. De qué trata esta tesis

Posiblemente la forma más antigua e históricamente más popular de determinar lo que sabe un aprendiz acerca de un tema sea la palabra, escrita o hablada. En las últimas décadas, la universalización de la educación y la extensión del período de aprendizaje a lo largo de la vida ha resultado, en todos los niveles, en un conjunto de profesores e instructores siempre insuficiente y sobresaturado de trabajo. La tarea de evaluación del conocimiento ha sido siempre pesada, tediosa y demandante en tiempo. La creciente disponibilidad de ordenadores y conectividad ha permitido implementar diversas formas de “preguntas cerradas”, hoy día comúnmente disponibles en cualquier plataforma de aprendizaje moderna. La razón principal para la adopción de estas formas de evaluación es que pueden ser corregidas automáticamente con muy poca o ninguna incertidumbre en la calificación. Sin embargo, aún el más cuidadosamente diseñado conjunto de preguntas cerradas resulta pobre frente a las posibilidades ofrecidas por una sola pregunta abierta. La pregunta abierta dirige la atención del estudiante hacia algunos aspectos, hechos o situaciones propias de un tema, y luego deja solo al estudiante para que exponga su conocimiento en texto libre, escribiendo a su antojo. La respuesta puede ser una oración breve o una larga composición.

Una aproximación simple a la calificación automática de respuestas a texto libre procesaría el texto de las respuestas tratando de extraer los conceptos y relaciones allí registrados, y verificaría si esos conceptos y relaciones son correctos, i.e. si están también registrados en lo que se considera una referencia autoritativa. Así, el problema deviene en comprender el texto escrito en lenguaje natural. La comprensión del lenguaje natural es tema del Procesamiento de Lenguaje Natural (PLN) y de la Inteligencia Artificial (IA). El problema general de comprensión del lenguaje es considerado IA-completo, que es una manera informal de decir que su dificultad es equivalente a construir un ordenador tan inteligente como un ser humano.

La calificación de respuestas a texto libre no es tan amplia, sin embargo el conocimiento está limitado a un cierto dominio, y el discurso acontece dentro de ese dominio; muchos aspectos del uso del lenguaje son predecibles, al menos hasta cierto punto. No todos los tipos de pregunta son adecuados para la calificación automática; las preguntas relativas a información fáctica donde no hay lugar para opiniones distintas parecen ser las más tratables. Restringiendo el tipo de preguntas, la extensión de las respuestas, y en alguna medida el estilo de la escritura, se han probado varias estrategias, y se han obtenido resultados promisorios.

La aproximación más común implica la aplicación de PLN y estadística a las respuestas
y al mismo tiempo a algún material de referencia, que puede ser un conjunto de respuestas calificadas, un conjunto de respuestas correctas escritas por los instructores, una colección de textos de referencia, o alguna clase de estructura de datos. Todas estas propuestas sufren de alguno de los siguientes inconvenientes: una parte significativa de las respuestas de los estudiantes debe ser corregida a mano para actuar como referencia; las respuestas escritas por los profesores no siempre están de acuerdo entre sí, ampliando indebidamente por esta vía el intervalo de aceptación; las respuestas corregidas manualmente también sufren este tipo de dispersión, ampliando la aceptación; una respuesta correcta no prevista pasa desapercibida, y puede ser calificada como errónea; la mayoría de los sistemas no son suficientemente adaptables como para asimilar una nueva respuesta correcta o corregir un error sin comenzar de nuevo desde el principio; los resultados estadísticos, aún con un nivel de confianza alto como el 95 %, pueden no resultar aceptables para instancias educativas, y ser impugnados por los estudiantes.

Aunque se han intentado otras aproximaciones, en las propuestas existentes pueden reconocerse uniformemente tres estadios: el procesamiento de lenguaje de las respuestas para detectar el conocimiento, el registro o reconocimiento de este conocimiento en alguna estructura de información o corpus de texto, y una comparación del conocimiento derivado de las respuestas con el conocimiento derivado del material de referencia considerado correcto. La comparación resulta en la calificación, generalmente un número, asignado a la respuesta.

El propósito de esta tesis sigue este derrotero general, pero intenta guiar la escritura de las respuestas de modo que el texto resultante no sea ambiguo dentro del dominio de conocimiento y, consecuentemente, más tratable. El conocimiento puede extraerse de estos textos en forma predecible, y almacenarse en alguna forma de representación de conocimiento. Una instancia de representación de conocimiento derivada de la respuesta se puede comparar con una instancia de representación de conocimiento del mismo tipo tomada como referencia de la respuesta correcta. Una medida de distancia entre estas dos instancias resulta en la calificación otorgada a la respuesta.

En esta tesis, la evaluación puramente automática de las respuestas a texto libre se considera inadecuada para la mayoría de los propósitos educativos. En su lugar, se propone una evaluación asistida. La instancia de representación de conocimiento tomada como referencia se diseñaba para ser dinámicamente ajustada: cuando en una respuesta aparece una estructura no presente en la referencia, el instructor puede incluir esta nueva estructura en la referencia, y así como estructura correcta o incorrecta. La próxima vez que esta estructura aparezca en una respuesta, será reconocida en la referencia, y calificada como corresponda. Un sistema que implemente este esquema puede “ser enseñado” por el instructor; la instancia de representación de conocimiento tomada como referencia puede crearse antes de comenzar la calificación, y ajustarse a medida que se realiza la calificación.

El propósito de esta tesis es el estudio y diseño de un sistema factible de evaluación asistida por ordenador para respuestas a texto libre mediante la guía en el uso del lenguaje en las respuestas, de modo de habilitar una extracción de conocimiento determinística y su comparación contra una instancia de representación de conocimiento de referencia.

Las dos áreas principales involucradas en este proyecto son el Procesamiento de Lenguaje Natural (PLN) y la Representación de Conocimiento (RC). Ambas áreas son vastas y complejas, y ambas proveen variedad de estrategias y herramientas que pueden ser usadas con éxito en el diseño e implementación de la solución propuesta. Aunque la prueba de concepto y el prototipado requieren optar entre estas posibles estrategias y herramientas, esta tesis intenta examinar las más promisorias entre ellas, tratando de evaluar su efectividad y el esfuerzo requerido para su desarrollo. En otras palabras, aunque esta tesis se limita a la prueba
de concepto y el prototipado, aporta alguna visión sobre el diseño y desarrollo de un futuro sistema de producción, a través de la evaluación de estrategias y herramientas alternativas. Consecuentemente, un sistema de producción puede, y seguramente será, implementado de maneras muy diferentes.

Otro principio a lo largo del cual esta tesis se desarrolla es el de mantener la usabilidad del sistema propuesto al alcance de instructores con habilidades modestas en tecnologías de la información. En el caso general, el diseño de un sublenguaje requiere un lingüista; el diseño de una representación de conocimiento requiere un ingeniero de conocimiento. Esta aproximación profesionalizada resultará seguramente en un sistema mucho más poderoso y extensamente utilizable, pero requiere al menos un consenso institucional, la contribución de varios expertos, y apoyo financiero para su desarrollo. Todo esto puede resultar muy difícil de obtener, más aún si no hay evidencia clara de la usabilidad del sistema. El sistema propuesto debe ser suficientemente simple como para permitir que un pequeño grupo de instructores no sólo lo usen, sino que entiendan lo que están haciendo y los beneficios subsidiarios de la escritura determinística.

La evaluación puede ser más completamente integrada en el proceso de aprendizaje, y resultar menos dolorosa para los estudiantes si se les da alguna participación en el proceso de su preparación. El sublenguaje y el esquema de representación de conocimiento pueden ser usados como herramientas de enseñanza, y los estudiantes pueden tomar parte activa en la construcción de las estructuras de conocimiento que serán utilizadas como referencia para la evaluación de sus respuestas. Esto está en línea con el uso de mapas conceptuales, mapas mentales, tormenta de ideas o esquemas similares para la adquisición y organización del conocimiento como estrategias de aprendizaje. Las mismas estructuras creadas para el aprendizaje pueden ser usadas como estructuras de referencia para la evaluación. Esta tesis propone una aproximación colaborativa involucrando a los estudiantes y profesores en la construcción de las estructuras de referencia que luego serán usadas en la evaluación, pero que han sido creadas primariamente para el aprendizaje en las actividades normales de un curso.

El resto de este capítulo desarrolla la motivación de esta tesis, formaliza su propósito, define el alcance, explica la aproximación metodológica, establece los principios según los cuales esta tesis fue desarrollada, y describe la organización de este documento.

20.2. Propósito

El propósito de esta tesis puede ser formulado sucintamente de esta forma:

Esta tesis intenta desarrollar una propuesta y construir una aplicación de software para asistir al docente en la evaluación de respuestas a texto libre, basada en el uso de un sublenguaje de dominio específico para escribir las respuestas, la extracción de una instancia de representación de conocimiento desde el texto de la respuesta, y una comparación de esta instancia de representación de conocimiento contra una instancia de representación de conocimiento de la respuesta correcta. Una medida de distancia definida entre las dos instancias de representaciones de conocimiento del mismo tipo determinan la calificación dada a la respuesta. Todo el esquema debe ser suficientemente simple como para ser comprendido, manejado y aplicado por docentes con poco más de las habilidades corrientes en el uso de las tecnologías de la información. Esta tesis también intenta explorar el uso de herramientas y proporcionar recomendaciones para construir un sublenguaje
20.3. Principios guía

de dominio específico, seleccionar y manejar una representación de conocimiento, definir una medida de distancia, y poner en marcha el sistema en el salón de clase.

20.3. Principios guía

El trabajo en esta tesis ha sido guiado por algunos principios en cuanto a la solución. Estos principios actúan como requerimientos no funcionales o restricciones en un proceso de desarrollo de software. La solución propuesta debe:

- ser fácil de entender y fácil de manejar.
- ser realmente útil para la enseñanza y el aprendizaje, no solamente para la evaluación.
- prestarse al aprendizaje colaborativo, comprometiendo al estudiante en la escritura de texto y la construcción de grafos a medida que se familiariza con el tema en estudio.
- ser practicable, estar al alcance de un pequeño grupo de instructores.
- diseñada con una perspectiva de ingeniería: su propósito último es proveer los fundamentos para la construcción de una aplicación de evaluación asistida.
- seguir una visión empírica de diseño, sin entrar en sutilezas ni discusiones teóricas de la educación, el lenguaje y el conocimiento. La visión final es una aplicación utilizable en el salón de clase, manejada por los profesores, y compartida con los alumnos.
- limitada al conocimiento fáctico, expresado como oraciones aseverativas o imperativas.

Este trabajo ha sido realizado en la convicción de que el conocimiento debe ser siempre de beneficio público; todo el software, recomendaciones y trabajo de esta tesis están destinados a ser usados y a estar disponibles para cualquiera.
21. La solución propuesta

**Resumen.** La factibilidad de este proyecto se basa en la posibilidad de compilar un sublenguaje adecuado, convertirlo en una instancia de representación de conocimiento, y comparar esta instancia contra una referencia. Como prueba de concepto, se desarrolló un sublenguaje llamado English 05 y un esquema de representación de conocimiento basado en sintaxis que produce un grafo semántico, conjuntamente con una aplicación prototipo en la cual se puede cargar y mostrar el funcionamiento del sublenguaje y el esquema de representación de conocimiento. Una visión general de los componentes del sistema, seguida de un ejemplo de uso, dan una idea de cómo se ve y trabaja. La aplicación lee desde ficheros el léxico, las reglas de producción y las reglas de conversión para transformar el texto en un grafo semántico. Esto permite a la aplicación ser usada con diferentes sublenguajes y esquemas de representación de conocimiento. El código del prototipo incluye rutinas para compilar un léxico, un editor para licenciar oraciones contra un sublenguaje, funciones para transformar texto en un grafo semántico, y una medida de distancia para comparar dos grafos semánticos. Pueden concebirse diferentes sublenguajes y esquemas de representación de conocimiento para diferentes propósitos, cada uno con sus pros y contras. Para una prueba en el salón de clase, los profesores deciden el sublenguaje y la representación de conocimiento a emplear, definen unidades de enseñanza con un único objetivo de aprendizaje cada una tal como se hace en los objetos de aprendizaje, entrenan a los estudiantes en el uso del sistema, promueven la creación colaborativa de las instancias de estructuras de conocimiento para las unidades de aprendizaje, preparan las preguntas basados en el conocimiento de las estructuras creadas, y los someten a los estudiantes para la evaluación. Los estudiantes responden las preguntas sobre las unidades de aprendizaje que han estudiado, registradas en las instancias de representación de conocimiento, escribiendo textos de la misma forma y en la misma aplicación que usaron durante el estudio. La propuesta queda así concebida no solo como un mero sistema de evaluación, sino como una actividad de aprendizaje completa, de la cual la evaluación es solamente el paso final.

Este capítulo describe la solución propuesta, como sistema y como metodología. Luego de recorrer los supuestos sobre los cuales la idea fue concebida, se evalúa la factibilidad del proyecto. Se describe brevemente la aplicación prototipo desarrollada para prueba de concepto, junto con algunas sugerencias sobre cómo estas ideas pueden ser llevadas al salón de clase. Se discuten dificultades potenciales y beneficios esperados. La mayoría de los aspectos mencionados aquí se desarrolla en los capítulos siguientes.
21. La solución propuesta

21.1. Factibilidad

Este proyecto propone un sistema para la evaluación asistida de respuestas a texto libre basada en el uso de un lenguaje restringido para escribir las respuestas, la extracción predecible de conocimiento desde las respuestas hacia una estructura de conocimiento, y la comparación de las estructuras de conocimiento para obtener una calificación. Fue concebida en los siguientes supuestos:

- es posible compilar un sublenguaje suficientemente expresivo para las necesidades de aprendizaje de un dominio específico, suficientemente similar al lenguaje natural corriente para ser fácilmente aprendido, y tan legible como los textos en lenguaje natural.

- es posible adoptar o adaptar un lenguaje de representación de conocimiento hacia una estructura de conocimiento suficientemente intuitiva para ser comprendida casi inmediatamente, tal como un mapa conceptual o una red semántica.

- es posible definir una transformación desde el texto escrito en el sublenguaje hacia una instancia de representación de conocimiento del tipo elegido, sin ambigüedad. La instancia de representación de conocimiento debe ser la misma o equivalente si las oraciones del texto aparecen en diferente orden, o las oraciones tienen diferente sintaxis, siempre que el conocimiento transmitido por estas oraciones sea el mismo.

- es posible comparar y determinar una medida de distancia entre dos instancias diferentes de representación de conocimiento. Cuando la medida se realiza entre una instancia correspondiente a la respuesta y una instancia tomada como referencia de la respuesta correcta, esta distancia es la calificación que puede darse a la respuesta.

A lo largo del proyecto, se desarrolló un lenguaje llamado English 05, se adoptó la red semántica como lenguaje de representación de conocimiento, se infirió una conversión de texto a grafo a partir de las estructuras sintácticas del sublenguaje, se asignó pesos a los nodos y arcos en el grafo, y se definió una medida de distancia sumando los pesos de nodos y arcos del grafo respuesta que también aparecen en el grafo referencia.

El sublenguaje basado en sintaxis propuesto es solo un posible desarrollo de un esquema de sublenguaje y representación de conocimiento. Fueron analizadas otras combinaciones de sublenguaje y representación de conocimiento: modelos de categoría e individuo (orientación a objetos), modelos secuenciales como una narración o historia, un modelo imperativo para instrucciones o manuales, y representaciones menos formales como mapas conceptuales y mapas mentales. Todos estos modelos pueden ser implementados con un esfuerzo moderado por un grupo de instructores, en tanto se les provea de las herramientas adecuadas. El uso de o la exportación a Topic Maps, un estándar ISO/IEC para la representación de conocimiento, fue también analizada. Aunque más complejos, los Topic Maps ofrecen gran potencial para proyectos educativos de largo alcance.

A lo largo del proyecto se desarrollaron algunas herramientas para ayudar en la compilación de un sublenguaje, y un prototipo de aplicación para licenciar texto y construir los grafos de representación de conocimiento. Estas herramientas y prototipos proveyeron facilidades suficientes para demostrar la propuesta, desde la escritura de texto hasta la construcción de la representación de conocimiento, la comparación de la respuesta contra la referencia, y el cálculo de la distancia. Una aplicación completa apta para la prueba en clase puede basarse en este código prototipo. Un pequeño grupo de dos o tres desarrolladores a medio tiempo
21.2. Visión del Sistema

Las herramientas y la aplicación prototipo fueron reunidas en una jerarquía de módulos Python bajo el nombre KLEAR, que significa “Knowledge and Language for Education, Assessment and Research” (Conocimiento y Lenguaje para Educación, Evaluación e Investigación).

La figura muestra un esquema del sistema. Los bloques superiores en color blanco son documentos de usuario, principalmente leídos, creados y vistos por los estudiantes. Los módulos del prototipo del proyecto KLEAR son cajas redondeadas en color amarillo. Los documentos de definición, que regulan el funcionamiento del sistema y se crean de acuerdo a reglas formales, se muestran en azul claro. Los objetos en verde claro son objetos de Python de diferentes clases definidos en el código del proyecto KLEAR.

Un léxico básico obtenido de listas de palabras, y un léxico específico de términos usados en el área temática pueden compilarse mediante las rutinas de léxico de KLEAR construyendo un objeto de léxico de KLEAR, el cual contiene esencialmente diccionarios de formas de palabra (“wordforms”), lemas (“headwords”), categorías léxicas (“part of speech”), y posiblemente definiciones. El objeto léxico puede, en cualquier momento, complementarse con palabras adicionales provenientes de libros de texto y documentos usados por los estudiantes. El documento de reglas de producción define las estructuras sintácticas aceptadas por el sublenguaje.

El documento de reglas de producción mantiene estrecha relación con otro documento, etiquetado “Text to Sm Objects” (texto a objeto semántico), que define cómo el texto se convertirá en los objetos KLEAR (Sm Objects) que soportan la estructura de conocimiento. Las reglas de producción dicen cómo el texto deberá escribirse; el documento de Text to Sm Object determina cómo el texto se transformará en los objetos que constituyen la red semántica, en la forma en que el proyecto KLEAR la implementa. Las reglas de producción y el objeto léxico definen el sublenguaje; el editor KLEAR acepta el texto de acuerdo al léxico y las reglas con que ha sido cargado.

Los textos de referencia para crear la red semántica, y las respuestas a texto libre a preguntas, se escriben ambas con la ayuda del editor. La salida del editor es un conjunto de árboles sintácticos correspondientes al texto ingresado. Los árboles sintácticos son elaborados por el módulo grafo de KLEAR en objetos SmGraph, la estructura de datos que el proyecto KLEAR usa para soportar las redes semánticas. Los objetos que representan las redes semánticas se construyen de acuerdo a las reglas establecidas en el documento Text to Sm Objects.

Un documento formal adicional, también cargado en el módulo grafo de KLEAR, etiquetado “Sm Objects to graphics”, define cómo se presentarán los objetos en formato gráfico. El módulo grafo puede mostrar la estructura del grafo en formato gráfico como imagen PNG.

Los objetos etiquetados “ref” y “ans” corresponden a los objetos SmGraph del grafo referencia y del grafo respuesta. El módulo KLEAR de distancia calcula la distancia entre los grafos de respuesta y de referencia, mostrando la calificación máxima en el objeto referencia, la calificación obtenida por el objeto respuesta, y el porcentaje de la respuesta en el grafo referencia.
21. La solución propuesta

Figura 21.2.1.: El proyecto KLEAR: esquema del sistema.

Los objetos “ref” y “ans” se pueden representar en formato gráfico, el gráfico referencia en gris y el gráfico respuesta coloreado sobre él.

La figura 21.2.2 muestra un gráfico referencia creado por estas oraciones:

Columbus was an explorer. Columbus was a navigator. Columbus was a colonizer. Columbus was born in Genoa. Columbus made [several voyages [across the ocean]].

Las siguientes oraciones de respuesta fueron reconocidas en el gráfico:

Columbus made [several voyages [across the ocean]]. Columbus was a colonizer. Columbus was a navigator.

La calificación total en el gráfico referencia fue 45 puntos, la respuesta obtuvo 30 puntos; la respuesta logró el 66.67 % del total.

21.3. Funcionamiento

El proyecto KLEAR fue concebido como una propuesta educativa, no limitada a la evaluación. La instancia de evaluación llega como un paso final del proceso de aprendizaje en el cual el sistema ha sido usado como una de las actividades que los estudiantes desarrollan a lo largo del curso, concretamente la construcción de instancias de representación de conocimiento de las diferentes unidades de aprendizaje que se han enseñado. Los siguientes pasos describen brevemente cómo fue diseñado el funcionamiento del sistema. El capítulo sobre despliegue del sistema en el documento original en inglés da sugerencias más detalladas sobre los requerimientos y procedimientos para realizar la actividad en clase.
21.3. Funcionamiento

1. **Preparación.** Los instructores acuerdan un esquema de sublenguaje y representación de conocimiento. El léxico del sublenguaje puede estar basado en una lista de palabras más vocabulario específico. Las reglas de producción dependen del modelo de representación de conocimiento elegido. English 05 y el esquema de representación de conocimiento basado en sintaxis descrito en los capítulos correspondientes del documento original en inglés son un ejemplo.

2. **Plan de curso.** El curso se divide en unidades de aprendizaje. Una unidad de aprendizaje implica un contenido de conocimiento que satisface un único objetivo de aprendizaje, en la misma forma en que se define un objeto de aprendizaje.

3. **Entrenamiento en el sublenguaje.** Se presenta a los estudiantes el sublenguaje: su motivación, propósito, estructuras, léxico, el editor, y unos cuantos ejemplos. Se debe informar a los estudiantes sobre las complejidades del lenguaje natural, y hacerlos conscientes de que escribiendo en el sublenguaje estarán escribiendo también en lenguaje natural correcto. Los estudiantes deben tener dominio de la gramática elemental subyacente a las reglas de producción, o ser guiados a través de trabajo de recuperación para dominarlas. El capítulo sobre construcción de un sublenguaje en el documento original en inglés contiene una lista de material de apoyo para el entrenamiento en el uso del sublenguaje.

4. **Construcción colaborativa.** Los estudiantes estudian todo el material correspondiente a una unidad de aprendizaje, y comienzan a escribir pequeños trozos de texto para describir sus contenidos. Con la guía y supervisión de los instructores, los estudiantes escriben en un editor que valida el texto en el sublenguaje, y dibuja los correspondientes grafos semánticos. La salida es uno o más grafos de referencia que contienen todo el conocimiento de esta unidad que se espera dominen los estudiantes.

5. **Preguntas.** Los instructores preparan las preguntas correspondientes a una unidad de
21. La solución propuesta

aprendizaje. Las preguntas abordan los contenidos registrados en los grafos de referencia para la unidad, posiblemente en menos pero nunca en más, y no abarcan conocimientos correspondientes a otra unidad de aprendizaje.

6. Evaluación. Los estudiantes contestan las preguntas escribiendo en el editor del sublenguaje, volcando el mismo conocimiento que han contribuido a registrar en los grafos de referencia para esa unidad de aprendizaje.

7. Realimentación. Los estudiantes obtienen una realimentación inmediata, viendo el grafo de referencia en gris, y destacadas sobre él en color las partes correspondientes a los items que lograron incluir en sus respuestas.

8. Evaluación de la enseñanza. Los errores y diferencias en la interpretación de los grafos de referencia pueden mostrar fallas en la enseñanza. Dado que la construcción de los grafos de referencia fue hecha en forma colaborativa y como actividad de aprendizaje, hubo suficientes oportunidades para corregir errores y acordar sobre diferencias. La responsabilidad compartida de estudiantes e instructores en el registro correcto del conocimiento se considera un valiosa experiencia de trabajo en grupo.

21.4. Discusión

La compilación de un sublenguaje no es una tarea trivial, pero existen recursos para hacer la tarea factible. Listas de palabras comunes pueden conformar la base de un léxico; muchos dominios del conocimiento tienen sus propios glosarios, y puede agregarse nuevas palabras dinámicamente. El diseño de un conjunto de reglas de producción cercanas a las estructuras comunes del lenguaje, con la expresividad necesaria, y simples de manejar, conlleva dificultades adicionales. El alcance de uso de un conjunto de reglas es mayor que el de un léxico, lo que permite su uso en diferentes dominios, reduciendo el esfuerzo. La contribución de un lingüista puede simplificar las cosas y ahorrar tiempo. Enseñar a los estudiantes a escribir en un sublenguaje diseñado tal como se describió no debería ser visto como una pérdida, ni en esfuerzo ni en utilidad: los estudiantes estarán aprendiendo a escribir correctamente en su lengua, aunque restringida, pero produciendo textos que cualquiera puede leer y entender sin esfuerzo particular; la mayoría de los lectores no notarán siquiera que los textos han sido escritos en un sublenguaje. Los hábitos de escritura introducidos durante el entrenamiento en un sublenguaje bien diseñado no deberían levantar más sospechas que el entrenamiento en la escritura técnica o científica. Un sublenguaje tiene mucho de eso, una forma de inducir un estilo claro, simple, directo, requerido por muchas profesiones. Un beneficio esperado de esta propuesta es una mejora en las capacidades de escritura de los estudiantes.

La integración de la evaluación con el estudio tal como se propone se considera una forma efectiva de reducir las tensiones propias de la instancia de evaluación como tal. En esta propuesta, los estudiantes llegan a la instancia de evaluación para hacer lo mismo que han hecho en anteriores sesiones de aprendizaje, con las mismas herramientas, y trabajando sobre los mismos contenidos. Escriben y ven los resultados de su escritura instantáneamente, en una presentación gráfica donde pueden comprobar y corregir. Construir un grafo semántico escribiendo texto en un editor será percibido probablemente como más atractivo que escribir un resumen o dibujar un mapa conceptual. Disponer de realimentación inmediata, tanto durante el aprendizaje como durante la evaluación, es universalmente reconocido como deseable.
Por último, pero no menos importante: no hay sesión de corrección para los instructores; la calificación se realiza automáticamente, sobre contenido bien conocido y acordado tanto por los instructores como por los estudiantes.

Ningún sistema funcionará bien fuera del alcance para el cual fue diseñado. Los esquemas de sublenguaje y representación de conocimiento se adaptan bien para el conocimiento fáctico, donde las oraciones aseverativas o imperativas son la forma natural de expresión. Mantenerse dentro del alcance previsto, un diseño sólido del esquema de sublenguaje y representación de conocimiento, un software de aplicación adecuado, y manejo competente de la clase, son todos factores esenciales para llevar exitosamente esta propuesta a clase.

21.5. Conclusiones

Luego de recorrer esta visión sumaria del sistema, quizás el punto principal a destacar sea la concepción de la propuesta como una actividad de aprendizaje, y no simplemente como un sistema de evaluación.

El uso en clase de las ideas propuestas en esta tesis se espera mejore el aprendizaje en general. La mayor parte del tiempo, los estudiantes estarán usando el sistema para aprender, que es definitivamente el objetivo principal, del cual la evaluación es simplemente una medida.
22. Contribuciones

Resumen. Las contribuciones originales de esta tesis radican en la combinación de ideas, métodos y técnicas de evaluación, procesamiento de lenguaje natural y representación de conocimiento, orientada a la calificación automática de respuestas a texto libre. En el ámbito de la evaluación, las contribuciones de esta propuesta comprenden: una determinación de los requerimientos educativos para un sistema de evaluación automática, aceptable tanto para los instructores como para los estudiantes; la responsabilidad compartida de instructores y estudiantes en la creación del material de evaluación; una integración efectiva de la evaluación en el aprendizaje, como una actividad de aprendizaje más. En el ámbito de lenguaje, las contribuciones de esta propuesta comprenden: la caracterización de un sublenguaje para educación; una metodología para la construcción de tal sublenguaje; un procedimiento para convertir texto en estructuras de conocimiento. En cuanto a léxicos, las contribuciones incluyen: identificación, auto verificación y comparación de varios recursos léxicos; la definición y prueba de una estructura común para soportarlos; una selección de palabras de función en idioma inglés manejadas separadamente de las palabras de contenido. Dos léxicos experimentales se presentan como contribución, basados en listas cortas de palabras, la General Service List of English Words y la Longman Communication 3000; su cobertura fue probada contra el corpus Brown. Esta tesis aporta varios conjuntos de reglas de producción, experimentales pero utilizables, tanto para el castellano como para el inglés, probados contra un conjunto reducido pero representativo de oraciones tipo. Las contribuciones en representación de conocimiento incluyen: una evaluación de técnicas y lenguajes en su potencial para la educación; un esquema de sublenguaje basado en sintaxis y representación de conocimiento, desarrollado y probado exitosamente en una aplicación prototipo, lo cual demostró la factibilidad extremo a extremo; una apreciación crítica de otros esquemas posibles de representación de conocimiento, que sugiere el tipo de relaciones de conocimiento que cada uno puede modelar mejor. Para la aplicación práctica de las ideas de esta tesis, se compiló una guía de despliegue, para prueba de campo y para uso en clase. Se considera que las contribuciones más valiosas de esta tesis son una propuesta practicable que efectivamente integra la evaluación en el proceso de aprendizaje, y un ensayo extremo a extremo del sistema satisfactorio, que prueba su factibilidad.

La propuesta de evaluación electrónica (eAssessment) de respuestas a texto libre realizadas en esta tesis condujo a contribuciones en las áreas de evaluación, lenguaje y representación de conocimiento. El presente capítulo describe estas contribuciones.
22. Contribuciones

22.1. Contribuciones

La evaluación automática de respuestas a texto libre, tal como se trata en esta tesis, se relaciona
con tres diferentes áreas de conocimiento: evaluación, procesamiento de lenguaje natural y re-
presentación de conocimiento. Aunque pueden asignarse a estos campos algunas contribuciones
originales, la originalidad de este trabajo reside más bien en la combinación de ideas, métodos
y técnicas existentes en estas áreas para dar una solución nueva a un problema existente,
solución que se considera mejor que las existentes. Esta es una aplicación de la metodología
propuesta por Andrión para la investigación en ingeniería de software [Andrión, 1993], adopta-
da en esta tesis tal como se establece en el capítulo Definición del Proyecto en el documento
original en inglés.

22.1.1. Evaluación

En el ámbito de la evaluación, las contribuciones de este proyecto incluyen:

- una determinación de los requerimientos para un sistema de evaluación satisfactorio
tanto para los estudiantes como para los instructores. Una solución efectiva no es solo
aquella que funciona, sino que también es aceptada como válida, útil y confiable. Las
expectativas de los instructores y de los estudiantes no son fácilmente compatibles; cono-
cer sus respectivas expectativas es un primer paso.

Esto fue hecho mediante una evaluación de importancia y consecuente selección de los
requerimientos a considerar para obtener un sistema confiable y aceptable. Primero es-
tablecemos lo que consideramos las más difundidas expectativas de los actores, junto
con los valores educativos a preservar en un sistema de respuesta abierta. Luego, la
consideración de los límites inherentes a un sistema de evaluación automático determi-
naron el tipo de respuestas a aceptar y el tipo de conocimiento a evaluar, i.e. oraciones
aseverativas y conocimiento fáctico. Recorriendo el proceso de evaluación, emergieron
las facilidades esperadas de una aplicación, lo cual determinó lo que consideramos un
sistema ideal. Los ítems anteriores fueron analizados, seleccionados y organizados de
acuerdo con nuestra propia visión. Aunque no son en modo alguno concluyentes, dieron
una orientación a nuestro trabajo, y ofrecen una base sólida para la discusión y el traba
jado futuro.

- la responsabilidad compartida de instructores y estudiantes en la evaluación. Un sistema
de evaluación asistida fue originalmente considerado la mejor opción, en lugar de un sis-
tema puramente automático. Finalmente, el sistema devino puramente automático para
la evaluación, pero como el instructor guió la creación de las estructuras de conocimiento
para la evaluación, no se considera necesaria una intervención ulterior. Los estudiantes y
los instructores comparten la responsabilidad de la creación correcta de las estructuras
de conocimiento a utilizar en la evaluación.

Aunque las aproximaciones colaborativas al aprendizaje son casi universalmente re-
comendadas, se hace difícil ponerlas en práctica en la evaluación: el estudiante debe
demostrar su conocimiento o habilidades, el docente debe determinar si esa demostración
es aceptable; un docente puede esmerarse en mostrar al estudiante sus errores o deficien-
cias, pero en última instancia el docente es juez. En nuestro sistema propuesto, todos los
22.1. Contribuciones

desacuerdos deben resolverse durante el proceso de aprendizaje, cuando se construyen las estructuras de conocimiento. La estructura de conocimiento de referencia contra la cual el conocimiento de los estudiantes será medido ha sido armada, y acordada por los estudiantes y los docentes juntos. Toda diferencia de apreciación sobre un ítem de un tema deberá tratarse durante el aprendizaje, en un momento en el cual puede recurrirse a material de referencia, otras opiniones, y la discusión constructiva. Las estructuras de referencia son de esta forma conocidas y acordadas tanto por los estudiantes como por los docentes; dado que el proceso de calificación es ahora enteramente automático, no queda lugar a dudas. Por su valor educativo, consideramos que esta característica es un punto muy fuerte a favor de nuestro sistema propuesto.

- la incorporación efectiva de la evaluación como una instancia más de aprendizaje. Esta propuesta implementa este concepto en su propia naturaleza, puesto que los estudiantes mismos construyen las estructuras a usar para la evaluación de su conocimiento.

Responder las preguntas de evaluación tradicionales, abiertas o cerradas, coloca al estudiante en una situación radicalmente diferente de lo que ha estado haciendo durante el aprendizaje: deberá contestar preguntas desconocidas eligiendo alternativas, escribiendo unas palabras o resultados, o escribiendo un trozo de texto. En muchos casos, incluso el tipo de preguntas no es conocido de antemano por el estudiante. Usualmente no hay instancias previas de entrenamiento, y el estudiante eventualmente se familiariza con la evaluación solo a través de la experiencia, a lo largo de su vida como estudiante. En algunos estadios de la educación esta forma de evaluación previamente desconocida puede aún ser deseable, puesto que en alguna medida emula lo impredecible de la vida misma, pero en la mayoría de los casos impone al estudiante una cuota de estrés e incertidumbre que compromete su desempeño. En nuestro sistema propuesto, la instancia de evaluación consiste en producir un texto que mapee hacia una estructura de conocimiento exactamente en la misma forma que lo hizo el estudiante cuando estudiaba el tema, en la certeza de que lo que produzca será comparado contra la misma estructura de conocimiento que conoció mientras estudiaba y que ayudó a construir. Nada nuevo, ninguna sorpresa aparecerá en la instancia de evaluación, sino la misma tarea que el estudiante ha venido realizando durante el estudio. Esto libera al estudiante del estrés de lo desconocido, y le da mayor confianza en su desempeño. Por esta razón, consideramos esta característica otro punto fuerte a favor de nuestro sistema propuesto.

22.1.2. Lenguaje

En el área de lenguaje, las contribuciones de esta propuesta comprenden:

- la caracterización de un sublenguaje para educación y para la escritura correcta. El sublenguaje propuesto se ubica entre los lenguajes controlados estrictos y las recomendaciones de estilo, imponiendo algunas limitaciones leves sobre el lenguaje natural y manteniendo el aspecto natural de los textos escritos, legibles como lenguaje natural correcto.

Salvo algunas propuestas para enseñanza de segunda lengua, no tenemos conocimiento de sublenguajes diseñados para propósitos educativos. Nuestra visión de las características que hacen deseable el uso de lenguajes restringidos para educación fueron expuestas en el capítulo sobre sublenguajes para educación del documento original en inglés, mientras
que en el capítulo sobre estado del arte de los sublenguajes fueron analizadas algunas de las dificultades y prejuicios que juegan en contra del uso de sublenguajes en educación, y se argumentó que la mayoría de estas dificultades y prejuicios pueden ser manejados exitosamente. Revisamos los lenguajes restringidos existentes, focalizando sobre los lenguajes de propósito general orientados a la comunicación humana, para seleccionar los principales requerimientos de un sublenguaje apto para educación (capítulo sobre construcción de un sublenguaje en el documento original en inglés). Agregamos algunos requerimientos adicionales para asegurar el determinismo (i.e. textos no ambiguos), una cualidad esencial para extraer conocimiento del texto. Esto resultó en una serie de reglas de escritura que favorecían la tratabilidad de los textos conservando la legibilidad y la escritura correcta (capítulo sobre construcción de un sublenguaje, en la versión original en inglés). Aunque no es mandatorio, y discutible en su importancia relativa, esta compilación de reglas de escritura proveen una base para el diseño de un sublenguaje para educación. Se requiere cierto conocimiento de gramática para escribir en tal sublenguaje, y dado que la escritura en un sublenguaje demanda cierta disciplina de parte de quien escribe, se detectaron como deseables, o aún imprescindibles, algunos recursos y herramientas para que el sistema resulte amigable para el usuario (capítulo sobre construcción de un sublenguaje). El análisis realizado, y los productos obtenidos, conforman una guía para el diseño de un sublenguaje para propósitos educativos, conservando la libertad del diseñador para elegir alguna de las propuestas existentes o diseñar la suya propia.

- **una primera aproximación a una metodología para la construcción de un sublenguaje**, incluyendo consideraciones de diseño, identificación de reglas generales de escritura, facilidades que una aplicación debe proveer, documentación y tutoriales, dificultades detectadas, y formas de resolverlas.

Basados en nuestra previa caracterización de un sublenguaje para educación, se desarrollaron varios ejemplos de sublenguajes (capítulo sobre sublenguajes basados en sintaxis), en un diseño propio. La experiencia adquirida en la construcción de estos sublenguajes ejemplo aparece registrada en el capítulo sobre construcción de un sublenguaje. Características tales como unidades de varias palabras, nombres propios, mayúsculas al inicio, números, fechas y medidas, pueden manejar mediante preprocesamiento del texto y la aplicación de reconocimiento de patrones. Una discusión de otras mejoras incluye completamiento de palabras, árboles sintácticos parciales, anticipación de categoría léxica, ayuda léxica, desambiguación de significados. Además de probar la factibilidad del emprendimiento, la experiencia de construir estos sublenguajes ejemplo permitió definir una primera aproximación a una metodología para la construcción de un sublenguaje que, en nuestra visión, ofrece una base útil para futuros desarrollos. La metodología más específica inferida del diseño de reglas de producción para una gramática generativa se discute en su propio parágrafo más adelante en este capítulo.

- **un procedimiento para convertir el texto escrito en un sublenguaje en una estructura de conocimiento**, incluyendo la identificación de tareas requeridas, el uso de ideas de gramáticas de dependencia, la definición de un conjunto de etiquetas simple, y el uso de etiquetas sustitutivas para desacoplar el léxico y las reglas.
22.1. Contribuciones

El diseño de un sublenguaje para escribir textos aptos para la representación de conocimiento debe ir en paralelo con el lenguaje de representación de conocimiento, puesto que se debe definir la conversión de cada una de las estructuras sintácticas en estructuras correspondientes del lenguaje de representación de conocimiento. Los pasos y tareas identificados en el capítulo sobre construcción de un sublenguaje basado en sintaxis fueron luego aplicados en la conversión de este sublenguaje en una representación de conocimiento en forma de grafo, como se describe en el capítulo sobre representación de conocimiento basada en sintaxis, y también en la discusión de los otros esquemas de sublenguaje y representación de conocimiento propuestos en el capítulo correspondiente.

En cuanto a léxicos, los recursos disponibles son mucho más abundantes y maduros para la lengua inglesa que para el castellano; los estudios de léxicos se limitaron al idioma inglés por esta razón. Sin embargo, las tareas desarrolladas y la metodología inferida pueden aplicarse igualmente a recursos léxicos en castellano o cualquier lenguaje occidental. La Wordnet, en particular, está disponible en inglés como recurso libre, en tanto el proyecto EuroWordnet requiere un pago. En cuanto a léxicos, esta tesis provee las siguientes contribuciones:

- **identificación de recursos libremente disponibles, evaluación de su usabilidad, y tareas que pueden realizar.** Verificaciones de consistencia interna, detección de errores, corrección de algunos errores simples sin alterar lo esencial de las listas originales.

Nuestro uso de recursos léxicos se limitó a los disponibles libremente en Internet: GSL (4 versiones), Simple English Wikipedia (relacionada con el Inglés Básico), VoA Special English, Longman Communication 3000, y las listas largas BNC, COCA, SCOWL, AGID, y PoS DB; aunque no estrictamente una lista de palabras, la Wordnet fue identificada como una oferta de gran potencial para desarrollos futuros (capítulo sobre recursos léxicos). A cada lista se le corrieron varias verificaciones de consistencia interna, como ser repeticiones, caracteres no alfabéticos, y anomalías de formato. Se encontraron algunos errores, aún en listas muy cuidadosamente mantenidas. Realizamos algunas correcciones menores, especialmente en las listas cortas que usamos (GSL Gilner y Longman 3000). El trabajo realizado sobre estas listas permitió determinar su cubrimiento, calidad, y tareas que pueden realizar. Una descripción de cada una, con ejemplos de las anomalías encontradas e información de tamaño, se encuentran en el capítulo sobre recursos léxicos. El software desarrollado para el manejo de léxicos se encuentra en el módulo de léxicos del proyecto KLEAR, descrito en el capítulo correspondiente. La documentación de las herramientas de software desarrolladas, los módulos Python para verificación, y los archivos doctest de las pruebas realizadas están disponibles en el sitio web de la tesis, [González-Barbome, 2012].

- **comparaciones entre recursos léxicos disponibles.** La comparación de las listas cortas de palabras comunes en cuanto a superposición, cobertura de listas cortas en listas largas, qué hacer para mejorarlas o hacerlas utilizables.

Se verificaron varias listas una contra otras, tal como palabras en las listas cortas contenidas en las listas largas, o comparación de las listas cortas entre sí. Las tablas del capítulo sobre recursos léxicos resumen los resultados. Los módulos Python para prueba, y los archivos doctest de las pruebas realizadas, están disponibles en [González-Barbome, 2012].
22. Contribuciones

- **una estructura de datos y funciones para soportar léxicos**, convirtiendo desde diferentes formatos originales. Una estructura de datos común permite la prueba y comparación de recursos léxicos, y su uso en aplicaciones experimentales.

La estructura de datos para soportar léxicos fue diseñada para cumplir las tareas de inclusión (si una palabra está en el léxico, ya sea como forma de palabra o como lema), lematización (eventualmente más de un lema para una forma de palabra), categorización léxica (eventualmente varias categorías léxicas por palabra), frecuencias de uso (por forma de palabra, por lema y categoría léxica, por lema solamente), definiciones (significados de una palabra), synsets relacionados a un lema. La eficiencia fue considerada, por lo que la mayoría de las estructuras están implementadas como diccionarios de Python, que son muy rápidos en el acceso por clave; se introdujo alguna redundancia controlada cuando se consideró necesario. Las tareas propias de un léxico se describen en el capítulo de compilación de léxicos; el módulo de léxico del proyecto KLEAR se describe en el capítulo del proyecto; la documentación del software está disponible en [González-Barbome, 2012].

- **dos léxicos experimentales de propósito general**, uno basado en la General Service List of English Words (GSL), y otro basado en la lista Longman Communication 3000. Ambas son listas de palabras de uso común.

Estos léxicos experimentales están disponibles como archivos pickle de Python en [González-Barbome, 2012]. Al leerlos quedan disponibles como objetos Python en la estructura de datos común definida para el manejo de léxicos; la documentación para el módulo de léxicos está disponible en [González-Barbome, 2012].

- **una primera aproximación a una metodología para compilar léxicos de propósito general o de dominio específico**: las palabras de función fueron analizadas separadamente, dado que tienen escaso contenido semántico, pero articulan las estructuras sintácticas que deben reconocerse.

La experiencia recogida durante la compilación de los lenguajes ejemplo desarrollados nos dio una primera aproximación para compilar léxicos de propósito general que sirvan de base para un sublenguaje de dominio específico. Las estructuras de datos para soportar léxicos se diseñaron para cumplir tareas tales como inclusión (si una palabra está en el léxico, ya sea como forma de palabra o como lema), lematización, etiquetado en categorías léxicas, frecuencia de uso, definiciones. Se compiló un léxico de propósito general basado en GSL, primero seleccionando las palabras de función, luego agregando las palabras de contenido, pero manteniendo ambos conjuntos disjuntos; esto requirió algunas decisiones de cuidado. Se consideraron algunos ajustes, tales como palabras con varias categorías léxicas, palabras de función que son también palabras de contenido, palabras de contenido dejadas fuera del léxico, consistencia interna, manejo de nombres propios, cantidades numéricas, fechas, inflexiones, y significados (capítulo sobre compilación de léxicos). La construcción de un léxico basado en GSL Gilner fue luego seguida por la construcción de otro basado en Longman 3000, siguiendo las mismas líneas; la aproximación metodológica probó ser igualmente útil en ambos casos.

- **herramientas y procedimientos para probar la cobertura contra corpora de textos**. Los
léxicos experimentales desarrollados fueron probados contra el corpus Brown.

Las herramientas y el software desarrollado en este proyecto son utilizables como tales o fácilmente adaptables para probar cobertura contra otros corpora; la adaptación se requerirá principalmente para acceder a estos otros corpora, que pueden estar usando diferentes juegos de etiquetas (tagsets), o presentarse en diferentes formatos. El corpus Brown está hoy desactualizado, pero está disponible libremente y es muy conocido; de la misma forma, es posible hacer pruebas de cubrimiento contra otros corpora, en tanto se logre la consistencia con el formato y con el juego de etiquetas (tagset).

En cuanto a reglas de producción para gramáticas generativas, se realizaron las siguientes contribuciones:

- *dos conjuntos de reglas para el idioma castellano para oraciones declarativas*: Spanish 01 se mantiene cerca de las estructuras sintácticas de la gramática tradicional; Spanish 05 requiere menor conciencia de la gramática, con una constitución de grano más grueso. Cada uno de estos sublenguajes comprende reglas y oraciones ejemplo. Fueron comprobados con léxicos mínimos, pero es posible agregar a la aplicación léxicos más amplios.

Estos ejemplos de sublenguajes son un diseño propio, y pueden ser usados como tales o como base para desarrollos futuros.

- *un conjunto de reglas para un sublenguaje en inglés con oraciones declarativas*, llamado English 05, que sigue la línea de Spanish 05 en sus requerimientos mínimos de conocimiento gramatical. Este conjunto de reglas fue probado con EnLong3k, un léxico de propósito general compilado a partir de la lista de palabras Longman Communication 3000.

Estos ejemplos de sublenguajes son un diseño propio, y pueden ser usados como tales o como base para desarrollos futuros.

- *una primera aproximación a una metodología para compilar un conjunto de reglas para sublenguajes*, basado en una lista de palabras de función previamente identificadas, y palabras de contenido tomadas de listas de palabras existentes, corpora, o agregadas manualmente.

En los sublenguajes ejemplo desarrollados, las diferentes versiones de sublenguajes en castellano exploraron diversos conjuntos de reglas alternativas, y permitieron conocer el proceso de construcción (apéndice sobre sublenguajes ejemplo, en la versión original en inglés). Los sublenguajes ejemplo para el inglés tomaron esa experiencia, y solo dos versiones fueron conservadas, con diferentes niveles de “conciencia gramatical” de parte de quien escribe (apéndice sobre sublenguajes ejemplo). En ambos casos, los juegos de etiquetas léxicas (tagsets) propuestos fueron hechos tan simples como fue posible, pero suficientes para nuestro propósito, y decentemente legibles, dado que quienes escribían deberán ser conscientes de las etiquetas léxicas (capítulo sobre sublenguajes basados en sintaxis). La construcción de las reglas para estos sublenguajes ejemplo, junto con sus correspondientes conjuntos de oraciones ejemplo, nos dieron una primera aproximación a una metodología para compilar conjuntos de reglas. El tratamiento por separado de
las palabras de función y las palabras de contenido, junto con el uso de oraciones de prueba para cada regla, son quizás las recomendaciones metodológicas más importantes.

**22.1.3. Representación de Conocimiento**

En el campo de la representación de conocimiento, se realizaron las siguientes contribuciones:

- **un estudio comparativo y evaluación de diferentes técnicas y lenguajes de representación de conocimiento**, en su utilidad para la educación y evaluación, basado en los roles tradicionales reconocidos para la representación de conocimiento establecidos originalmente por Davis [Davis and Szolovits, 1993].

Los cinco roles tradicionales atribuidos a la representación e conocimiento fueron considerados y evaluados en su importancia relativa para representar conocimiento con propósitos educativos. Esta evaluación fue usada como guía para calificar los diferentes lenguajes de representación de conocimiento para su uso en educación, a saber: Lógica de Primer Orden, sistemas basados en reglas, sistemas basados en marcos (frames), mapas conceptuales y redes semánticas, ontologías, y lenguajes restringidos (capítulo sobre representación de conocimiento para educación). Condujimos nuestra discusión en una perspectiva amplia, basados en el hecho reconocido de que un único lenguaje de representación de conocimiento no puede representar todas las clases de conocimiento que la mente humana es capaz de manejar. Además de esta calificación inicial, la selección de un lenguaje de representación de conocimiento dependerá del campo de conocimiento y el propósito perseguido. Sin embargo, algunos lenguajes de representación de conocimiento parecen más apropiados que otros, y los diferentes grados de evolución y disponibilidad también pesan en la decisión.

- **un esquema de representación de conocimiento basado en sintaxis**, que define la transformación de las estructuras sintácticas en nodos y arcos de un grafo semántico. Las estructuras sintácticas fueron las de English 05, el sublenguaje desarrollado anteriormente, pero la transformación se basa en ideas de gramática de dependencias que pueden ser aplicadas a diferentes sublenguajes. Una medida de distancia permite reconocer el conocimiento proveniente de un texto nuevo en el grafo de referencia existente.

En nuestra aproximación, usamos ideas de las gramáticas de dependencia para detectar en el texto los conceptos y relaciones que luego serán los nodos y arcos de un grafo. El sublenguaje empleado fue English 05, descrito en el capítulo de lenguajes basados en sintaxis en la versión en inglés de este documento. La conversión de las construcciones sintácticas de English 05 en un grafo aparecen en el capítulo sobre representación de conocimiento basada en sintaxis. Algunas consideraciones sobre cómo evitar la redundancia, e ideas más avanzadas tales como agrupamiento de conceptos (concept clustering), también fueron discutidos. Esta elaboración muestra es solo una elaboración posible, y pueden usarse otros esquemas. El esquema empleado mostró ser suficiente para nuestros propósitos, y puesto que sigue de cerca las estructuras gramaticales comunes, debe resultar fácil de dominar por cualquier persona familiarizada con la gramática de la escuela secundaria. Es posible pensar en una evolución posterior de English 05 junto con las reglas de transformación correspondientes, ampliando la sintaxis del sublenguaje, e introduciendo las relaciones apropiadas en las reglas de transformación.
22.1. Contribuciones

- una guía para el desarrollo de un esquema de sublenguaje y representación de conocimiento para la distinción de categoría e individuo, en la forma de la Programación Orientada a Objetos, pero usando expresiones de lenguaje cotidiano y extrayendo de ellas la distinción entre categorías e individuos implicadas en esas expresiones. Una propuesta que adapta la metodología de Orientación a Objetos al aprendizaje.

Una discusión de los aspectos principales de un sublenguaje con distinción de categorías e individuos, y la conversión de construcciones sintácticas en un diagrama UML o un grafo equivalente, se proporcionan como una guía de diseño para un desarrollo completo. Se identificaron varias construcciones sintácticas del lenguaje común para definir categorías, definir la pertenencia de un individuo a una categoría, definir subclases, asignar propiedades a categorías o individuos, integración de partes en un todo, y establecimiento de asociaciones nominadas. Las construcciones sintácticas están en el capítulo sobre esquemas de representación de conocimiento; como puede verse, la mayoría de estas construcciones son expresiones comunes de uso diario en el lenguaje, lo que hace que el sublenguaje suene muy natural. Sin embargo, resulta en una construcción de conocimiento muy elaborada que exhibe las principales características de un modelo orientado a objetos.

- una guía para desarrollar otros esquemas de sublenguaje y representación de conocimiento, para desarrollos secuenciales y manuales de instrucción.

Una discusión de las ideas principales para otros tipos de conocimiento se proporcionan como guías para un desarrollo completo, dando ideas para la solución de algunos aspectos específicos de cada una. Una crónica o secuencia de hechos requiere un orden cronológico; se indican diferentes construcciones de lenguaje para fijar referencias en el tiempo, definir períodos de tiempo, e insertar eventos en un grafo de línea temporal. Los manuales de instrucción o descripciones de procedimientos requieren el manejo de oraciones imperativas, y la habilidad de mantenerlas en un cierto orden. Las oraciones aséverativas pueden tomarse de nuestras propuestas ejemplo; el secuenciamiento puede inspirarse en la discusión precedente sobre líneas temporales, aunque simplificadas; el agregado de condicionales y bifurcaciones en el curso de eventos es también posible. Esto completa las ideas principales para el diseño de un esquema de sublenguaje y representación de conocimiento para escribir respuestas que describan cómo se realiza una tarea.

Las posibilidades e inconvenientes de la representación de conocimiento en forma de mapas conceptuales, mapas mentales y otros lenguajes de representación de conocimiento más o menos informales, fueron también discutidos. El alto potencial de los Topic Maps, un lenguaje de representación de conocimiento regulado por un estándar, los muestra como una alternativa promisoria en la representación de conocimiento para propósitos educativos, aunque este emprendimiento requiere un proyecto de largo alcance.

22.1.4. Prototipos y desarrollo

Aunque no estrictamente trabajo de investigación, el desarrollo de código alcanzó una dimensión considerable, está bien documentado y es mayormente reutilizable, ya sea para pruebas o para desarrollo de aplicaciones. Por esta razón, se lo lista aquí como contribución, junto con
22. Contribuciones

algunas consideraciones y sugerencias acerca de cómo desplegar el sistema.

- estructuras de datos y funciones para léxicos, para cargar recursos léxicos existentes hacia un formato común para la auto verificación, comparación y uso en la construcción de sublenguajes.

- una aplicación prototipo para escribir en un sublenguaje, dibujar una red semántica a partir de texto, y reconocer texto en una red semántica. El léxico, las reglas de producción, los datos de dependencias para la conversión de árboles sintácticos en árboles de dependencia y los atributos gráficos para la presentación visual se cargan todos de ficheros, lo cual permite que la aplicación sea utilizada para probar diferentes esquemas de sublenguaje y representación de conocimiento.

- una guía de desarrollo, con sugerencias sobre los requerimientos y los pasos necesarios para llevar exitosamente la propuesta al salón de clase en una primera etapa de prueba. La propuesta se presenta como una actividad de aprendizaje, y requiere un cuidadoso desarrollo, aún para una prueba.

22.2. Conclusiones

La idea original de esta tesis fue usar alguna forma de lenguaje restringido para la escritura de textos, para poder construir instancias de representación de conocimiento entre las cuales pudiera medirse una distancia. Se realizaron contribuciones en evaluación, construcción y uso de sublenguajes, propuestas de esquemas de representación de conocimiento, y prueba de concepto. La mayor parte del código de prueba es reutilizable, la aplicación prototipo provee una demostración extremo a extremo del sistema, y puede considerarse una primera versión de un sistema para producción en prueba beta. Es posible prever con trabajo futuro aplicaciones mucho más amigables y con mejor ayuda para facilitar la escritura y mejorar la visualización.

Se consideran como más valiosas contribuciones de esta tesis la propuesta de un sistema practicable que efectivamente integra la evaluación al proceso de aprendizaje, y una prueba satisfactoria de principio a fin del sistema, lo que demuestra su factibilidad.
23. Conclusiones

Resumen. Los métodos tradicionales de aprendizaje se renuevan con dificultad. Nuevos desarrollos casi universalmente reconocidos como valiosos rara vez llegan al salón de clase como herramientas de uso diario. Este proyecto intentó lograr una solución practicable, no solo a través de su factibilidad, sino también a través de su aceptación. La creación de estructuras de conocimiento a partir de textos escritos en un sublenguaje es una actividad de aprendizaje equivalente a resumir o dibujar mapas conceptuales. Las estructuras creadas son las que se usarán en la evaluación, lo cual resulta en un sistema justo y aceptable para los estudiantes. La evaluación se prepara en una actividad de aprendizaje, la calificación es automática, y la re-alimentación a los estudiantes es inmediata. Los instructores pueden dedicar el tiempo de calificación a la enseñanza, y librarse del penoso trabajo de corregir. Un sublenguaje bien diseñado, aplicado correctamente dentro de su dominio objetivo, y un conocimiento claro de lo que este sublenguaje es, debe ser suficiente para apartar cualquier sospecha de limitaciones a la libertad de expresión. Ningún sublenguaje puede ni pretende sustituir al lenguaje natural; cada sublenguaje tiene un propósito, y no debe ser usado con otro propósito. El conocimiento es un fenómeno humano extremadamente complejo, y los diferentes esquemas de representación de conocimiento dan cuenta de los muchos modos en que la mente humana captura el mundo real. Un determinado esquema de representación de conocimiento no será útil para todas las situaciones; diferentes esquemas de sublenguaje y representación de conocimiento pueden ser usados ventajosamente si se seleccionan adecuadamente para cada situación. La aplicación de esta propuesta induce a los estudiantes a escribir correctamente, les enseña a organizar su conocimiento, provee una experiencia colaborativa, e integra la evaluación en el aprendizaje, reduciendo drásticamente el sentimiento de estrés asociado a los exámenes. Basado en el trabajo de este proyecto, investigaciones y desarrollos futuros en el uso de sublenguajes y representación de conocimiento para la educación y la evaluación se consideran un esfuerzo valioso a realizar.

Este capítulo recoge las conclusiones emergentes del trabajo realizado a lo largo de esta tesis. Como la evaluación se concibe como una actividad de aprendizaje, las conclusiones abarcan los ámbitos de la evaluación, educación, procesamiento de lenguaje natural y representación de conocimiento.

23.1. Cumplimiento

La metodología de ingeniería seguida a lo largo de este trabajo fue la propuesta por Adrion en 1993, e implicó los siguientes pasos:

1. evaluación crítica de las propuestas existentes para la calificación automática de respuestas a texto libre;
23. Conclusiones

2. diseño de una solución para resolver las carencias o mejorar los sistemas existentes;
3. prueba, evaluación o validación de la solución propuesta;
4. iteración de los procesos anteriores incorporando mejoras.

En el documento original en inglés se indican los capítulos de la tesis en los cuales se dio cumplimiento a los objetivos fijados, conforme la metodología expuesta.

23.2. Una propuesta practicable

En una época tan lejana como el año 1967, el profesor L. G. Alexander lamentaba que “los métodos tradicionales de enseñar una lengua extranjera murieran tan lentamente” (“traditional methods of learning a foreign language die hard”) [Alexander, 1967], en referencia a las importantes observaciones sobre aprendizaje realizadas por el Dr. Harold Palmer en 1921, el acuerdo general sobre la validez de esas observaciones, y el desacuerdo general sobre cómo llevarlas a la práctica. Aunque el profesor Alexander se refería a las lenguas extranjeras, lo mismo puede decirse en muchos campos. El advenimiento de la era digital trajo una profusión de técnicas nuevas, muchas de ellas destacables por su valor para el aprendizaje y su costo razonable, como es el caso de la simulación, en la cual el estudiante interactúa con una computadora como si estuviera en la cabina del piloto o desplegando una red de comunicaciones. Menos impresionantes pero igualmente valiosas, las herramientas disponibles hoy en un notebook común y corriente permiten el uso de técnicas de enseñanza y aprendizaje que, si no son nuevas, son mucho más realizables que antes.

A pesar del valor de muchas propuestas de aprendizaje, resulta alarmante el escaso número de ellas que llegan a aplicarse en clase en un curso real. Los mapas conceptuales pueden ser un caso paradigmático: un enorme número de artículos, libros, tutoriales y páginas web los describen, y promueven su utilización describiendo sus ventajas. Sin embargo, una simple conversación con colegas instructores en diferentes lugares muy probablemente muestre que la mayoría de ellos sabe de los mapas conceptuales, los consideran muy interesantes y valiosos, pero no los usan en sus clases. Quizás algunos de esos colegas han probado los mapas conceptuales un par de veces en sus clases, pero es difícil encontrar un instructor que sistemáticamente use mapas conceptuales en su enseñanza. Naturalmente, puede haber instituciones que los han adoptado, y puede haber instructores en el mundo que los usen regularmente, pero los mapas conceptuales están muy lejos de ser universalmente utilizados. Curiosamente, podría decirse que han sido aceptados universalmente pero raramente son usados.

Un principio guía de esta tesis fue producir una propuesta practicable. En una primera aproximación, esto significa algo que pueda ser puesto en práctica por un pequeño equipo, con recursos limitados, sin requerir expertos en computación o ingenieros de conocimiento para cada curso, aunque su apoyo pueda ser invaluable en el desarrollo de una aplicación. Más adelante, una propuesta práctica se presentó como aquella que pudiera ser aceptada simultáneamente por estudiantes e instructores, que ambos estuviesen dispuestos a probar, a trabajar con ella y eventualmente a adoptar. Una mirada a las muchas propuestas existentes, durmiendo en los estantes o girando en la web, debería ser suficiente para convencer a cualquiera de que se trata de un objetivo muy difícil.

Se hicieron dos previsiones para la aceptación: la integración de la evaluación en el aprendizaje, y el compromiso compartido entre instructores y estudiantes en ambas actividades.
23.3. Una actividad de aprendizaje

El material de evaluación es reunido por los estudiantes mientras aprenden; aprenden a través de la creación de estructuras que serán la referencia para la evaluación. Los instructores se liberan de laboriosas preparaciones de exámenes; solo deben incluir en una pregunta aquellas partes de la estructura de referencia que desean que los estudiantes reproduzcan. Los instructores “preparan el examen” cuando guían el aprendizaje de sus estudiantes, un uso del tiempo mucho mejor.

El trabajo colaborativo es universalmente reconocido como deseable en educación. Aquí los estudiantes trabajan colaborativamente en la construcción de estructuras de conocimiento, junto con sus compañeros, bajo la guía de los instructores. Instructores y estudiantes comparten la responsabilidad por la calidad de las estructuras de conocimiento creadas. El proceso compartido de construcción ofrece oportunidades para aclarar dudas, decidir sobre puntos discutibles, acordar en el alance. Una vez cumplidas las actividades de aprendizaje, la instancia de evaluación no debería traer sorpresas, estrés, ni demasiado trabajo.

23.3. Una actividad de aprendizaje

Este proyecto comenzó con el objetivo de mejorar un sistema de evaluación, pero el trabajo de investigación, diseño y prueba de prototipos, junto con la experiencia de años dedicados a la enseñanza, gradualmente condujeron a la idea de que un sistema sería insuficiente como solución práctica y educativamente valiosa. Lo que se necesitaba era una propuesta educativa. Los sistemas que trabajan sobre respuestas a texto libre escritas en lenguaje natural, aún si son exitosos, no representan ninguna contribución particular al proceso de aprendizaje. Además de enfrentar la dificultad de textos escritos en lenguaje natural completo, con todas sus complejidades y ambigüedades, estos sistemas deben enfrentar un problema diferente e inesperado: el mal uso del lenguaje por parte de los estudiantes. En sistemas que trabajan sobre conjuntos de entrenamiento, el mal uso del lenguaje puede pasar totalmente inadvertido; el sistema aprenderá cualquier cosa que aparezca en el conjunto de entrenamiento, y calificará conforme las nuevas respuestas. En sistemas basados en estructuras previamente cargadas, o en respuestas correctas escritas por los instructores, o en libros de texto, el sistema esperará respuestas escritas correctamente y calificará como erróneo lo que no pueda reconocer, sin hacer diferencia entre la mala escritura y el conocimiento errado.

Las consideraciones anteriores conducen a algunas preguntas cruciales:

- ¿Qué tan malo es el uso del lenguaje en las respuestas de los estudiantes? El nivel educativo, el medio social de los estudiantes, las áreas temáticas en las cuales han sido formados, la escala para medir el buen uso del lenguaje, son solo algunos de los factores a considerar en una respuesta. Siendo muy conservadores, puede decirse que el uso del lenguaje en los textos escritos por los estudiantes está, en general, por debajo de los estándares esperados para cada nivel educativo.

- ¿Debe un sistema de evaluación involucrarse con el uso del lenguaje? El propósito de un sistema de evaluación es determinar el nivel de logro en algún conocimiento o destreza, no enseñar. Un sistema puede sufrir en su exactitud debido al mal uso del lenguaje, especialmente si trabaja con conjuntos de entrenamiento, pero en una definición de roles estricta, no es responsabilidad del sistema de evaluación asegurar el correcto uso del lenguaje.
23. Conclusiones

- **¿Qué significa “buen uso del lenguaje”?** Los criterios pueden diferir abruptamente entre instituciones, comunidades, e individuos; no hay esperanza de acuerdo. La adopción de un estándar es una forma de superar el estancamiento de las discusiones. Algunos de los lenguajes discutidos en el capítulo de Sublenguajes del documento original en inglés ofrecen léxico, reglas de gramática, material de lectura y tutoriales que en su conjunto garantizan el buen uso del lenguaje.

Uno de los principios bajo los cuales fue desarrollada esta tesis fue obtener una propuesta realmente útil para la enseñanza y el aprendizaje, no solamente para la evaluación. En esta perspectiva, el buen uso del lenguaje es una consideración. El proyecto evolucionó hacia una propuesta educativa, en la forma de una actividad útil tanto para el aprendizaje como para la evaluación, que integra efectivamente ambos procesos, y en la cual la evaluación llega como una consecuencia natural de las actividades de aprendizaje desarrolladas previamente.

**23.4. La cuestión del lenguaje**

La idea de restringir el uso del lenguaje es, a priori, muy cuestionable, al punto de ser considerada inaceptable en algunos contextos. Dos aspectos muy importantes pueden convertir lo negro en blanco: el propósito y la calidad.

Ningún sublenguaje, lenguaje controlado, o versión restringida del lenguaje, puede considerarse un reemplazo del lenguaje natural en toda circunstancia. Los sublenguajes emergen naturalmente en una comunidad por las necesidades de sus miembros; los lenguajes controlados se diseñan con un propósito, en una especificación formal; los sublenguajes en este proyecto están concebidos como diseñados para un propósito pero no necesariamente adhiriendo a un estándar reconocido, aunque esto pueda eventualmente suceder, y ser ventajoso.

Escribir en un sublenguaje no significa escribir mal. Por el contrario, puede contribuir efectivamente a la buena escritura. Un sublenguaje es un subconjunto del lenguaje natural; las oraciones escritas en un sublenguaje serán un subconjunto de todas las oraciones correctas escritas en lenguaje natural. En este contexto, correcto no significa solamente cumplimiento léxico y sintáctico, sino también satisfacer el concepto de uso correcto del lenguaje que tiene el hablante en general, por ejemplo tal como se enseña en las escuelas o como segunda lengua.

Un mal a veces invocado es la limitación en la libertad de expresión. Esto no tiene fundamento, al menos por dos razones: primero, algunos sublenguajes están destinados a la comunicación humana, se enorgullecen de su claridad, simplicidad y expresividad, y solo establecen limitaciones formales para el buen uso del lenguaje. Segundo, un sublenguaje tiene un propósito, y en consecuencia un contexto de aplicación, dentro de una comunidad que acepta ese sublenguaje como una conveniencia, porque sus miembros ven el beneficio de su adopción. Como ya se dijo, ningún sublenguaje intenta reemplazar el lenguaje natural. Nadie pensaría en imponer un sublenguaje en la literatura, la filosofía o la política, ni tampoco en las salas de chat, las redes sociales, los correos electrónicos o los mensajes de texto en los celulares.

Hacer que los estudiantes sean conscientes del léxico, la gramática, las ambigüedades y complejidades del lenguaje, pueden también ser vistos como una barrera en un programa escolar siempre sobrecargado. Aquí la línea divisoria puede trazarse según el esfuerzo requerido de los estudiantes: una propuesta de un sublenguaje para la evaluación tal como se concibe en esta tesis no debería demandar un esfuerzo extra significativo más allá del requerido por los cursos de lenguaje en la enseñanza secundaria. Los sublenguajes propuestos en base a sintaxis y basados en esquemas de representación de conocimiento propuestos en esta tesis se consideran...
23.5. Adquisición de conocimiento

Uno de los aspectos del aprendizaje es la adquisición de conocimiento. La propuesta de esta tesis concierne a un tipo particular de conocimiento: aquel que puede expresarse en oraciones aséverativas, tales como eventos, hechos, conceptos abstractos o sucesos imaginarios. Este "conocimiento declarativo" es solo una parte de todo el conocimiento que el estudiante debe adquirir en la mayoría de los cursos. Aunque limitado, este conocimiento declarativo es el más tedioso de evaluar; generalmente pide al estudiante, de diferentes formas, recordar lo que ha aprendido. Las preguntas cerradas están dirigidas a este tipo de conocimiento. El uso de respuestas a texto libre, incluso respuestas breves consistentes en al menos una oración completa, obligan al estudiante a producir algo más elaborado de lo que requieren las preguntas cerradas. Aunque un conjunto de preguntas de múltiple opción cuidadosamente preparadas pueden dejar a un buen estudiante en la duda, su respuesta será una marca en un pequeño cuadro. Una respuesta de texto, incluso si es una sola oración, es un tipo de respuesta más creativa y completa en sí misma. Una oración dice algo por sí misma, puede existir sola, es una pequeña creación, y ofrece una mayor recompensa. Unas pocas oraciones en una respuesta de texto pueden decir mucho sobre lo que un estudiante sabe y es capaz de expresar.

Las consideraciones anteriores no fueron hechas para desvalorizar las preguntas cerradas, sino para remarcar el valor de los estudiantes produciendo algo más elaborado que el mero hecho de elegir o aparear. El uso de respuestas a texto libre en la evaluación, incluso en pequeña escala, con respuestas cortas, puede verse como algo más demandante, una prueba más informativa del conocimiento que el estudiante posee, y si es capaz de comunicarlo.

Los textos y los grafos son instrumentos de aprendizaje, y se usan del mismo modo que las notas de los estudiantes, los resúmenes o esquemas de conceptos y relaciones principales de la unidad en estudio. Para los instructores que fomentan el uso de los mapas conceptuales, esta propuesta simplemente agrega el uso del sublenguaje para su creación. La construcción de las estructuras de conocimiento es muy similar a la construcción de un mapa conceptual, aunque con las restricciones impuestas por el esquema de sublenguaje y representación de conocimiento elegido. Como fue discutido en el capítulo sobre esquemas de representación de conocimiento en el documento original en inglés, esto puede ser beneficioso, incorporando algunas reglas formales de representación a las convenciones más o menos laxas de los mapas conceptuales. Diferentes esquemas de lenguaje y representación de conocimiento son posibles, unos mejor adaptados que otros para modelos de conocimiento que jerarquizan distintas clases de relaciones entre conceptos. Estos diferentes modos de estructurar el conocimiento reflejan algunas de las muchas formas que tiene la mente humana de ahondar el mundo.

al alcance de estudiantes de secundaria a nivel medio, con algunas horas de entrenamiento y práctica.

Este proyecto intenta mejorar la evaluación asistida por ordenador de las respuestas a texto libre imponiendo restricciones leves sobre el uso del lenguaje natural para hacer posible la extracción de conocimiento sin ambigüedades. Al principio esto fue visto solo como una forma de convertir un problema difícil en algo tratable, como un inconveniente necesario. Sin embargo, las posibilidades educativas de un sublenguaje bien diseñado soportado en una aplicación amigable surgieron casi inmediatamente. El uso de sublenguajes en la enseñanza secundaria y más allá, en la enseñanza de segundas lenguas, y en la construcción de pequeñas ontologías, merece más atención que la recibida hasta el momento.
Ningún esquema de representación único puede ser el mejor en todos los casos, ningún esquema único puede dar cuenta de las muchas formas en que piensa la mente humana. Las áreas temáticas, los objetivos de un curso, el nivel de instrucción, harán que algunos esquemas de representación sean más efectivos que otros. Una propuesta educativa debe considerar estos diferentes esquemas, y elegir el mejor para cada circunstancia.

Otros lenguajes de representación de conocimiento, tales como marcos o reglas, pueden jugar un papel en educación y evaluación. La adecuación de diferentes lenguajes de representación de conocimiento fue discutida en el capítulo sobre representación de conocimiento aplicada a la educación. Para reglas y marcos, un esquema de sublenguaje y representación de conocimiento puede desarrollarse a lo largo de las líneas expuestas en el capítulo sobre construcción de un sublenguaje y en el capítulo de esquemas de representación de conocimiento, en el documento original en inglés, donde fueron analizados varios esquemas ejemplo.

Una lección aprendida en la representación de conocimiento es mantenerse abierto a alternativas, estudiar diferentes lenguajes, considerar sus posibilidades, y evaluar su adecuación para el propósito actual. No es posible avanzar firmemente en la representación de conocimiento, y en la evaluación basada en representación de conocimiento, sin ser consciente de la relación esencial entre el tipo de conocimiento y el esquema elegido de representación. Ni los estudiantes ni los instructores deben convertirse en especialistas cognitivos; un conocimiento general debe ser suficiente, con cierta conciencia del lenguaje y de los dispositivos de representación de conocimiento requeridos por los diferentes tipos de conocimiento.

Se están desarrollando sublenguajes para la creación de ontologías; éstos pueden eventualmente evolucionar para ser usados en educación, o una versión simplificada de ellos. La creación de topic maps desde texto es otra alternativa potente y de gran alcance. Aunque sería un proyecto más ambicioso, todo el material de un curso puede tejerse en una red de topic maps; los recursos se pueden integrar en las estructuras de conocimiento, agregando confiabilidad, compartiendo e incorporando, con los beneficios propios de adherir a un estándar.

23.6. Síntesis

A lo largo de este trabajo, la idea de usar un sublenguaje y representación de conocimiento para la calificación de respuestas a texto libre demostró ser factible, al menos hasta donde un trabajo de investigación y el esfuerzo de un solo individuo puede alcanzar. Las herramientas y el prototipo desarrollado dan razones para afirmar que un sistema de producción puede construirse, y puede realizarse una prueba en clase, en un proyecto de un año llevado adelante por un pequeño grupo de instructores y desarrolladores.

La propuesta se presenta como una actividad de aprendizaje que involucra a los estudiantes e instructores en trabajo colaborativo. Los estudiantes construyen las estructuras bajo la guía de los profesores, en una actividad equivalente a la confección de resúmenes o el dibujo de mapas conceptuales. Las estructuras de conocimiento de referencia son construidas por los estudiantes, que comparten la preparación de la instancia de evaluación; esto resulta en un sistema justo, y reduce en gran manera el estrés del examen. La evaluación no requiere prácticamente tiempo de preparación a los instructores, la calificación es automática, y la realimentación a los estudiantes inmediata.

El uso de un sublenguaje bien diseñado contribuye a mejorar el uso del lenguaje natural y el desarrollo de habilidades de comunicación. Dibujar grafos escribiendo texto legible ofrece una especie de desafío similar a un juego de ordenador, lo cual la hace más atractiva frente a
23.7. Una última palabra

otras actividades de aprendizaje.

La combinación de un esquema de sublenguaje y representación de conocimiento emerge como una técnica promisoria para propósitos educativos, y merece más atención de la recibida hasta el momento.

23.7. Una última palabra

La aplicación de esta propuesta incentivará a los estudiantes hacia la escritura correcta, les enseñará a organizar su conocimiento, les proveerá una experiencia colaborativa, e integrará la evaluación en el aprendizaje, reduciendo drásticamente el estrés del examen.

Basado en el trabajo realizado en este proyecto, se considera un esfuerzo valioso continuar la investigación en el uso de sublenguajes y representación de conocimiento para la educación y la evaluación.
Part VIII.

Appendices
This part includes the following appendices:

- Bibliography,
- rules and example sentences for the various example sublanguages developed,
- a summary of the KLEAR project documentation.
- Glossary.
Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


A. The Example Sublanguages

The following sections describe the tagsets used in the example sublanguages, the rules of each, and the results of the tests. Spanish02 is not described; it was experimental, and discarded at some point.

A.1. Tagsets

The tagset for terminals and non terminals is shown in the following boxes. Terminals include the square brackets, used to delimit functional constituents, necessary to avoid a number of possible syntactic ambiguities. In non terminals, an ad-hoc grouping is defined to the same purpose within phrases, such as “adjective group” or “noun group”; this grouping provides a more fine grained control on the syntax of sentences.

<table>
<thead>
<tr>
<th>Terminales / Terminals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Np : nombre propio / proper noun</td>
</tr>
<tr>
<td>Nc : nombre común / common noun</td>
</tr>
<tr>
<td>Adj : adjetivo / adjective</td>
</tr>
<tr>
<td>Adv : adverbio / adverb</td>
</tr>
<tr>
<td>Vcop : verbo copulativo / linking verb</td>
</tr>
<tr>
<td>Vpred : verbo predicativo / predicative verb</td>
</tr>
<tr>
<td>Vintr : verbo intransitivo / intransitive verb</td>
</tr>
<tr>
<td>Vtran : verbo transitivo / transitive verb</td>
</tr>
<tr>
<td>PrepCI : preposición complemento indirecto ('a', 'para') / indirect object prepositions ('to', 'for')</td>
</tr>
<tr>
<td>Prep : preposiciones / prepositions</td>
</tr>
<tr>
<td>Det : determinante / determinant</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Terminales requeridos por la sintaxis / terminals required by syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0 -&gt; '['</td>
</tr>
<tr>
<td>P1 -&gt; ']'</td>
</tr>
</tbody>
</table>
A. The Example Sublanguages

<table>
<thead>
<tr>
<th>No terminales / non terminals</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 : oración / sentence</td>
</tr>
<tr>
<td>SN : sintagma nominal, GN grupo nominal / noun phrase, noun group</td>
</tr>
<tr>
<td>SAdj : sintagma adjetival, GAdj grupo adjetival / adjective phrase, adjective group</td>
</tr>
<tr>
<td>SAdv : sintagma adverbia, GAdv grupo adverbial / adverbial phrase, adverbial group</td>
</tr>
<tr>
<td>SPrep : sintagma preposicional / prepositional phrase</td>
</tr>
<tr>
<td>SV : sintagma verbal / verb phrase</td>
</tr>
<tr>
<td>SVtran : sintagma verbal transitivo / transitive verb phrase</td>
</tr>
<tr>
<td>SVintr : sintagma verbal intransitivo / intransitive verb phrase</td>
</tr>
<tr>
<td>SVcop : sintagma verbal copulativo / linking verb phrase</td>
</tr>
<tr>
<td>SVpred : sintagma verbal predicativo / predicative verb phrase</td>
</tr>
<tr>
<td>CN : complemento del nombre / noun complement</td>
</tr>
<tr>
<td>CD : complemento directo / direct object</td>
</tr>
<tr>
<td>CI : complemento indirecto / indirect object</td>
</tr>
<tr>
<td>CC : complemento circunstancial / circumstancial complement</td>
</tr>
<tr>
<td>Atr : atributo / attribute</td>
</tr>
</tbody>
</table>

The following tags are used to convert wordforms in a sentence into part of speech tags which substitute those wordforms, so that the parser acts on the sentence with this small set of 'PoS tag wordforms'; the syntactic tree obtained is reconverted so that the original wordforms appear in it. This helps decouple the lexicon from the parser.

<table>
<thead>
<tr>
<th>Etiquetas de sustitución / substitution tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>Np -&gt; 'np'</td>
</tr>
<tr>
<td>Nc -&gt; 'nc'</td>
</tr>
<tr>
<td>Adj -&gt; 'adj'</td>
</tr>
<tr>
<td>Adv -&gt; 'adv'</td>
</tr>
<tr>
<td>Vcop -&gt; 'vcop'</td>
</tr>
<tr>
<td>Vintr -&gt; 'vintr'</td>
</tr>
<tr>
<td>Vtran -&gt; 'vtran'</td>
</tr>
<tr>
<td>PrepCI -&gt; 'prepci'</td>
</tr>
<tr>
<td>Prep -&gt; 'prep'</td>
</tr>
<tr>
<td>Det -&gt; 'det'</td>
</tr>
</tbody>
</table>

A.2. Spanish01

The rules of production for this sublanguage are given in the following lines.
Sp01Rules.txt: rules of production for Spanish01

### O oración
O -> SN SVcop | SN SVtran | SN SVintr

### SPrep Sintagma preposicional
SPrep -> P0 GPreq P1
GPreq -> Prep Np
GPreq -> Prep GN | Prep GN CN | Prep GN CN CN
GPreq -> Prep Det GN | Prep Det GN CN | Prep Det GN CN CN

### SAdj Sintagma Adjetival
SAdj -> GAdj | GAdj SPrep
GAdj -> Adj | Adv Adj

### SAdv grupo adverbial
SAdv -> GAdv | GAdv SPrep
GAdv -> Adv | Adv Adv

### SN, sintagma Nominal
SN -> Np
SN -> Det GN | Det GN CN | Det GN CN CN
GN -> Nc | Nc GAdj | GAdj Nc | GAdj Nc GAdj
CN -> SPrep

### Atr, atributo, para sintagma verbal copulativo
Atr -> SAdj | GPreq | GN | GN CN | Det GN | Det GN CN

### CC, Complemento circunstancial
CC -> SPrep | P0 SN P1 | P0 SAdv P1
# [ de tarde ]; [ el domingo ]; [ ayer ], [ totalmente ]

### SVcop, sintagma verbal copulativo
# Atr con paréntesis permite OC sin ambigüedad
SVcop -> Vcop Atr | Vcop P0 Atr P1 | Vcop P0 Atr P1 CC

### CD, Complemento directo
CD -> P0 SN P1

### CI, complemento indirecto
CI -> P0 PrepCI SN P1 | P0 P1
# en SVtran CI superpone en formato con OC, por eso PrepCI
# CI vacío permite definir bien cuanto CI no está

### SVtran, sintagma verbal transitivo
SVtran -> GVtran CD | GVtran CD CI | GVtran CD CI CC | GVtran CD CI CC OC
SVtran -> GVtran CD OC | GVtran CD OC OC | GVtran CD CC OC OC
GVtran -> Vtran | Vtran GAdv

### SVintr sintagma verbal intransitivo
SVintr -> GVintr | GVintr CI
SVintr -> GVintr CI OC | GVintr CI OC OC | GVintr CI CC OC CC
SVintr -> GVintr CC | GVintr OC OC | GVintr CC CC CC
GVintr -> Vintr | Vintr GAdv
### A. The Example Sublanguages

The following set of sentences summarize parsing results: each line gives the number of wordforms in the sentence, the number of licensed syntactic trees, and the sentence itself. Sentences with no licensed tree, or with more than 1 licensed tree, are marked with *. A small group of sentences is provided for testing different constituents.

```plaintext
===> validate.py, testing with Sp01Lex.txt Sp01Rules.txt

<table>
<thead>
<tr>
<th>Wordforms</th>
<th>Licensed Trees</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1</td>
<td>la mesa [ de madera ] es nueva</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>la mesa [ de madera lustrada ] es nueva</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>la mesa [ de muy fina madera totalmente lustrada ] es nueva</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>la mesa [ de muy fina madera totalmente lustrada [ de roble ] ] es nueva</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>la casa [ de Pedro ] es blanca</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Juan está [ en la casa ]</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>Juan está [ en la casa [ de Pedro ] ]</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>Juan está [ en la casa [ de la vieja esquina rosada [ de Pedro ] ] ]</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>Juan está [ en la casa [ de madera [ de roble ] ] ]</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>Juan está [ en la casa [ de madera [ de roble ] ] [ de Pedro ] ]</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Juan es bueno</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Juan es muy bueno</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Juan es bueno [ en arquitectura ]</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>Juan es muy bueno [ en arquitectura ]</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>Juan es muy bueno [ en la vieja arquitectura [ de Madrid ] ]</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>el jarrón es blanco</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>el jarrón es totalmente blanco</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Juan caminó lentamente</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Juan caminó muy lentamente</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] ]</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Pedro es chino</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>el jarrón es chino</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>el muy viejo jarrón totalmente blanco es chino</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>el muy viejo jarrón totalmente blanco [ de muy fina porcelana china ] es nuevo</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Juan es [ estudiante ]</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>Juan es [ muy buen estudiante chino ]</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>Juan es [ un muy buen estudiante chino ]</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Juan es [ estudiante [ de arquitectura ] ]</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>Juan es [ muy buen estudiante chino [ de arquitectura ] ]</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>Juan es [ muy buen estudiante chino [ de arquitectura ] [ en Madrid ] ]</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>Juan es [ muy buen estudiante chino [ de la arquitectura [ de Madrid ] ] ]</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Juan es [ estudiante [ en Madrid ] ]</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Juan es [ viejo ]</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Juan es [ muy viejo ]</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Juan es [ de Madrid ]</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>Juan es [ de la ciudad [ de Madrid ] ]</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Juan pintó [ la casa ]</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Juan pintó totalmente [ la casa ]</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>Juan pintó [ la casa [ para Pedro ] ]</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>Juan pintó [ la casa [ de la esquina ] [ para Pedro ] ]</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>Juan pintó totalmente [ la casa [ de la esquina ] [ para la vieja madre china [ de Pedro ] ] ]</td>
</tr>
</tbody>
</table>
```
13 1 Juan pintó [ la casa ] [ para la madre [ de Pedro ] ] [ el domingo [ de tarde ] ]

23 1 Juan pintó totalmente [ la vieja casa [ de madera lustrada ] [ de la esquina ] ] [ para la vieja madre [ de Pedro ] ] [ el domingo [ en la tarde ] ]

17 1 Juan pintó totalmente [ la casa ] [ para la madre ] [ con la estudiante [ de arquitectura ] ] [ el domingo [ de tarde ] ]

8 1 Juan pintó [ la casa ] [ el domingo [ de tarde ] ]

12 1 Juan pintó [ la casa [ de madera [ de roble ] ] ] [ el domingo [ de tarde ] ]

15 1 Juan pintó totalmente [ la casa ] [ con la estudiante [ de arquitectura ] ] [ en una tarde ] [ el domingo ]

2 1 Juan caminó

3 1 Juan caminó lentamente

4 1 Juan caminó muy lentamente

5 1 Juan caminó [ hacia la casa ]

7 1 Juan caminó [ hacia la casa [ de madera ] ]

10 1 Juan caminó [ hacia la casa [ de madera [ de la esquina ] ] ]

15 1 Juan caminó muy lentamente [ hacia la nueva casa blanca [ de madera clara ] [ de la esquina ] ]

19 1 Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] [ de la esquina [ de Pedro ] ] [ ayer [ de tarde ] ]

16 1 Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] ] [ con Pedro ] [ ayer [ de tarde ] ]

19 1 Juan caminó muy lentemente [ hacia la casa [ de madera [ de roble ] ] ] [ de la madre ] [ con Pedro ] [ ayer [ de tarde ] ]

4 1 Juan corrió [ a Pedro ]

7 1 Juan corrió [ a Pedro ] [ de la casa ]

9 1 Juan corrió [ a Pedro ] [ de la casa [ el domingo ]]

14 1 Juan corrió [ a Pedro ] [ de la casa ] [ con un palo ] [ el domingo [ de tarde ]]


53 1 el gran perro totalmente negro [ de la buena madre [ de los amigos [ de Pedro ] ] ] corrió [ a el gato muy blanco [ de el médico viejo [ de el nuevo hospital psiquiátrico ] ] [ por la casa grande [ de buena madera [ de roble ] ] [ de la tradicional esquina rosada [ de la vieja calle [ de Madrid ] ] ] [ el domingo [ de tarde ] ]

A.3. Spanish03

The rules of production for this sublanguage are given in the following lines.
### Sp03Rules: rules of production for Spanish03

#### Oración

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 -&gt; SN SV</td>
<td></td>
</tr>
</tbody>
</table>

#### SV, sintagma verbal:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SV -&gt; SVcop</td>
<td>STran</td>
</tr>
</tbody>
</table>

#### SPrep, sintagma preposicional:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPrep -&gt; P0 GPrep P1</td>
<td>GPrep -&gt; Prep GN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPrep -&gt; Prep GN P0 GPrep P1</td>
<td>Prep Det GN P0 GPrep P1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPrep -&gt; Prep Det GN P0 GPrep P1 P0 GPrep P1</td>
<td></td>
</tr>
</tbody>
</table>

#### SAdv, sintagma adverbial:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAdv -&gt; GAdv</td>
<td>GAdv SPrep</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAdv -&gt; Adv</td>
<td>Adv Adv</td>
</tr>
</tbody>
</table>

# SAdv = [Cuant] Adv ; más específico, Cuant es un adverbio de cantidad

#### SAdj, sintagma adjetival (o adjetivo):

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAdj -&gt; GAdj</td>
<td>GAdj SPrep</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAdj -&gt; Adv</td>
<td>Adv Adv</td>
</tr>
</tbody>
</table>

#### SN, sintagma nominal:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN -&gt; Np</td>
<td>SN -&gt; Det GN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GN -&gt; Np</td>
<td>Nc</td>
</tr>
</tbody>
</table>

#### Atr, atributo, para sintagma verbal copulativo

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atr -&gt; GAdj</td>
<td>GN</td>
</tr>
</tbody>
</table>

#### CC, Complemento circumstancial

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC -&gt; SPrep</td>
<td>P0 SAdv P1</td>
</tr>
</tbody>
</table>

#### SVcop, sintagma verbal copulativo:

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVcop -&gt; Vcop Atr</td>
<td>Vcop Atr OC</td>
</tr>
</tbody>
</table>

#### CD, Complemento directo

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD -&gt; P0 SN P1</td>
<td>SPrep</td>
</tr>
</tbody>
</table>

#### CI, complemento indirecto

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CI -&gt; P0 PrepCI SN P1</td>
<td></td>
</tr>
</tbody>
</table>

# CI -> SPrep : superpone con CC, por eso PrepCI -> ’a’ | ’para’

#### STran, sintagma verbal transitivo

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STran -&gt; GVtran CD</td>
<td>GVtran CD CI</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STran -&gt; GVtran CD</td>
<td>GVtran CD OC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GVtran -&gt; Vtran</td>
<td>Vtran GAdv</td>
</tr>
</tbody>
</table>

#### Sintagma verbal intransitivo

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIntr -&gt; GVintr</td>
<td>GVintr CI</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIntr -&gt; GVintr CI OC</td>
<td>GVintr CI OC OC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIntr -&gt; GVintr CC</td>
<td>GVintr OC OC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GVintr -&gt; Vintr</td>
<td>Vintr GAdv</td>
</tr>
</tbody>
</table>
The following set of sentences summarize parsing results: each line gives the number of wordforms in the sentence, the number of licensed syntactic trees, and the sentence itself. Sentences with no licensed tree, or with more than 1 licensed tree, are marked with *. A small group of sentences is provided for testing different constituents.

```python
=> validate.py, testing with Sp01Lex.txt Sp03Rules.txt

### SPrep, sintagma preposicional
## en sujeto como complemento de nombre
6 1 la casa [de Pedro] es blanca
6 1 la mesa [de madera] es nueva
7 1 la mesa [de madera lustrada] es nueva
10 1 la mesa [de muy fina madera totalmente lustrada] es nueva
12 1 la mesa [de muy fina madera totalmente lustrada] [de Pedro] es nueva
14 0 *la mesa [de muy fina madera totalmente lustrada] [de roble] [de Pedro] es nueva
10 1 la mesa [de madera [de roble]] [de Pedro] es nueva

### SAdj, sintagma adjetival
## en sujeto de oración copulativa
6 1 el viejo jarrón blanco es chino
7 1 el muy viejo jarrón blanco es chino
8 1 el muy viejo jarrón totalmente blanco es chino
13 1 el muy viejo jarrón totalmente blanco [de la madre [de Juan]] es chino

### SAdv, sintagma adverbial
# como complemento de verbo
3 1 Juan caminó lentamente
4 1 Juan caminó muy lentamente

### SN, sintagma nominal
# como sujeto
3 1 Juan es chino
4 1 el jarrón es chino
8 1 el muy viejo jarrón totalmente blanco es chino
9 1 el jarrón [de muy fina porcelana china] es viejo
8 1 el jarrón [de porcelana] [de Juan] es viejo
8 1 la mesa [de madera [de roble]] es nueva
# como atributo en oración copulativa
4 1 Juan está [con Pedro]
5 1 Juan está [en la casa]
7 1 Juan está [en la casa] [con Pedro]
8 1 Juan está [en la casa [de la esquina]]
13 1 Juan está [en la casa [de la esquina [de la calle [de Madrid]]]]
16 1 Juan está [en la casa [de la esquina [de la calle [de la ciudad [de Madrid]]]]]
10 1 Juan está [en la casa [de la esquina]] [con Pedro]
18 1 Juan está [en la casa [de la esquina [de la calle [de la ciudad [de Madrid]]]]]
[con Pedro]

### SVcop, sintagma verbal copulativo
## con atributo nominal
3 1 Juan es médico
```
A. The Example Sublanguages

6 1 Juan es médico [de el hospital]
11 1 Juan es médico [de el hospital [de la ciudad [de Madrid]]]
5 1 Juan es muy buen médico
7 1 Juan es muy buen médico [de hospital]
8 1 Juan es un muy buen médico [de hospital]
10 1 Juan es un muy buen médico [de hospital] [en Madrid]
3 1 Pedro es estudiante
6 1 Pedro es muy buen estudiante chino
5 1 Pedro es estudiante [ de arquitectura ]
5 1 Pedro es estudiante [ en Madrid ]
11 1 Pedro es muy buen estudiante muy nuevo [ de arquitectura ] [ en Madrid ]
5 1 Pedro es estudiante [ de arquitectura ]
5 1 Pedro es estudiante [ en Madrid ]
12 1 Pedro es muy buen estudiante muy nuevo [ de la arquitectura [ de Madrid ] ]

## con atributo adjetival
3 1 Juan es bueno
4 1 Juan es muy bueno
6 1 Juan es muy bueno [como médico]
5 1 Juan es muy buen médico
7 1 Juan es muy buen médico [de hospital]

## con atributo preposicional
16 1 Juan está [en la casa [de la esquina [de la calle [de la ciudad [de Madrid]]]]]
5 1 Juan es [de la ciudad]
7 1 Juan es [de la ciudad [de Madrid]]
7 1 Juan está [en la casa] [con Pedro]
10 1 Juan está [en la casa [de la esquina]] [con Pedro]
22 1 Juan estuvo [en la casa [de la esquina [de la calle [de la ciudad [de Madrid]]]]]
[con Pedro] [ayer [en la tarde]]

### SVtran, sintagma verbal transitivo

## con CD
4 1 Juan pintó [la casa]
6 1 Juan pintó casi totalmente [la casa]
6 1 Juan pintó [la casa] [casi totalmente]

## con CD y CI
6 1 Juan pintó [la casa] [ para Pedro ]
9 1 Juan pintó [la casa [ de la esquina ] ] [ para Pedro ]
16 1 Juan pintó casi totalmente [la casa [ de la esquina ] ] [ para la vieja madre china [ de Pedro ] ]

## con CD, CI y CC
13 1 Juan pintó [la casa ] [ para la madre [ de Pedro ] ] [ el domingo [ de tarde ] ]
23 1 Juan pintó totalmente [la vieja casa [ de madera lustrada ] [ de la esquina ] ] [ para la vieja madre [ de Pedro ] ] [ el domingo [ en la tarde ] ]
17 1 Juan pintó totalmente [la casa ] [ para la madre ] [ con la estudiante [ de arquitectura ] ] [ el domingo [ de tarde ] ]

## prueba con CD y CC
8 1 Juan pintó [la casa ] [ el domingo [ de tarde ] ]
A.4. Spanish04

12 1 Juan pintó [ la casa [ de madera [ de roble ] ] ] [ el domingo [ de tarde ] ]
15 0 *Juan pintó totalmente [la casa] [con la estudiante [de arquitectura]] [ en una tarde] [el domingo]

### SVintr, Sintagma verbal intransitivo
## sin complementos
2 1 Juan caminó
3 1 Juan caminó lentamente
4 1 Juan caminó muy lentamente
5 1 Juan caminó [ hacia la casa ]

## con C C
7 1 Juan caminó [ hacia la casa [ de madera ] ]
10 1 Juan caminó [ hacia la casa [ de madera [ de la esquina ] ] ]

## con GAdv, C C
15 1 Juan caminó muy lentamente [ hacia la nueva casa blanca [ de madera clara ] [ de la esquina ] ]
19 1 Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] [ de la esquina [ de Pedro ] ] ] [ ayer [ de tarde ] ]
16 0 *Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] ] [ con Pedro ] [ ayer [ de tarde ] ]
19 0 *Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] [ de la madre ] ] [ con Pedro ] [ ayer [ de tarde ] ]

## prueba con CI y C C
4 1 Juan corrió [ a Pedro ]
7 1 Juan corrió [ a Pedro ] [ de la casa ]
9 1 Juan corrió [ a Pedro ] [ de la casa ] [ el domingo ]
14 0 *Juan corrió [ a Pedro ] [ de la casa ] [ con un palo ] [ el domingo [ de tarde ] ]
12 1 el perro negro corrió [a el gato blanco] [el domingo [de tarde]]
20 1 el perro totalmente negro corrió [a el gato muy blanco] [por la casa [de la esquina]] [el domingo [de tarde]]

### Oraciones largas
36 1 el muy buen estudiante chino [ de la vieja arquitectura [de Madrid]] caminó muy lentamente [hacia la casa [de madera lustrada [de roble claro]] [de la vieja madre [de Pedro]]] [ayer [en la tarde clara [de domingo]]]
50 1 el gran perro totalmente negro [de la buena madre [de Pedro]] corrió [a el gato muy blanco [de el médico viejo [de el nuevo hospital psiquiátrico]]] [por la casa grande [de buena madera [de roble]] [de la tradicional esquina rosada [de la vieja calle [de Madrid]]]] [el domingo [de tarde]]

A.4. Spanish04

This sublanguage is a modification of Spanish03 trying to avoid differentiating transitive and intransitive verbs, treating both of them as predicative verbs. The rules of production for this sublanguage are given in the following lines.
A. The Example Sublanguages

<table>
<thead>
<tr>
<th>Sp04Rules: rules of production for Spanish04</th>
</tr>
</thead>
<tbody>
<tr>
<td>### Oración</td>
</tr>
<tr>
<td>0 -&gt; SN SV</td>
</tr>
</tbody>
</table>

| ### Sintagma verbal: |
| SV -> SVcop | SVpred |

| ### Sintagma preposicional: |
| & these rules admit nesting of prep phrases: |
| SPrep -> P0 GPrep P1 |
| GPrep -> Prep GN | Prep Det GN |
| GPrep -> Prep GN P0 GPrep P1 | Prep Det GN P0 GPrep P1 |
| GPrep -> Prep Det GN P0 GPrep P1 P0 GPrep P1 |

| ### Sintagma adverbial: |
| SAdv -> GAdv | GAdv SPrep |
| GAdv -> Adv | Adv Adv |
| # SAdv = [Cuant] Adv ; más específico, Cuant is an adverbio de cantidad |

| ### Sintagma adjetival (o adjetivo): |
| SAdj -> GAdj | GAdj SPrep |
| GAdj -> Adv | Adv Adv |

| ### Sintagma nominal: |
| SN -> Np |
| SN -> Det GN | Det GN SPrep | Det GN SPrep SPrep |
| GN -> Np | Nc | GAdj Nc | Nc GAdj |

| # Atributo, para sintagma verbal copulativo |
| Atr -> GAdj | GN | Det GN SPrep | SPrep |

| # Complemento circunstancial |
| CC -> SPrep | P0 SAdv P1 | P0 SN P1 |

| ### Sintagma verbal copulativo: |
| SVcop -> Vcop Atr | Vcop Atr OC | Vcop Atr OC OC |

| # Complemento directo |
| CD -> P0 SN P1 | SPrep |

| # Complemento indirecto |
| CI -> P0 PrepCI SN P1 |

| ### Sintagma verbal predicativo |
| Vpred -> Vtran | Vintr |
| GVpred -> Vpred | Vpred GAdv |
| # transitivo |
| SVpred -> GVpred CD | GVpred CD CI | GVpred CD CI CC | GVpred CD CI CC OC |
| SVpred -> GVpred CD | GVpred CD OC | GVpred CD CC |
| # intransitivo |
| SVpred -> GVpred | GVpred CI |
| SVpred -> Gpred CI CC | GVpred CI OC OC |
| SVpred -> GVpred CC | GVpred OC OC |
The following set of sentences summarize parsing results: each line gives the number of wordforms in the sentence, the number of licensed syntactic trees, and the sentence itself. Sentences with no licensed tree, or with more than 1 licensed tree, are marked with *. A small group of sentences is provided for testing different constituents.

```python
===> validate.py, testing with Sp01Lex.txt Sp04Rules.txt
pals árbs oración
### SPrep, sintagma preposicional
## en sujeto como complemento de nombre
6 1 la casa [de Pedro] es blanca
6 1 la mesa [de madera] es nueva
7 1 la mesa [de madera lustrada] es nueva
10 1 la mesa [de muy fina madera totalmente lustrada] es nueva
12 1 la mesa [de muy fina madera totalmente lustrada] [de Pedro] es nueva
14 0 *la mesa [de muy fina madera totalmente lustrada] [de roble] [de Pedro] es nueva
10 1 la mesa [de madera [de roble]] [de Pedro] es nueva
### SAdj, sintagma adjetival
## en sujeto de oración copulativa
6 1 el viejo jarrón blanco es chino
7 1 el muy viejo jarrón blanco es chino
8 1 el muy viejo jarrón totalmente blanco es chino
13 1 el muy viejo jarrón totalmente blanco [de la madre [de Juan]] es chino
### SAdv, sintagma adverbial
## como complemento de verbo
3 1 Juan caminó lentamente
4 1 Juan caminó muy lentamente
### SN, sintagma nominal
## como sujeto
3 1 Juan es chino
4 1 el jarrón es chino
8 1 el muy viejo jarrón totalmente blanco es chino
9 1 el jarrón [de muy fina porcelana china] es viejo
8 1 el jarrón [de porcelana] [de Juan] es viejo
8 1 la mesa [de madera [de roble]] es nueva
## como atributo en oración copulativa
4 1 Juan está [con Pedro]
5 1 Juan está [en la casa]
7 1 Juan está [en la casa] [con Pedro]
8 1 Juan está [en la casa [de la esquina]]
13 1 Juan está [en la casa [de la esquina] [de la calle [de Madrid]]]
16 1 Juan está [en la casa [de la esquina] [de la calle [de la ciudad [de Madrid]]]]
10 1 Juan está [en la casa [de la esquina]] [con Pedro]
18 1 Juan está [en la casa [de la esquina] [de la calle [de la ciudad [de Madrid]]]] [con Pedro]
### SVcop, sintagma verbal copulativo
```
A. The Example Sublanguages

## con atributo nominal
3 1 Juan es médico
6 1 Juan es médico [de el hospital]
11 1 Juan es médico [de el hospital [de la ciudad [de Madrid]]]
5 1 Juan es muy buen médico
7 1 Juan es muy buen médico [de hospital]
8 1 Juan es un muy buen médico [de hospital]
10 1 Juan es un muy buen médico [de hospital] [en Madrid]
3 1 Pedro es estudiante
6 1 Pedro es muy buen estudiante chino
5 1 Pedro es estudiante [ de arquitectura ]
5 1 Pedro es estudiante [ en Madrid ]
11 1 Pedro es muy buen estudiante muy nuevo [ de arquitectura ] [ en Madrid ]
5 1 Pedro es estudiante [ de arquitectura ]
5 1 Pedro es estudiante [ en Madrid ]
12 1 Pedro es muy buen estudiante muy nuevo [ de la arquitectura [ de Madrid ] ]

## con atributo adjetival
3 1 Juan es bueno
4 1 Juan es muy bueno
6 1 Juan es muy bueno [como médico]
5 1 Juan es muy buen médico
7 1 Juan es muy buen médico [de hospital]
7 1 Juan está cómodamente sentado [en el sofá]

## con atributo preposicional
16 1 Juan está [en la casa [de la esquina [de la calle [de la ciudad [de Madrid]]]]]
5 1 Juan es [de la ciudad]
7 1 Juan es [de la ciudad [de Madrid]]
7 1 Juan está [en la casa] [con Pedro]
10 1 Juan está [en la casa [de la esquina]] [con Pedro]
22 1 Juan estuvo [en la casa [de la esquina [de la calle [de la ciudad [de Madrid ]]]]]
[con Pedro] [ayer [en la tarde]]

### SVtran, sintagma verbal transitivo
## con CD
4 2 *Juan pintó [la casa]
6 2 *Juan pintó casi totalmente [la casa]
6 2 *Juan pintó [la casa][casi totalmente]

## con CD y CI
6 1 Juan pintó [ la casa ] [ para Pedro ]
9 1 Juan pintó [ la casa [ de la esquina ] ] [ para Pedro ]
16 1 Juan pintó casi totalmente [ la casa [ de la esquina ] ] [ para la vieja madre china [ de Pedro ] ]

## con CD, CI y CC
13 1 Juan pintó [ la casa ] [ para la madre [ de Pedro ] ] [ el domingo [ de tarde ] ]
23 1 Juan pintó totalmente [ la vieja casa [ de madera lustrada ] [ de la esquina ] ] [ para la vieja madre [ de Pedro ] ] [ el domingo [ en la tarde ] ]
17 0 *Juan pintó totalmente [ la casa ] [ para la madre ] [ con la estudiante [ de arquitectura ] ] [ el domingo [ de tarde ] ]
## prueba con CD y CC
8 2 *Juan pintó [ la casa ] [ el domingo [ de tarde ] ]
12 2 *Juan pintó [ la casa [ de madera [ de roble ] ] ] [ el domingo [ de tarde ] ]
15 0 *Juan pintó totalmente [la casa] [con la estudiante [de arquitectura]] [en una tarde] [el domingo]

### SVintr, Sintagma verbal intransitivo
#### sin complementos
2 1 Juan caminó
3 1 Juan caminó lentamente
4 1 Juan caminó muy lentamente

#### con CC
5 2 *Juan caminó [ hacia la casa ]
7 2 *Juan caminó [ hacia la casa [ de madera ] ]
10 2 *Juan caminó [ hacia la casa [ de madera [ de la esquina ] ]]

#### con GAdv, CC
15 2 *Juan caminó muy lentamente [ hacia la nueva casa blanca [ de madera clara ] [ de la esquina ] ]
19 2 *Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] [ de la esquina [ de Pedro ] ] [ ayer [ de tarde ] ]]
16 1 Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] [ con Pedro ] [ ayer [ de tarde ] ]]
19 1 Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] [ de la madre ] [ con Pedro ] [ ayer [ de tarde ] ]]

#### prueba con CI y CC
4 1 Juan corrió [ a Pedro ]
7 0 *Juan corrió [ a Pedro ] [ de la casa ]
9 1 Juan corrió [ a Pedro ] [ de la casa ] [ el domingo ]
14 0 *Juan corrió [ a Pedro ] [ de la casa ] [ con un palo ] [ el domingo [ de tarde ] ]
12 0 *el perro negro corrió [a el gato blanco] [el domingo [de tarde]]
20 1 el perro totalmente negro corrió [a el gato muy blanco] [por la casa [de la esquina]]
[el domingo [de tarde]]

### Oraciones largas
36 2 *el muy buen estudiante chino [ de la vieja arquitectura [de Madrid]] caminó muy lentamente [hacia la casa [de madera lustrada [de roble claro]] [de la vieja madre [de Pedro]]] [ayer [en la tarde clara [de domingo]]]
50 1 el gran perro totalmente negro [de la buena madre [de Pedro]] corrió [a el gato muy blanco [de el médico viejo [de el nuevo hospital psiquiátrico]]] [por la casa grande [de buena madera [de roble]] [de la tradicional esquina rosada [de la vieja calle [de Madrid]]]] [el domingo [de tarde]]

A.5. Spanish05

This sublanguage is a modification of Spanish04 trying to avoid differentiating transitive and intransitive verbs, treating both of them as predicative verbs. The rules of production for this sublanguage are given in the following lines.
A. The Example Sublanguages

<table>
<thead>
<tr>
<th>Sp05Rules: rules of production for Spanish05</th>
</tr>
</thead>
<tbody>
<tr>
<td>### Oración</td>
</tr>
<tr>
<td>O -&gt; SN SV</td>
</tr>
<tr>
<td>### Sintagma verbal:</td>
</tr>
<tr>
<td>SV -&gt; SVcop</td>
</tr>
<tr>
<td>### Sintagma preposicional:</td>
</tr>
<tr>
<td>## these rules admit nesting of prep phrases:</td>
</tr>
<tr>
<td>SPrep -&gt; P0 GPrep P1</td>
</tr>
<tr>
<td>GPrep -&gt; Prep GN</td>
</tr>
<tr>
<td>GPrep -&gt; Prep GN P0 GPrep P1</td>
</tr>
<tr>
<td>GPrep -&gt; Prep Det GN P0 GPrep P1 P0 GPrep P1</td>
</tr>
<tr>
<td>### Sintagma adverbial:</td>
</tr>
<tr>
<td>SAdv -&gt; GAdv</td>
</tr>
<tr>
<td>GAdv -&gt; Adv</td>
</tr>
<tr>
<td># SAdv = [Cuant] Adv ; más específico, Cuant es un adverbio de cantidad</td>
</tr>
<tr>
<td>### Sintagma adjetival (o adjetivo):</td>
</tr>
<tr>
<td>SAdj -&gt; GAdj</td>
</tr>
<tr>
<td>GAdj -&gt; Adj</td>
</tr>
<tr>
<td>### Sintagma nominal:</td>
</tr>
<tr>
<td>SN -&gt; Np</td>
</tr>
<tr>
<td>SN -&gt; Det GN</td>
</tr>
<tr>
<td>GN -&gt; Np</td>
</tr>
<tr>
<td>## Atributo, para sintagma verbal copulativo</td>
</tr>
<tr>
<td>Atr -&gt; GAdj</td>
</tr>
<tr>
<td>## Complemento circunstancial</td>
</tr>
<tr>
<td>CC -&gt; SPrep</td>
</tr>
<tr>
<td>### Sintagma verbal copulativo:</td>
</tr>
<tr>
<td>SVcop -&gt; Vcop Atr</td>
</tr>
<tr>
<td>### Sintagma verbal predicativo</td>
</tr>
<tr>
<td>GVpred -&gt; Vpred</td>
</tr>
<tr>
<td>GVpred -&gt; GVpred</td>
</tr>
</tbody>
</table>

The following set of sentences summarize parsing results: each line gives the number of wordforms in the sentence, the number of licensed syntactic trees, and the sentence itself. Sentences with no licensed tree, or with more than 1 licensed tree, are marked with *. A small group of sentences is provided for testing different constituents.

```python
==> validate.py, testing with Sp05Lex.txt Sp05Rules.txt
(83, 808, 0, 1)
pals árbs oración
```
### SPrep, sintagma preposicional
## en sujeto como complemento de nombre
6 1 la casa [de Pedro] es blanca
6 1 la mesa [de madera] es nueva
7 1 la mesa [de madera lustrada] es nueva
10 1 la mesa [de muy fina madera totalmente lustrada] es nueva
12 1 la mesa [de muy fina madera totalmente lustrada] [de Pedro] es nueva
14 0 *la mesa [de muy fina madera totalmente lustrada] [de roble] [de Pedro] es nueva
10 1 la mesa [de madera [de roble]] [de Pedro] es nueva

### SAdj, sintagma adjetival
## en sujeto de oración copulativa
6 1 el viejo jarrón blanco es chino
7 1 el muy viejo jarrón blanco es chino
8 1 el muy viejo jarrón totalmente blanco es chino
13 1 el muy viejo jarrón totalmente blanco [de la madre [de Juan]] es chino

### SAdv, sintagma adverbial
## como complemento de verbo
3 1 Juan caminó lentamente
4 1 Juan caminó muy lentamente

### SN, sintagma nominal
## como sujeto
3 1 Juan es chino
4 1 el jarrón es chino
8 1 el muy viejo jarrón totalmente blanco es chino
9 1 el jarrón [de muy fina porcelana china] es viejo
8 1 el jarrón [de porcelana] [de Juan] es viejo
8 1 la mesa [de madera [de roble]] es nueva
## como atributo en oración copulativa
4 1 Juan está [con Pedro]
5 1 Juan está [en la casa]
7 1 Juan está [en la casa] [con Pedro]
8 1 Juan está [en la casa [de la esquina]]
13 1 Juan está [en la casa [de la esquina [de la calle [de Madrid]]]]
16 1 Juan está [en la casa [de la esquina [de la calle [de la ciudad [de Madrid]]]]]
10 1 Juan está [en la casa [de la esquina]] [con Pedro]
18 1 Juan está [en la casa [de la esquina [de la calle [de la ciudad [de Madrid]]]]] [con Pedro]

### SVcop, sintagma verbal copulativo
## con atributo nominal
3 1 Juan es médico
6 1 Juan es médico [de el hospital]
11 1 Juan es médico [de el hospital [de la ciudad [de Madrid]]]
5 1 Juan es muy buen médico
7 1 Juan es muy buen médico [de hospital]
8 1 Juan es un muy buen médico [de hospital]
10 1 Juan es un muy buen médico [de hospital] [en Madrid]
3 1 Pedro es estudiante
6 1 Pedro es muy buen estudiante chino
A. The Example Sublanguages

5 1 Pedro es estudiante [de arquitectura]
5 1 Pedro es estudiante [en Madrid]
11 1 Pedro es muy buen estudiante muy nuevo [de arquitectura] [en Madrid]
5 1 Pedro es estudiante [de arquitectura]
5 1 Pedro es estudiante [en Madrid]
12 1 Pedro es muy buen estudiante muy nuevo [de la arquitectura] [de Madrid]

## con atributo adjetival
3 1 Juan es bueno
4 1 Juan es muy bueno
6 1 Juan es muy bueno [como médico]
5 1 Juan es muy buen médico
7 1 Juan es muy buen médico [de hospital]
7 1 Juan está cómodamente sentado [en el sofá]

## con atributo preposicional
16 1 Juan está [en la casa [de la esquina [de la calle [de la ciudad [de Madrid]]]]]
5 1 Juan es [de la ciudad]
7 1 Juan es [de la ciudad [de Madrid]]
7 1 Juan está [en la casa] [con Pedro]
10 1 Juan está [en la casa [de la esquina]] [con Pedro]
22 1 Juan estuvo [en la casa [de la esquina [de la calle [de la ciudad [de Madrid]]]]]
[con Pedro] [ayer [en la tarde]]

### SVtran, sintagma verbal transitivo
## con CD
4 1 Juan pintó [la casa]
6 1 Juan pintó casi totalmente [la casa]
6 1 Juan pintó [la casa][casi totalmente]

## con CD y CI
6 1 Juan pintó [la casa] [para Pedro]
9 1 Juan pintó [la casa [de la esquina]] [para Pedro]
16 1 Juan pintó casi totalmente [la casa [de la esquina]] [para la vieja madre china [de Pedro]]

## con CD, CI y CC
13 1 Juan pintó [la casa] [para la madre [de Pedro]] [el domingo [de tarde]]
23 1 Juan pintó totalmente [la vieja casa [de madera lustrada] [de la esquina]] [para la vieja madre [de Pedro]] [el domingo [en la tarde]]
17 1 Juan pintó totalmente [la casa] [para la madre] [con la estudiante [de arquitectura]] [el domingo [de tarde]]

## prueba con CD y CC
8 1 Juan pintó [la casa] [el domingo [de tarde]]
12 1 Juan pintó [la casa [de madera [de roble]]] [el domingo [de tarde]]
15 1 Juan pintó totalmente [la casa] [con la estudiante [de arquitectura]] [en una tarde] [el domingo]

### SVintr, sintagma verbal intransitivo
## sin complementos
2 1 Juan caminó
3 1 Juan caminó lentamente
4 1 Juan caminó muy lentamente

## con CC
5 1 Juan caminó [ hacia la casa ]
7 1 Juan caminó [ hacia la casa [ de madera ] ]
10 1 Juan caminó [ hacia la casa [ de madera [ de la esquina ] ] ]

## con GAdv, CC
15 1 Juan caminó muy lentamente [ hacia la nueva casa blanca [ de madera clara ] [ de la esquina ] ]
19 1 Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] [ de la esquina [ de Pedro ] ] [ ayer [ de tarde ] ] ]
16 1 Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] [ con Pedro ] [ ayer [ de tarde ] ]]
19 1 Juan caminó muy lentamente [ hacia la casa [ de madera [ de roble ] ] [ de la madre ] [ con Pedro ] [ ayer [ de tarde ] ]]

## prueba con CI y CC
4 1 Juan corrió [ a Pedro ]
7 1 Juan corrió [ a Pedro ] [ de la casa ]
11 1 Juan es médico [ de el hospital [ de la ciudad [ de Madrid ] ] ]
5 1 Juan es muy buen médico
7 1 Juan es muy buen médico [ de hospital ]
8 1 Juan es un muy buen médico [ de hospital ]
10 1 Juan es un muy buen médico [ de hospital ] [ en Madrid ]
9 1 Juan corrió [ a Pedro ] [ de la casa ] [ el domingo ]
14 1 Juan corrió [ a Pedro ] [ de la casa ] [ con un palo ] [ el domingo [ de tarde ] ]
12 1 el perro negro corrió [ a el gato blanco ] [ el domingo [ de tarde ] ]
20 1 el perro totalmente negro corrió [ a el gato muy blanco ] [ por la casa [ de la esquina ] [ el domingo [ de tarde ] ]]

### Oraciones largas
50 1 el gran perro totalmente negro [ de la buena madre [ de Pedro ] ] corrió [ a el gato muy blanco [ de el médico viejo [ de el nuevo hospital psiquiátrico ] ] ] [ por la casa grande [ de buena madera [ de roble ] ] [ de la tradicional esquina rosada [ de la vieja calle [ de Madrid ] ] ] ] [ el domingo [ de tarde ] ]

A.6. English05

This is an English version of Spanis05, built along the same lines. The tagset and rules of production for this sublanguage are given in the following boxes.
## Tags for English05

### terminals

- **Np**: noun, proper
- **Nc**: noun, common
- **Adj**: adjective
- **Adv**: adverb
- **Vlk**: verb, linking
- **Vintr**: verb, intransitive
- **Vtran**: verb, transitive
- **PrepIO**: indirect object prepositions ('to', 'for')
- **Prep**: preposition (except 'to' and 'for')
- **Det**: determiner

```
P0 -> '['
P1 -> ']'
```

### non terminals

- **S**: sentence, start symbol
- **NP**: nominal phrase, NG nominal group
- **AdjP**: adjective phrase
- **AdjG**: adjective group
- **AdvP**: adverb phrase
- **AdvG**: adverb group
- **PrepP**: prepositional phrase
- **VP**: verb phrase
- **VactP**: active verb phrase
- **VtranP**: transitive verb phrase
- **VPintr**: intransitive verb phrase
- **VlnkP**: sintagma verbal copulativo
- **Attr**: attribute of linking verb
### Sentence, declarative
S -> NP VlnkP | NP VactP

### Sentence, imperative
S -> VlnkP | VactP

### PrepP, prepositional phrase:
## these rules admit nesting of prep phrases:
PrepP -> P0 PrepG P1
PrepG -> Prep NG | Prep Det NG
PrepG -> Prep NG P0 PrepG P1 | Prep Det NG P0 PrepG P1
PrepG -> Prep Det NG P0 PrepG P1 P0 PrepG P1

### AdvP, adverb phrase:
AdvP -> AdvG | AdvG PrepP
AdvG -> Adv | Adv Adv

### AdjP, adjective phrase:
AdjG -> Adj | AdjG Adj
AdjG -> Adj Adj | AdvG Adj Adj | Adj AdvG Adj | AdvG Adj Adj

### NP, noun phrase:
NP -> Np
NP -> Det NG | Det NG PrepP | Det NG PrepP PrepP
NG -> Np | Nc | AdjG Nc
## Attribute, for linking verb phrase
Attr -> AdjG | Det NG | PrepP # | Det NG PrepP
Attr -> AdjG | Det NG | PrepP
## verb complements
CC -> PrepP | P0 NP P1 | P0 AdvP P1

### VlnkP, linking verb phrase:
VlnkP -> Vlnk Attr | Vlnk Attr CC | Vlnk Attr CC OC | Vlnk Attr CC CC

### VactP, active verb phrase
VactG -> Vact | Vact AdvG
VactP -> VactG | VactG CC | VactG CC OC | VactG CC CC | VactG CC OC CC | VactG CC CC CC

The following set of sentences summarize parsing results: each line gives the number of wordforms in the sentence, the number of licensed syntactic trees, and the sentence itself. Sentences with no licensed tree, or with more than 1 licensed tree, are marked with *. A small group of sentences is provided for testing different constituents.

```bash
$pals árbs oración
5 1 Jack walked [to the house]
8 1 Jack walked [to the house [in the corner]]
11 1 Jack walked [to the house [in the corner [of the street]]]
13 1 Jack walked [to the nice white house [in the corner [of the street]]]
16 1 Jack walked [to the house [in the corner [of the street [of the city [of London]]]]]
5 1 the man kissed [the maiden]
6 1 the tattered man kissed [the maiden]
```
A. The Example Sublanguages

7 1 the very tattered man kissed [the maiden]
8 1 the very tattered torn man kissed [the maiden]
8 1 the tattered very torn man kissed [the maiden]
13 1 the very tattered very torn man kissed [the very forlorn very shy maiden]
3 1 Jack walked slowly
4 1 Jack walked very slowly
7 1 Jack walked very slowly [to the house]
9 1 the man [from the old house] kissed [the maiden]
10 1 the man [from the old white house] kissed [the maiden]
23 1 the very tattered very torn man [from the very old totally white house] kissed
 [the very forlorn very shy maiden] [in the morning]
24 1 the very tattered very torn man [from the very old totally white house] kissed
 [the very forlorn very shy maiden] [yesterday [in the morning]]
30 1 the very tattered very torn man [from the very old totally white house] kissed
 [the very forlorn very shy maiden] [yesterday [in the morning]] [at the corner [of the street]]
4 1 Jack is a physician
7 1 Jack is a physician [in the hospital]
12 1 Jack is a physician [from the hospital [in the city [of London]]]
16 1 Jack is a very old very nice physician [from the hospital [in the city [of London]]]
4 1 the man is tattered
5 1 the man is very tattered
4 1 the man is tattered
7 1 the man is very tattered very torn
3 1 Jack is seated
7 1 Jack is comfortably seated [on the sofa]
8 1 Jack is very comfortably seated [on the sofa]
5 1 Jack is [in the house]
16 1 Jack is [in the house [at the corner [of the street [of the city [of London]]]]]
20 1 Jack is [in the house [at the corner [of the street [of the city [of London]]]] [with the tattered man]
25 1 Jack is [in the house [at the corner [of the street [of the city [of London]]]] [with the tattered man] [near the totally white house]
31 1 Jack is [in the house [at the corner [of the street [of the city [of London]]]] [with the tattered man] [near the very nice totally white house [of the forlorn maiden]]
]
10 1 Jack was [in the house] [with the forlorn maiden] [yesterday]
4 1 Jack built [a house]
8 1 Jack built [a totally white very nice house]
7 1 Jack built [a house] [for the maiden]
9 1 Jack built very swiftly [a house] [for the maiden]
9 1 Jack built [a house] [for the forlorn maiden] [yesterday]
23 1 Jack built [a house [in the corner [of the street [of the city [of London]]]] [for the forlorn maiden] [yesterday [in the morning]]]
27 1 Jack built [a house [in the corner [of the street [of the city [of London]]]] [for the forlorn maiden] [yesterday [in the morning]] [with the tattered man]
5 1 Jack built [a house] [yesterday]
9 1 Jack built very swiftly [a house] [in the morning]
10 1 Jack built very swiftly [a house] [yesterday [in the morning]]
14 1 Jack built very swiftly [a house] [yesterday [in the morning]] [with the tattered man]
2 1 Jack walked
3 1 Jack walked slowly
4 1 Jack walked very slowly
5 1 Jack walked [towards the house]
10 1 Jack walked [towards the house [of wood] [at the corner]]
14 1 Jack walked [towards the house [of wood] [at the corner]] [yesterday [in the morning]]
18 1 Jack walked [with the tattered man] [towards the house [of wood] [at the corner]] [yesterday [in the morning]]
20 1 Jack walked very slowly [with the tattered man] [towards the house [of wood] [at the corner]] [yesterday [in the morning]]
B. The KLEAR Project Documentation

This appendix shows the module hierarchy of the software developed for experimentation, testing and demonstration. The complete documentation is available at the web page of this thesis [González-Barbome, 2012].

The KLEAR project (Knowledge and Language Environment for Assessment and Research) intends to provide tools and methods to create knowledge structures from texts written in a Domain Specific Sublanguage (DSS).

Package klear

The KLEAR project module.

The KLEAR project (Knowledge and Language Environment for Assessment and Research) intends to provide tools and methods to create knowledge structures from texts written in a Domain Specific Sublanguage (DSS).

Modules

- initeng05: Initialization module for the KLEAR English 05 sublanguage.
- kldemo: A module for the demonstration of the KLEAR project.
- kleditor: Defines and tests class KLEditor, the KLEAR text editor.
- klgraph: A test module for the KlGraph class, the KLEAR graph builder.
- lexicon: Data and functions to compare wordlists and create lexicons for English DSSs (Domain Specific Sublanguages).
  - be1500: Basic English combined wordlist, used by Simple Wikipedia.
  - bnc: Frequency wordlists from the British National Corpus.
  - bncfreq: Frequency wordlists from the British National Corpus.
  - closedcats: Function words for the English language, in closed categories.
  - coca: Corpus of Contemporary American English (COCA) wordlists.
  - explexicons: Functions and Classes for experimental lexicons.
  - gsl: General Service List of English Words (GSL) from several sources.
  - gslbauman: Frequency list for the GSL Bauman wordlist.
  - gsklickins: GSL Dickins wordlist.
  - gsltextutor: GSL Lextutor wordlist.
  - kevinat: Several of Kevin Atkinson’s very large wordlists.
B. The KLEAR Project Documentation

- **lextests**: Tests for lexicon classes.
- **longman**: Longman Communication 3000 wordlist.
- **qrywnet**: Experimental functions to query Wordnet.
- **voa**: Classes for Voice of America (VoA) Special English wordlists.
- **voawords**: Classes for Voice of America (VoA) Special English wordlists.
- **wordlist**: Lexicon management and support for several wordlists.
- **semgraph**: Classes and libraries to build, compare and display semantic graphs.
  - **deptree**: Functions to build dependency trees and add to graphs.
  - **deptree0**: Functions to build dependency trees and add to graphs.
  - **deptreematch**: A module to recognize in a graph the dependency tree from a sentence.
  - **deptreetst1**: A test module for dependency tree construction.
  - **deptreetst4**: A module to test recognition in a graph of dependency tree from sentence.
  - **py2dot**: Auxiliary functions to generate a Graphviz dot file for a graph.
  - **smgraph**: Functions and Classes for the generation of a semantic graph.
  - **smtree**: Classes and functions to build dependency trees.
  - **tagseten**: A module with tagsets for English and display control variables.
  - **tagsetsp**: A module with tagsets for Spanish.
- **sublang**: Classes and auxiliary functions to process sentences against a context free grammar.
  - **lexicon**: A module for lexicons.
  - **sentences**: Module to build a syntactic tree from a sentence using a context free grammar.
  - **utils**: Utility functions for general use.
  - **validate**: Classes and functions to validate a set of sentences against a grammar.
- **utils**: KLEAR utility functions for general use.
C. Glossary

The following explanations state the meaning of each term in the context of this work. No attempt at formal definitions has been made, but to explain as clearly as possible what each term refers to.

**Affix**: a morpheme which may be added to a stem to complement the stem meaning. [Chapter 5]

**AI-complete problem**: Artificial Intelligence complete problem, a problem which is as difficult as solving the general problem of building a computer as intelligent as a human being. [Chapter 4]

**Assessment**: an activity which determines to what extent a candidate has achieved the assessment criteria. [Chapter 2]

**Assessment criteria**: statements which define what the learner must do during the assessment instance in order to demonstrate that a learning objective has been achieved. Assessment determines to what extent a candidate has achieved the assessment criteria. [Chapter 2]

**Assessment objective**: a single unit of knowledge, skills or understanding that a test is designed to assess in a candidate. This single unit of knowledge, skills or understanding defines a learning objective. Learning objectives and assessment objectives are usually part of a program of study. [Chapter 2]

**Assessment specification**: a detailed description of methods, processes, tasks and criteria used to assess a learning objective. [Chapter 2]

**Automatic Assessment**: assessment in which the marks are given by a computer system, with no intervention of the teacher. [Chapter 2]

**Bigram**: a 2-gram (see N-gram). [Chapter 5]

**Closed questions**: a type of question or task in which the range of possible responses or outcomes the student can give are limited. Assessment by closed questions is called objective assessment. Closed questions are apt for automatic marking, because the given answers can be compared to a correct answer. [Chapter 2]

**Closed word classes**: lexical categories which exhibit little or null variation in time. For most western languages, closed word classes are determiners, pronouns, prepositions, and conjunctions. Closed word classes are essentially function words, with little meaning themselves, but used to establish grammatical relations among words. [Chapter 5]

**Computer Assisted Assessment**: assessment in which the computer system is a tool which helps the teacher at her request and under her control; in Computer Assisted Assessment the teacher can see and modify the markings according to her own criteria. [Chapter 2]
C. Glossary

**Concept map:** a graphical tool for representing knowledge by drawing concepts as labeled circles or boxes, and relationships among concepts as labeled links connecting the corresponding circles or boxes. [Chapter 9]

**Context dependent synonyms:** the collection of words that can be considered as reasonable substitutes of the ones mostly accepted within a particular field of knowledge. [Chapter 2]

**Controlled language:** see Controlled Natural Language. [Chapter 4]

**Controlled natural language:** an engineered subset of natural languages where lexicon and grammar have been restricted to reduce the ambiguity and complexity of full natural languages. The short form term “controlled language” has been superseded by “controlled natural language”, to make explicit that they are subsets of natural languages, and not related to languages associated with programming or logics, which are indeed controlled languages, not always “natural”. [Chapter 4]

**Controlled vocabulary:** a list of terms with an unambiguous interpretation of each. Ambiguity is dealt with by not accepting more than one meaning for each term, or distinguishing different meaning for a term in some clear way. [Chapter 9]

**Coverage:** a measure of the capacity of a word list to account for the occurrences of words in a text. Coverage is determined by summing all the times a word in the list appears in the text, and dividing by the length of the text in words. Coverage is usually expressed as a percentage. [Chapter 6]

**Data:** the qualitative or quantitative values of attributes or variables, expressed as symbols. These symbols may be characters, numbers, images, sound, or any output which can be perceived. Symbol: something that by association or convention stands for or represents something else. [Chapter 9]

**Domain** (in a Knowledge Representation context): an area of concern or interest. A domain may also be called a field, in the sense of a topic or subject of academic or educational interest. [Chapter 9]

**EAssessment:** the use of Information and Communications Technologies for assessment. EAssessment comprises the presentation of assessment activities and the recording of responses. According to JISC, eAssessment is an end to end process for learners, tutors, learning establishments, awarding bodies and regulators, and also for the public general. [Chapter 2]

**Edit distance:** the minimum number of character substitutions required to convert a word into another. [Chapter 2]

**Essay:** a short literary composition dealing with a subject, usually presenting the personal views of an author. [Chapter 2]

**Essay question:** a type of question which calls for a written answer of some specified length, usually a short essay. [Chapter 2]

**Evaluate:** ascertain or fix the value or worth of something; it also means judge carefully, appraise something. See also validate. [Chapter 10]
**Expressiveness** (of a language): see Expressive power (of a language). [Chapter 6]

**Expressive power** (of a language): the scope of ideas that can be expressed in that language. A language will be more expressive if it can express more ideas, or if it can express the same ideas without recurring to unwieldy, far fetched constructions. [Chapter 6]

**Formal language**: a language defined by a generative grammar. [Chapter 5]

**Free text answers**: candidates’ responses consisting of pieces of text of some length. Words or short phrases are not considered free text answers; these short pieces of text can be marked automatically by much simpler techniques such as pattern matching. Free text answers are long phrases or sentences, paragraphs, or a short essay. [Chapter 2]

**Generative grammar**: a description of a language in terms of rules which define all the “well formed” strings in the language, i.e. the strings that comply with the rules. A language defined by a generative grammar is called a formal language. [Chapter 5]

**Glossary**: a list of terms peculiar to a field of knowledge with their definitions. [Chapter 6]

**Headword**: see Lemma. [Chapter 6]

**Inference**: the act or process of deriving logical conclusions or making logical judgements from premises or propositions accepted to be true. Inferred conclusions emerge from incidental evidence or from previous conclusions rather than from direct observation; it is the result of thinking. [Chapter 9]

**Infections** (of a word): variations of a headword, such as “dogs” from “dog”, or “makes, made” from “make”. Also, inflected forms of the word. [Chapter 6]

**Information extraction**: the identification of entities, relations and events in free text, by the automatic processing of extraction rules reflecting a user’s interest. [Chapter 4]

**Information retrieval**: activity which consists of finding objects relevant to a user’s query, almost always expressed in words. The limits of statistically based methods seem to have been reached, and language oriented methods are presently being tried. [Chapter 4]

**Information**: symbols with a meaning. Data becomes information when some sense is given to the symbols of data. A person’s name, age, sex and photograph may be collected as characters, numbers and an image; these characters, numbers and image become information when they are identified as name, age, sex and photograph, and associated into a personal record. [Chapter 9]

**Knowledge based system**: a system that can assimilate new information and adjust its behavior accordingly. [Chapter 9]

**Knowledge base**: the symbolic representation of a collection of propositions related to some purpose (the intentional stance of the agent). [Chapter 9]

**Knowledge Representation**: the use of formal symbols to represent the propositions considered true by some agent. [Chapter 9]. Knowledge Representation may be a field of
C. Glossary

study, a category of entities conceived to express or record knowledge, one of those individual entities (an instance of the class of knowledge representations. A knowledge representation instance is expressed in a knowledge representation language.

Knowledge: the collection of information intended to be useful for some purpose. When a teacher looks at the personal record of a new student and memorizes the information contained therein, she has acquired some knowledge of the student. When this teacher finally meets the student, and appropriately addresses her by name an title (Mr or Ms), the teacher is using her knowledge of the student. In formal contexts, the relation between an agent (person or machine) and the idea expressed in a proposition. [Chapter 9]

KR: Knowledge Representation.

Language: any system of signals used for communication. The term language may be applied to the formal expressions of mathematics, logics or computer systems, the natural languages we speak and write in, and to any system of signals used for communication. [Chapter 4]

Learning objective: a single unit of knowledge, skills or understanding. [Chapter 2]

Lemma: one form of the lexeme selected as representative of the whole lexeme. The lemma is also called the citation form, canonical form or dictionary form of the lexeme. Word “find” is the lemma of lexeme “find, finds, found, finding”. In some contexts, the lemma is called headword, to differentiate it from the wordforms, which are all the forms in a lexeme, including its headword. [Chapter 5]

Lemmatization: the operation of determining the headword corresponding to a wordform. [Chapter 6]. The task of determining the lemma corresponding to a given form of a lexeme. [Chapter 5]

Lexeme: the set of different forms of a word with the same meaning is called a lexeme; “find, finds, found, finding” are the forms in the English lexeme “find”. [Chapter 5]

Lexical categories: the different roles a word may play in a sentence, or the part of speech (PoS) for that word. [Chapter 5]

Lexicon: the collection of terms in a language, or the collection of terms used in a profession or subject; also, vocabulary. [Chapter 6]

Logical inference: the process by which a new proposition comes out as a logical conclusion of some previous propositions. [Chapter 9]

Morpheme: the minimum language unit with a meaning. A word may be formed by several morphemes. [Chapter 4]

Morphological parsing: decomposing a word into its morphemes. [Chapter 5]

Natural Language Processing (NLP): a field of knowledge that tries to understand human language in such a way that a machine can extract meaningful information from texts or speech. It comprises several subfields: morphology studies the decomposition of words
in meaningful pieces, syntax studies how one word relates to others in the structure of a sentence, semantics tries to determine the knowledge contained in a piece of text or speech, pragmatics deals with the somewhat elusive subject of determining the goals and intentions of the speaker, discourse studies how sentences relate to one another in a long piece of text or speech. [Chapter 5]

**N-gram:** a subsequence of N items within a sequence, where N is an integer. [Chapter 5]

**N-gram model:** a model which attempts to predict the next item in a sequence, given some subsequence. N-gram models are useful in PoS tagging, when a single wordform may belong in several part of speech categories: neighbour wordforms may help decide which part of speech the wordform is playing in a particular sequence of wordforms. [Chapter 5]

**NLP:** Natural Language Processing.

**Objective assessment:** assessment by closed questions. [Chapter 2]

**Ontology:** an explicit and formal specification of concepts and relations in a domain of knowledge. [Chapter 4]

**Open questions:** a type of question or task with no predetermined response or outcome. Assessment by open questions is called subjective assessment. Open questions are difficult to mark automatically. [Chapter 2]

**Open word classes:** lexical categories which constantly acquire new members while others fade away. For most western languages, open word classes are nouns, verbs, adjectives and adverbs. [Chapter 5]

**Part of speech (PoS):** lexical category. [Chapter 5]. Part of speech (PoS): the class of all the words that may occur in the same places in a sentence, or accomplish similar functions in context. [Chapter 6]

**Part of Speech tagging:** the operation of recognizing the part of speech of each word in a sentence. [Chapter 6]

**PoS tagging:** Part of Speech tagging. [Chapter 6]

**PoS:** Part of Speech.

**Proof of concept:** the realization of a certain method or idea to demonstrate its feasibility. A proof of concept is frequently small, not complete, and not apt for real use. A prototype may be constructed as a proof of concept; this prototype need not be a first version of the product, its only purpose is to show its working possible. [Chapter 16]

**Proposition:** a declarative sentence that can be true or false. [Chapter 9]

**Reasoning:** the process of drawing conclusions, inferences, or judgements, based on previous knowledge. Reasoning is a way to go from some ideas to other ideas through thinking, in a process considered valid or legitimate to establish such relations. One of these valid processes of thinking is the use of Logic. [Chapter 9]
C. Glossary

**Representation:** an arrangement of symbols which stand in place of some domain. [Chapter 9]

**Restricted language:** short for restricted natural language. [Chapter 4]

**Restricted natural language:** a sublanguage or a controlled natural language. In this work, a restricted sublanguage will always be a subset of a natural language, and will exhibit a behavior similar to a natural language. [Chapter 4]

**Semantic network:** a graphic notation for representing knowledge in the form of nodes and arcs. [Chapter 9]

**Sentence segmentation:** the task of dividing a text into sentences. [Chapter 5]

**Stemming:** the task of stripping affixes from a word leaving its stem. Stemming goes beyond variations in a word form to obtain the main meaning of the word. [Chapter 5]

**Stem:** the main morpheme of a word, also called the base or root form of the word. [Chapter 5]

**Subjective assessment:** assessment by open questions. [Chapter 2]

**Sublanguage:** any proper subset of expressions in a language which exhibits some systematic behavior, i.e. it is used like a language. Sublanguages tend to appear spontaneously in communities sharing a common interest and interchanging information on some specific subjects [Chapter Kittredge:2003:SublangsContrLangs]. Sublanguages have no strict rules as controlled natural languages have. [Chapter 4]

**Sublanguage for knowledge representation:** a sublanguage compiled for knowledge extraction apt to build a knowledge representation instance in some knowledge representation language. A sublanguage for knowledge representation lies somewhere in the middle of sublanguages and controlled natural languages: though it is more formal than a sublanguage as formerly defined, since its vocabulary and grammar rules are determined, the users are allowed to modify them; there is not a formal specification to follow, and the scope of application may be very restricted, such as a bunch of learning units in a certain subject. [Chapter 4]

**Synset:** a data structure used to describe a single concept. A synset is associated with several words usually recognized as referring to this concept; these words are considered synonyms, since they can be exchanged in a context with no alteration of meaning. The name synset suggests “set of synonyms”, which is the form a concept is characterized in the Wordnet. [Chapter 6]

**Tagset:** the collection of labels or tags used to refer to parts of speech. [Chapter 6]

**Thesaurus:** a list of terms and their synonyms. [Chapter 9]

**Token:** a string of symbols of an alphabet, formed according to certain rules. Though a token may be considered a word when dealing with text, it is a wider concept, since it may be just a sequence of characters, as in some formal languages. [Chapter 9]

**Tokenization:** word segmentation. [Chapter 5]
**Trigram:** a 3-gram (See N-gram). [Chapter 5]

**UML profile:** a customization of standard UML for a particular domain or platform defined in accordance with UML extension mechanisms. [Chapter 9]

**Understanding:** knowledge plus the capacity to synthesize new knowledge, as when a child applies her knowledge of arithmetic operations and time to determine that a day is $12\times60\times60=43200$ seconds long [Chapter Bellinger2004:DataInfoKnowWisdom]. [Chapter 9]

**Validate:** to establish the soundness of something, corroborate or confirm. Validation can be conceived as a part of an evaluation: if the proposal can be validated, i.e. its soundness can be proved, then the value of the proposal is increased. See also *evaluate*. [Chapter 16]

**Vocabulary:** see Lexicon. [Chapter 6]

**Weak ontology:** an ontology not so rigorous as to allow machine processing and inference. [Chapter 10]

**Word:** a unit of language that native speakers can identify. A word may designate a word that appears as an entry in a dictionary as well as all its variations. The word used as an entry in a dictionary is called a headword, lemma or citation form of the word. [Chapter 6]

**Wordbook:** a reference book containing a list of words with their definitions. [Chapter 6]

**Wordform:** the headword and any of its variations is called a word form, usually written as wordform. [Chapter 6]

**Word segmentation:** the process of dividing a sentence into its compounding wordforms. [Chapter 5]