

Knowledge Graph Theory and Structural Parsing

Lei Zhang

2002

Ph.D. thesis
University of Twente



Twente University Press

Also available in print:

<http://www.tup.utwente.nl/catalogue/book/index.jsp?isbn=9036518350>

**KNOWLEDGE GRAPH THEORY
AND
STRUCTURAL PARSING**



Twente University Press

Publisher:

Twente University Press, P.O. Box 217, 7500 AE Enschede, The Netherlands,
www.tup.utwente.nl

Print: Océ Facility Services, Enschede

© L. Zhang, Enschede 2002

No part of this work may be reproduced by print,
photocopy or any other means without the permission
in writing from the publisher.

ISBN 9036518350

KNOWLEDGE GRAPH THEORY
AND
STRUCTURAL PARSING

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. F. A. van Vught,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op woensdag 20 november 2002 te 16:45 uur

door

Lei Zhang

geboren op 10 maart 1964
te Xi'an, China

Die proefschrift is goedgekeurd door de promotoren

prof. dr. C. Hoede

en

prof. dr. X. Li

Acknowledgements

In the first place I express my gratitude to Prof. Cornelis Hoede for inviting me to work under his inspiring supervision. I have learned a lot from him, not only from a scientific point of view. With him I have had many extremely interesting and stimulating discussions. He encouraged and helped me in everything I did for my thesis, by his patience, understanding and caring. He flooded my email box to help me continue my research while I was in China and we could not work face to face temporarily. All those emails, discussions and talks are unforgettable and were always a great motivation. It has been most pleasant to work with him.

I consider myself very lucky to have had Prof. Xueliang Li as my cosupervisor, who gave me the opportunity to visit to Twente University in the Netherlands. He has been of great help in introducing interesting areas of knowledge graph theory to me. This thesis benefited much from his detailed criticism and advice.

Working with other people speeds up the research process. I like to thank all my colleagues for giving me a pleasant atmosphere to work. Especially I like to thank Dr. Broersma, Carla and Dini for all their help. Also I like to thank the other Ph.D. students at Twente University, namely Xiaodong Liu and Shenggui Zhang for making my time in Twente so enjoyable.

Finally, very special thanks go to my family and all my friends. In particular I thank my father, my mother, my husband and my two sisters. I am indebted to my parents

for their great support and encouragement. Most especially, I would like to thank my son for his sensibilities and love. It would have been impossible to have written this thesis without their unconditional support.

October 2002, Enschede

Lei Zhang

Preface

This thesis makes a contribution to a theory of knowledge representation by means of graphs. The theory belongs to the broad spectrum of semantic networks. In the eighties two related theories developed. One was started by Sowa, who published a book on *conceptual graphs* in 1984. Concepts are represented by labeled vertices connected by labeled arcs, that are also represented by vertices, the so-called total graph form in the terminology of Harary [Harary, 1972]. The types of the arcs express relationships between concepts, like AGENT or INSTRUMENT. The other theory started in 1982 and was called a theory of *knowledge graphs*. It was developed by Hoede and Stokman, who wanted to extract knowledge from medical and sociological texts, in order to obtain expert systems. In first instance only three types of relationships were distinguished, of which the causal relationship was the most important one.

Their first PhD-student was Bakker [Bakker, 1987], who, in his thesis, developed an information system, that included a so-called path algebra, to obtain implied relationships. The second thesis, by de Vries [de Vries, 1989], deals mainly with the problem of extracting causal relationships from a text. In a third thesis Smit [Smit, 1991] investigated robustness and consistency of extracted knowledge graphs. In the beginning of the project de Vries Robbé [de Vries Robbé, 1987] participated as well, but then started a rather large program of his own that led to the development of the medical expert system MEDES. The knowledge graphs in that system had 18 types of

relationships.

The increase in the number of types of relationships, very much in line with what can be seen in semantic networks, is due to the fact that many sentences of a text have to be left unprocessed if one considers only three types of relationships. The information in these sentences may, however, be considered to be too valuable to delete.

This was one of the reasons that after the focus had been on the structuring of knowledge, leading to the first three theses, the knowledge graph project was continued, focusing on the representation of knowledge in general. Now the problem of the ontology of knowledge graphs came forward. Arbitrary linguistic sentences should be representable by knowledge graphs. This led to considerable extension of the number of types of relationships and to the introduction of so-called frames. The thesis of Willems [Willems, 1993] was titled “Chemistry of Language” and started off the rather ambitious project of representing language by knowledge graphs. In principle every word should have a *word graph*, a sentence should be represented by a *sentence graph*. In fact, man is considered to have in mind a *mind graph*, a huge knowledge graph, representing both structured impressions from the “outer” world and his “inner” world. The thesis of van den Berg [Berg, 1993] was titled “Logic and Knowledge Graphs: one of two kinds” and deals with the problem of representing logical systems in terms of knowledge graphs. This work can be seen as an extension of the work of Peirce [Peirce, 1885] on *existential graphs*, that also stands at the basis of Sowa’s theory of conceptual graphs.

The construction of word graphs was investigated by Hoede and students of the University of Twente. Several hundreds of the words frequently used in English were represented within the knowledge graph formalism. The lexicon of word graphs is a prerequisite for any further investigation of language by means of graphs. By coincidence Li [Li, 1991] wrote a thesis in Twente on the purely mathematical subject of “Transformation Graphs”. Back in China he proposed to start a joint project in which two students, Liu and Zhang, would study Chinese by means of knowledge graphs. The interesting point here is that English and Chinese are significantly different as languages. If the paradigm that language can be expressed by knowledge graphs is to be defended, then also specific features of Chinese should be representable within the theory. Liu wrote a thesis that focuses on these specific

features as well as on other problems, like the extraction of causal relationships, which was done before, for English, by de Vries [de Vries, 1989].

In this thesis the focus is on the extraction of sentence graphs, both for English and for Chinese sentences. A prerequisite for this is a lexicon of word graphs. Three papers were dedicated to this. Hoede and Li [Hoede & Li, 1996] wrote a paper on a first set of words; verbs, nouns and prepositions. Hoede and Liu [Hoede & Liu, 1998] wrote a paper on a second set of words; adverbs, adjectives and Chinese classifiers or quantity words. Hoede and Zhang [Hoede & Zhang, 2001a] wrote a paper on a third set of words; the logic words, which is part of this thesis, see Chapter 4. In all three papers both Chinese and English words are considered. The contents of the first two papers are summarized in appendixes of this thesis. Mapping a sentence on a sentence graph, which is called *structural parsing*, constitutes the main theme of this thesis. A paper by Hoede and Zhang [Hoede & Zhang, 2001b] contains a shortened version of Chapter 5. Concepts involved in this thesis are e.g. *semantic word graph*, and *syntactic word graph*, *utterance path* and *chunk*. Utterance paths, studied in Chapter 6 and *partial structural parsing*, involving chunks and studied in Chapter 7, can be seen as first extensions of the developed theory in theoretical direction respectively applied direction.

The main *results* and *conclusions* coming forward from our research are the following.

i) Although there are many ontologies for knowledge representation, till now no one can be called universal, because it can replace the others. This is why *knowledge graph theory* was put forward and has been developed gradually. Chapter 3 focuses on the knowledge graph ontology and makes comparisons with a few other well-known ontologies. As knowledge is expressed by language the theory should be able to represent *any* language. The focus was therefore, in the beginning, on the representation of Chinese. Very specific aspects of Chinese turned out to be representable, see also Liu [Liu, 2002]. Also from our research we may conclude that the representation by means of knowledge graphs seems indeed independent of the language considered.

ii) We propose how to express words by *word graphs* both semantically and syntactically in Chapter 4. In the knowledge graph project the focus was first on the

semantic word graphs only, so that we obtained an extension of the theory. We argue that the structural parsing developed in Chapter 5 of this thesis could be used both in English and in Chinese. For mapping a sentence on a sentence graph, both syntactic and semantic information is needed. As one of the goals is to develop translation systems, with the main steps: structural parsing, transformation of the sentence graph and uttering the sentence graph in the target language, an important result is that chunks can be used to develop computer programs for structural parsing. Given a sentence graph there are usually several ways how such a graph can be brought under words, i.e. can be *uttered*. Languages differ in the way the words occurring in the sentence are ordered. Chapter 6 studies the problem of determining rules for uttering a sentence graph both in English and in Chinese. The main conclusion from these three chapters is that indeed a translation system can be developed from knowledge graph theory.

iii) *Applications* of knowledge graph theory are a challenge, especially in NLP. Based on the theory developed in this thesis, Chapter 7 develops a method for carrying out Information Extraction (IE). This was a third major item, that came along in a later stage of our research. Again the importance of considering chunks of sentences, and sentence graphs, should be mentioned. Here too the idea of structural parsing turned out to be fruitful.

Contents

<i>ACKNOWLEDGEMENTS</i>	<i>V</i>
<i>PREFACE</i>	<i>VII</i>
<i>CONTENTS</i>	<i>XI</i>
<i>CHAPTER 1 INTRODUCTION</i>	<i>1</i>
1.1 NATURAL LANGUAGE PROCESSING	2
1.2 A SEMANTIC MODEL OF NATURAL LANGUAGE PROCESSING	2
1.3 OUTLINE OF THIS THESIS	4
<i>CHAPTER 2 NATURAL LANGUAGE PARSING</i>	<i>7</i>
2.1 INTRODUCTION	7
2.2 DEVELOPMENTS IN PARSING	8
2.3 ASPECTS OF PARSING	11
2.3.1 Top-down parsing	11
2.3.2 Bottom-up parsing	13
2.3.3 Search techniques.....	15
2.4 PARSING METHODS	15
2.4.1 Traditional methods	15
2.4.2 Parsing with knowledge graphs	17

CHAPTER 3	<i>THEORY OF KNOWLEDGE GRAPHS</i>	19
3.1	FORMAL DESCRIPTION OF KNOWLEDGE GRAPHS	19
3.1.1	Concept	20
3.1.2	Basic relations	22
3.1.3	Relations in general	26
3.2	ONTOLOGICAL ASPECTS	27
3.2.1	Aristotle, Kant and Peirce	28
3.2.2	Logic	30
3.3	SEMANTICS IN NATURAL LANGUAGE PROCESSING	32
3.3.1	Fillmore's case grammar	33
3.3.2	Expressing semantics with knowledge graphs	34
3.3.3	Structure is meaning	36
3.3.4	Elimination of ambiguity in natural language	37
3.3.5	A limited set of relation types	38
3.4	CONCLUSION	39
CHAPTER 4	<i>WORD GRAPHS: THE THIRD SET</i>	41
4.1	INTRODUCTION	41
4.2	CLASSIFICATION OF WORDS	42
4.3	LOGIC WORDS AND THEIR CLASSIFICATION	43
4.3.1	Classification criteria	44
4.3.2	Classification of logic words of the first kind	50
4.3.3	Classification of logic words of the second kind	53
4.4	WORD GRAPHS FOR LOGIC WORDS	59
4.4.1	Proposition operators	59
4.4.2	Modal logic operators	60
4.4.3	Quantification	61
4.4.4	Logic words based on set comparison	62
4.4.5	Logic words referring to space and time	63
4.4.6	Logic words due to mental processes	65
4.4.7	Words used in other logics	66
4.4.8	Words linking sentences	67
4.5	CONCLUSION	67
CHAPTER 5	<i>STRUCTURAL PARSING</i>	69
5.1	INTRODUCTION	69

5.2 SYNTACTIC AND SEMANTIC WORD GRAPHS	70
5.2.1 Definitions of syntactic and semantic word graphs	70
5.2.2 Word types for Chinese and English.....	72
5.2.3 Syntactic word graphs for word types	77
5.3 GRAMMARS FOR CHINESE AND ENGLISH	82
5.4 STRUCTURAL PARSING	87
5.4.1 A traditional parsing approach	87
5.4.2 Utterance paths and chunks	88
5.4.3 Chunk indicators	90
5.4.4 Examples of structural parsing.....	92
5.5 CONCLUSION.....	106
CHAPTER 6 UTTERANCE PATHS	107
6.1 INTRODUCTION	107
6.2 UTTERANCE PATHS AND GENERATIVE GRAMMAR	110
6.3 UTTERING AN EXTENDED EXAMPLE SENTENCE GRAPH.....	113
6.4 UTTERING A SENTENCE GRAPH WITH REFERENCE WORDS	116
6.5 UTTERING GRAPHS CONTAINING FRAMES.....	117
6.6 UTTERING QUANTIFICATION.....	121
6.6.1 All, any, each and every	122
6.6.2 Uttering the word “dou1”	125
6.7 UTTERING AND GRAMMAR.....	133
6.7.1 An introductory example.....	134
6.7.2 Uttering rules from production rules.....	137
6.7.2.1 Rules involving word types only	138
6.7.2.2 Rules involving phrases.....	140
6.7.3 Uttering paths for the extended example	141
CHAPTER 7 INFORMATION EXTRACTION.....	143
7.1 INTRODUCTION	143
7.2 THE STATE OF THE ART OF IE	144
7.3 OVERVIEW OF THE APPROACH.....	148
7.4 DESCRIPTION OF KG-EXTRACTION.....	150
7.4.1 Partial structural parsing	150
7.4.2 An example of representing patterns with knowledge graphs: KG-Structure.....	151

7.4.3 Named entity recognition.....	152
7.4.4 Automatic pattern acquisition	153
7.4.5 Inference and merging	154
7.4.6 Generating templates	154
7.4.7 A worked out example	154
7.4.8 Chunk graphs for the example	163
7.4.9 Discussion	179
BIBLIOGRAPHY	183
APPENDIX I WORD GRAPHS: THE FIRST SET	191
I.1 INTRODUCTION	191
I.2 WORD GRAPHS FOR SOME SIMPLE PREPOSITIONS	192
I.3 WORD GRAPHS FOR OTHER PREPOSITIONS	194
I.4 WORD GRAPHS FOR VERBS AND NOUNS	196
APPENDIX II WORD GRAPHS: THE SECOND SET	199
II.1 INTRODUCTION	199
II.2 ADWORDS	201
II.2.1 Adjectives	201
II.2.1.1 The FPAR-adwords.....	203
II.2.1.2 The PAR-adwords	204
II.2.1.3 The CAU-adwords.....	206
II.2.1.4 The ALI-adwords.....	208
II.2.2 Adverbs.....	208
II.2.3 Classifiers in the Chinese Language	210
II.2.3.1 FPAR-classifiers	210
II.2.3.2 Other classifiers	211
INDEX	213
SUMMARY	215
CURRICULUM VITAE.....	217

Chapter 1

Introduction

For more than half a century, *Artificial Intelligence* (AI) has gradually got attention of more and more scholars, and has become an interdisciplinary and frontal science subject increasingly. With the development of the software for and the hardware implementation of computers, a computer already can store much information and carry out fast information processing. During the last decades, AI makes further applications to more and more fields, see [Rich, 1983, Graham, 1979].

Knowledge representation is a central topic in AI. Whether problem solving, or task describing, or expressing experience knowledge, or inferring and decision making, all of these are based on knowledge. Therefore, the research on knowledge representation propels the information age to change and develop from the elementary stage, mainly with data processing, to the high level stage, mainly with knowledge processing. It has important influence in fields like pattern recognition, natural language understanding, information processing, machine study, robotics, automatic theorem proving, automatic programming, expert systems, etc.

Although now there are many methods for knowledge representation, such as production rules, logic, semantic network, frame, script, its problems are not solved completely yet. Thus, to explore new methods for knowledge representation is still

one of the important subjects in AI, see [Rich, 1983, Graham, 1979].

Fortunately, *knowledge graphs*, as a new method, expand the knowledge representation methods. It establishes a semantic explanation model for human perception and information processing, based on philosophy and psychology.

1.1 Natural Language Processing

Generally, natural language (used by humans) is the most direct method and the symbol system most in use to express human ideas and pass on information. There is a gap between formal languages (used by a computer) and natural languages. Communication between computers and humans is only possible when much research is aimed at bridging this gap. The approach to bridge the gap is often named *natural language processing (NLP)*, *natural language understanding*, or *computational linguistics* [Allen, 1987, Harris, 1985].

Naturally, describing and modeling natural language is the base for the development of natural language understanding, and it determines the research process and the direction in the field of natural language understanding.

Moreover, today, information on INTERNET is growing unimaginably. This requires an intelligent information system, to search for information automatically, but also to filter, refine, and translate information, on a high level of understanding. The processing of these high level understandings must be and can only be based on semantics.

Knowledge graphs, as a kind of representation for NLP, points out a new way for natural language describing and modeling, and also makes a big step forward to the semantic understanding of “know it and know why”.

1.2 A Semantic Model of Natural Language Processing

A *concept* is an important component of human thought, and is the thinking unit that refers to objective things and their peculiar properties. The formation of a concept is a procedure that has the direction “from special to general”. Considering various objects that are “special” cases, one determines a “general” set of properties that form the

components of the concept. In the other direction an object can only be described by the name of the concept if these general features are present in the object.

With the mind's operation of forming concepts, the meaning of words and phrases has realized. The essentials of the meaning of a word are determined by the perception of reality, which belongs to both the category of thought and the category of language. Therefore, concept forms have a correspondence with the meaning of a word. The key of language information processing is to handle the meaning of a word. For instance, a person establishes the concept of some objective thing step by step, during he grows up, through practice and study. When he meets a certain concept in language, where a concept is expressed by a word, he will think of every aspect that is related to this concept.

For instance, when we meet the word "apple", we can associate the information related to its shape, color and taste, etc. Certainly, on different occasions or for different persons, a difference in level and depth to understand the same concept is possible. When we say "shoes", "boots" and "socks", we will associate "wear the thing on foot" as the common feature of these three. The common point of "shoes" and "boots" is "be used for walking"; the common feature of "boots" and "socks" is "be tube-shaped". By connecting in the mind, these three are considered together. This can be described as *thinking is linking somethings*. So, the course of human cognition is to establish "connection" for some concepts already in the brain, in order to form a network of related concepts. These smaller networks form a larger information network via connections.

Knowledge graph theory is based on the procedure of natural language processing that is assumed to be made by mankind, as mentioned in the above example. So, it is a kind of appropriate pattern of semantic understanding. It expresses a concept by a word graph, which is operational through the connection of word graphs. As the result of operating, it generates the information network of a greater semantic piece, the sentence graph. Sentence graphs form the network of larger semantic information again through *joining sentences*. Word graph formation (expressing the subjective meaning of a concept stored in a person's brain) and connection operation (carrying out thought by a person's brain), finally, lead to a larger semantic information graph, after carrying out the joint operation continuously.

1.3 Outline of This Thesis

In this thesis we study knowledge graph theory and its applications, especially in natural language processing. On the one hand, we present a new class of words that are used in natural language, and which are called *logic words*. On the other hand, we propose a new kind of NLP technology, which is called *structural parsing*. At the same time, this new method is further applied in extracting information from texts.

In Chapter 2 the outline of parsing is given, and some special problems on parsing are discussed. Furthermore we are concerned with its applications in natural language processing.

In Chapter 3 the basic theory of knowledge graphs is outlined from the point of view of ontology. In contrast with logic, it is claimed that knowledge graph theory is original, general and valid in representing knowledge. Also it is posited that knowledge graphs are more general and more original than conceptual graphs, due to the fact that the number of its relation types is very limited.

In Chapter 4 we propose the concept of logic word and classify logic words into groups in terms of semantics. A start is made with the building of the lexicon of logic words in terms of knowledge graphs.

In Chapter 5 structural parsing, which is based on the theory of knowledge graphs, is introduced. Under consideration of the semantic and syntactic features of natural language, both *semantic and syntactic word graphs* are formed. Grammar rules are derived from the syntactic word graphs. Due to the distinctions between Chinese and English, the grammar rules are given for the Chinese version and the English version of syntactic word graphs respectively. By traditional parsing a parse tree can then be given for a sentence, which can be used to map the sentence on a sentence graph. This is called structural parsing. The relationship with *utterance paths* is discussed. As a result, *chunk indicators* are proposed to guide structural parsing.

In Chapter 6 the problem of *uttering* a sentence graph, bringing a sentence graph under words, is discussed. The order of uttering words determines an utterance path. The rules for utterance paths are investigated.

In Chapter 7 we apply structural parsing to information extraction. We propose a

multiple level structure based on knowledge graphs for describing template information. As a result, the relationships with the 10 functionalities mentioned by Hobbs [MUC-5, 1993] are discussed.

In the summary, the direction of further research is focused upon. One of the goals is an automatic information extraction system, based on knowledge graphs, for Chinese.

Chapter 2

Natural Language Parsing

This chapter will aim at producing a general overview of some of the problems associated with parsing. A little bit of history of automatic parsing is mentioned, with the aim of describing the current state of the art and of outlining new research.

The first section covers some basic terminology and defines concepts of the theory. The second section is essentially historical, covering early attempts to study aspects of parsing and recent developments in the field of syntax. The basic strategies of the parser involved are given. The final section discusses the role of semantics within parsing, and an evaluation of the different ways in which semantic information can be incorporated into a parser.

2.1 Introduction

To understand something is to transform it from one representation into another. This is done by a process, which is called *parsing*. To parse a sentence, it is necessary to use a *grammar* that describes the structure of strings of words in a particular language.

A sentence in a natural language can be analyzed from two points of view, *syntax* and *semantics*.

Syntax, or grammar, is concerned with the form of the sentence. The sentence

“He saw any airplane.”

is syntactically incorrect, since English syntax states that the word “any” usually is used in “negative sentences”. Note that the syntax says nothing about the meaning of the sentence.

Semantics, on the other hand, is concerned with the meaning of a sentence. The sentence

“The blue idea dreams.”

is meaningless if the words in it are given their usual interpretations, since the idea neither has a color, nor can dream. On the other hand, the sentence is perfectly grammatical, and we can easily analyze it into “determiner + adjective + noun + verb”.

Syntax allows us to identify word patterns without concerning about their meanings. However, semantics is more important, because we are interested in the meaning of sentences. Consider the two sentences

a) “He saw the girl with a telescope.”

b) “He saw the girl with the red hair.”

Sentence a) is structurally ambiguous, since the adjunct ‘with a telescope’ can be either a modifier of ‘the girl’ or an instrumental modifier of the verb ‘saw’. If a parser has only syntactic information, it is likely to find a) and b) ambiguous in exactly the same way, since the disambiguation comes from semantic information. Hence, a parser should be able to parse a sentence not only syntactically but also semantically.

In general, many natural language processing applications (e.g. information extraction, machine translation, etc.) require fast and robust parsing of large numbers of texts.

2.2 Developments in Parsing

In the earliest years of the research of natural language parsing, Chomsky’s grammar theory influenced the field. Chomsky in 1956, with further elaboration in 1958 and

1959, first introduced the idea that languages could be interpreted as sets of strings. The rewriting rules (the *grammar*) can define an infinite set of possibilities, including many that have never been encountered before. Chomsky's approach was to shift from an emphasis on the language to an emphasis on the grammar, i.e. on rules that could generate the language. The grammatical formalism first introduced was known as the Chomsky hierarchy of grammar theory.

Example Assume that the language is limited to a set $\{a, b, c, d\}$.

Rules could be

$$(i) S \rightarrow AB$$

$$(ii) S \rightarrow aSa$$

$$(iii) A \rightarrow a$$

$$(iv) B \rightarrow b.$$

Then we can use the rules to generate a sentence "aaba". We show the procedure here:

$$(1) S$$

$$(2) S \rightarrow aSa \quad (\text{rule } ii)$$

$$(3) S \rightarrow AB \quad (\text{rule } i)$$

$$(4) A \rightarrow a \quad (\text{rule } iii)$$

$$(5) B \rightarrow b \quad (\text{rule } iv).$$

Chomsky's work had been preceded by the work of Harris [Harris, 1968, 1982], who had already introduced the idea of "transformations". However, it was Chomsky who was able to recognize the broader significance of these developments and to bring them into focus for the linguistic community. He developed a new theory, *transformational grammar*, which represents a major theoretical reformulation for the field of linguistics.

The *argument transition network* model of Woods [Woods, 1970], usually abbreviated to ATN, has enjoyed considerable success in computational linguistics. Such information is easily available in the literature: the original 1970 article by Woods, see

[Bates, 1978] for a tutorial overview; [Grimens, 1975] on ATNs as a medium for linguistic descriptions; [Kaplan, 1975] and [Stevens & Rumelhart, 1975] on ATNs as model of human sentence processing; [Waltz, 1978] and [Woods *et al.*, 1972], summarized in [Woods, 1977], on ATNs as front-end processor for database interrogation.

The ATN idea has had considerable influence in the field of natural language parsing, and is still an active one. It has provided a useful tool, which explicates left-to-right processing, and separates the parser from the grammar that is being applied. More specifically, ATNs, in contrast to Transformational Grammar (TG) in its then Standard Theory (ST) form, provided a set of actions offering a procedurally neat way to produce the deep structure of sentences.

Throughout the 1960s, there was an emphasis on purely syntactic parsers (e.g. [Kuno, 1965], [Thorne *et al.*, 1968]), but this was followed, in the early 1970s, by a desire to design ‘wholly semantic’ sentence-analysers. The more semantically-oriented work of Riesbeck, Wilks and Winograd [Winograd *et al.*, 1972] can be seen as a great progress in natural language parsing.

In the early 1980s, an important theoretical development has made the formal grammar theory become more spread. This can also be illustrated with the introduction of formalisms like Lexical Functional Grammar (LFG), Generalized Phrase Structure Grammar (GPSG) and Functional Unification Grammar (FUG). These new linguistic theories stressed the role of the lexical information in automatic parsing. For example, the Word Expert Parser (WEP) has been developed with particular attention paid to the wide variety of different meaning roles of words when analyzing fragments of natural language text.

On the other hand, there has been a resurgence of statistical or empirical approaches to natural language processing since the late 1980s. The success of such approaches in areas like speech recognition [Rabiner, 1989], part-of-speech tagging [Charniak *et al.*, 1993], syntactic parsing [Ratnaparkhi, 1999]; [Manning & Carpenter, 1997]; [Charniak, 1996]; [Collins, 1997]; [Pereira & Shabes, 1992], and text or discourse segmentation [Litman, 1996] is evidential.

2.3 Aspects of Parsing

To parse a string, a sentence, according to a grammar, means to reconstruct the parse tree that indicates how the given string can be produced from the given grammar.

The parse tree is a basic connection between a sentence and the grammar from which the sentence can be derived. To reconstruct the parse tree corresponding to a sentence one needs a parsing technique. There are dozens of parsing techniques, but only two basic types are reviewed in this section, one is bottom-up parsing, the other is top-down parsing.

Also, two search techniques, depth-first search and breadth-first search, are mentioned in this section.

2.3.1 Top-down parsing

In top-down parsing, we start with the start symbol S and try to deduce the input sentence by constructing the parse tree, which describes how the grammar was used to produce the sentence.

Suppose we have the following simple grammar for natural language, and suppose the sentence is “He hits the dog”.

$$S \rightarrow NP VP$$

$$NP \rightarrow the N$$

$$NP \rightarrow PN$$

$$VP \rightarrow V$$

$$VP \rightarrow V NP$$

$$N \rightarrow dog$$

$$PN \rightarrow he$$

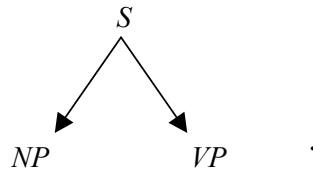
$$V \rightarrow hit.$$

First we try the top-down parsing method.

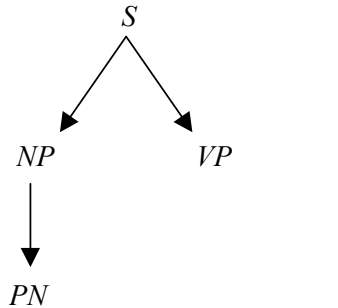
We know that the production tree must start with the start symbol:

S .

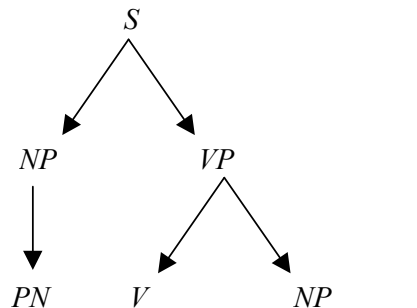
We only have one rule for S : $S \rightarrow NP VP$:



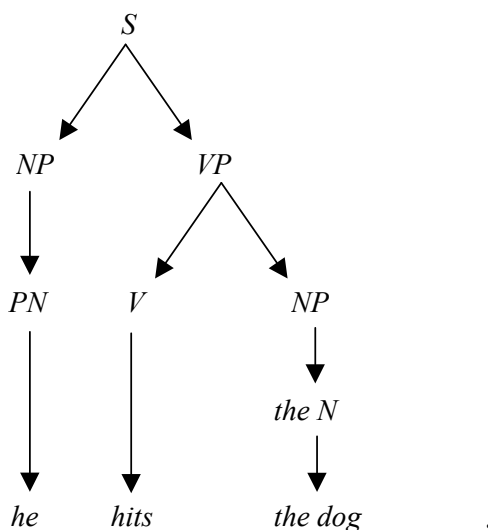
We have two rules for NP : $NP \rightarrow the N$ and $NP \rightarrow PN$. The first rule would require “the noun” for some noun, the second rule would require a “pronoun”; this leads to the choice of applying the second rule and we obtain:



Again two rules may be applied for VP : $VP \rightarrow V$ and $VP \rightarrow verb NP$. The second one is fit for this sentence:



We continue this process by applying the first rule for NP and the sentence is deduced by substituting the actual words:



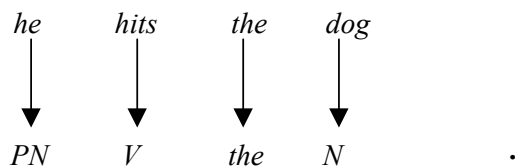
Top-down parsing tends to identify the production rules in prefix order, in which a sentence is deduced by using production rules from the left-hand side to the right-hand side. Note that we have to choose the proper rules to reach our goal. There is a search problem.

2.3.2 Bottom-up parsing

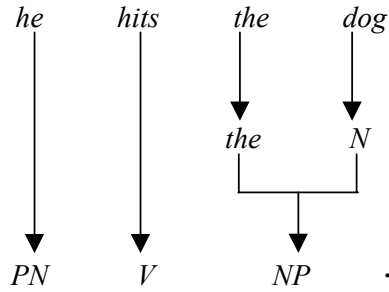
In bottom-up parsing, we start with the sentence as the input string and try to reduce it to the start symbol that is usually expressed by the symbol *S*. Here the keyword is reduce. We reduce the input (sentence) to the substring (segment) that is the result of the last step by applying an inverse rule of grammar in postfix order. When we find that the right-hand side of a rule can match with a segment, we replace the segment with the left-hand side of the rule and repeat the process, until only the start symbol is left.

Suppose we have the same grammar as above and suppose the sentence is also “He hits the dog”. Now we try the bottom-up parsing method.

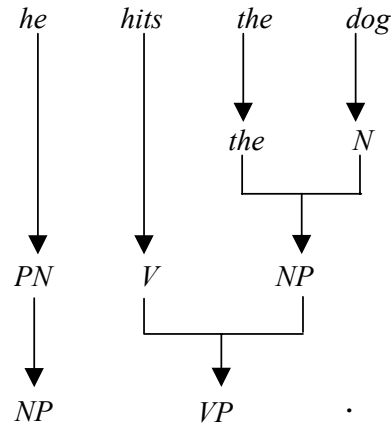
The first step is to recognize the word type for each word as follows:



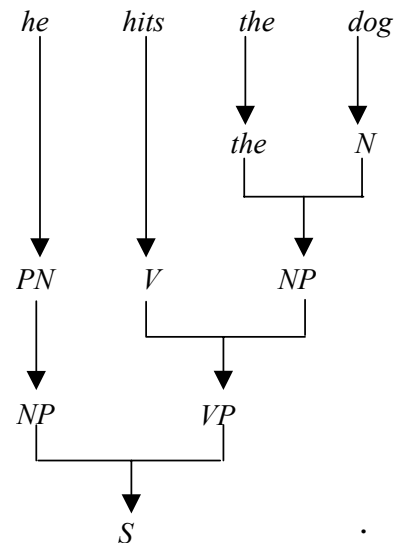
Then we recognize the word “he” as derived by $NP \rightarrow PN$, the word “the” and “dog” as derived by $NP \rightarrow the\ N$. Hence we try



Again we find only one recognizable substring, namely “ $V\ NP$ ” that can be derived by $VP \rightarrow V\ NP$. So here we are forced to construct



And also our last reduction step leaves us no choice:



We have obtained the same parse tree, with arcs in opposite order.

Both parsing techniques can be used in structural parsing. Note that also in bottom-up parsing we may have more possibilities to choose from.

2.3.3 Search techniques

Search techniques are used to guide the parsing through all its possibilities to find one or all parsings.

There are in general two methods for searching, which are depth-first search and breadth-first search in the production tree of partially generated solutions.

Suppose there are several alternatives for further processing a partially solved problem. In depth-first search we concentrate on one alternative, and continue with that alternative until we reach a dead end. Then we go back in the production tree to choose another alternative. In breadth-first search we keep all the alternatives for each partially solved problem, unless we reach a dead end, in which case we choose one alternative to repeat the same procedure.

The distinction between breadth-first search and depth-first search is rather evident. Both of them are valid, we can just choose one of both.

2.4 Parsing Methods

2.4.1 Traditional methods

Numerous parsing methods have been developed. Generally, in order to understand a sentence three phases of processing are distinguished.

- In the syntactic phase, a sentence is processed, using syntactic and morphological knowledge, into a structural description (such as the syntactic parse tree), which is used to represent the syntax structure of the sentence.
- In the semantic interpretation phase, the structural description is mapped, using semantic knowledge, into a semantic model that represents the meaning of the sentence (independent of the context).

- In the contextual interpretation phase, this representation is mapped onto a final representation of the sentence that is the real meaning of the sentence in the whole context.

One of the main issues in constructing a parser is whether to use an approach to separate *syntactic* and *semantic* processing of a sentence or to use an integrated approach. Parsing methods can be roughly classified into three categories, according to the way in which they handle syntax and semantics:

- The *syntax-first* approach: the first step is to build a full syntactic structure for the sentence, and the second step to map this to a semantic representation. The main advantage of this approach is the simplicity of design of the program. The main disadvantage of this approach is that the semantic information is not used when a sentence is parsed syntactically. This can lead to a combinatorial explosion of the number of syntactic representations that are possible. Perhaps the best known ‘syntax-first’ program was the question-answering system of Woods [Woods *et al.*, 1972].
- The *non-syntactic* approach: there is no syntax/semantics distinction, and a semantic structure is built directly from the sentence. Throughout the 1960s, there was an emphasis on purely syntactic parsers (e.g. [Kuno, 1965], [Thorne *et al.*, 1968]), but this was followed, in the early 1970s, by a desire to have *wholly semantic* sentence-analysers. One of the best known *non-syntactic* system was that of Riesbeck [Riesbeck, 1974, 1975a, 1975b], which built a *conceptual dependency* structure [Shank, 1972, 1975] while scanning a sentence from left to right. The main distinguishing feature is the lack of a formal separation, and all the grammatical information is treated as being qualitatively the same. A simple example is parsing according to a semantic grammar, where syntactic categories are replaced by semantic categories in the grammar rules.
- The *integrated* approach: the syntactic and semantic processing takes place simultaneously, throughout the parsing process. The main advantage of this approach is that parsing rules operate with both syntactic and semantic information, and semantic information is used to limit the number of syntactic parses. The main disadvantage of this approach is that it is impossible to construct parsing rules that can be applied to syntactic categories in general.

2.4.2 Parsing with knowledge graphs

A new parsing method, which is called *structural parsing*, has been developed in the framework of knowledge graph theory. The approach usually distinguishes two processors: a syntactic processor and a semantic interpreter. The syntactic processor converts a sentence into a *syntactic sentence graph* of the sentence. The semantic interpreter gives a *semantic sentence graph* that represents the meaning of the sentence.

A very important knowledge source for structural parsing is the lexicon of word graphs. The lexicon is a list of *syntactic word graphs* and *semantic word graphs*. Depending on this lexicon, a syntactic sentence graph of the sentence that is processed can be constructed. Thus a semantic sentence graph of the sentence is obtained.

In our structural parsing theory, we would like to set up a more flexible method that can both be used for the *syntax-first* approach and for the *integrated* approach, based on syntactic word graphs and semantic word graphs. Moreover, we pay more attention to *semantic chunks* (i.e. partial meanings can be constructed as the sentence is processed from left to right) and *utterance paths*.

Chapter 3

Theory of Knowledge Graphs

Knowledge graph theory is a kind of new viewpoint, which is used to describe human language, while focussing more on the semantic than the syntactic aspects. Ontological aspects of knowledge graphs are discussed by comparing with important other kinds of representations. It is expounded that knowledge graphs have advantages, which are stronger ability to express, to depict deeper semantic layers, to use a minimum relation set and to imitate the cognition course of mankind etc. Its appearance gave a new way to the research of computer understanding of human language.

3.1 Formal Description of Knowledge Graphs

Knowledge graphs, as a new method of knowledge representation, belongs to the category of semantic networks.

In principle, the composition of a knowledge graph is including *concept* (tokens and types) and *relationship* (binary and multivariate relation).

3.1.1 Concept

(1) Tokens

Understanding human language lies in the perception of the real world (including general concept and individual instantiations). The thing that can arouse man's perception in the real world is said to give rise to a token in the mind. In knowledge graph theory, we use the symbol \square to express this token. In fact, that people observe a thing is to signify there is such a thing in our real world. Therefore, in knowledge graph theory, everything will have a corresponding token.

Definition 3.1 A token is a node in a knowledge graph, which is indicated by \square . It expresses that we experience a thing in the real world or an existent concept in our inner world.

(2) Types

According to the viewpoint of subjectivism, different persons may describe experiences of the real world by different tokens. Some persons can experience some existent things in the world, that some other persons can not experience in the same way. So, in the inner world of a person, the token that can express existence may not exist in the inner world of another person. If a token can be shared by most of the persons, or can be shared by all persons, we have one objective picture.

The most basic nature of perception is to divide different tokens into similarity classes. We observe that there is an identical type of the tokens that are in the same class, and that we can introduce types to express these tokens. For instance, if we see a dog or a tree, there is the token of dog or tree in the knowledge graph. So, we divide nodes into two categories that are types and tokens. In knowledge graph theory we may distinguish three kinds of *marks*. Mark \square shows a concept, and plays a role that is similar to the argument in logic. Type is the mark that is doing the labeling. It expresses the general concept that is determined by its property set. Therefore, type can be regarded as general information. Another kind of mark expresses the instantiation. That mark gives an example of the type, and it expresses the individual that is considered within the domain.

In knowledge graph theory, we use the directed ALI relation between type and token

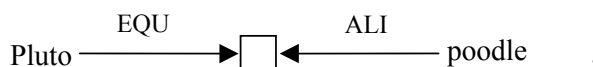
to express that the token has that certain type, and the directed EQU relation to express the instantiation.

Definition 3.2 If a token is related to a mark by an ALI relation, which points to this token, this token is said to have the *type* expressed by the mark.

The symmetric ALI relation expresses that a kind of thing and another kind of thing are similar. In graphic representation, the directed ALI link is used to point at a concept from a label, to give the concept a type with this label. However, A ALI B means that the concept B is similar to the concept A.

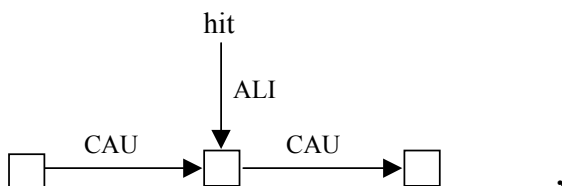
Types of tokens are part of a *type hierarchy*. This hierarchy involves the well-known ISA relationship. If type 1 ISA type 2 then type 2 is more general than type 1. The most general type we consider is “something”, which may be seen as the name of the single token. “Type ISA something” holds for all types.

An example should clarify the things so far. If the poodle Pluto is considered then we represent this by the knowledge graph



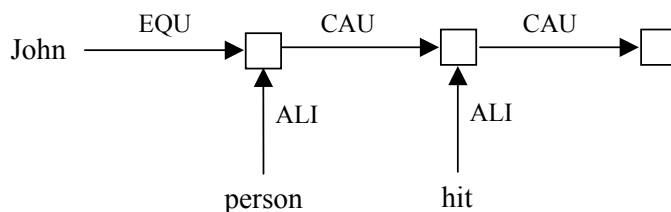
Here Pluto, \square and poodle are *marks*. We know that a poodle ISA dog, in knowledge graph theory the concept poodle, which is a graph, is considered to contain the concept dog, which is a graph too. The two graphs are related in the sense that the graph of “dog” is a subgraph of the graph of “poodle”. In the theory this is described by the FPAR relation, so dog —FPAR→ poodle describes that poodle ISA dog. The FPAR relation will be discussed later.

In most graphs that we consider the tokens are typed, i.e., there is an ALI-link towards that token. However, it may occur that a token without indicated type occurs. In “John hits”, the object is not mentioned. A transitive verb like “hit” is represented by



in which only the central token is typed. The left token describes the subject and may

e.g. be instantiated by “John”, but the right token may be left unspecified, knowing that John in an instantiation of a person, we would describe “John hits” by



3.1.2 Basic relations

To describe the real world, we need to distinguish the relationships between tokens. In knowledge graph theory, the most important principle is to use a very limited set of relationships. These relation types are required to be basic. The more independent these types are semantically, the better. It should not be possible to deduce one relation type from others. The meaning of these relations can be described through considering the relationship with the real world. These basic relation types can be used for establishing more complex concepts and relations. How to choose these basic relations in knowledge graph theory?

A basic relation is the relation of cause (CAU). In the initial stage of the knowledge graph project, this relation was the only relation type that people were interested in. In fact, in medical and social science, this one plays a very important role.

Definition 3.3 The *causal relation* between two tokens is expressed by the arc labeled CAU.

This relation expresses the relationship between a cause and an effect, or a thing influencing another thing. This relation type, not only in knowledge graph theory but also in other representation methods, is the relation that was distinguished most early, and is a basic relation which is used in a lot of inferences as occurring in a diagnosis system. The famous expert system MYCIN concerns IF THEN-rules only based on this one type. This relation points from the concept that produces the influence to the concept that has been affected, in the graphic representation.

There are also various situations in which the causal relation occurs. A thing or person can arouse an incident or course. An incident or course can also arouse another incident or course. An incident or course can arouse a state. Therefore, the complex structure that contains the causal relation is used to describe complex concepts, such as agency, purpose, reason, tool and result. For instance, in English the phrase “to hit with a stick” can be described as “to cause the stick to move resulting in a contact state”.

When we discuss set theory, considering the relations between sets, we discover that four should be basic relations due to the following. Given “A” and “B” to express two sets respectively, we can distinguish the following relations: $A = B$, $A \subset B$, $A \cap B \neq \phi$, as well as $A \cap B = \phi$. The four relations show respectively: A and B are identical, A is a subset of B, A and B have common elements and A and B are completely disjoint. If we will regard A and B as the property sets of some special designated things, we must introduce relations to express these four relation types.

Definition 3.4 A mark is a *value*, if it is connected to a token through a directed EQU-relation. An EQU-relation between two tokens expresses that two sets are *equal*.

In graphic representation, this relation can express the concept naming through the arc from label to concept. This relation also can be used for thing assignment, for instance, red as value assigned to color. For a symmetrical relation, such as the equaling relation in set theory, we use the symmetric EQU-relation to join A and B.

Special values are the perceptions by a person as individual, therefore, a mark is a very special value, and this special value can be expressed by an EQU-relation, this EQU-relation being directed from the mark that expresses the value to the token that indicates the perception. The reason for using the EQU-relation is that the special value is the assignment for the studied token.

A very important relation is that a thing is a part of another thing.

Definition 3.5 If there are two tokens that express two sets respectively, and one is a *subset* of another, then there is a SUB-relation between the two tokens.

Note that there is a subtle difference between the SUB-relationship and the ISA-relationship mentioned before. For a SUB b, there are two different interpretations.

- Concept a is a part of concept b. For instance, tail SUB cat. This expresses that the tail of a cat can be regarded as a *part* of the cat because the molecules of the tail form a subset of the molecules of the cat.
- Concept a is more general than concept b, therefore, concept b contains at least all features of concept a. For instance, “mammal SUB cat” expresses that cat is a kind of mammal and has more information than that involved in the general mammal.

Note that we use a concept as a set of properties here. If the concept is expressed as a graph the elements of this graph will be said to be in an FPAR-relation with the graph as a whole. In the second interpretation of $a \text{---SUB---} b$, that in which a and b are seen as *property sets*, we have a contamination of terminology. As *sets* a and b are related by the SUB-relation, but as concepts, represented by graphs, we prefer to say that a is a *property* of b, as are all other subgraphs of the graph representing b. For this relationship between a and b we used the FPAR-relation for its description.

Definition 3.6 The ALI-relation is used between two tokens for which there exist *common* elements.

Definition 3.7 The DIS-relation is used to express that two tokens are in *no relation* to each other.

In set theory, $A \text{ DIS } B$ expresses $A \cap B = \emptyset$. Because of the symmetry, the DIS-relation is described by an edge instead of an arc. The same holds for the EQU and the ALI relation.

When humans think about something they judge and ascribe certain attributes to things. For example, “the ball is red” indicates a relation like “red is the color of the ball”. This led to including a new relation between an attribute and an entity.

Definition 3.8 The PAR-relation expresses that something is an *attribute* of something else.

This relation expresses that a certain thing is attributed to or the external nature of another thing. In graphic representation, this relation is from the attribute concept to the entity concept.

Another relation that needs to be considered is the order relation. This relation expresses ordering with respect to a certain scale, like space, time, place, length,

temperature, weight, age, etc. With this relation it is also possible to represent different tenses of language, by relating the time of an event with the time of speaking, see Chapter 6. In our concept world, this relation is a basic type of relation.

Definition 3.9 The ORD-relation expresses that two things have a certain *ordering* with respect to each other.

When comparing the order of two things, we use this relation. This relation is usually used for showing the order of time and space; but it also can be used to express “<” relation in mathematics. When considering an ordering relation, the ORD-arc usually points from the token with “low” value of the concept to the token with “high” value of the concept.

The basic goal of the knowledge graph project is to use a limited number of relation types. Only if the relation types are not enough to express something, we will be forced to add a new relation type. To express *the dependency relation*, we must consider a new relation type, which corresponds to *mappings*. Though in natural language there are many words to express mapping, we still choose one relation type to express this relation, which is called SKO (Skolem)-relation. We particularly refer to van den Berg [Berg, 1993]. For informational dependency in mathematics we use the words function, functional or just mapping. In natural language words like “depends on” are used.

Definition 3.10 A token in a knowledge graph has an incoming SKO-relation from another token if it is *informationally dependent* on that token.

The meaning of the SKO-relation is based on the concept of *informational dependency*. This involves an aspect of *choice*, see also Section 6.6.

In information transmission, we discover the mutual connection between information, as one of the basic relations that we must consider. It expresses a relation as between “saying” and “what is said”. A change in the “saying” causes a change in “what is said”, but “what is said” is informationally dependent on the “saying” process. On the syntactic level we encounter a similar situation. In “man hits dog”, we choose to relate “man” with “hit” and “hit” with “dog” by a CAU-relation. But what is the type of relation between the subject and the verb, respectively the verb and the object? That something is an object or a subject depends on its *functional* relationship with the verb.

For that reason these syntactic relationships are also modeled by the SKO-relation.

Of course, knowing, perception, feeling, may also be modeled with this relation.

Apparently, it is impossible to express everything in the world with only binary relations. To solve this problem, in the first stage of the knowledge graph project, the frame relation FPAR was introduced. In the second stage of the knowledge graph project, people were led to the three other frame relations. At present in knowledge graph theory there are four frame relations in total, see [Reidsma, 2001].

In fact, the FPAR-relation is the initial frame type, which is used to express a complex concept or to express the word “and” in logic.

Definition 3.11 A *frame* is a labeled node. A frame relation expresses that the labeled node is actually a frame around some complex graph. All nodes and arcs within the frame are connected to the frame node by the FPAR-relation.

Note that the graph can be interpreted as an n-ary relationship, just like an arc can be interpreted as a binary relationship. The FPAR relationship expresses that some subgraph of the graph is part of the whole graph that was formed. The “animal” graph, itself a frame, is part of the “cat” graph. Hence, animal — FPAR → cat. We already discussed the possibility to use a SUB-relationship here.

To express negation, and the possibility and the necessity in modal logic, three kinds of relation types are introduced.

Definition 3.12 *NEGP* expresses the *negation* of the contents of the frame.

Definition 3.13 *POSP* expresses the *possibility* of the contents of the frame.

Definition 3.14 *NECP* expresses the *necessity* of the contents of the frame.

Note that the contents of the frame may form the graph representation of a proposition.

3.1.3 Relations in general

Like the concept essentially is a graph, the relation between two concepts is also a graph, namely some graph containing both concepts. An example that should make

this clear is the relationship “married to”. This is clearly a concept in itself and therefore there must be a graph that is the meaning of this concept. In that graph two concepts occur that are the entities that are married to each other. If “John is married to Mary”, both the concept “John” and the concept “Mary” must occur in the graph that represents this sentence.

Definition 3.15 A *relationship* between two concepts a and b is a graph in which both a and b occur.

It will be clear that different graphs containing a and b in principle determine different relationships. However, homonymy is abundant in language. There are many graphs possible for one concept name. For “married to” there are also many definitions possible. It is also clear why there is a definite need for determining basic relationships. If one does not pay attention to such a basic set of notions, which is usually called an *ontology*, the number of types of relationships will grow indefinitely, as is so obvious from the field of semantic networks.

3.2 Ontological Aspects

We recall only the most essential parts for our discussion. The word graph ontology consists, up till now, of the token, represented by a node, eight types of binary relationships and four types of n-ary relationships, also called frame relationships.

The eight binary types describe:

- Equality : EQU
- Subset relationship : SUB
- Similarity of sets, likeness : ALI
- Disparateness : DIS
- Causality : CAU
- Ordering : ORD
- Attribution : PAR
- Informational dependency : SKO.

They are seen as means, available to the mind, to structure the impressions from the outer world, in terms of awarenesses of somethings. This structure, a labeled directed graph in mathematical terms, is called *mind graph*. Any part of this graph can be framed and named. Note that here WORDS come into play, the relationships were considered to be on the sub-language level so to say, on the level of processing of impressions by the brain, using different types of neural networks.

Once a subgraph of the mind graph has been framed and named another type of relationship comes in, that between the frame as a unit and its constituent parts. The four n-ary frame-relationships are describing:

- Focusing on a situation : FPAR
- Negation of a situation : NEGPARG
- Possibility of a situation : POSPAR
- Necessity of a situation : NECPARG.

The situation is always to be seen as some subgraph of the mind graph. It will already be clear that word graphs for logic words will mainly be constructed using the second set of four n-ary relationships.

3.2.1 Aristotle, Kant and Peirce

Let us compare our ontology with two of the many ontologies proposed in history. The first one is of course that of Aristotle. He distinguished:

- Quantity ● Relation ● Time ● Substance ● Doing
- Quality ● Location ● Position ● Having ● Being affected.

These ten basic notions clearly focus on the physical aspects of the impressions, as do the first eight notions of word graph ontology. The focus there is on the way the world is built. The second ontology to consider is that of Kant, who distinguished twelve basic notions:

QUANTITY	QUALITY	RELATION	MODALITY
Unity	Reality	Inherence	Possibility
Plurality	Negation	Causality	Existence
Totality	Limitation	Commonness	Necessity

Note that Kant clearly focuses on the logical aspects, including modal logic concepts like possibility and necessity. Of course negation is included as well. Together with the “and” concept, which is simply two tokens framed together in knowledge graph theory, the negation gives a functionally complete set of logical operators for predicate logic. The two other frame relations give a way of describing all known systems of modal logic by means of knowledge graphs, as was shown by van den Berg [Berg, 1993].

Here some remarks are due concerning the work of C.S.Peirce [Peirce, 1885]. Describing logic by graphs, called *existential graphs* by him, was introduced by Peirce before 1900, starting with the idea of simply indicating *and* (\wedge) and *negation* (\neg) by two different types of frames. The work of van den Berg can be seen as a direct continuation of this setup. It has often been said that Peirce was guided by the ontology of Kant, who presented the twelve basic notions in four triples, see above, when he introduced the notions of firstness, secondness and thirdness of a concept. Peirce’s definitions are not very easy to understand. We quote from Sowa [Sowa, 1994].

- Firstness: The conception of being or existing independent of anything else.
- Secondness: The conception of being relative to, the conception of reaction with, something else.
- Thirdness: The conception of mediation, whereby a first and a second are brought into relation.

From the point of view of knowledge graph theory the following stand is taken.

For any concept, token (or node) of a mind graph, we can distinguish:

- The token itself, which usually has an inner structure, the definition of the concept.

- The token together with its neighbors, inducing a subgraph of the mind graph, that we call the *foreground* knowledge about the concept.
- The whole mind graph, considered in relation to the concept, also including what we call the *background* knowledge about the concept.

In this view Kant's triples do not correspond precisely to Peirce's notions and we have the idea that the triple of knowledge graph theory: concept, foreground knowledge, background knowledge, is all that Peirce's notions are about. What is extra in our theory is the fact that the mind graph is not a fixed entity but depends on the particular mind (human) and for one human even on the particular circumstances in which the concept word is to be interpreted. Also the *intension* of the concept, its definition, is often not uniquely determined, although it is one of the major goals in science to get at least the definitions straight. The variation in meaning, possible for a word, is an intrinsic and essential part of knowledge graph theory.

3.2.2 Logic

(1) *The graphic view of first order logic*

Symbolic logic was started by mathematicians, and has been applied in many fields. Peirce gave a graphic representation for symbolic logic. For instance, suppose that p , q , and r are predicates in symbolic logic, the graphic symbol of each of them being given by itself. We list the graphic symbols and standard predicate formulas as follows:

Graphic Symbol	Standard Logic Symbol
$\boxed{p \quad q \quad r}$	$(p \wedge q \wedge r)$
$\neg \boxed{p \quad q \quad r}$	$\neg(p \wedge q \wedge r)$
$\neg \boxed{\neg \boxed{p \quad q \quad r}}$	$\neg(\neg(p \wedge q \wedge r))$.

In symbolic logic, we know that any predicate formula can be changed into a

conjunction form. For instance, $p \vee q$ can be represented by the equivalent formula “ $\neg(\neg p \wedge \neg q)$ ”. Some complex predicate formulas are listed:

Graphic Symbol	Standard Logic Symbol
\neg \neg p \neg q \neg r	$p \vee q \vee r$
\neg p \neg q \neg r	$p \rightarrow (q \vee r)$
\neg p q \neg r s	$(p \wedge q) \rightarrow (r \wedge s)$.

According to the above, the disjunction, conjunction as well as various predicate formulas derived from them can be expressed conveniently by graphic symbols. Since the universal quantifiers can be converted into the existential quantifiers in first-order logic, the graph of a logic formula is also called an existent picture. Extending the frame concept, not only first-order logic but also other logic, such as modal logic and tense logic, can be expressed by these graphic symbols. These kinds of graphic symbols can also be applied to conceptual graphs and knowledge graphs. Therefore, the graphic symbol for standard logic of Peirce has established the logical foundation of both knowledge graph theory and conceptual graph theory.

(2) Knowledge graphs and logic

The FPAR-frame and the NEG-frame in knowledge graph theory correspond with the structure of Peirce’s graphs. Besides, in knowledge graph theory there also exists the SKO-loop to express the universal quantifier; including the POS-frame and the NEC-frame knowledge graphs are able to express modal logic, such as possibility and necessity; the ORD-relation can express tense logic. Following we give the corresponding knowledge graphs.

The connection words in proposition logic “and”, “or”, “not”, “if then”, etc., as well as the necessity and the possibility in modal logic etc., have word graphs that are shown in Figure 3.1-3.6.

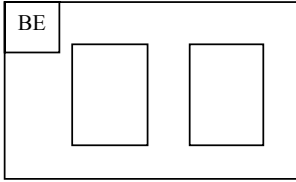


Figure 3.1 “and”.

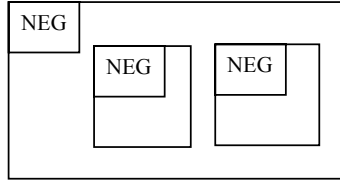


Figure 3.2 “or”.



Figure 3.3 “not”.

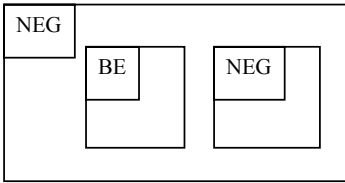


Figure 3.4 “if...then”.

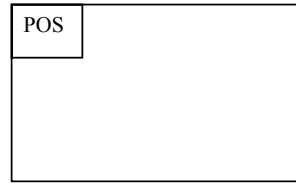


Figure 3.5 “possibility”.



Figure 3.6 “necessity”.

The existential quantifier is expressed by a distinct knowledge graph. The SKO-loop is used to express the universal quantifier, as Figure 3.7 shows.

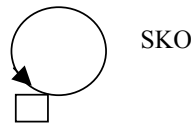


Figure 3.7 “universal quantifier”.

In a word, knowledge graphs can not only express propositional logic, but can also express predicate logic and other logics, such as tense logic and modal logic, so the theory has a very strong knowledge representation ability.

3.3 Semantics in Natural Language Processing

In this section, we study the knowledge graph method in natural language understanding. Somewhat special problems in Chinese natural language understanding are analyzed with knowledge graph theory. At the same time, the problem of ambiguity in natural language processing is studied by knowledge graphs. It is explained that the knowledge graph is a new method of natural language understanding, which expresses the meaning with structure, depicts the semantic meaning from deep layers, as well as reduces possible ambiguities.

3.3.1 Fillmore's case grammar

In natural language understanding, how to describe the semantic structure of a sentence? In this respect, people often focus on the case grammar of Fillmore. The key of case grammar is that the deep layer structure of a simple sentence is composed of the proposition part and the modality part. The modality part includes the concepts: tense, aspect, mood, form, modal, essence, time, manner. These concepts themselves can have different instantiations. For example, tense can be past, present or future, modal can be can, may, or must, etc.

- TENSE (PAST, PRESENT, FUTURE)
- ASPECT (PERFECT, IMPERFECT)
- MOOD (DECLARATIVE, INTERROGATIVE, IMPERATIVE)
- FORM (SIMPLE, EMPHATIC, PROGRESSIVE)
- MODAL (CAN, MAY, MUST)
- ESSENCE (POSITIVE, NEGATIVE, INDETERMINATE)
- TIME (ADVERBIAL)
- MANNER (ADVERBIAL).

The proposition part is formed by a verb and some noun phrase. Every noun phrase connects with verbs by a certain kind of relation, and such a connection is called a "case". Fillmore focuses on the research of the proposition part, and does not discuss the modality part.

There are many special problems in Chinese natural language processing. This is determined by the properties of Chinese linguistics. Summarized, the important features of Chinese are the following:

- There is no change in the shape of a word. For example, "shi2 xian4" (implement) can be used both as a noun and as a verb.
- The part-of-speech is not simply corresponding to the sentence composition.
- The structure of a sentence and the structure of a phrase are consistent basically.

- There are some special sentence patterns.

If case grammar aims at describing the semantics of a Chinese sentence, it has to expand to more cases. Now, a complete system for Chinese natural language understanding is the “lexicon semantic driven theory (LSD)”, developed by Yao [Yao, 1995]. He expanded case grammar further, which led to 49 kinds of semantic relations in total. However, the 49 kinds of relations can be expressed completely by the 8 kinds of binary relations and the 4 kinds of frame relations used in knowledge graphs. A detailed comparison is considered by Liu [Liu, 2002]. This shows that knowledge graph theory is more basic for describing the semantic structure of a sentence.

In fact knowledge graph theory, from a psychological point of view, describes this semantic structure with a limited number of relations (8 + 4 kinds).

3.3.2 Expressing semantics with knowledge graphs

To establish a model for natural language understanding, it is necessary to be able to express the meaning of a word or a sentence when the knowledge graph is used. The meaning of a sentence is a function of the meaning of each of its parts. This is usually called the *compositionality principle*. Therefore, to know the meaning of a sentence is to first know the meaning of each word, then gather all words into a sentence, in order to know the meaning of the entire sentence.

Here, we first talk about the meaning of a word. The word meaning is to be expressed through linking some concepts to other concepts. Consider for instance the word “single man”, in Chinese this is expressed by one word “dan1 shen1 han4”. We might connect the two concepts “man” and “not married”. The knowledge graph can simulate this. It puts in correspondence the “man” concept to a structure, as indicated by the dotted frame in Figure 3.8, and the concept “not married” in correspondence with another structure, as indicated by the drawn frame in Figure 3.8 in its simplest form. Then, connecting the two structures, we obtain a new structure that expresses the meaning of “single man”. Note that Figure 3.8 has given a most simple kind of sketch that expresses this word by a knowledge graph. According to different requirements, we can “expand” the concept “man” or “married” for further development to get a more complicated structure. This depends on the degree of

complexity required. Also note that we have used numbers after the Chinese words. These numbers denote the four ways Chinese words can be pronounced.

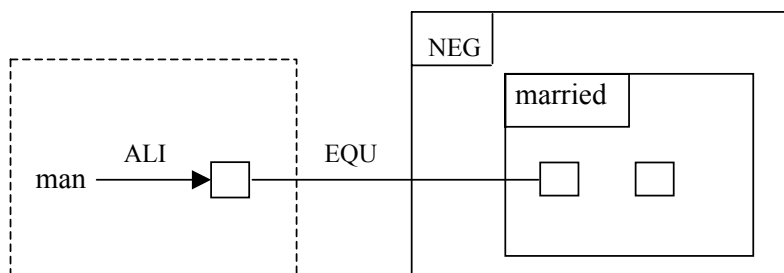


Figure 3.8 “single man”.

We should give a more thorough discussion of the idea of “expansion” of knowledge graphs.

Suppose that for the words that we use a lexicon of word graphs exists. Then in these word graphs certain concepts may occur, for which the lexicon has a word graph too. Now, expansion means that some concept, a single token, is replaced by the word graph in the lexicon that is given for it. This replacement leads to a more complex word graph for the concept in which it occurred. There are some problems here about the way such a more complex graph is to be included instead of the simple token, but we will not discuss this here.

After the expansion the word graph for the concept considered has been enlarged. However, it is still a meaning of the corresponding word. This is a central feature of knowledge graph theory. The meaning of a word is not fixed. A more elaborate discussion of the concept indicated by the word gives more “meaning” to that word. The very simple graph for “married” just indicated that two tokens were involved. Of course the meaning of “married” is much more complex and involves various ways the two tokens are related.

Also from the psychological point of view this subjective aspect of meaning is relevant. When a person ponders on some word, different structures may come to his mind. Each of the mental structures that the person can have in a correspondence with the word is a possible meaning, interpretation, of that word.

Making a lexicon of word graphs, may lead to a collection of graphs that are very simple or that are rather complex. It should be noted that this is also the case with

dictionaries and encyclopedia. A certain word gets explanations of different size in these books too.

Again, for instance, in Chinese the word “jie4 zhu4”, has the meaning that Figure 3.9 shows. This word expresses something like “make use of”, it literally says “borrow...to help self”. To understand the meaning of “jie4 zhu4” an “expanded” graph is needed. Figure 3.9 can be explained by noting that the tokens for both the agent of “borrow” and the objective of “help” are the same. The active “borrow” occurs before the action “help”.

Without this analysis it is not quite easy to understand “borrow... to help self”. In Chinese often word combinations force the listener to construct, in the mind, rather complex pictures, i.e., knowledge graphs. As was found by Liu [Liu, 2002], this is even more so in ancient Chinese.

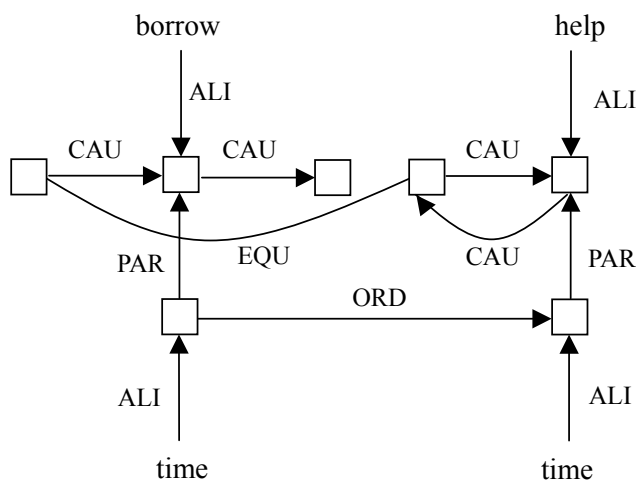


Figure 3.9 “jie4 zhu4”.

3.3.3 Structure is meaning

Knowledge graph theory emphasizes that “the structure is the meaning”. The meaning of a word is expressed by a word graph; the meaning of a sentence is expressed by a sentence graph. The graph of a sentence is composed of the word graphs that express the meanings of words in this sentence by the operations of *concept identification*,

concept integration and relation integration, see [Bakker, 1987].

In Chinese, as also in English, the same meaning may be expressed by various sentences. Although these sentences look different, the meaning expressed by them is identical, so in knowledge graph theory they are corresponding to one and the same sentence graph. In this way, the knowledge graph can restrict the redundancy of semantics considerably.

Consider for instance, the following two sentences (a) and (b). Though the structure of the surface layer of the sentence looks different, the deep layer meaning of these two sentences is identical, therefore their sentence graphs should have identical structure, the one that Figure 3.10 shows.

- Example**
- (a) Tai2shang4 zuo4zhe zhu3xi2tuan2.
 (rostrum) (sit) (presidium)
- (b) Zhu3xi2tuan2 zai4 tai2shang4 zuo4zhe.
 (presidium) (on) (rostrum) (sit)

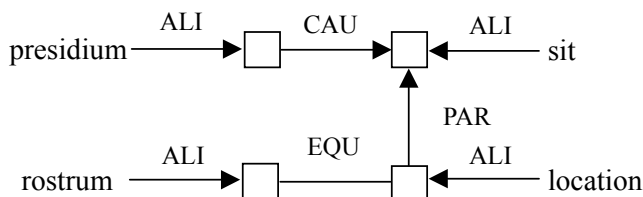


Figure 3.10 Sentence graph of sentences (a) and (b).

What we meet here is that a sentence graph can be uttered in more ways than one. This phenomenon will be discussed in greater detail in Chapter 6.

3.3.4 Elimination of ambiguity in natural language

In natural language there are many cases in which the meaning of a sentence is ambiguous.

For instance, in Chinese the sentence “xie3 de hao3” has two meanings at least, one meaning is “write well”, another is “to be able to write well”. These two different meanings can be expressed clearly by different sentence graphs. Figure 3.11 shows

“write well”; Figure 3.12 shows the sentence “to be able to write well”. The structure of these two graphs is different. Each graph has a distinct meaning, and the ambiguity is not due to the sentence graphs. The ambiguity comes in when the two different graphs are uttered in the same way.

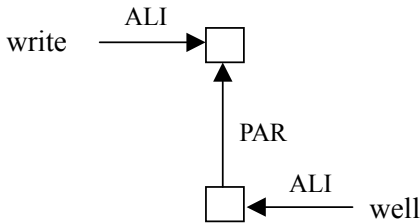


Figure 3.11 “xie3 de hao3” (1).

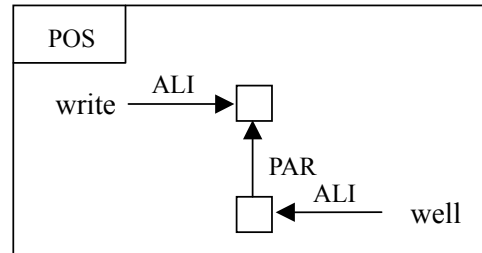


Figure 3.12 “xie3 de hao3” (2).

3.3.5 A limited set of relation types

In conceptual graph theory and in semantic network theory the number of relation types is not limited. Whenever some type of relation is needed, this relation will be added. In knowledge graph theory, the types of relation are limited in number, only the eight relations and four frames should be enough to express all semantics. This is the major difference between the two theories. We refer to [Willems, 1993] for a detailed comparison of the two theories.

If the set of relations is not restricted, there exists the problem of overlapping relations in semantics. A relation can be derived from another relation. We explain this problem in more detail with the following example.

Example Consider the sentence “man hits dog”. The conceptual graph and the knowledge graph are shown respectively in Figure 3.13 and 3.14.

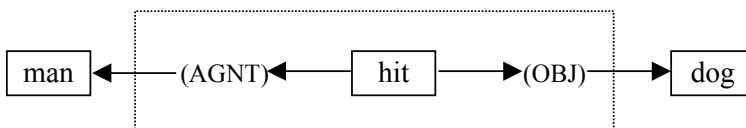


Figure 3.13 The conceptual graph of “man hit dog”.

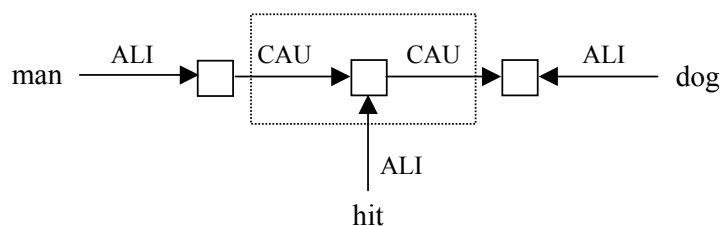


Figure 3.14 The knowledge graph of “man hit dog”.

Making a comparison between the two parts that are framed with dotted lines, it is clear that Figure 3.13 has used AGENT and OBJECT as two relations, but Figure 3.14 has used only one relation, namely the CAU-relation. In fact, in conceptual graphs, the AGENT relation expresses the agent (the doer of the action in a sentence), and the OBJECT relation expresses the object (the object of the action in a sentence). But in knowledge graphs, using only a CAU-relation, both the agent and the object can be expressed. If a token is related to a CAU-arc that points out, then this token is an agent. If a token is related to a CAU-arc that points in, then this token is an object.

This fact claims that in conceptual graphs the structure easily leads to redundancy. However, in knowledge graphs the relation set is limited, so that the redundancy in structures is considerably eliminated. Therefore, the knowledge graph is an appropriate semantic model of natural language understanding. It should be the first selection for language information processing of researchers to express the meaning of natural language.

3.4 Conclusion

For a knowledge representation method, besides of its ability to express things, the most important is its inference method. Whether an inference system is convenient, as well as efficient, directly affects its application level. The inference system of knowledge graphs has a very tight mathematical foundation, it has surmounted an obstacle for semantic networks in solving the complex problem of inference. This is one of the notable advantages of the new knowledge representation method. This chapter has not introduced the basic concept of the inference on knowledge graph. For details on the inference problem, see [Blok, 1997].

The meaning of a sentence, besides of the key part that concerns the logic and

semantics of a sentence, also includes the extension of a sentence, such as e.g. speaker's tone, mentioned as parts of the modality of a sentence in Section 3.3.1. It is part of the program to try to solve these problems with knowledge graphs.

Chapter 4

Word Graphs: The Third Set

This chapter deals with the third part of a series of word graphs in terms of knowledge graphs. A word is a basic unit in natural language processing. This is why we study word graphs. Word graphs were already built for prepositions and adverbs (including adjectives, adverbs and Chinese quantity words) in two other papers [Hoede & Li, 1996], [Hoede & Liu, 1998]. In this chapter, we propose the concept of logic word and classify logic words into groups in terms of semantics and the way they are used in describing processes. A start is made with the building of the lexicon of logic words in terms of knowledge graphs.

4.1 Introduction

Natural language is kind of special symbol system that is used to express human ideas and pass on information. Each natural language has evolved into a kind of traditional symbol system, which has its own word types, and word structures, in which different components have some relationship and these related components play a role as a whole that has some meaning. In science a particularly important part of the special symbol system is the set of words used in logic. It is therefore necessary to study the logic phenomena in natural language and the common rules in different languages in

order to reveal and explain pure logical forms, other logical structures and logical rules in sentences of a natural language text.

In particular in knowledge graph theory the meaning, which is considered to be identical with the graph structure, of a sentence is a function of the meanings, graph structures, of its various parts. Therefore, to understand the meaning of a sentence one first needs to understand the meaning of each word that occurs in the sentence thus gaining the meaning of the whole sentence from these words and their order, one of the syntactic aspects of the sentence. Finally, combining the sentence graphs we can obtain the meaning of the whole text.

This chapter will study the set of words called logic words, which have a logical function in a sentence or even a paragraph. In Section 4.2 we discuss aspects of classification of words. In Section 4.3 the classifications of logic words are proposed. By relating the analysis of linguistics with the analysis of logic, we try to reveal common properties of logic words that are independent of the particular language. In Section 4.4 we discuss word graphs for logic words. The first set and second set are given in Appendixes I and II, to make this thesis self-contained.

4.2 Classification of Words

It is necessary to classify the logic words before we study them. This section is inserted because there are more ways to classify than the one used in traditional Chinese linguistics. How to classify a word depends on the specific purposes that classifiers have. Different purposes will result in different classifications. In order to study the structure of a set of single words it is usual to classify the words into monosyllabic, disyllabic and polysyllabic words according to the number of syllables. In order to study the formation of words, we classify words into simple words and compound words according to the number of morphemes. However, this is not the classification of grammar. The classification of grammar is according to the grammatical function of a word. Its purpose lies in explaining the structure of language and the usage of a word. For example, in traditional Chinese linguistics, two categories in the classification according to grammar are the notional word and the function word.

In order to make a distinction between the purposes of traditional linguistic and general purposes we ought to divide the classification into two parts. One is the word classification in the narrow sense, which is according to traditional linguistics, another is the word classification in the general sense, which is according to some general purpose.

Definition 4.1 A *narrow* word classification is a classification according to the grammar of traditional linguistics.

Definition 4.2 A *general* word classification is a classification according to the purpose of the classifiers.

There are several criteria in a narrow word classification. In Chinese we have three main criteria. Words are considered:

- in terms of the shape feature
- in terms of the syntax feature
- in terms of the meaning of the word.

Of these criteria the second one is considered to be the key one because Chinese has no distinct changes in the shape feature and the criterion of a word's meaning may generate ambiguities. In line with the knowledge graph idea of meaning as a word graph, particularly in Chinese the meaning of a word strongly depends on the context. The specific meaning attached to a word is affected by the philosophy, the social aspects, the ethical aspects, etc., involved in the discussion. Nevertheless, the main goal of knowledge graph theory is to construct a lexicon of word graphs.

For the purpose of natural language processing, we investigate logic words and classify them, but not according to the narrow classification. The classification will be general, the purpose being to make distinction according to the ontology of knowledge graph theory.

4.3 Logic Words and their Classification

The words mentioned in traditional logic should be logic words. For example, “and”, “or”, “not”, “if... then...” and “if and only if”, which are five words describing

proposition logic, are typical logic words. Words like “possible”, “necessary”, “ought” and “permitted” which are used in modal logic and deontic logic are of course also logic words. But, many other words used in natural language, such as “therefore”, “since”, “while”, “but”, “before”, etc., are related to logical aspects of utterances as well. So the classification of logic words might include two parts, which are *pure logic words* and *other logic words*. The pure logic words are then, by definition, those words that are mentioned in traditional logic (including proposition logic, predicate logic, tense logic, modal logic, deontic logic and fuzzy logic). This class of words seems quite easy to recognize. The other logic words are words that are somehow related to logical aspects as well. This class turns out not to be as easy to define and structure. We are therefore in need of more precise classification criteria.

4.3.1 Classification criteria

Before discussing possible classification criteria we should mention the objects that we wanted to classify. A corpus of 2000 English words with the property that they occurred most frequently in a set of 15 texts was established by Holland and Johansson [Holland & Johansson, 1982]. Our main goal in this thesis is to develop a system of structural parsing, by means of which from a lexicon of word graphs the sentence graph of a given sentence can be constructed. So first the sentence is taken apart, and then from a representation of the parts, the words, a representation of the sentence is constructed.

a) Subjective classification

As a start of the general lexicon these 2000 words should be included. For that reason, and to have a natural restriction for the set of words, we considered these 2000 words and tried to classify them on a five-point scale:

- definitely a logic word
- clearly related to logic, but not basic
- related to some form of logical reasoning
- having a vague logical flavor
- no relation to logic.

As a result we classified the words into classes C_{11} , C_{12} , C_{13} , C_{14} , C_{15} , C_{22} , C_{23} , C_{24} , C_{25} , C_{33} , C_{34} , C_{35} , C_{44} , C_{45} and C_{55} . The two indices indicate the scale values mentioned by two classifiers. We give only the first class resulting from this subjective coding process, in order of the frequency of the words.

C_{11} : and, if, no, then, must, might, right, however, every, possible, difference, cannot, necessary, therefore, probably, true, thus, nor, everything, else, unless, truth, impossible, neither, doesn't, wouldn't, everyone, ought, isn't possibly, nevertheless, possibility, existence, maybe, equal, equivalent, necessarily, hence.

b) Classification by using Kant's ontology

Another way of classifying would be to use Kant's ontology and decide whether a word belongs to one of his twelve categories, i.e. expresses something of which the main feature is one of these twelve concepts. As an example, let us consider those words in the class C_{11} that we determined, that would fall in the category of "possibility". We would choose "might", "possible", "cannot", "probably", "impossible", "possibly", "possibility" and "maybe" as elements of this class.

c) Classification by using knowledge graph theory ontology

Looking at these words in b) from the knowledge graph point of view we discover that people, students in our case, making a word graph for those words, use the POSPAR-frame in all cases. This prompts another way of classifying, namely according to the occurrence of FPAR, NEGPAR, POSPAR and NECPAR-frames in the word graphs of the word. Note that these frames correspond to Kant's categories existence, negation (seen as a quality by Kant), possibility and necessity.

Definition 4.3 A *logic word of the first kind* is a word, the word graph of which contains one of the four types of frames in the knowledge graph ontology.

The existence of two somethings, seen as two components of a frame, put them in an FPAR-relationship with the frame. That frame can be named "and". Similarly, something in a NEGPAR-frame is put in a NEGPAR-relationship with that frame that now can be named "not". By functional completeness the other connectives from proposition logic follow from equivalences like $p \vee q \Leftrightarrow \neg(\neg p \wedge \neg q)$ for the "or" connective " \vee ".

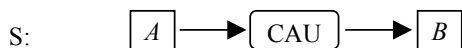
When we consider something, say a situation S (German: Sachverhalt) in the form of a graph a POSPAR-frame may be considered around it. We may describe this by saying “ S is a possibility”, “It is possible that S ” or “Possibly we have S ”. In Chinese these three utterances are translated as

- “ke3 neng2 S ”, literally “possible S ”
- “ S shi4 ke3 neng2 de”, literally “ S is possible”
- “you3 ke3 neng2 S ”, literally “have possible S ”,

respectively.

As mentioned before, we have chosen to give the Chinese sentences and words in this thesis in spelling form, *pin1 yin1*, followed by a number 1, 2, 3 or 4, indicating the four forms of intonation.

Subtle differences come forward due to the choice of the POSPAR-frame. Suppose S is given by the following graph, in so-called total graph form in which the arc is also described by a vertex,

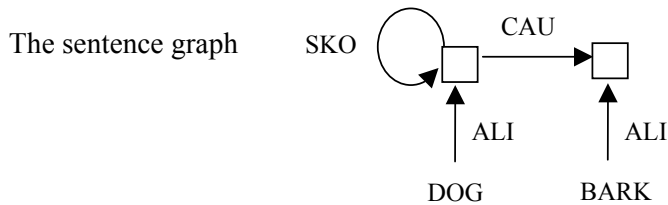


A POSPAR-frame around the whole of S would describe “(There is the) possibility (that) A causes B ”. A frame around $\boxed{A} \longrightarrow \boxed{\text{CAU}}$, which by itself reads, “cause A ”, would lead to a description of “ A (is a) possible cause (of) B ”. A frame around $\boxed{\text{CAU}} \longrightarrow \boxed{B}$ would describe, “ A possibly causes B ”. In English “possible” is an adjective and “possibly” an adverb. The decision which word to use depends here on “cause” being a noun and “causes” a form of the verb “cause”. In Chinese, the three sentences are translated

- “ A yi3 qi3 B shi4 k3 neng2 de”, literally “ A cause B is possible of”,
- “ B ke3 neng2 you2 A yin3 qi3”, literally “ B possible have A cause”
- “ A ke3 neng2 yin3 qi3 B ”, literally “ A possible cause B ”.

The reader should remark that the existential and universal quantifiers, “there exists” and “for all”, are not falling under Definition 4.3. In fact, Peirce already pointed out that making the statement S on the paper of assertion (in whatever form), is equivalent

to existential quantification (for closed formulae). That is why we put “There exists” between brackets in our example sentences. The universal quantifier in knowledge graph theory is expressed by the SKO-loop on a token, that should be read as “for all”.



is to be read as “all dogs bark” or “for all dogs (holds) dog bark(s)”. So “all” is not a logic word of the first kind according to Definition 4.3. We should note here that “all” is falling in the category “totality” of Kant’s ontology. “Exist” is a logic word of the first kind as its word graph is the “be”-frame, the empty frame, filled with “something SUB world”. In the framing and naming process, a subgraph of the mind graph is framed and in that way the definition of a concept C is given. The description is “ C is a ...”. Here “is” is a logic word of the first kind too as it describes the FPAR-relationship between C and its frame content. The famous “ISA”-relationship, like in “A dog is a(n) animal”, expresses the FPAR-relationship between “animal” (part of the definition of “dog”) and the “dog”-frame. So here IS in ISA is a logic word of the first kind too.

For the other logic words the classifying feature in their word graphs is chosen to be one of the other eight types of, binary, relationships. An important example is “causality”, basic in words like “cause” or “because”, which also is a category of Kant’s ontology. In knowledge graph theory we would classify according to the occurrence of a CAU-arc. The ALI-link, for “alike” concepts, corresponds to concepts in the “commonness”-category of Kant’s ontology and determines a set of words like “like”, “as ... as”, etc. Other logic words, so words with one of the eight binary relationships as dominant link in the word graph, are not really expressing aspects of pure logic, but then pure logic is not the whole of thinking, that we like to describe as “linking of somethings”.

In expert systems the question “why C ?” can be answered if causations are known. If B is a possible cause for C and A a possible cause for B , then the answer may be “Possibly because A ”. If we analyze this thinking process, we see that we start with C ,

and then note that we know that $B \xrightarrow{\text{CAU}} C$ and $A \xrightarrow{\text{CAU}} B$. By linking these data we obtain $A \xrightarrow{\text{CAU}} B \xrightarrow{\text{CAU}} C$, and have found A as possible cause. This process of linking is particularly interesting when the concepts are expanded, i.e. they are replaced by the content, of their frame, that is embedded in the rest of the graph considered. This replacement poses problems of its own, but we are not interested in them here. The expansion process plays an important role in thinking. Given some statements, say in mathematics, and the problem to prove that some goal-statement G is true if the given statements are true, then the way to find the proof may be the following. Expand all given statements as far as possible, with available further knowledge, till a graph A is obtained that has the graph G of the goal statement as a subgraph. The basic process of reasoning is namely that whenever a graph is considered to be true, each of its subgraphs must be true (under specific conditions on the structure of these subgraphs).

In trying to find the (answer) graph A one meets the difficulty of expanding the graph in the right direction. From the given statements expansion, combining of “true” graphs, may lead to many answer-graphs A that are all true, but none of which contains the goal graph G as a subgraph.

Both in this general process of reasoning and in the case of expert systems, a “rule based” version can be given. “If a graph A is true then its subgraph G is true” is the rule in the first case, but in natural language we would use the word “so” (which by the way turned up in class C_{12}): “ A so G ”. “If A then B ” and “If B then C ” are rule-versions for natural language descriptions like “ B because of A ” and “ C because of B ”. It is for this reason of almost equivalence in description that it makes sense to speak about other logic words. The rule-version has the pure logical setting in which the pure logic words are used. The statements have to be well-formed closed formulae. In natural language the thinking process is often described by non-well-formed statements that nevertheless correspond to certain subgraph of the mind graph that can be used in the description of the expansion process. We will see that this deviation from pure logic allows for dealing with some other linguistic aspects as well.

Definition 4.4 *Logic words of the second kind* are words the word graph of which contains one of the eight types of binary relationships of the knowledge graph ontology as dominant link.

We have given a restriction here by demanding that the relationship, e.g. the CAU-relationship, is a dominant link as otherwise all words would be logic words. Meant are those words that describe the linking process in its basic form. Word graphs with more than one type of binary relationship are to be excluded, unless one link is clearly dominant. In the first paper in the series on word graphs [Hoede & Li, 1996], the 15 different word graphs for the Chinese word for “in” were given. In them a SUB-link was clearly dominant. A preposition like “in” can therefore also be seen as a logic word of the second kind, used often in thinking about structuring the world. To determine which link is dominant we need some measure for dominance. In graph theory measures have been developed for the concept of dominance and they can be used to decide whether a word can be called a logic word of the second kind or not.

Finally, some discussion is due on the words “truth” and “true”, definitely words often used in logic. There are two ways of looking at these words. First there is the comparison of a statement or proposition p with a model of the situation expressed by p . The outcome of the comparison determines the truth-value of p , which in two-valued logic is “true” or “false”. For our knowledge graph view, what is happening is comparison of two mind graphs, one for p and one for the model. “Truth” can then be seen as equality of certain frames, one for p and one for (a part of) the model, and hence is a logic word of the first kind according to Definition 4.3. The truth values “true” and “false” are nothing but instantiations of the outcome of the comparison that may be replaced by, for example, the numbers 1 and 0, as is often done. These are not logic words.

A second way of looking at “truth” is as an attribute of a framed part of the mind graph. That part may represent the content of contemplation, or, more close to what was said before, the model of a situation, as perceived. Such a perceived situation may be held to be true or false, i.e. truth is the fact that both statement and model are parts of the mind graph. The model is also held to be a correct description of the state of affairs, whether this state of affairs is due to a presupposed “outer world”, as in physics, or due to ideas in an “inner world”, as in mathematics, where e.g. axioms are simply considered to be true. The statements describing axioms are not considered to be in need of comparison.

The knowledge graph theory slogan “the structure is the meaning” is in line with the

second way of looking at truth. The statement “it is raining outside”, a standard example, does not need comparison with a model. The structure of the part of the mind graph associated by the listener with the statement is all that matters, as far as meaning attribution is concerned. The truth of the statement is depending on the comparison, with the outer world. In so-called *truth conditional semantics* this comparison is stressed. In our *structural semantics*, the outcome of such a comparison is irrelevant. As a major consequence of our stand even statements that are not well formed also have a well-defined semantics as far as the corresponding mind graph frames are well-defined. A statement like “ $x < 5$ ” is considered to have no well-defined truth conditional semantics even when a model is given, with proper domain and interpretations, because x is free. Any knowledge graph constructed by a mind as corresponding to the statement is the meaning of that statement in structural semantics.

4.3.2 Classification of logic words of the first kind

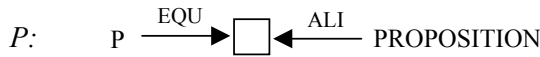
Due to our chosen Definition 4.3 we would have to construct the word graph to decide whether a word is a logic word of the first kind. We already discussed the operators from proposition logic and the quantification operators. Also modal logic operators were discussed. Of all these only the universal quantification did not involve a frame, but was expressed by a SKO-loop and was therefore a logic word of the second kind. However, universal quantification can also be represented by frames. We will come back to this in Chapter 6.6. The first set of *logic words of the first kind* corresponds to the four frames themselves and is given in Table 4.I.

WORD GRAPH	WORD	PARA-PHRASE	CHINESE WORD	LITERALLY
Frame	Inherence	be with	gu4 you3 xing4	primitive gender
Negframe	Negation	be not	fei1	not
Posframe	Possibility	be possible	ke3 neng2 xing4	possible gender
Necframe	Necessity	be necessary	bi4 ran2 xing4	necessary gender

Table 4.I The first set of logic words of the first kind.

Remark first the use of the word xing4, “gender”, which is used to describe the occurrence of an alternative. Literally, possibility is circumscribed by “possible male/female”, where male/female only functions to express the two values for possibility, possible/impossible. Secondly, the word “Fei1”, for “negation”, used in the context of logic, literally must be translated as “not”. We have chosen the words inherence and negation as these are two of Kant’s categories. Note that the word “negation” has a subjective undertone. Similarly, any subgraph that is framed and named gives a concept with the subgraph as *inherent* property set. The word “inherence” clearly expresses more than just “being”.

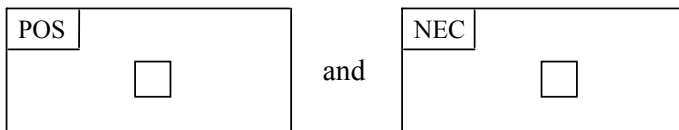
The second set of logic words of the first kind has graphs containing one of the four frames next to other parts. Let a graph P , corresponding to a proposition p , be contained in a frame, which may be described by “it is so that p ”, or simply by “being p ” or even just “ p ”. In the knowledge graph formalism the graph



would be given, or, simpler, \boxed{P} . A frame containing the frames \boxed{P} and \boxed{Q} is the representation of $p \wedge q$, or p AND q in natural language. The word graph for “and”, “he2”, respectively “ \wedge ”, “yu3”, is

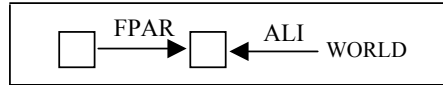


without specifying the contents of the two inner frames. By functional completeness other connectives in proposition logic are expressible by the “and”-frame and the NEG-frame. Consider a NEG-frame containing a proposition graph P , then it can be described by “it is not so that p ”, or simply by “negation p ” or “not p ”. Omitting p from the graph the word graph for the word “not” results. In Chinese this is described by, “bu2 shi4 p ”, literally “not be p ”. Likewise the POS-frame and the NEC-frame allow expressing “possible p ” respectively “necessary p ”. Omitting p again the word graphs for “possible”, “ke3 neng2 de”, and “necessary”, “bi4 ran2 de”, result as



Note that in Chinese the word “de” is used to express the fact we are dealing with an adjective.

The frame corresponding to “being” may contain the graph of something considered to exist in the world, so essentially the graph



describes the word “existence”, “cun2 zai4 xing4”, literally “existence gender”. Further information on the “world” considered, which may e.g. be a set of numbers, may be added. We now have word graphs for the logic words of the first kind describing logical operators, see Table 4.II.

LOGICAL OPERATOR	WORDS	CHINESE WORDS	LITERALLY
Proposition logic	And	yu3	and
	Not	fei1	not
	Or	huo4	or
	If...then	ru2 guo3...ze2	if...then
	If and only if	dang1 qie3 jin3dang1	when and only when
Predicate logic	Existence (of)	cun2 zai4	exist
Modal logic	Possible	ke3 neng2 de	possible of
	Necessary	bi4 ran2 de	necessary of

Table 4.II The second set of logic words of the first kind.

Note that for “if”, the word “ru2 guo3”, is used in “if... then”, whereas in “if and only if”, the word “dang1”, is used, literally meaning “when”.

We should take a stand with respect to other logic like tense logic, deontic logic, or fuzzy logic and the words used therein. Consider for example the word “obligation” in deontic logic. It may be argued that it is a word like “possibility” or “necessity”, but also that obligation is just an attribute of something, say an attribute of an act, attached to the act by the speaker, according to his norm system. If we follow the first

argument, we might introduce an OBL (obligation)-frame, analogous to the POS-frame and the word would have to be considered a logic word of the first kind. In that case we consider “obligation” to refer to a given set of rules. We follow the second argument and do not consider “obligation” to be a logic word of the first kind or even an logic word of the second kind, but to be similar to a word like “beauty”.

Words used in tense logic are words like “present”, “past”, “future”. They are expressible in terms of the ORD-relationship concerning time values, so at most fall under logic words of the second kind in case the ORD-arc is really dominant in the word graph. “Fuzziness” basically involves unprecise information about values of something. A fuzzy word from natural language is for example “youth”. No specific frame or basic relationship seems present here. The word graph for “youth” will clearly be quite complex.

Concluding, the logic words of the first kind are those mentioned in the Tables 4.I and 4.II. For those words that have word graphs very close to those for these pure logic words we take the stand that although frames are used, the essential meaning is not expressed by the frame. One example should suffice here. The word “both” is used in expressions like “both a and b are numbers”, meaning “a is a number and b is a number”. “Number(a) \wedge Number(b)” may be the formulation in predicate logic. The essential meaning of “both” is that the predicate holding for a and b is the same, the “and”-connective has the, different, meaning of combining two propositions that may have no further commonness at all. “Both” is therefore not considered to be a logic word of the first kind. Whether it should be seen as a logic word of the second kind, because the EQU-relationship is clearly present, will be discussed in the next section.

4.3.3 Classification of logic words of the second kind

For the logic words of the first kind the four frames in the ontology of knowledge graph theory were chosen as defining entities. For the logic words of the second kind we choose the eight binary relationships.

The representation we choose is that of the total graph in which the arc is also represented by a, labeled, vertex. The reason for this is that we can then consider subgraphs of the graph $\square \longrightarrow \boxed{\text{TYPE}} \longrightarrow \square$, where the label TYPE may be

one of the eight types we distinguish. Table 4.III gives the words corresponding to the whole graph of three vertices and the graph in which an encompassing be-frame is considered as well. For the type EQU the graph is considered to represent the word “equal”, whereas “being equal” is considered to be a synonym of “equality”.

TYPE	WORD	WORD(+BE)	CHINESE WORD	CHINESE WORD(+BE)
EQU	Equal	Equality	xiang1 deng3	xiang1 deng3 xing4
SUB	In	Containment	li3	bao1 han2 xing4
ALI	Alike	Community	xiang1 xiang4	gong4 xing4
DIS	Distinct	Disparateness	bu4 tong2	cha1 yi4 xing4
ORD		Ordering	shun4 xu4	pai2 xu4
CAU	Causation	Causality	qi3 yin1	yin1 guo3 xing4
PAR		Attribution		te4 zheng1
SKO		Dependency	ying4 she4	yi3 lai4 xing4

Table 4.III The words corresponding to the graph $\square \longrightarrow \boxed{\text{TYPE}} \longrightarrow \square$.

Some remarks are due here. First, we could not determine an English or Chinese word for the PAR-link. Secondly, we notice that for the ORD-link, within a BE-frame, in Chinese the alternative indicating word xing4, “gender”, is not used. Thirdly, the Chinese words tend to extend the description considerably, i.e., a more complicated word graph is expressed. “Bao1 han2”, literally means “around inside”, hence “containment”. This hints at the fact that also the English word “containment” expresses more than just a SUB-link in a BE-frame. This explains why the two columns for Chinese words given are so different apart from the words for “equal” and “equality”. It is simply so that not every knowledge graph has a precisely describing word. We might even have left more places open in the table for “lack of words”. Now let us consider the subgraphs of the form $\square \longrightarrow \boxed{\text{TYPE}}$ and the corresponding words, given in Table 4.IV, and those of the form $\boxed{\text{TYPE}} \longrightarrow \square$, given in Table 4.V.

Note, as discussed in [Hoede & Li, 1996], that the merological stand is that SUB,

PAR and FPAR are the three merological relationships, mixed up in the English language by the fact that the words “of” and “with” are used in all three cases, like in the examples:

- Part *A* of *B*, respectively *B* with part *A*,
- Attribute *A* of *B*, respectively *B* with attribute *A*,
- Property *A* of *B*, respectively *B* with property *A*.

TYPE	WORD	CHINESE WORD	LITERALLY
EQU	Of...	... de yi1 bu4 fen4	... of one part
SUB			
ALI			
DIS	From	cong2	from
ORD			
CAU	By	you2 ... yin3 qi3	have ... cause
PAR	Of	... de te4 zheng1	... of attribute
SKO	Dependent(on)	yi3 lai4	depend on

Table 4.IV The words corresponding to the subgraphs of form $\square \longrightarrow \text{TYPE}$.

TYPE	WORD	CHINESE WORD	LITERALLY
EQU	Equal (to)	deng3 yu2	equal (to)
SUB	With	ju4 you3...	have ...
ALI	Similar(to)	xiang1 si4 yu2	similar to
DIS	Distinct(from)	bu4 tong2 yu2	not equal to
ORD	To	dao4	to
CAU	With	ju4 you3 ... te4 zheng1	have ... attribute
PAR			
SKO			

Table 4.V The words corresponding to the subgraphs of form $\text{TYPE} \longrightarrow \square$.

Or, more concretely:

- The tail of the dog,
- The beauty of the dog,
- The barking of the dog,

if barking is part of the definition of “dog”. Some people define a dog as “something that barks and sniffs”.

The graphs $\square \longrightarrow \boxed{\text{FPAR}}$ and $\boxed{\text{FPAR}} \longrightarrow \square$ are worded “of” and “with” too. In Chinese the two graphs can be described by “de” respectively “you3”. In a more elaborate description they can be uttered by “...de xing4zhi” respectively “ju4you3... de xing4zhi”, as semantically the FPAR-relationship is used to modal a property-relationship. In Chinese many words are used to describe what, in English, would be described by “part”, “attribute” or “property”. Like for the different words for “in”, we can make word graphs for all these words describing merological relationships.

It is quite remarkable that all these words used in Chinese to describe merological relationships, which fact supports our choice of only three merological relationships can indeed be modeled with the three types occurring in the knowledge graph ontology. This was not so apparent from English, where only a few words are at our disposal.

The EQU, ALI and DIS-relationships are symmetric. This is probably the reason why in Table 4.IV no words are given, which is due to the fact that they do not seem to be present in language.

If y is a function of x , $x \xrightarrow{\text{ALI}} \square \longrightarrow \boxed{\text{SKO}} \longrightarrow \square \xleftarrow{\text{ALI}} y$, we say that y depends (is dependent) on x . Special attention should be paid to the SKO-loop that is considered, by van den Berg and Willems, to represent the words “all”, “sou3you3”, “each”, “mei3ge4”, “every”, “mei3ge4”, and “any”, “ren4yi4”. The authors consider the four words to be slightly different so that four different graphs should be presented, although the SKO-loop, indicating something that is informationally dependent only on itself and hence can be anything, clearly stands central. In Chinese three different words are used. We will come back to this in Chapter 6.

Also note that the CAU-relationship, with chosen word “by” is somewhat out of line

with the other relationships that seem more basic as structuring relationships. “You2 ... yin3 qi3”, literally means “by ... cause”.

Of Kant’s categories we can, with some difficulty, recognize the following six: Inherence, Negation, Possibility, Necessity, Limitation and Causality. “Existence” we consider as “being in the world”, so to be less basic than “being”, “Reality” in our subjectivistic theory is an assumption about correspondence between our image in the mind and a presupposed outer world. We do not discuss the quantity categories of Kant: unity, plurality and totality. Rather, we want to focus on commonness, that we take to be synonymous with likeness, and that puts the ALI-link central.

The ALI-link might be called a first among equals as the process of concept formation is seen to result from discovering similarity between a set of perceived objects. A word evokes different subgraphs in different mind graphs, although probably with great similarity. Using the word for a part of its complete meaning, for the speaker, allows to use the word metaphorically. The listener may yet understand the used word in the proper way by searching for other concepts that contain as part of their word graph the same part as envisaged by the speaker. “The big bird landed on Schiphol airfield” uses “bird” for “something that flies”. The listener may see this part of speaker’s word graph for “bird” as the intended focus of his word choice and may see it as part of listener’s word graph for “plane”. Speaker is considered by listener to have used the word for a part of a concept to indicate the whole concept.

Like for the logic words of the first kind, there is the problem to determine the borderline of the logic words of the second kind. “Alike” is the word with word graph $\square \overset{\text{ALI}}{\text{---}} \square$, and hence a logic word of the second kind by Definition 4.4, but what about “like”? “*A* and *B* are alike” is a rather clean statement, but in “he works like a horse” the ALI-link stands central but in the sentence graph the words “works” and/or “horse” have to be expanded in order to localize the parts that are similar. The simple expansion of “horse” to “working horse”, one of the things a horse can do, already enables the localization, although the specific feature, on which the likeness meant by the speaker is based, namely working hard, is not yet part of the expansions. The point is that next to the ALI-link more structure is needed in order to give a proper word graph for the word “like”. This is similar to the situation for the word “in”, discussed in [Hoede & Li, 1996]. The single SUB-arc gives the word graph for

“in” in English, but for the fifteen different Chinese words for “in” we have fifteen different word graphs. We would prefer to call “like” and the Chinese words for “in” logic words of the second kind, but that implies that more complex word graphs than those existing merely of one of the basic types of arcs or parts of them are to be called word graphs of logic words. That again brings forward the problem to determine a borderline. In tense logic the word “past” clearly involves an ORD-arc to time of the speech act from the time of the act described by the verb. In “I say (at time t_0) John worked (at time t_1)” we have $t_1 < t_0$ on the time scale, or $t_0 \xrightarrow{\text{ORD}} t_1$ in short notation. We say that the verb “work” has the past tense. The essence in “past” is as much the ordering as the fact that we are talking about time. In “the lower floors of the building” the word “lower” involves an ordering too, but now concerning spatial coordinates. Both “past” and “lower” may be considered to be logic words of the second kind, as the ORD-arc stands central. However, it is increasingly becoming more difficult to classify the word as a logic word when the graph becomes more complex. A linking of concepts by one of the eight basic types of arcs is predominant in prepositions, these are logic words of the second kind according to Definition 4.4. It will also be clear that the classification formed by subjective coding of two coders and leading to the classes C_{11} to C_{55} mixes up logic words of the first kind and logic words of the second kind as distinguished in this section. Apparently in our minds no sharp borderline is present to distinguish logic words from other words. One might even take the stand that all words that are not noun, verb or adword are forming the material used to link these basic words. Here one should remark that with the noun its, named, instantiations are included. The number “2” is an instantiation of the noun “number”. Likewise all plant names are instantiations of the noun “plant name”. All plants in the extension of “plant”, however indicated, would also belong to the set of basic words. An objection to this stand is that there are verbs like “be”, “can” and “must” that are clearly pure logic words. But then, these verbs are not, without reason, called auxiliary verbs. In fact in Chinese “can” and “must” are not considered to be verbs. The verb “be”, “shi4”, is often missing in a sentence, no other verbs being present. The nouns indicating the categories in Kant’s ontology, as far as we already interpreted them in the tables, would have to be excluded as well and this poses a more serious problem, as for example “possibility” was considered a typical word of the first kind. But then, it might be considered to be a concept that expresses something attributed by the mind, like “beauty”.

Due to our view on word formation a classification should be based on the concept. Although representing a concept is affected by sociology, philosophy, psychology and so on, it is independent of the difference in languages such as Chinese and English. Whether we choose Chinese or English to express a concept, that is the same in the minds of a Chinese or an Englishman, the structure of the concept is the same. There may of course be language specific concepts. Before taking a more firm stand on the choice of the classification principle, we will try to give knowledge graphs for some potential logic words, as knowledge graphs are specifically designed for representing the structure of concepts.

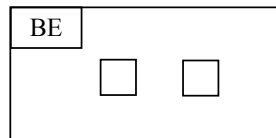
4.4 Word Graphs for Logic Words

For our discussion about the classification problem for logic words it is useful to give a survey of the actual word graphs. For the process of structural parsing we need to have word graphs for all words anyhow.

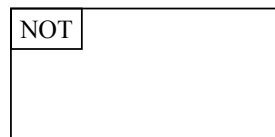
We already considered as first set prepositions, [Hoede & Li, 1996], then Chinese classifiers and adwords [Hoede & Liu, 1998] next to the many nouns and verbs that, in first instance, have very simple word graphs, just describing the word as type. In the following listing we distinguish types of structures of logic words. Types of frames are indicated in the left upper corner.

4.4.1 Proposition operators

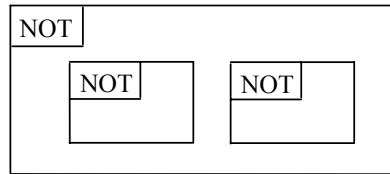
AND:



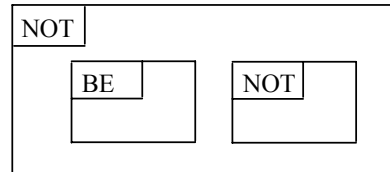
NOT:



OR:



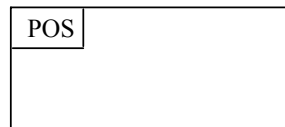
IF ... THEN ...



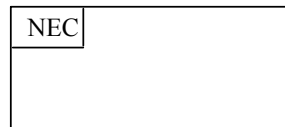
According to Definition 4.3 these are word graphs for logic words of the first kind.

4.4.2 Modal logic operators

POSSIBLE:

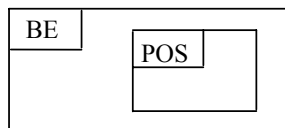


NECESSARY:

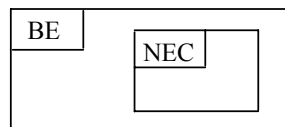


The nouns “possibility” and “necessity” have word graphs, corresponding to “being possible” and “being necessary”.

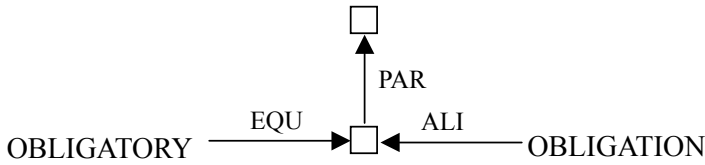
POSSIBILITY:



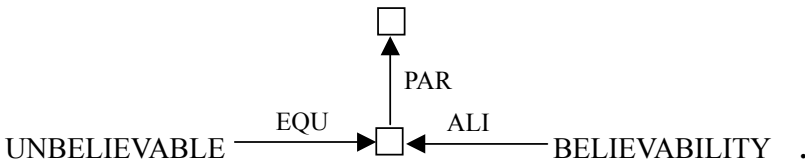
NECESSITY:



Within the frames, graphs, that describe propositions, can be present and we have a formalism, strictly parallel to “normal” notation for logic, see van den Berg [Berg, 1993]. If we would allow two other types of frames, according to “obligatory” and “believable”, analogous graphs could be given for “obligation” and “believability”. However, whether something is obligatory or believable is a subjective matter. Hence a PAR-arc, for attribution, seems more appropriate and we would have word graphs like



and



The adjectives are seen as instantiations of the nouns. If “possibility” and “necessity” are seen as judgments, analogous word graphs could be given, but we would clearly get away from the formalism of logic.

4.4.3 Quantification

Existential quantification is expressed by an explicit knowledge graph in which a variable is instantiated, i.e. a token has been vaulted. Unevaluated tokens correspond to free variables. Note that there may be many graphs of the same structure only differing in the instantiation. All these graphs correspond to “there is an x such that...” by explicitly mentioning the “x”. We follow van den Berg and Willems and represent universal quantification by an SKO-loop:



However, we will come back to universal quantification in Section 6.6.

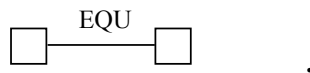
4.4.4 Logic words based on set comparison

As we distinguish eight types of binary relationships, we would have eight classes of logic words of the second kind to consider. However, as was discussed in [Hoede & Li, 1996], the types are to be divided into three groups, four based on set comparison, two based on the structure of space-time and two based on mind processes.

(1) The equ-link

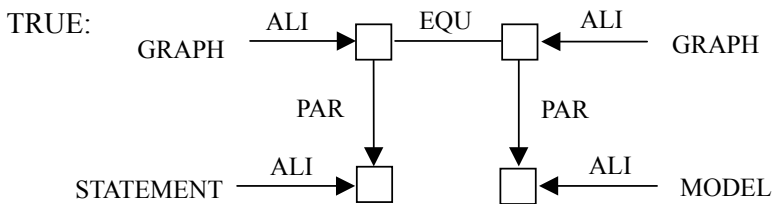
The basic word here is of course

EQUAL:



With a BE-frame we obtain the word graph for “equality”. There are quite a few words in which the word “equal” occurs, like e.g. equivalent. The only word for which we would like to give a word graph here is “true”. The graph of the statement compared with the graph of the model in which the statement is interpreted should be *equal* to make the statement true.

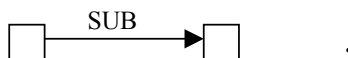
Hence



could be given as the word graph for “true”. Note the central position of the EQU-link. “Truth” is then again obtained by putting a BE-frame around the word graph for “true”.

(2) The sub-link

IN:



This preposition was extensively discussed in [Hoede & Li, 1996]. The SUB-arc is typically structuring entities. There are many words of which “sub” is a part. Should

all these words be called logic words? Is “subset” mainly describing a set or is the word mainly expressing a “part of” relationship? In a general classification, according to Definition 4.2, the specific purpose of the classifier is decisive. When we want to gather all words showing structuring aspects we should include all “sub”-words in the set of logic words of the second kind and give predominance to the “part of” aspect.

(3) *The ali-link*

The basic word is of course “alike”



We recall that this type of link may be considered the first among equals as it is considered to be the basis of the framing and naming process. Having discovered the likeness in the examples of a species the mind may form a prototype structure and frame and name this. We mention the ALI-relationship as playing an important role in metaphors. It occurs in the word graphs of word like “similar”, which is almost a synonym of “alike”, or a word like “seem”.

(4) *The dis-link*

The example of two sets with empty intersection, disjoint sets, is appropriate here.



Like for “sub”, there are many words including “dis”. For the same reason, if we want to list all structuring words as logic words of the second kind, all these words are considered to be so, even if this means including verbs like “disrupt” or “discover”.

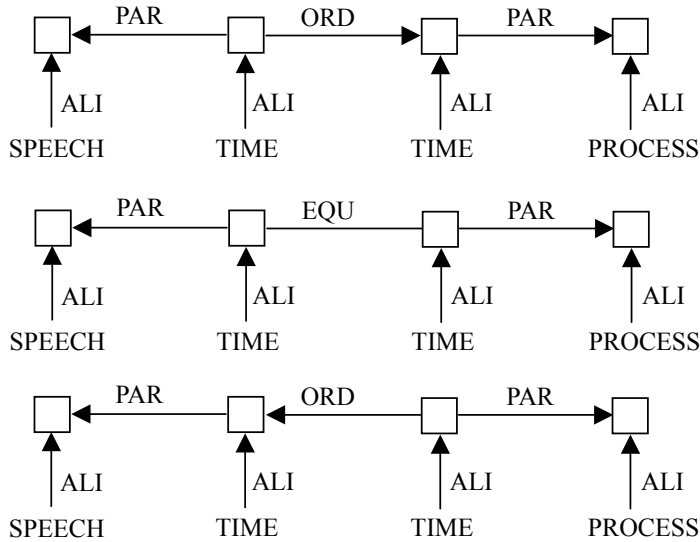
4.4.5 Logic words referring to space and time

For the reflection upon space and time we assume that two basic types of relationships suffice, the ORD-link and the CAU-link, although the CAU-link may be a composite according to the philosopher Hume, whom we are inclined to follow.

(1) *The ord-link*

Here again we refer to the first paper on prepositions, in which the ORD-link is the main link. Words like “from”, “to”, “before”, “after”, “under”, “behind”, “above” etc.

all have word graphs in which the ORD-link stands central. The words used in temporal logic mainly refer to some kind of ordering. Interesting examples are the tenses of a verb; present, past and future. Like we said before, two values of time are involved, the time of speaking and the time of the process described by the verb.



are describing that the process occurs after, during, respectively before the speech act. In a rather complicated way these graphs will be present when “future tense”, “present tense”, respectively “past tense” would have to be described, as “tense” refers to the form of the verb that describes the process. We do not give such graphs but rather mention that these three graphs without the token for process, so in smaller form, would be word graphs for “at a time after speech”, “at the time of speech”, respectively “at a time before speech”, which is usually described by “later”, “now” respectively “earlier”. These words, and analogous triples like “tomorrow”, “today” and “yesterday” are considered to be logic words of the second kind. The ORD-link typically occurs in word graphs for words that are used in comparisons.

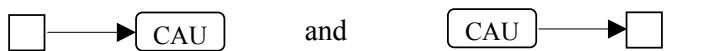
(2) The cau-link

We do not have a good word for the graph



the word “causing” seems to come closest. With a BE-frame around it we might have

the word graph for “causation”. This again sheds some doubt on the CAU-link being basic, now in a different way. Subgraphs of the total graph like



however, may be seen as word graph, for “cause” and “effect”, “something that is causing” and “something that is caused”. Causal relationships are very important in building expert systems or decision support systems. First expert systems in medicine were rule-based systems where the rules were formulated in the “if A then B ” form instead of “ A causes B ”. For this reason and because we chose the CAU-link in our ontology, the words with word graphs in which the CAU-link stands central are considered to be logic words of the second kind as well.

4.4.6 Logic words due to mental processes

The last two types are the PAR-link for attribution and the SKO-link for informational dependency.

(1) *The par-link*

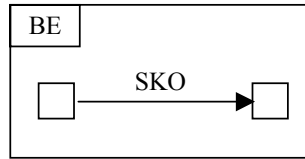
Next to the FPAR-link and the SUB-link, this is the third merological relationship. All three occur in the word graphs of prepositions like “of” and “with”. The PAR-link, with BE-frame describing “attribution”, is typically used for adjectives, see [Hoede & Liu, 1998]. It is structuring thought like the other types of link do, but also has syntactic aspects, in that it links certain types of words, adverbs, to other types of words like nouns and verbs. The SKO-link, which we use for describing the functioning of verbs, has such a syntactic aspect as well, in that it determines subject and object (if present). Although the ending “-ly” in adverbs refers to the attribution to verbs, and hence to the PAR-link, we are not inclined to say that the PAR-link stands central in adverbs and do not call these words logic words.

(2) *The sko-link*

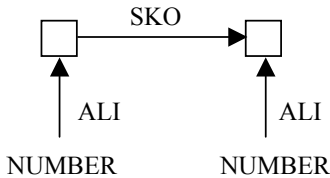
The SKO-loop was used for universal quantification. The typical example for the SKO-arc is mapping or function, as known from mathematics. If $y = f(x)$ indicates that y is a function of x , we say that y is informationally dependent on x , which is essentially described by:

DEPENDENT ON:  ,

DEPENDENCY being described by:



In natural language words for which the SKO-link stands central in the word graph are rather rare. The concept “function”, as mapping from numbers to numbers, is used in mathematical language and has word graph:

FUNCTION:  .

Like in the case of “functional” and “mapping”, for mapping of arbitrary objects to numbers respectively mapping of arbitrary objects to objects, the SKO-link stands central in the word graph for “function”. Words like these are called logic words of the second kind as well.

4.4.7 Words used in other logics

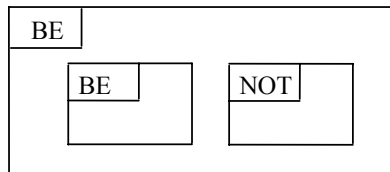
We have seen in Section 4.4.5 that words used in temporal logic would not be called logic words in our classification according to Definitions 4.3 and 4.4. However, the important role played by the ORD-link makes us say that many of these words fall in the category of logic words of the second kind. There are other logics like deontic logic or fuzzy logic, where according to our classification we would not speak of logic words as none of the basic types of n-ary or binary relationship plays an important role, although we have mentioned that for deontic logic an OBL-frame might be introduced. In that case words like obligatory would be logic words of the first kind. We prefer not to do this, as ethical valuations are essentially subjective. Likewise, for words with a fuzziness aspect like “youth”, “old” or “somewhat” we do not have the structuring aspect that other logic words have. A word that seems to stand central here is “extent”, which is basically a measure for inclusion. The value of this measure may

be called the fuzziness of the inclusion. With “inclusion” we may think of sets, but the word is used in a broader sense here. The quantitative aspect stands central. For this reason we would not speak of logic words in case of fuzziness.

4.4.8 Words linking sentences

The last group of logic words is special in that they link sentences. Examples are “however”, “nevertheless” or “but”. These words are frequently used in a reasoning process. Within the same sentence the word “although” links sub-sentences in the same way. Let p and q represent two sentences connected by the word “but”. Then the meaning is considered to be “it is not so that p implies q”, or $\neg(p \rightarrow q)$ in logical notation. As this is equivalent to $\neg(\neg p \vee q)$, which is equivalent to $(p \wedge \neg q)$, the word graph would be

BUT:



and therefore we should classify such words as logic words of the first kind, although two types of frames are occurring.

4.5 Conclusion

It will be clear for which type of classification we have chosen; type c), classification by using knowledge graph ontology. In Section 4.4 we have tried to apply our Definitions 4.3 and 4.4 for classifying pure logic words and other logic words. The main problem that we encounter is to make a decision about the demand “stands central” with respect to a specific type of link. The problem is similar to the problem of delineating the meaning of a word. In knowledge graph theory the structure, of the graph, is the meaning, but what are the boundaries of the specific subgraph of the mind graph? The extreme stand was taken that the whole graph focused upon, that contains the word graph, is the meaning. This encompasses the word and all its associations, i.e. background knowledge. If the same stand would be taken with respect to logic words, their word graphs would not have natural boundaries as well.

The taken stand is rather off line with the mathematical logical tradition in which it is the goal to give a meaning that is as concise as possible. In knowledge graph theory interpretation this means that the word graph given should be the smallest graph still considered to give the meaning of the word, one would say the essential meaning.

For our classification problem we face the problem that we would like to call a word a logic word or not. Here some frame or some basic type of link may be embedded in a larger graph that may still be the smallest graph giving the essential meaning of the word. The question is: does the word graph contain the link like any other type of link or is its structure so that the link grasps the essentials of the meaning. The smaller the graph, the easier it is to answer this question. We have made choices in Section 4.4, calling some words logic words and other words not. This unsatisfactory feature is accepted because for the process of structural parsing that we have in mind the distinction between logic words and other words becomes irrelevant.

The words and linguistic elements remaining, the fourth set, will have to contain graphs for linguistic elements like question marks, exclamation signs, to name a few.

Chapter 5

Structural Parsing

Parsing is an essential part of natural language processing. In this chapter, structural parsing, which is based on the theory of knowledge graphs, is introduced. Under consideration of the semantic and syntactic features of natural language, both semantic and syntactic word graphs are formed. Grammar rules are derived from the syntactic word graphs. Due to the distinctions between Chinese and English, the grammar rules are given for the Chinese version and the English version of syntactic word graphs respectively. By traditional parsing a parse tree can then be given for a sentence that can be used to map the sentence on a sentence graph. This is called *structural parsing*. The relationship with utterance paths is discussed. As a result, chunk indicators are proposed to guide structural parsing.

5.1 Introduction

A natural language processing system always contains a parser, which is a device that has a natural language sentence as input string and that produces a representation of the sentence when it is acceptable. Parsing is the process of structuring a representation of a natural language sentence usually in accordance with a given grammar. There are two important points here; one is that we require a representation

as an interlingua (inter-transmittal language) that is standing between the natural language sentence accepted and its access structure in a computer, the other is that we require grammars with which the natural language acts in accordance. The former point is independent of the specific language, the latter is dependent on the specific language.

We chose knowledge graphs as the interlingua, due to their advantageous properties in NLP. Since based on knowledge graph theory, parsing is more special than traditional parsing methods and is called structural parsing. The structural parsing that is introduced aims at transferring (or giving a meaning to) this sentence. A sentence graph is built from word graphs, which just stand for meanings of the words contained in this sentence. This means that word graphs are at the base of parsing.

This chapter will discuss the theory of structural parsing, and then take Chinese and English sentences as examples to support our theory. In Section 5.2 semantic and syntactic word graphs are introduced and the grammars for English and Chinese, based on them, are given in Section 5.3. Section 5.4 is the central section and contains the discussion of utterance paths and their relation with parsing by chunks, which is our approach to structural parsing. Some examples are discussed.

5.2 Syntactic and Semantic Word Graphs

We are interested in word graphs in term of which a sentence will be analyzed. It is fortunate that word graphs were already discussed for prepositions, adwords and logic words. Here, another aspect of word graphs is discussed, namely the semantic and the syntactic representation of a word by word graphs.

5.2.1 Definitions of syntactic and semantic word graphs

To analyze a sentence to obtain a sentence graph, two pieces of information are necessary; one is the meaning of the words that constitute this sentence, the other is the function of the words, which is called syntactic information. Considering the semantic and the syntactic information of natural language, we develop semantic word graphs for the meanings of words and syntactic word graphs for the syntactic function of words. We give the definitions for semantic and syntactic word graphs.

Definition 5.1 A *semantic word graph* is a word graph, which expresses the meaning of a word.

The three papers [Hoede & Li, 1996], [Hoede & Liu, 1998] and [Hoede & Zhang, 2001a] on word graphs concern semantic word graphs.

Definition 5.2 A *syntactic word graph* is a word graph, which expresses the syntactic functions of a word.

For example, in Chinese the word “wo3” means “I”, and this pronoun has at least the following three usages:

- subject
- object
- attribute.

We can express its functions with three different knowledge graphs as in Figure 5.1.

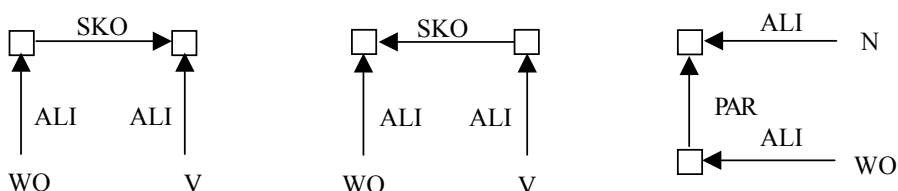


Figure 5.1 Syntactic word graphs for the pronoun “wo3” corresponding to “I”, “me” and “my” in English.

The word graphs in Figure 5.1 give the different syntactic functions of the word “wo3”. We have chosen to represent the subject function respectively the object function by a SKO-arc, to respectively from a verb V, where on the semantic level we would choose a CAU-arc. The possessive use of “wo3” is expressed by a PAR-arc. We can also express the three syntactic functions with one knowledge graph like in Figure 5.2, which is called the syntactic word graph of the word “wo3”.

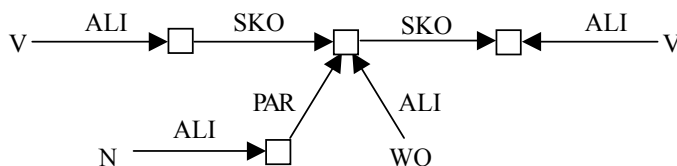


Figure 5.2 Syntactic word graph for the pronoun “wo3”.

The meaning of the word “wo3” is expressed by another word graph, which is called the semantic word graph of the word “wo3”, see Figure 5.3.

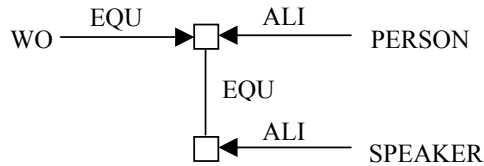


Figure 5.3 Semantic word graph for “wo3”.

Consider another Chinese word “ta1”, in English “he”, “him”, or “his”. We obtain that the syntactic word graph of the word “ta1” is the same as for the word “wo3”. In fact many other words, in this case pronouns, have the same syntactic functions in a sentence, so their functions can be expressed by the same syntactic word graph. The question now is how many different syntactic word graphs there are. This depends on how many word types there are, which will be discussed in the next section. Syntactic word graphs are essentially semantic word graphs for word types. Willems [Willems, 1993] introduced syntactic graphs, but took prepositions and function descriptions like subject or object as labels of arcs. We use the basic ontology of knowledge graphs for all word types, including e.g. prepositions.

5.2.2 Word types for Chinese and English

In Chinese the problem of word types is more complex than in English. There is no Chinese dictionary with word types till now. But if there are no word types in our lexicon, the intended structural parsing is impossible. Therefore, it is necessary that we first classify the types of Chinese words. In English there is no problem, we do not need to reclassify the words.

Definition 5.3 *Word types* are the types of words, classified in terms of their syntactic functions.

It is a problem how to divide Chinese words into types. One of the reasons is that Chinese words show no change of shape feature of words, the type of a word always changes according to the context. Let us consider the following two Chinese sentences:

(1) Ta1 you2yong3 le.
 (He swam, literally “He swim past time”.)

(2) You2yong3 you3yi4yu2 jian4 kang1.
 (Swimming is good for the health, literally “Swim has good for health”.)

The Chinese word “you2yong3”, is the same in these two sentences, but has different word types. In sentence (1), “you2yong3” is a verb, “le” indicates that the sentence is in past tense. In sentence (2), “you2yong3” is a noun. There is no change in word shape. What about English? Just look at the same sentence. There is the word “swam” to express the past tense and another word “swimming” to be recognized as a noun.

Although no dictionary mentions word types, there are many views about how to classify Chinese words. According to Zhu, a Chinese linguist, Chinese words are to be classified into 22 word types [Zhu, 1984]. We list them in Table 5.1, representing Chinese word types with pin1yin1 and also give the names of these types in English.

The types listed here are based on the syntactic functions of words. For example, on the one hand, in the sentence “ta1 zuo2tian1 lai2 le (He came yesterday.)”, the word “zuo2tian1 (yesterday)” here modifies a verb “lai2 (come)”, and looks like an adverb syntactically. On the other hand, in the sentence “zuo2tian1 shi4 qing2tian1 (yesterday was sunny)”, the same word “zuo2tian1 (yesterday)” looks like a noun syntactically. For this reason it is neither an adverb nor a noun, it is considered by Zhu to belong to the type of “time”.

Based on the 22 word types of Zhu, 71 word types were classified by Yao [Yao, 1995]. He reclassified each of the above 22 word types into many subtypes, because he has to pay more attention to the word types semantically in order to make natural language processing possible.

In our theory we represent semantic and syntactic features of natural language separately by forming semantic word graphs for semantic features of a word and syntactic word graphs for syntactic features of a word. Typically semantic aspects, like time or location, should not occur on the syntactic level of word types.

When we classify word types, our purpose is to build a grammar that is independent of semantic features and dependent only on syntax. Due to this, we do not need to mention the semantic aspect of a word and can just concentrate on syntax. This is why

in principle we agree with the types of Table 5.I that refer to the syntactic view to classify word types. Due to this we can reduce the bigger set of word types into a smaller one.

CHINESE	ENGLISH
ming2 ci1	noun
dong4 ci1	verb
xing2 rong2 ci1	adjective
dai4 ci1	pronoun
shu4 ci1	numeral
liang4 ci1	classifier
shi2 jian1 ci1	time
zhu4 ci1	auxiliary
jie4 ci1	preposition
fu4 ci1	adverb
zhuang4 tai4 ci1	state
lian2 ci1	conjunction
fang1 wei4 ci1	orientation
yu3 qi4 ci1	modal particle
tan4 ci1	interjection
chu4 suo3 ci1	location
xiang4 sheng1ci1	onomatopoeia
qu1bie2 ci1	comparative
qian2 zhui4	prefix
hou4 zhui4	suffix
biao1 dian3 fu2 hao4	punctuation
ci2 su4	morpheme

Table 5.I Classification of Chinese words according to Zhu.

From our view, as a first reduction, we do not think a “mark”, such as “!”, “?”, “...”, etc. should be a type of word, although they play a very important role in expressing meaning. This belongs to another problem area that we call sentence patterns. We will not discuss sentence patterns in this thesis.

Secondly, we know that prefix, suffix as well as morpheme are very important linguistic concepts, especially in English. However, they belong to the word building problem. We leave these out as well.

Thirdly, conjunction is always used to describe a connection between sentence and sentence, playing a key role in reasoning. Here we just consider simple sentences, so we leave it to further research too.

Fourthly, modal particle words are very interesting ones. They are very small words and are only used in Chinese. They have at least four usages. Let us give the four usages and example words, because they are unique features in Chinese. We cannot give corresponding English words.

Modal particle words are used for:

- statement : such as “de”, “le”, “ni”, “ba le”, “a”
- question : such as “ma”, “ni”, “a”
- suggestion : such as “ba”, “le”, “a”
- exclamation : such as “la”, “a”.

Now we give four sentences to explain these distinctive usages. We also try to express each sentence in English, according to the association with the modal particle word.

(3) Wo₃men₂ qu₄ you₂yong₃ le.

(We went to swim.)

(4) Wo₃men₂ qu₄ you₂yong₃ ma?

(Do we go for a swim?)

(5) Wo₃men₂ qu₄ you₂yong₃ ba.

(Let's go to swimming.)

(6) Wo₃men₂ qu₄ you₂yong₃ la!

(It's very nice that we are going to swim!).

Note that all the modal particle words happen to appear at the end of a sentence, so that there is no problem in parsing. Modal particle words can be cut off from the sentence. As for distinguishing the meaning, the semantic word graphs can express this. Also note that e.g. “ma” can both function as a question mark and as a tone word.

Finally, we want to make structural parsing theory as clear and concise as possible. We therefore start with the simplest and basic situation. The main idea is to first give the main word types and build a grammar based on these word types, and later expand our system by refining the word types and the grammar rules in order to process more complex sentences. Because we would not like our system to be too complicated to work with at the beginning, we just chose the main word types as our target to begin structural parsing.

Therefore we classify Chinese words into 8 word types, given in Table 5.II with the terminology in English as well as the symbols that are used in the word graphs.

CHINESE	ENGLISH	SYMBOL
ming2 ci1	noun	N
dong4 ci1	verb	V
xing2 rong2 ci1	adjective	adj
dai4 ci1	pronoun	PN
shu4 ci1	numeral	num
liang4 ci1	classifier	cl
jie4 ci1	preposition	prep
fu4 ci1	adverb	adv

Table 5.II Restricted set of Chinese word types.

In English we also chose 8 word types, but the “classifier” type is replaced by the “determiner” type. We do not give a table.

5.2.3 Syntactic word graphs for word types

The surface structure of a sentence is to be expressed by its syntactic sentence graph, and the deep structure of a sentence is to be expressed by its semantic sentence graph. Our further analysis will be based on 8 syntactic word graphs for the 8 word types given above. In syntactic analysis, they will be combined to construct the surface structure of a sentence. The Figures 5.4 and 5.5 give the syntactic word graphs for the 8 word types given in Table 5.II. These graphs are constructed by expressing the various functions. If, for example, a classifier classifies a noun, see the graph for “cl”, then in the graph for the noun N the used PAR-arc should also be included. Noun and verb have the most complicated syntactic word graphs. Note that in the syntactic word graph for a preposition the type of arc is indicated by T, as there are several possibilities for the way that arcs link nouns, also see [Berg, 1993].

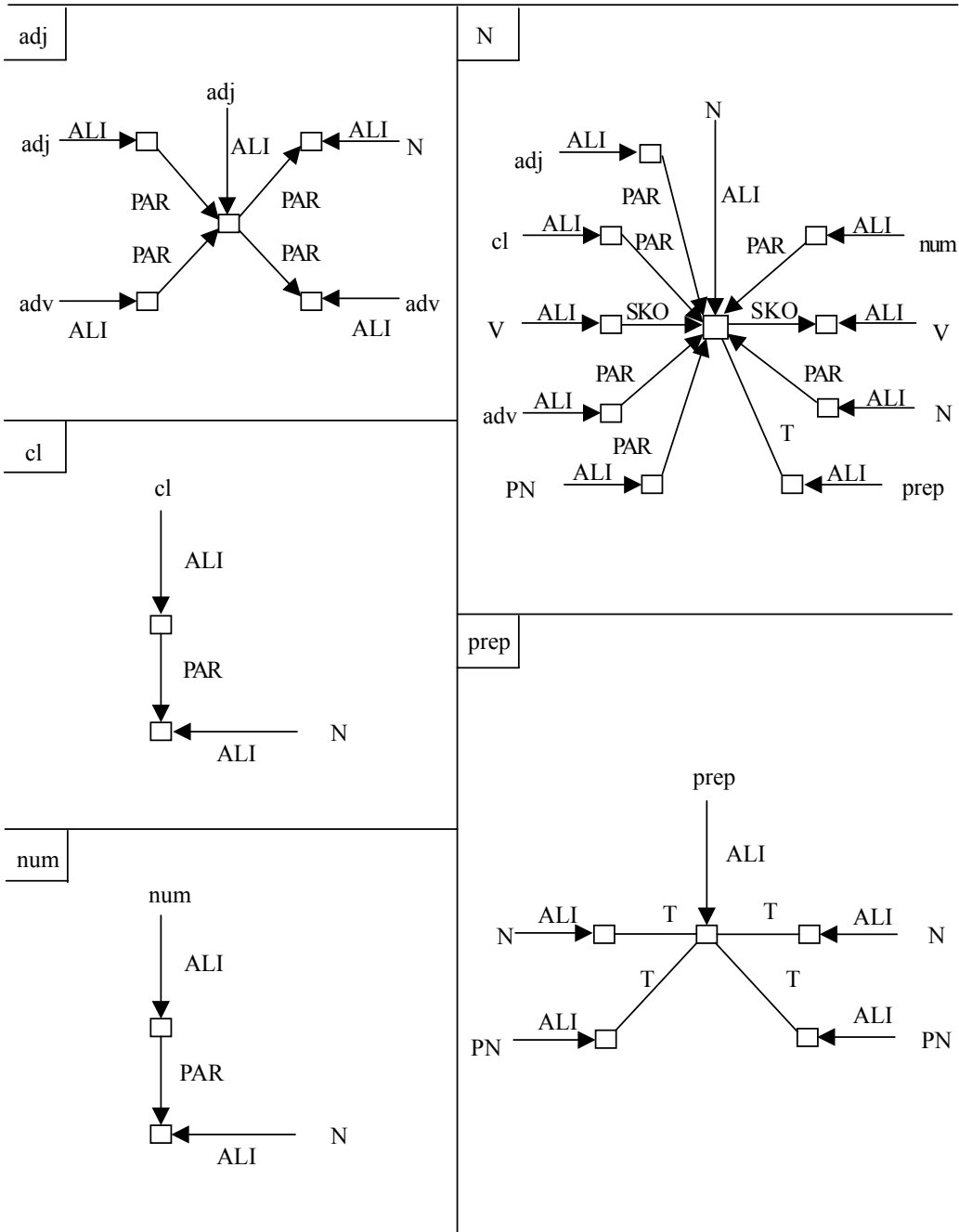


Figure 5.4 Syntactic word graphs for 5 word types of Chinese.

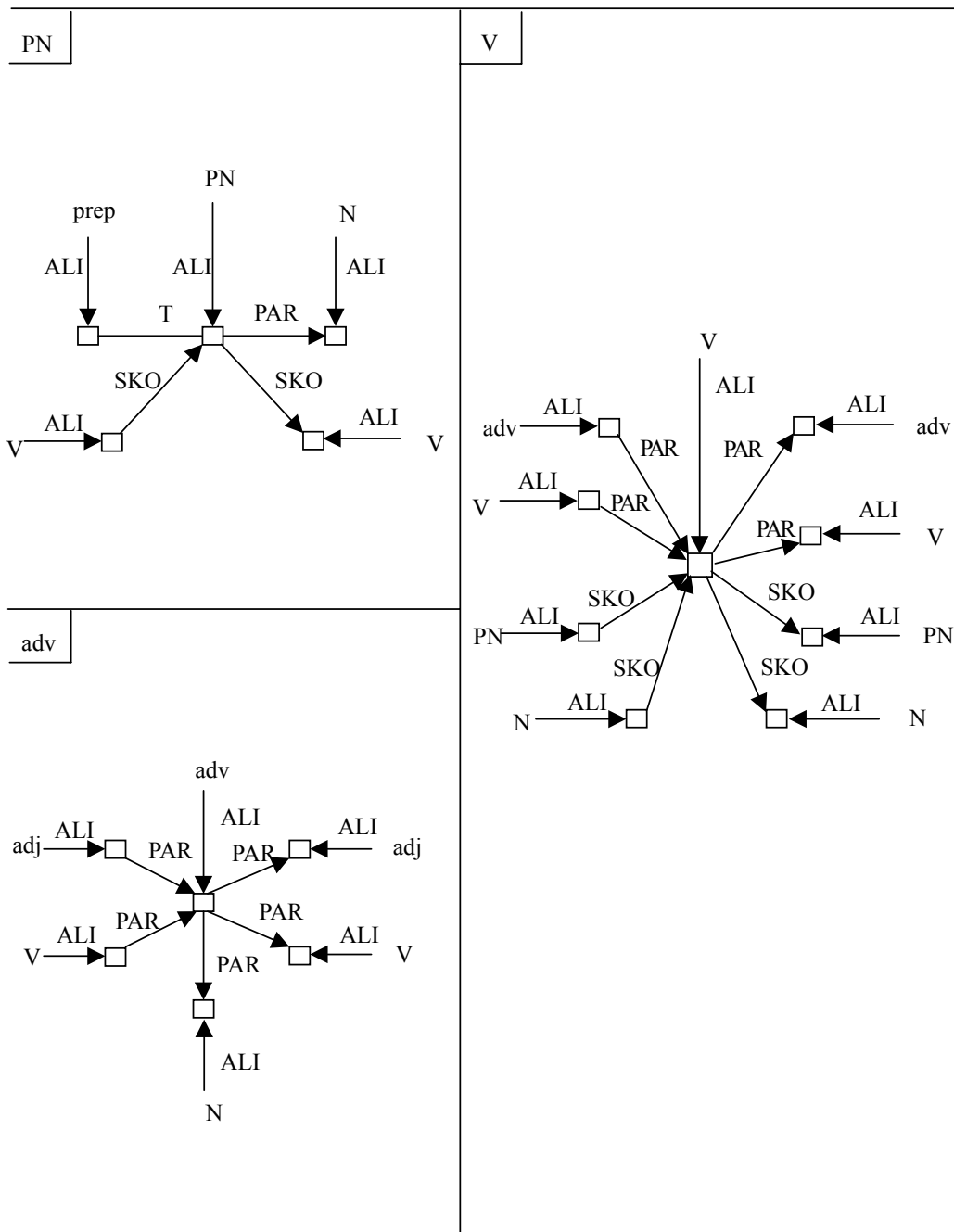


Figure 5.5 Syntactic word graphs for 3 word types of Chinese.

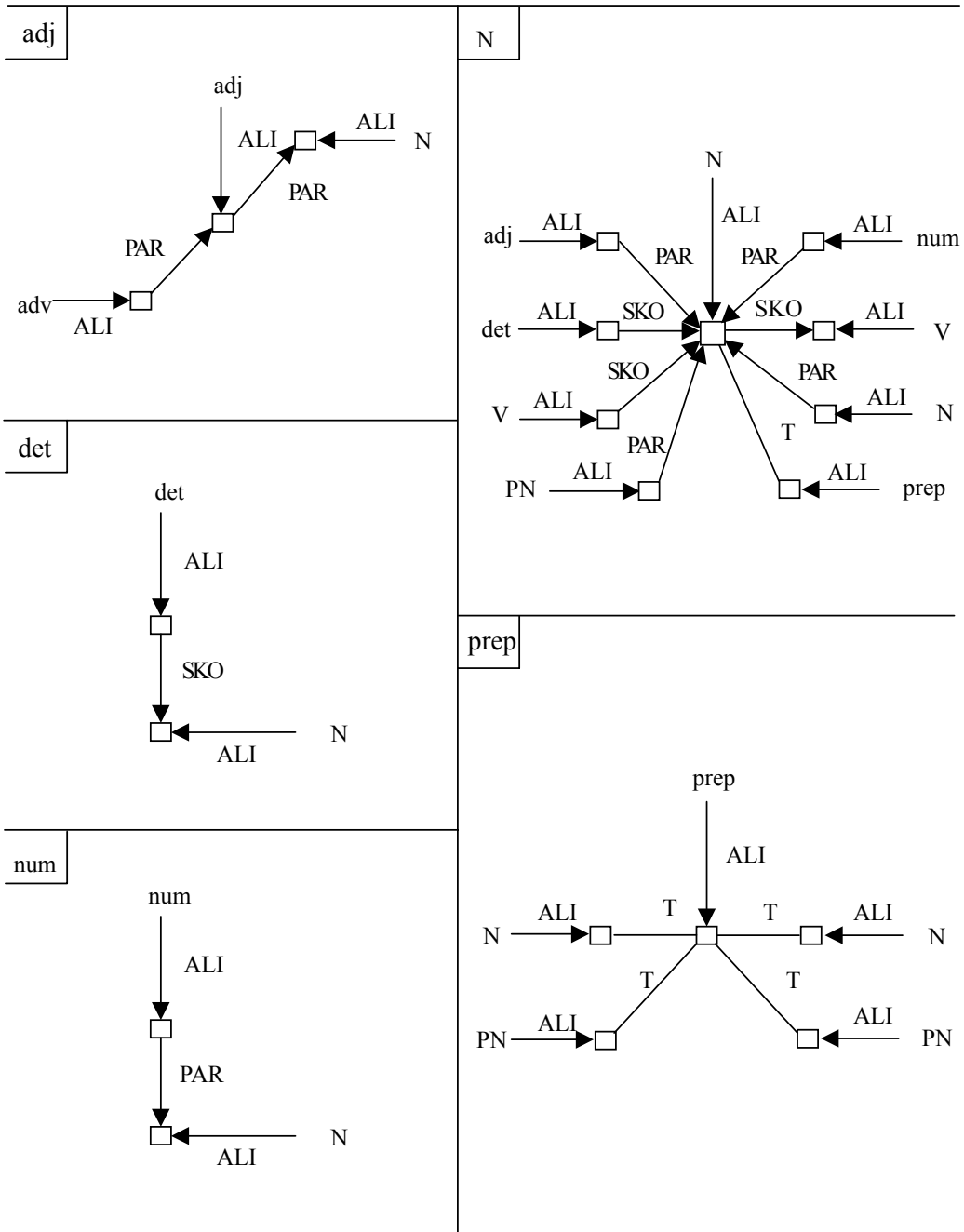


Figure 5.6 Syntactic word graphs for 5 word types of English.

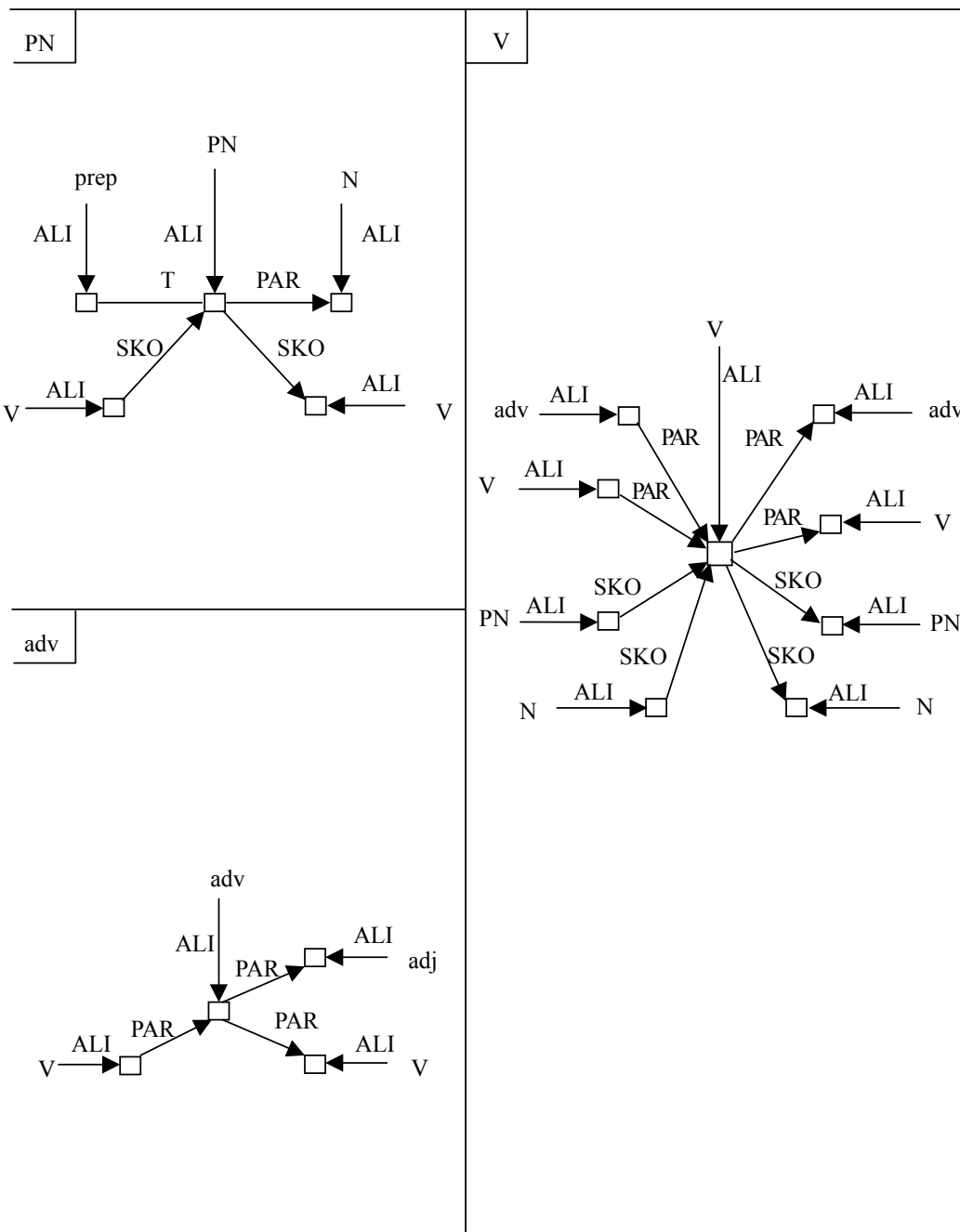


Figure 5.7 Syntactic word graphs for 3 word types of English.

5.3 Grammars for Chinese and English

We try to derive a grammar, for English and Chinese respectively, from the syntactic word graphs, given in Section 5.2.

There were 8 word types in Chinese and 8 word types in English, which were given according to the syntactic functions of words. In terms of the word types, rules in a grammar indicate in what order the words can be combined. Such a combination of words should be possible as far as the syntactic word graphs are concerned.

First consider Chinese. The above 8 word types have many ways to be combined in a sentence. It is obvious that a noun can be combined with an adjective, a classifier, a numeral or a preposition, such as “hong2 hua1 (red flower)”, “yi4 ke1 shu4 (a tree)”, in which “ke1” is a classifier, “san1 tian1 shi2 jian1 (three days of time)”, or “zai4 jiao4 shi4 li3 (in classroom)”. Also a noun can be combined with a verb as a subject or an object. The following sentences are examples:

(7)	Xue2	you2yong3	hen3	rong2yi4.
	V	N	adv	adj
	(Learn	swim	very	easy.)

(8)	Xia4wu3	lai2	zhun3	xing2.
	N	V	adv	adj
	(Afternoon	come	probably	well.)

Here the verb “xue2” combines with the noun “you2yong3” in the first sentence, and the noun “xia4 wu3” combines with the verb “lai2” in the second sentence.

So there may be grammar rules where on the right hand side we read V N or N V.

Consider a verb, it also has many ways to combine with other words, such as an adverb, an adjective, etc.

Take the following sentence as an example:

(9)	Ta1	gang1	zou3.
	PN	adv	V
	(He	just	go.)

The adv “gang1” combines with the verb “zou3”. This example shows the possibility of a rule with adv V on the right hand side.

Sometimes a noun also can be combined with an adverb. Together they can play the role of a predicate in a sentence. For example,

(10)	Ta1	cai2	shi2liu4sui4.
	PN	adv	N
	(He	only	16 years age.)

In the sentence the adverb “cai2” combines with the noun “shi2 liu4 sui4”. No verb is mentioned, a rather common feature in Chinese, see [Hoede & Liu, 1998].

We now describe the grammar rules that we can derive from the syntactic word graphs for Chinese. Note that they are considerably different from the syntactic word graphs for English as we have to take into account very specific forms of sentence building, like in the last example where no verb is mentioned. To enable the generation of such a sentence we need a production rule like $V \rightarrow N$. In “He baker”, considered non-well-formed in English, we see a pronoun followed by a noun. From $S \rightarrow NP VP$ we can obtain this sentence by using $NP \rightarrow PN$, $VP \rightarrow V$ and $V \rightarrow N$ as rules. We decided to include the rule $VP \rightarrow N$.

The general way to find the rules is to check whether two graphs can be coupled. For example, the PAR-arc from adj to N in the graph for adj is found in the graph for N as well. This tells us that the ordered pair adj N can occur in a sentence. There should therefore be a rule $X \rightarrow \text{adj N}$, or, as we prefer because of our aim to develop structural parsing, an inverse rule $\text{adj N} \rightarrow X$.

X is chosen to be N as nouns and verbs are the dominant word types in language. Not without reason the first rule of grammar is $S \rightarrow NP VP$, where NP can be seen basically as an N and VP as a V, to which various other parts of the sentence graph, that is to be expressed, are added.

Considering all pairs of word types we obtain a set of rules that are such that, when applied in a parsing process, guarantee that the corresponding word graphs, syntactic or semantic, can be coupled.

Special rules are needed to generate NP, VP and also PP, the prepositional phrase.

8. N → PN N
9. N → N N
10. NP → V N
11. NP → N V
12. NP → V PN
13. NP → PN V
14. AP → prep N
15. AP → prep PN
16. VP → V
17. VP → adj
18. VP → N
19. V → adv V
20. V → V adv
21. V → AP V
22. V → V AP
23. V → V V
24. VP → V N
25. VP → N V
26. VP → V PN
27. VP → PN V
28. adj → adj adj
29. adj → adv adj
30. adj → adj adv .

In English there is no word type of “classifier”, that is a particular word type in Chinese. Determiners, like “the” and “a”, are more often used to modify a noun in an English sentence. So, we replace the classifier type by the determiner type in the set of English word types. Based on the 8 word types that we found, in an analogous way, the following English grammar rules were found:

1. S → NP VP
2. NP → PN
3. NP → N
4. N → N N
5. N → adj N
6. N → det N
7. AP → prep N
8. N → num N
9. N → PN N
10. AP → V N
11. VP → V
12. VP → V N
13. VP → V PN
14. V → V V
15. V → adv V
16. V → V adv
17. V → V AP
18. adj → adv adj .

Note that we had to use more rules for Chinese. The main reason is that certain sentences, that would not be considered correct English sentences, had to be parsable in Chinese.

As a final example, an adjective or an adverb can never function as a predicate in English.

Consider the following sentences in Chinese:

(13) Tian¹ gao¹, lu⁴ yuan³,
 N adj N adj
 (Sky high road far)

(14) Ta¹ hen³ gao¹.
 PN adv adj
 (He very tall).

The underlined parts are respectively an adjective “gao1 (high)” or “yuan3 (far)”, or the combination of an adverb “hen3 (very)” and an adjective “gao1 (tall)”. In both cases they are used like a verb.

5.4 Structural Parsing

Parsing in natural language processing is defined as transforming a sentence or text into a representation of that sentence or text. However, this representation can also be a knowledge graph, which includes the concept of a word graph and that of a sentence graph. A word graph expresses the meaning of a word with a structure, and a sentence graph expresses the meaning of a sentence with a structure. A word graph is then a basic unit in natural language processing; a sentence graph is to be formed from the word graphs of the words that appear in this sentence. Knowledge graphs for words can be both syntactic and semantic, as we have seen in Section 5.2. Parsing by representing a sentence with a knowledge graph is a new field, which we can call structural parsing.

Definition 5.4 *Structural parsing* is the mapping of a sentence on a semantic sentence graph.

The goal of structural parsing, the semantic graph of a sentence, is in principle obtained as follows:

- A grammar is used to construct one or more parse trees for the sentences.
- A syntactic sentence graph is derived from syntactic word graphs using a parse tree.
- A semantic sentence graph is derived from the found syntactic sentence graph.

Note that usually many syntactic sentence graphs can be derived by the grammar, but that often only one syntactic graph is suitable semantically, unless there is essential ambiguity.

5.4.1 A traditional parsing approach

The following procedure could be used:

- In a lexicon for each word a semantic and a syntactic word graph is given for each use of the word.
- The set of grammar rules, discussed in Section 5.3, is used in traditional parsing, which leads to one or more parse trees.
- Syntactic word graphs are combined to a syntactic sentence graph according to bottom-up parsing.
- Each syntactic sentence graph is transformed into a semantic sentence graph by combining corresponding semantic word graphs.

Both traditional bottom-up parsing or top-down parsing can be used to analyze a sentence. One of the most difficult problems for traditional parsing techniques is to get rid of ambiguities. We can produce many syntactic sentence graphs that make no sense, if we use the grammars listed in Section 5.3. We do not like to produce many syntactic sentence graphs for complexity reasons. For this reason, we would like to give our own parsing method, that is adapted to our knowledge graph theory. The key to our parsing method lies in a discussion of utterance paths.

5.4.2 Utterance paths and chunks

A sentence expresses a sentence graph. The graph is “brought under words”. The speaker chooses an order in which these words are uttered. Corresponding with this order is an ordering of subgraphs of the sentence graph, the word graphs. With such an ordering of subgraphs usually one or more paths can be indicated, depending on whether consecutive words have overlapping word graphs or not.

We will make the concept of utterance path clear by an example sentence:

- The volcano, that lies in Alaska, 130 kilometers from Anchorage, erupted in 1992.
- The volcano, that erupted in 1992, lies in Alaska, 130 kilometers from Anchorage.
- 130 kilometers from Anchorage, Alaska, lies the volcano, that erupted in 1992.
- In Alaska, 130 kilometers from Anchorage, lies the volcano, that erupted in 1992.

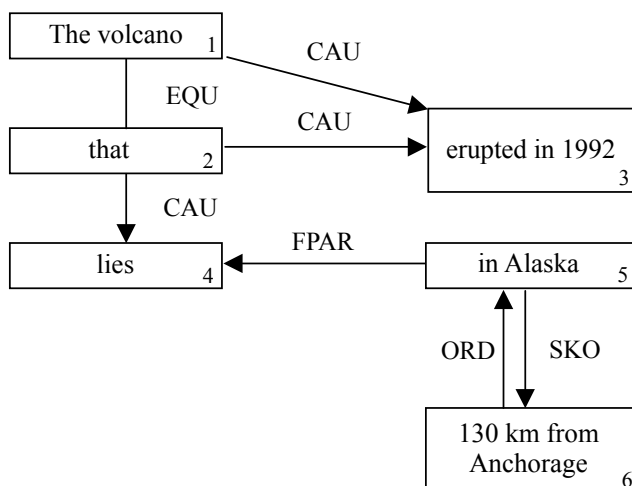


Figure 5.8 Semantic sentence graph with display of “chunks”.

In all four sentences for one sentence graph we recognize typical paths. “130 kilometers from Anchorage” is one such path. “Erupted in 1992” is another path occurring in all four sentences, and “the volcano, that” also. The ordering “in Alaska” does not occur in the third sentence, but could have been used therein. In the simplified sentence graph in Figure 5.8 these paths can be read off as texts in the frames.

The remarkable feature is that the six indicated frames, that occur as connected graphs in the non-simplified sentence graph, are expressed as what might be called chunks of the sentence, see also the paper of Abney [Abney, 1991] on parsing by chunks. Abney states that people tend to express a sentence in chunks of words, and we see that chunks of the sentence graph are brought under words in some specific order. The four sentences can be described as an ordering of the expressed chunks, 1 to 6:

- 1→2→4→5→6→3
- 1→2→3→4→5→6
- 6→5→4→1→2→3
- 5→6→4→1→2→3

Note that “jumps” occur, consecutive chunks, not linked in the sentence graph. In the

first sentence there is a jump $6 \rightarrow 3$, in the second sentence a jump $3 \rightarrow 4$, in the third a jump $4 \rightarrow 1$ and in the fourth there are two jumps: $6 \rightarrow 4$ and $4 \rightarrow 1$.

As our goal is to construct the sentence graph from a sentence, the fact that chunks of the graph are expressed as chunks of the sentence leads us to want to read off chunks from the sentence, for which chunks of the sentence graph seem to be easily constructable. A problem for finding chunks of a sentence is that of finding begin point and end point of a chunk. With the interpretation of a sentence, as expressing a sentence graph, as a guide line we will try to find chunk indicators.

5.4.3 Chunk indicators

Our reasoning behind the choice of indicators is the following. In terms of knowledge graph theory, frame words, such as: be, can, may, must, see Chapter 4, which are auxiliary verbs, and modify the whole sentence, should be a chunk indicator, where the chunk is the whole sentence. For example in the following sentence:

“Can I have a listing of all flights from Amsterdam to Beijing?”

the auxiliary verb “can” modifies the whole sentence. In the sentence graph this is expressed with a POS-frame. We have discussed frame words like BE-frame, NEC-frame, NOT-frame, OR-frame, IF-THEN-frame or POS-frame in Chapter 4. The auxiliary verb “have” is essentially “be with”. The BE-frame can be seen as a chunk indicator too, so that what remains for structural parsing is “I with a listing of all flights from Amsterdam to Beijing”.

Now consider reference words, such as: it, that, the, she, he, her, his, this, ..., etc. They are used to avoid repetition of mentioning something, and hint at a chunk. Consider the sentence:

“Every woman thinks she raises children better than her mother.” and

“The triangle has a right angle, its sides are 3, 4 and 5 cm, its circumference is 12cm”.

The words “she” and “her” are chunk indicators in the first sentence, the word “its” is a chunk indicator in the second sentence. In Section 5.4.2, where “the volcano” occurred as a chunk, we had the possibility to cut the sentence into two sentences by

replacing “that” by “the volcano”. Likewise we might replace “it” by “the triangle” and obtain three sentences. These sentences, like all sentences, are clearly chunks.

If two consecutive words can not be combined, they hint at a “jump”. Therefore they should belong to different chunks, such as in the following sentence, where the word “up” cannot be combined with “earlier”.

“She gets up earlier than John.”

Prepositions are very useful in natural language and always link other words. If a preposition is met in a sentence, it hints at a chunk, e.g., in “from Amsterdam to Beijing” or in “in Alaska”, see Section 5.4.2.

Of course comma pairs, in written language, are clearly chunk indicators too, as are pairs of period signs, indicating a whole sentence, or a pair of comma and period sign.

Summing up we list the chunk indicators as follows:

- Indicator 0: Pairs of comma and/or period signs;
- Indicator 1: Frame words, including auxiliary verbs;
- Indicator 2: Reference words;
- Indicator 3: “Jumps”, with respect to grammar;
- Indicator 4: Link words, including prepositions.

In structural parsing, we do not think complete parse trees are necessary. If chunks are recognized, we can give the graphs of these chunks by combining word graphs. After that, we link these chunk graphs into a sentence graph.

Of course now there are three problems:

- To what chunks of the sentence do the indicators lead;
- How to make chunk graphs for the found sentence chunks;
- How to link chunk graphs into a sentence graph.

We will not develop a general theory for answering these questions in this thesis, but will consider a few examples as an experiment, in which also other points discussed in this thesis should become clear.

5.4.4 Examples of structural parsing

In this section there are two example sentences: one is in English, the other is in Chinese. We will give a detailed analysis, indicating the different phases.

Example 1

“The volcano, that lies in Alaska, 130 kilometers from Anchorage, erupted in 1992.”

Phase 1

The preparatory phase contains two parts.

First, we chunk the sentence by checking indicators, which were discussed in Section 5.4.3, one by one.

According to indicator 0, commas and period signs, we get four chunks directly. Next we cut chunks into sub-chunks according to the other indicators.

We do not use indicator 1, as there is no auxiliary verb in this sentence.

The indicator 2 is about reference words. There are two reference words, “the” and “that”. A determiner combines with the noun following. “The volcano” is therefore a “complete” chunk, there are no sub-chunks. Other reference words, like pronouns, are separate chunks: “that” is a sub-chunk.

As for the indicator 3, there are three jumps; between “lies” and “in”, “kilometers” and “from”, as well as “erupted” and “in”. These jumps cut sub-chunks into smaller sub-chunks.

There are three prepositions, “in”, “from” and “in”. Prepositions combine with the noun following. This takes into account indicator 4.

As there are no further chunk indicators, there is no further chunking.

We get in this way the resulting chunks and sub-chunks:

- | | | |
|----|------------------------------------|--------------------|
| 1. | [.[The volcano], | CHUNK1 |
| 2. | [[that][lies][in Alaska]], | CHUNKS 2, 3, and 4 |
| 3. | [[130 kilometer][from Anchorage]], | CHUNKS 5 and 6 |

4. [[erupted]][in 1992]].] CHUNKS 7 and 8.

Second, for all the words in this sentence, semantic as well as syntactic word graphs should be listed in a lexicon. Since the syntactic word graphs have been listed in Section 5.2, here we indicate them with word type abbreviations, see Figure 5.9.

We construct syntactic chunk graphs chunk by chunk, like in Figure 5.10. The syntactic word graphs used are only represented with relevant arcs. The other arcs are indicated by dotted lines.

So now the sentence is expressed like “. CH1, CH2 CH3 CH4, CH5, CH6, CH7 CH8.” We will combine syntactic chunk graphs into a bigger one when they can be linked syntactically. We use a new number to indicate a chunk, which may be a combination of sub-chunks.

CH9 = CH1

CH10= CH2 CH3 CH4

CH11= CH5 CH6

CH12= CH7 CH8.

Checking these chunks, from CH9 to CH12, we found that only CH2 and CH3 can be combined into one chunk, others allow no linking syntactically in this phase. The following figure shows the combination of CH2 and CH3.

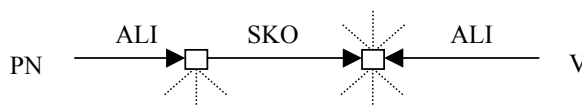


Figure 5.11 The syntactic graph of CH2+CH3.

Since there are no further linkings syntactically, we now will construct semantic chunk graphs by using the simple semantic word graphs given in Figure 5.9. Note that we might have given expanded versions of these semantic word graphs.

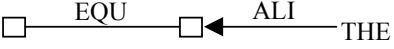
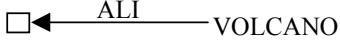
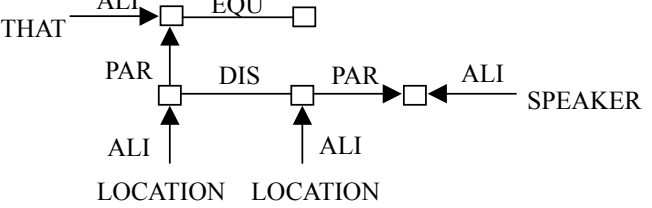
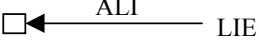
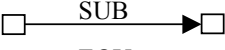
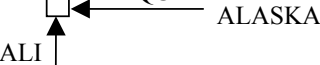
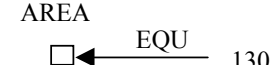
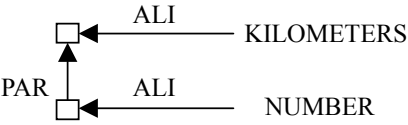
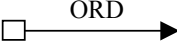
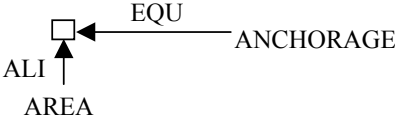
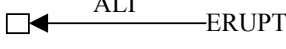
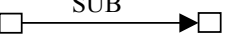
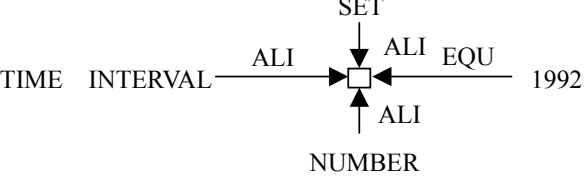
WORDS	SEMANTIC WORD	WORD TYPES
THE		det
VOLCANO		N
THAT		PN
LIE		V
IN		prep
ALASKA		N
130		num
KILOMETERS		N
FROM		prep
ANCHORAGE		N
ERUPT		V
IN		prep
1992		N

Figure 5.9 Lexicon of Example 1.

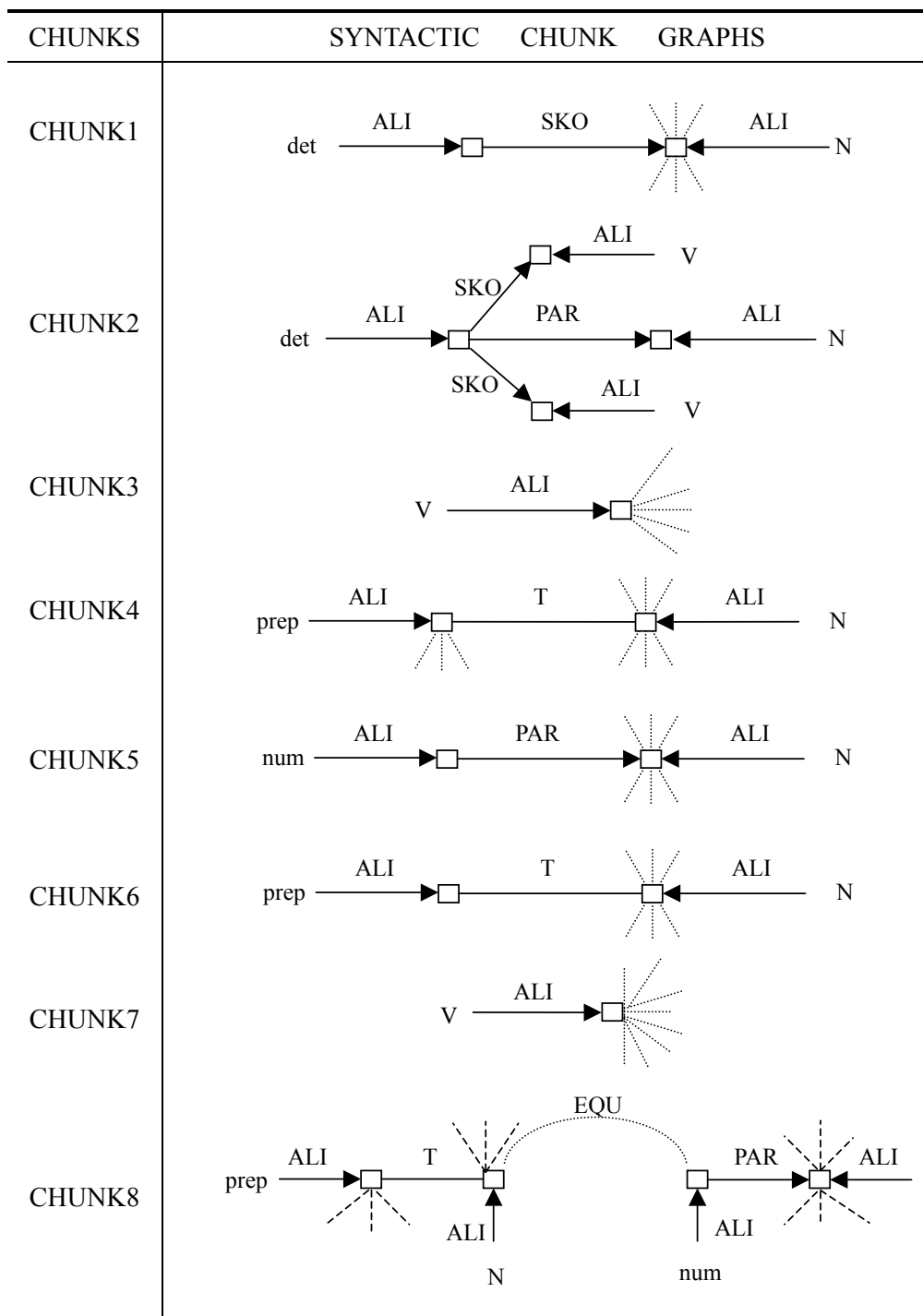


Figure 5.10 Syntactic chunk graphs of Example 1.

Phase 2

In order to make things clear, we renumber all the chunks as follows:

CH9 = CH1

CH10a = CH2 CH3

CH10b = CH4

CH11a = CH5

Ch11b = CH6

CH12a = CH7

CH12b = CH8.

Now we give the formation of semantic chunk graphs in Figure 5.12 and describe this in more detail. We do not consider word changes like “lies” instead of “lie”.

- For CH9, which is “the volcano”, the semantic chunk graph can be given directly.
- With simple semantic word graphs, the verb “lie” cannot be combined with “in Alaska” directly, so we have semantic chunk graphs of CH10a and CH10b separately.
- The semantic chunk graph of “130 kilometers” can be obtained from the simple semantic word graphs of “130” and “kilometers”.
- The semantic chunk graph of “from Anchorage” can be obtained by identification of tokens as indicated by the EQU-link from the token in the preposition “from” to that in the noun “Anchorage”.
- The semantic chunk graph of “erupt” is very simple.
- In the semantic word graph of “in”, the tokens are not specified, so the right hand token can be identified with another token, like that for the number 1992. Note that we expressed “1992” with an ALI-arc to indicate the set nature of the time interval.

Due to the fact that the used semantic word graphs are so simple, we also have some subchunks which cannot be combined in this phase. To achieve this we need some background knowledge. This is introduced by expanding the simple semantic word graphs into more complex ones.

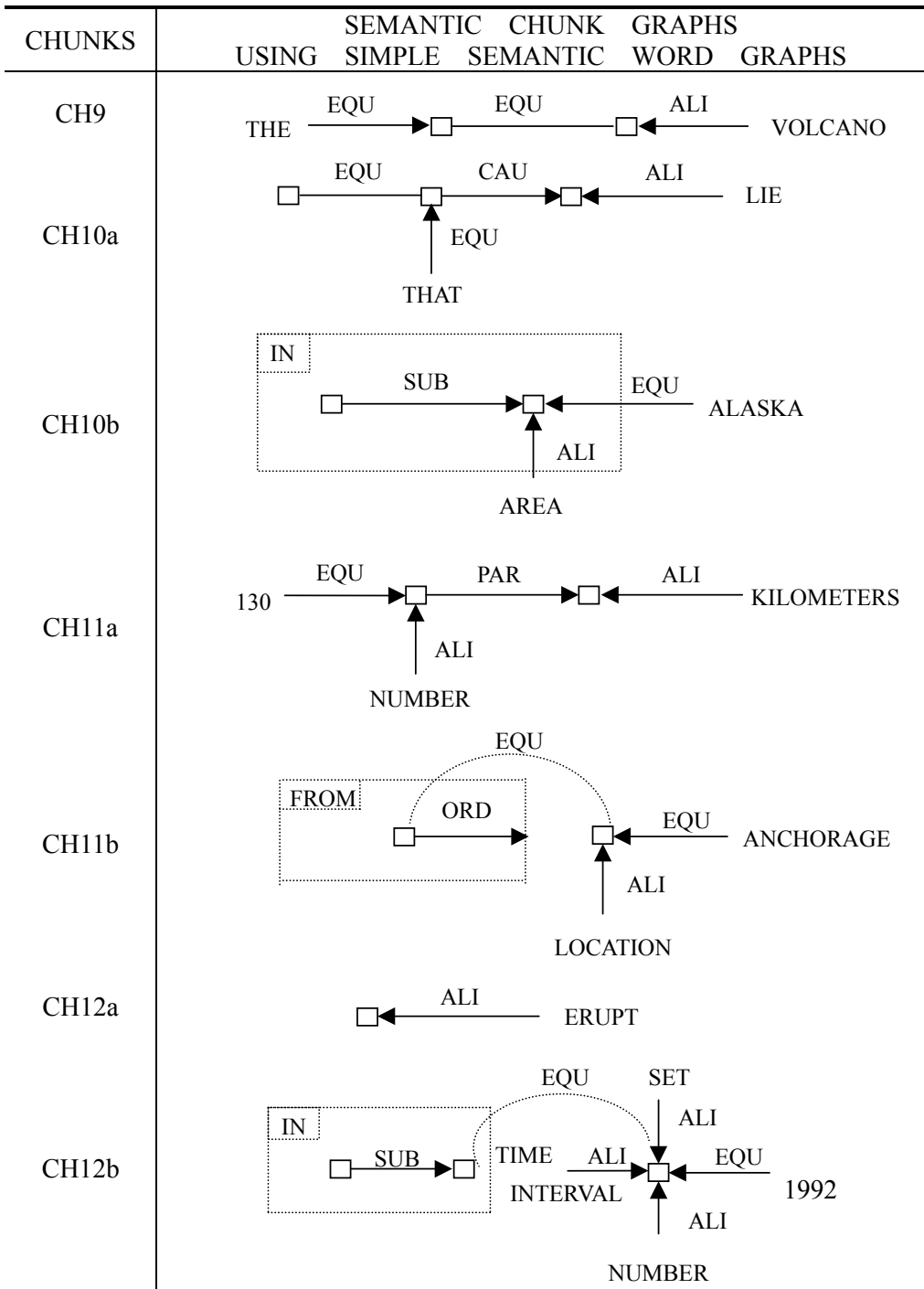


Figure 5.12 Semantic chunk graphs using simple semantic word graphs from Figure 5.9.

Phase 3

First we expand “lie” and “in”. For “lie” we add two FPAR-arcs from tokens of type “area”. For “in” the tokens are given type “area”. We do this, because semantically “lie” and “in” are both related to areas. Then CH10a and CH10b can be combined into CH10 as in Figure 5.13.

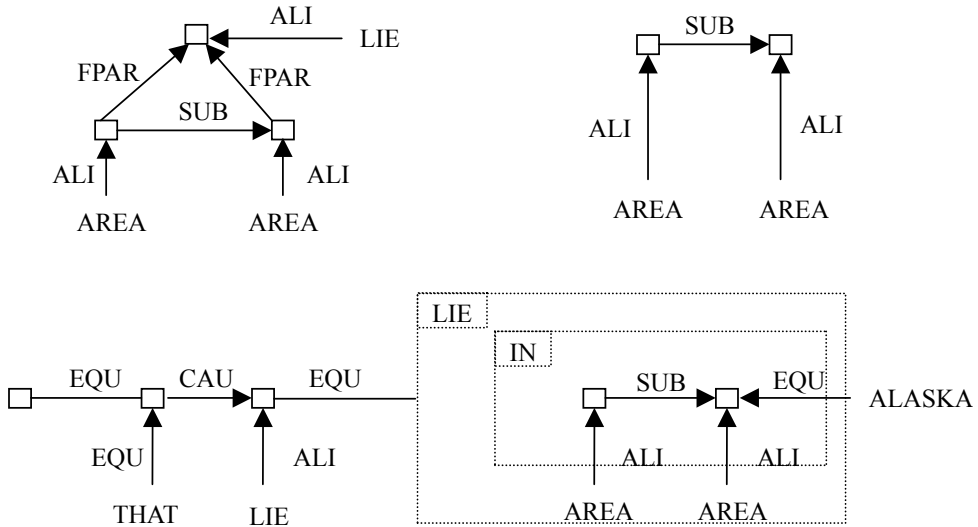


Figure 5.13 The semantic chunk graph of CH10.

We expand “from” with an ALI-arc linking to “location”, and expand “130 kilometers” with a PAR-arc linking to “distance” that has two SKO-arcs both to “location”. So, CH11a and CH11b can be combined into CH11 as in Figure 5.14;

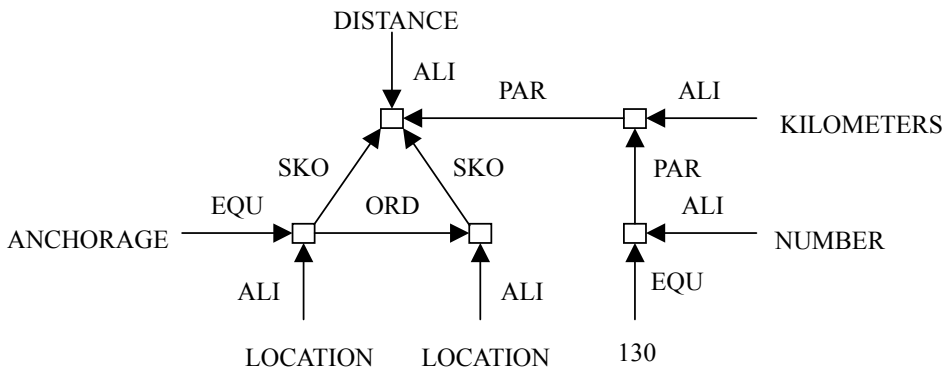


Figure 5.14 The semantic chunk graph of CH11.

We expand “erupt” with a PAR-arc and a CAU-arc. There are two points to mention here: “erupt” is a verb, it should have a CAU-arc coming in; semantically “erupt” is related to a “location”. Linking “in” to “erupt” with a PAR-arc is the only possibility to combine them. CH12a and CH12b should be combined into CH12 like in Figure 5.15.

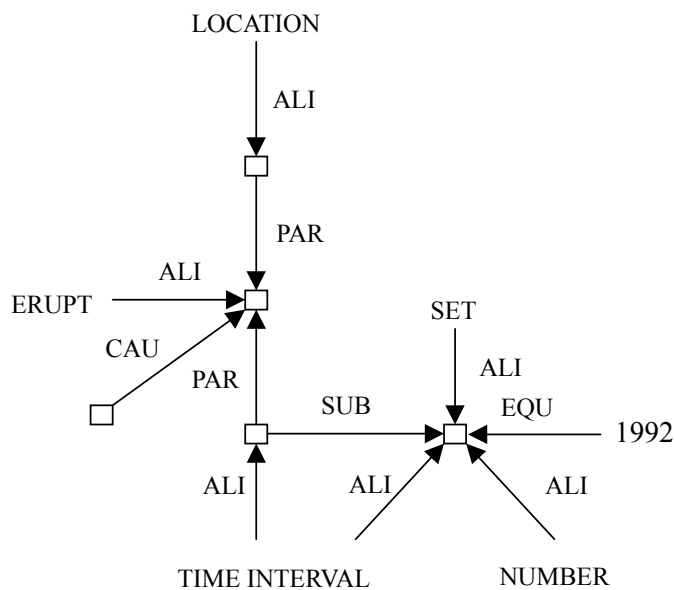


Figure 5.15 The semantic chunk graph of CH12.

Now we get the result:

$S = CH9, CH10, CH11, CH12.$

Here the sentence has 4 chunks, which have corresponding semantic chunk graphs.

Finally, we link the four chunk graphs from the left hand side to the right hand side, unless there is a jump, and we obtain the semantic sentence graph, see Figure 5.16, where some arcs used in the analysis, like the ALI-arcs to “set” or “number”, have been omitted for reason of clarity.

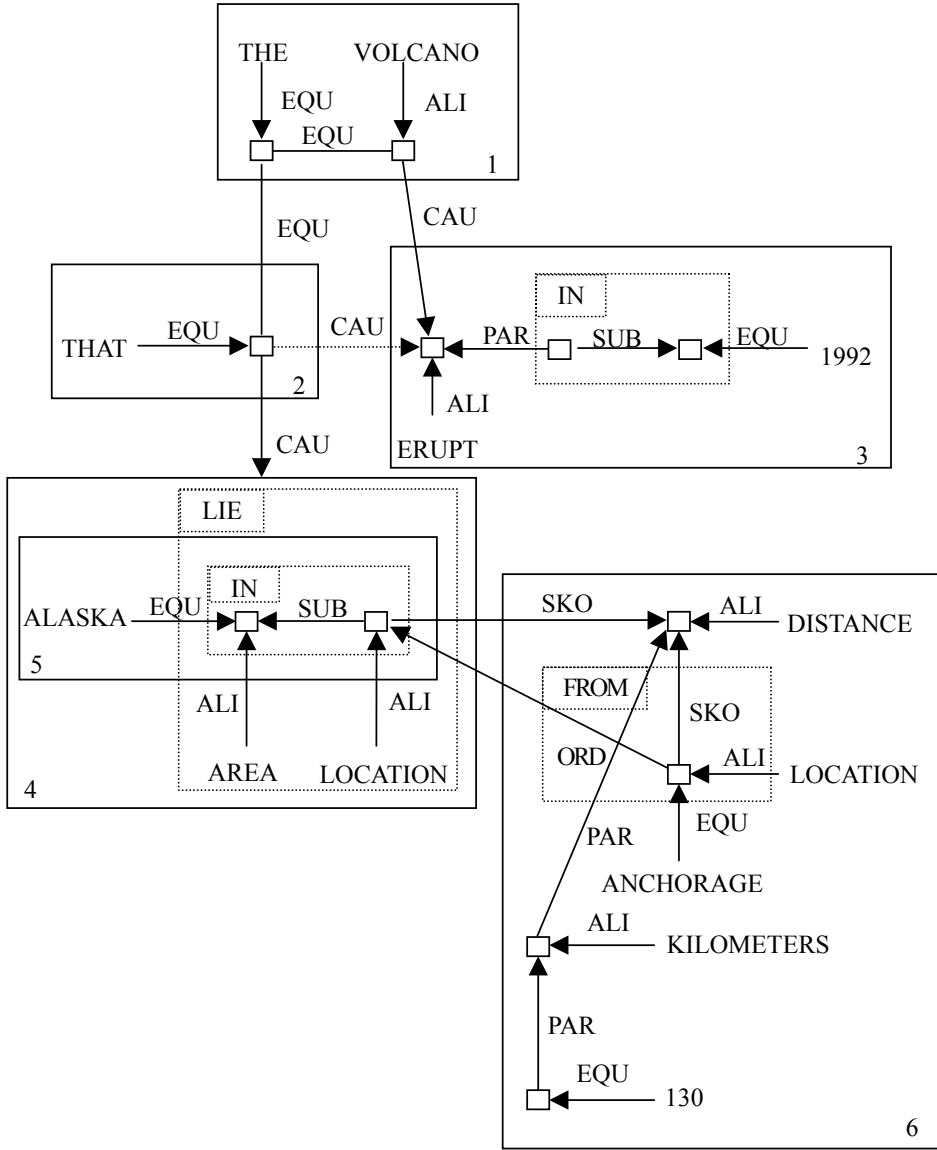


Figure 5.16 The semantic sentence graph for the English example sentence.

Now let us take a Chinese sentence as an experiment to test the 5 phases of structural parsing.

Example 2

Wo3 de xiao3 di4di zai4 XIAN de Xi1Bei3Da4Xue2 shang4xue2.

(I of small brother in XIAN of west north university study.)

“My small brother studies in Northwest University of XIAN.”

- In Phase 0, a lexicon of this sentence is given in Figure 5.17.
- In Phase 1, the sentence is chunked as follows, according to chunk indicators.
 1. [wo3]
 2. [de]
 3. [xiao3 di4di]
 4. [zai4 XIAN de]
 5. [Xi1Bei3Da4Xue2 shang4xue2].
- In Phase 2, using syntactic word graphs, we obtain syntactic chunk graphs like in Figure 5.18.
- In Phase 3, syntactic chunk graphs are linked into bigger ones, see the upper part of Figure 5.19.
- In Phase 4, CH6 and CH7 are expressed by semantic word graphs, like in the lower part of Figure 5.19.

Note that expansion is needed for linking “zai4” with XIAN, and “ren2” has been added for CH6. There are two problems: “zai4” has an unspecified token; in the chunk graphs “ren2” is not linked. Both problems are solved by expanding the semantic word graphs. For “di4di” and for “shang4xue2”, expansions are chosen as in Figure 5.20. Then the “wrong” CAU-arc from XBDX to “shang4xue2” is cut, because “ren2” does not occur in the expansion of XBDX. The CAU-arc “looks for” “ren2” and therefore links with “di4di”. The “wei4zhi4” of “shang4xue2” “fills” the unspecified token in “zai4”.

- In Phase 5, the semantic sentence graph is completed, see Figure 5.21.

WORDS	SEMANTIC WORD	WORD TYPES
wo3	<pre> graph LR wo3[wo3] -- EQU --> Node1[] ren2[ren2] -- ALI --> Node1 Node1 -- EQU --> Node2[] shuo1hua4zhe3[shuo1hua4zhe3] -- ALI --> Node2 </pre>	PN
de	<ol style="list-style-type: none"> <pre> graph LR Node1[] -- PAR --> Node2[] N[N] -- ALI --> Node2 </pre> <pre> graph LR Node1[] -- PAR --> Node2[] PN[PN] -- ALI --> Node2 </pre> 	prep
xiao3	<pre> graph LR xiao3[xiao3] -- EQU --> Node1[] nian2ling4[nian2ling4] -- ALI --> Node1 Node1 -- PAR --> Node2[] </pre>	adj
di4di	<pre> graph LR Node1[] -- ALI --> di4di[di4di] </pre>	N
zai4	<pre> graph LR Node1[] -- SUB --> Node2[] wei4zhi4_1[wei4zhi4] -- ALI --> Node1 wei4zhi4_2[wei4zhi4] -- ALI --> Node2 </pre>	prep
XIAN	<pre> graph LR XIAN[XIAN] -- EQU --> Node1[] cheng2shi4[cheng2shi4] -- ALI --> Node1 </pre>	N
XBDX	<pre> graph LR Node1[] -- EQU --> XBDX[XBDX] da4xue2[da4xue2] -- ALI --> Node1 </pre>	N
shang4xue2	<pre> graph LR Node1[] -- CAU --> Node2[] shang4xue2_2[shang4xue2] -- ALI --> Node2 </pre>	V

Figure 5.17 Lexicon of Example 2.

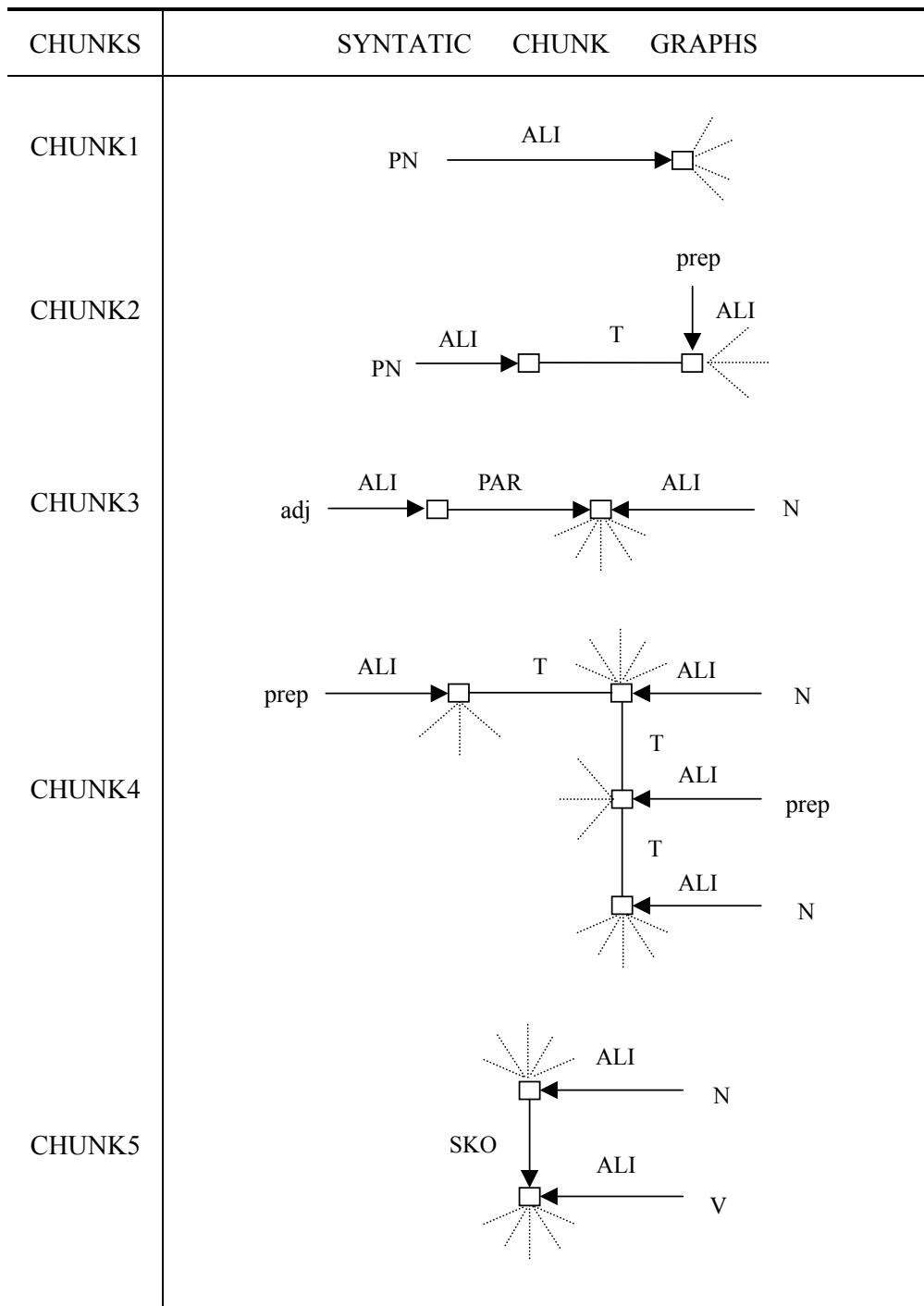
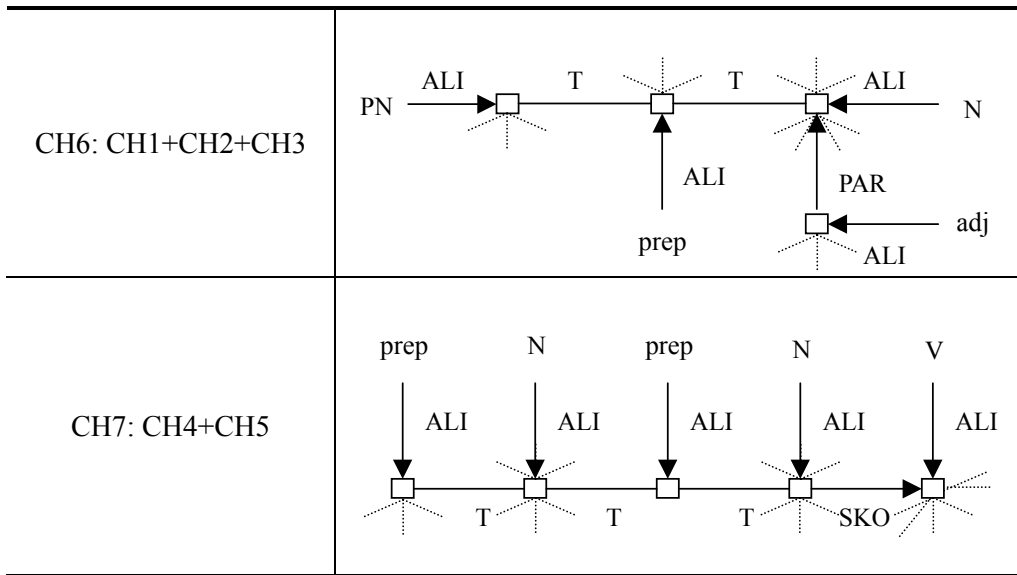


Figure 5.18 Syntactic chunk graphs of Example 2.



Bigger syntactic chunk graphs

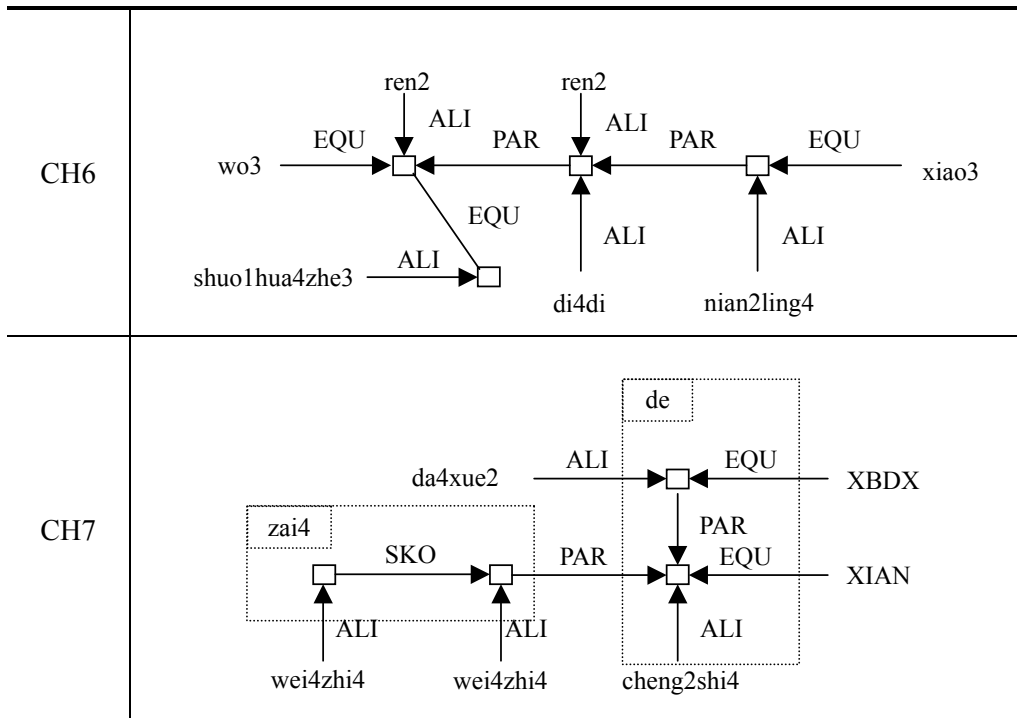


Figure 5.19 Semantic chunk graphs of CH6 and CH7.

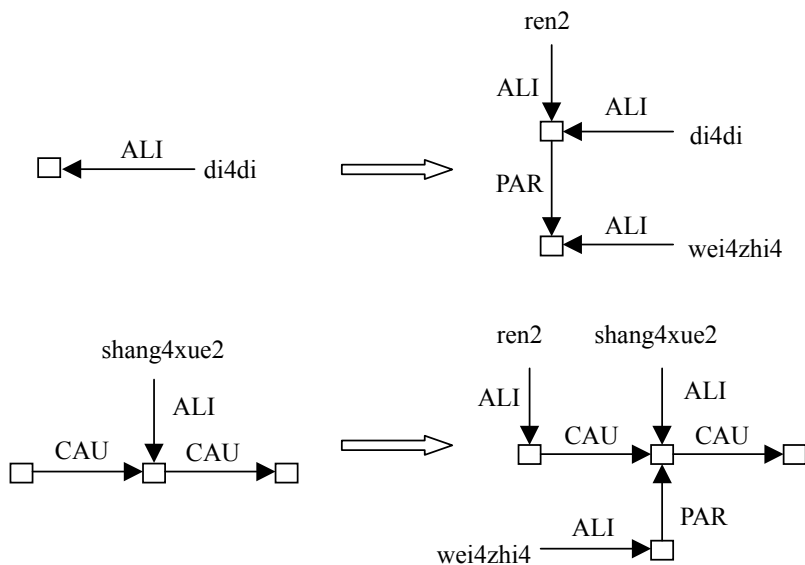


Figure 5.20 Expansions for “di4di” and “shang4xue2”.

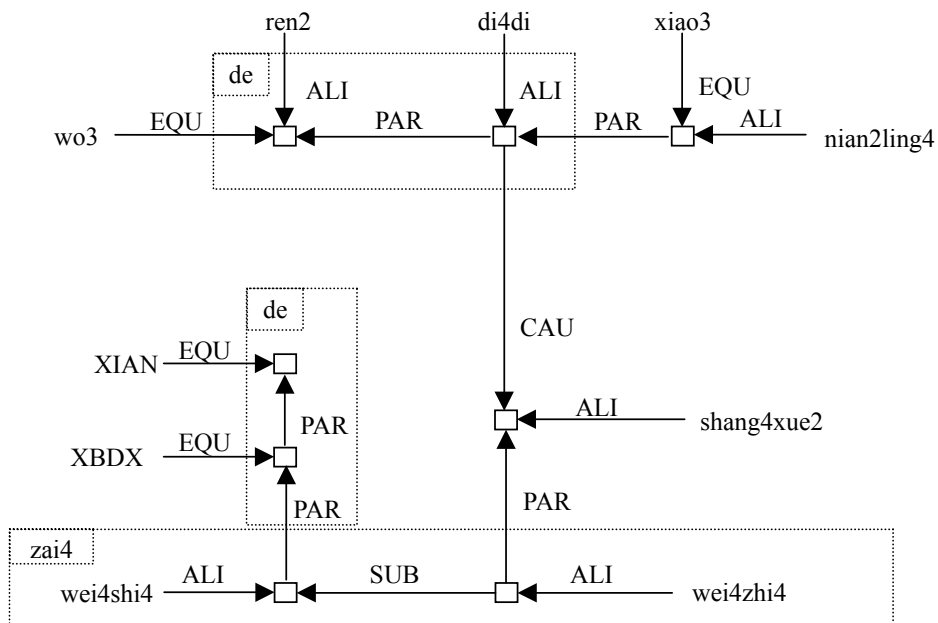


Figure 5.21 The semantic sentence graph for the Chinese example sentence.

5.5 Conclusion

The standard way of parsing is to find a generation of a sentence by a grammar. The most elementary aspects of mapping a sentence on a so-called parse tree were mentioned in Chapter 2.

Traditional parsing focuses on the syntactic aspects of language and then faces the problem of dealing with semantics. Truth conditional semantics involves a comparison with a model, i.e. the sentence statement is interpreted within a model.

In knowledge graph theory the approach is from the side of semantics. The meaning of a word or a structure is considered to be a graph, i.e. the structure is the meaning. Structural parsing then is the mapping of a sentence on a graph. This starts with word graphs for the words, which are to be combined into a sentence graph. The new concept introduced here is that of a syntactic word graphs for a certain type of word. The word graphs originally considered in the theory are called semantic word graphs. From the syntactic word graphs grammars can be derived, which is done for both Chinese and English.

As syntactic word graphs express the way words function with respect to each other, which has corresponding linking of semantic word graphs, a traditional parse tree obtained with the derived grammars will allow corresponding linking of semantic word graphs to obtain a semantic sentence graph.

It turns out, however, that the preparatory step of obtaining a syntactic sentence graph need not be carried out according to the traditional approach. Sentences are uttered in “chunks”, for which a traditionally flavored theory was designed by Abney [Abney, 1991]. By investigating so-called “utterance paths”, we find that parts of the semantic sentence graph are brought under words in such a way that “chunks” of the graph are expressed. This led to the idea that chunks of a sentence have corresponding chunks of the graph. With a chosen set of indicators a Chinese sentence and an English sentence were investigated. The idea turned out to be quite fruitful. Semantic sentence graphs were obtained, be it that background knowledge concerning the words had to be included by expanding the semantic word graphs. That this is possible, however, is considered to be a point of strength of knowledge graph theory.

As we will see in Chapter 7, the approach makes very clear where the aspects of artificial intelligence are located.

Chapter 6

Utterance Paths

6.1 Introduction

We have mentioned utterance paths in Chapter 5 in the sentence “The volcano, that lies in Alaska, 130 kilometers from Anchorage, erupted in 1992”. In this chapter the concept of utterance path will be investigated in more detail. Here we study the problem of determining rules for uttering the sentence graph. Given a sentence graph there are usually several ways how such a graph can be brought under words, i.e. can be *uttered*. The sentences arising from these ways of uttering consist of words occurring in the sentence graph in a specific order. Languages differ in the way the words occurring in the sentence are ordered. We investigate several sentences both in English and in Chinese.

Let us start with a very simple sentence like “man hit(s) dog”, the sentence graph is given in Figure 6.1.

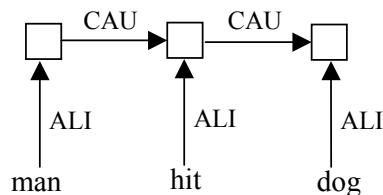


Figure 6.1 Sentence graph for “man hit(s) dog”.

Suppose that this sentence graph is given. Then we may ask what the sentence looks like that utters the situation expressed by the graph. There are $3! = 6$ ways to utter the graph. We might say:

- man hit dog
- man dog hit
- hit man dog
- hit dog man
- dog hit man
- dog man hit.

In English, and in Chinese, the first *utterance path* is used. However, in other language other orderings, other utterance paths occur. In Japanese the verb is usually put at the end as in the second and sixth way to utter the graph. The six orderings are usually described by the syntactic function of the words, “man” is the subject (S), “hit” is the verb (V) and “dog” is the object (O) grammatically. English and Chinese are therefore often called SVO-languages, as that ordering has developed for these two languages. We want to stress that, therefore, our considerations about utterance paths are language dependent.

The graph in Figure 6.1 must, in English, be uttered as “man hit(s) dog”. This sentence starts with a noun. Any grammar, in English, with production rules, starts by the rule $S \rightarrow NP VP$, where S stands for “sentence”, NP for “noun phrase” and VP for “verb phrase”. In our simple example the NP is “man” and the VP is “hit(s) dog”. Uttering the graph in Figure 6.1 should therefore start with “the noun in the noun phrase”, which is, of course, “man”. But how can the noun phrase be recognized in the graph?

There are various ways to find out whether a word is a noun. First, a lexicon of words may explicitly say that “man” is a noun, as is “dog”. Second, we could use the method described by Radford [Radford, 1988], who discussed test sentences like:

—— can be a pain in the neck.

If a word can be placed in the slot indicated by —— to give a sentence that makes

sense, the word is a noun. Indeed both “man” and “dog” pass this test, but hit (s) does not. The problem with this method is that the outcome comes from the human being, who has to decide whether the sentence makes sense. We therefore should point out that there is a third way to find out the word type involved here, *from the structure of the graph*.

A token with an incoming and an outgoing CAU-arc can only be a transitive verb, as only verbs are represented with the help of CAU-arcs. This makes hit(s) a verb. An intransitive verb would only have an incoming CAU-arc. The tokens from which and to which the CAU-arcs are coming respectively going, must be labeled by words that are nouns. This too is due to the way word graphs are used, see the syntactic and semantic word graphs in Chapter 5.

For our utterance problem we now know how to proceed. Find the verb, looking at the CAU-arcs, and find the noun from which there is a CAU-arc towards that verb. We find “man”. Then, because of the rule $S \rightarrow NP VP$, start by uttering “man”. As English is a SVO-language we know that now first “hit(s)” and finally “dog” has to be uttered.

In the graph we see that we follow the path from the token “man” to the token “dog” via the token “hit(s)”. The utterance path has been found for our simple example sentence graph. Note that the ordering of the CAU-arcs, with our rule for uttering, does not lead to “dog hit(s) man”.

It is, however, not only the syntactic interrelation of words that plays a role. Also semantic concepts play an important role. To make our point clear, let us consider an example given in [Radford, 1988]. He mentions that in Serbocroatian the four words {Peter, read, book, today} may be put in any of the $4! = 24$ possible orderings without changing the meaning and, what is particularly interesting, all these utterance “paths” are allowed, i.e. are considered to be grammatical.

What we meet here is a phenomenon, that does not occur in English or Chinese. There only 4 or 5 of the 24 orderings are good. For example

* Peter today book reads.

is not allowed in English. Such non-grammatical sentences are indicated with a star:
*.

We can give an explanation why the 24 utterance paths are equally well possible. The sentence graph is the same for all these 24 sentences and is given in Figure 6.2.

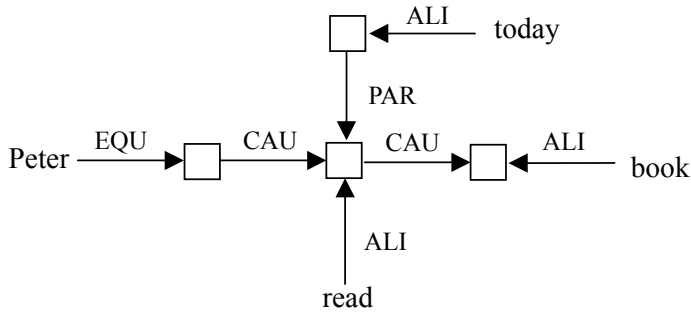


Figure 6.2 Sentence graph with the word set {Peter, read, book, today}.

The solution is that “Peter” cannot be read and “book” cannot read, which puts these pronoun and noun in the position of subject and object, purely on semantic grounds. In the case of “man hit(s) dog”, exchanging the positions of “man” and “dog” gave a semantically completely different sentence. But here exchanging the position of “Peter” and “book” does not have any consequence, for the meaning of the sentence, as the person reading or hearing the four words reconstructs the sentence graph in a unique way. Also the word “today” can only be attached to the verb “read”.

We conclude that there is, in this particular case, hardly any utterance rule. Just utter the four words, in any order.

6.2 Utterance Paths and Generative Grammar

A sentence is a linearly ordered set of words. In traditional parsing a parse tree is generated following grammar rules. Radford [Radford, 1988] calls the parse tree a *phrase marker*. In the parse tree phrases are usually easily recognized as subtrees.

Let us consider the sentence “Peter read(s) (the) book today”, again. In order to discuss the relationship between utterance path and generative grammar, let us recall that there is just one sentence graph, namely the one given in Figure 6.2. This sentence graph is the meaning of any ordering of the four words that is admissible, i.e. can be generated by the grammar. We may generate one example sentence by the rules:

S → NP VP
 NP → PN
 VP → V AP
 V → V N
 AP → adv
 PN → Peter
 V → Read
 N → Book
 adv → today.

The resulting parse tree is:

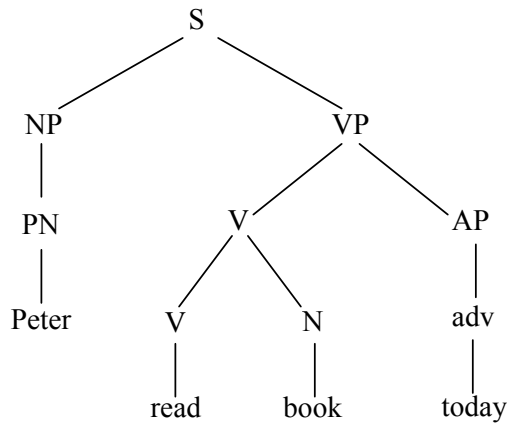


Figure 6.3 Parse tree for one particular uttering.

It is clear that, if the sentence graph is brought under words by an utterance path so that an admissible ordering, for the generative grammar, is obtained, the grammar has rules that justify the utterance path chosen. As a very simple example we compare the utterance “the book” with * “book the”. In English the determiner “the” must be uttered before the noun, and there is a grammar rule $N \rightarrow \text{det } N$, not a rule $N \rightarrow N \text{ det}$.

So we see that utterance paths must follow the rules of the generative grammar of the particular language considered. Suppose now that there is a language X in which there is only one way to make a sentence out of the four words {Peter, read, book, today}, and let this be, for example, the sentence “Peter read(s) (the) book today”. Then we

have one sentence graph, Figure 6.2, but in Serbocroatian there are 24 uttering possibilities, in English or Chinese only a handful, 4 or 5, and in X there is only one possibility.

Note that these numbers depend on the grammars for these languages and are numbers of phrase markers/parse trees that correspond to sentences with identical meaning.

The main conclusion for the problem of utterance paths for sentence graphs is that it is very difficult to give general rules for bringing a sentence graph under words.

In the language X there is no flexibility at all. The ordering of the words is precisely prescribed. X is a typical SVO-language. In Serbocroatian there is high flexibility. The ordering of the uttered words is not dependent on the structure of the sentence graph. Yet it may turn out that some of the 24 possible utterings of the sentence graph are more often used than others. It is not unlikely, for a European language, that SVO orderings are more often used.

To enable a generative grammar to generate all 24 possible orderings of the four words, it must have many different production rules. Language X only allows one ordering and must therefore have only a restricted set of production rules. There is a corresponding difference in the numbers of possible utterance paths for the sentence graph.

On the other hand we may say that the number of utterance *rules* in Serbocroatian, for this example sentence, is very small;

- “Utter some word”
- “Let a word be followed by a next word”.

In X the number of utterance rules is large.

- “Start with this specific word”.
- “Then let it be followed by that specific word”, etc.



There are $n!$ permutations of n words. If all are admissible, for some grammar, then, again, the number of rules for uttering a sentence consisting of these n words is very small; 2. If only one permutation is admissible the number of rules is large; n .









6.3 Uttering an Extended Example Sentence Graph

We have seen in the last section that the number of ways to produce a sentence, given a unique sentence graph, depends strongly on the language used. We will therefore not focus on general rules, but just investigate whether in English or Chinese patterns can be found in the way admissible utterance paths reflect the structure of the graph. Figure 6.1 can only be uttered as “man hit(s) dog”, using only these three words, following the SVO-structure of sentences in English. We start with the token for the subject noun in the graph and follow the path determined by the CAU-arcs. This example is now extended to the following sentence:

“The mean tall man hit(s) the poor small dog in the garden with a very big stick”.

Next to the verb “hit” there are the NPs “the mean tall man” and “the poor small dog”, and the APs “in the garden” and “with a very big stick”. We want to see in which order these phrases can be uttered.

First we remark that the graph in Figure 6.1 can also be uttered as “dog **is** hit **by** man”. When uttered this way, the “be” frame that can be considered to be present around any sentence graph, and therefore omitted, is now explicitly mentioned. As the object “dog” is mentioned first, thus suggesting a CAU-arc going out from this token, the fact that it is the “man” who is the agent is expressed by the incoming CAU-arc of “hit”. The word graph for “by” can be taken to be: . Something that is doing the act, that is expressed by the verb that must follow the outgoing arc, can be brought under words as “by (something)”. This extends our possibilities to order the phrases together with the central verb. However, the verb has to be sandwiched by the two NPs also in this way of uttering. The two APs can occur on any of the eight places indicated by the symbol  in :

  man   hit   dog   .

In written language we will use commas and in spoken language pauses will be taken. For example one might write “In the garden the mean tall man, with a very big stick, hit the poor small dog.” Or: “The mean tall man, with a very big stick, hit(s) the poor small dog, in the garden.” Or: “The mean tall man hit(s) the poor small dog, in the garden, with a very big stick.”

There is high flexibility in uttering the four phrases, apart from the positioning of the two NPs with respect to the verb. Like for the example considered in Section 6.2, this can be understood on semantic grounds. The APs are adverbial phrases for the verb. Before considering the phrases themselves let us give the complete sentence graph in Figure 6.4.

Note that “the” with word graph $\square \xrightarrow{\text{EQU}} \square$ (3 times) and “a” with word graph element $\xrightarrow{\text{ALI}} \square \xrightarrow{\text{PAR}} \square \xleftarrow{\text{ALI}}$ set are not mentioned in the graph as we wanted to avoid frames in the figure. Also the prepositions “with” with word graph $\square \xrightarrow{\text{PAR}} \square$ and “in” with the word graph $\square \xrightarrow{\text{SUB}} \square$ are not mentioned, for the same reason.

The four phrases correspond to four subgraphs of the sentence graph, that all four have tree structure. The two NPs have corresponding subgraphs that are identical in structure. The determiner has to be uttered first, the two adjectives can then be uttered in arbitrary order, but before the head noun, unless commas are allowed, as then one might write “the dog, small, poor”. The AP “with a very big stick” shows an extra feature. The connecting preposition is uttered first and then the determiner. But then the subgraph corresponding to “very big stick” shows a clear utterance path, starting as the outermost token for “very” via that for “big” to that for “stick”. Its arcs are of the PAR-type. Let us call the subgraph induced by the two CAU-arcs the main structure. Then we can say that, in English, the main structure follows the SVO pattern. The phrases are extensions of the head nouns or the verb. We already saw how the corresponding subgraphs are uttered. Most importantly, we must remark that these subgraphs are uttered “in one piece”. One cannot utter one of its constituent words while uttering another subgraph. These subgraphs are phrases that correspond to the chunks that we considered in Chapter 5. The rather arbitrary order in which these phrases can be uttered, when pauses are used in spoken language, is possible due to the fact that the syntactic function of the APs is fixed; they attach to the verb. We recall that chunks were introduced, by pointing out that people tend to make pauses in speaking.

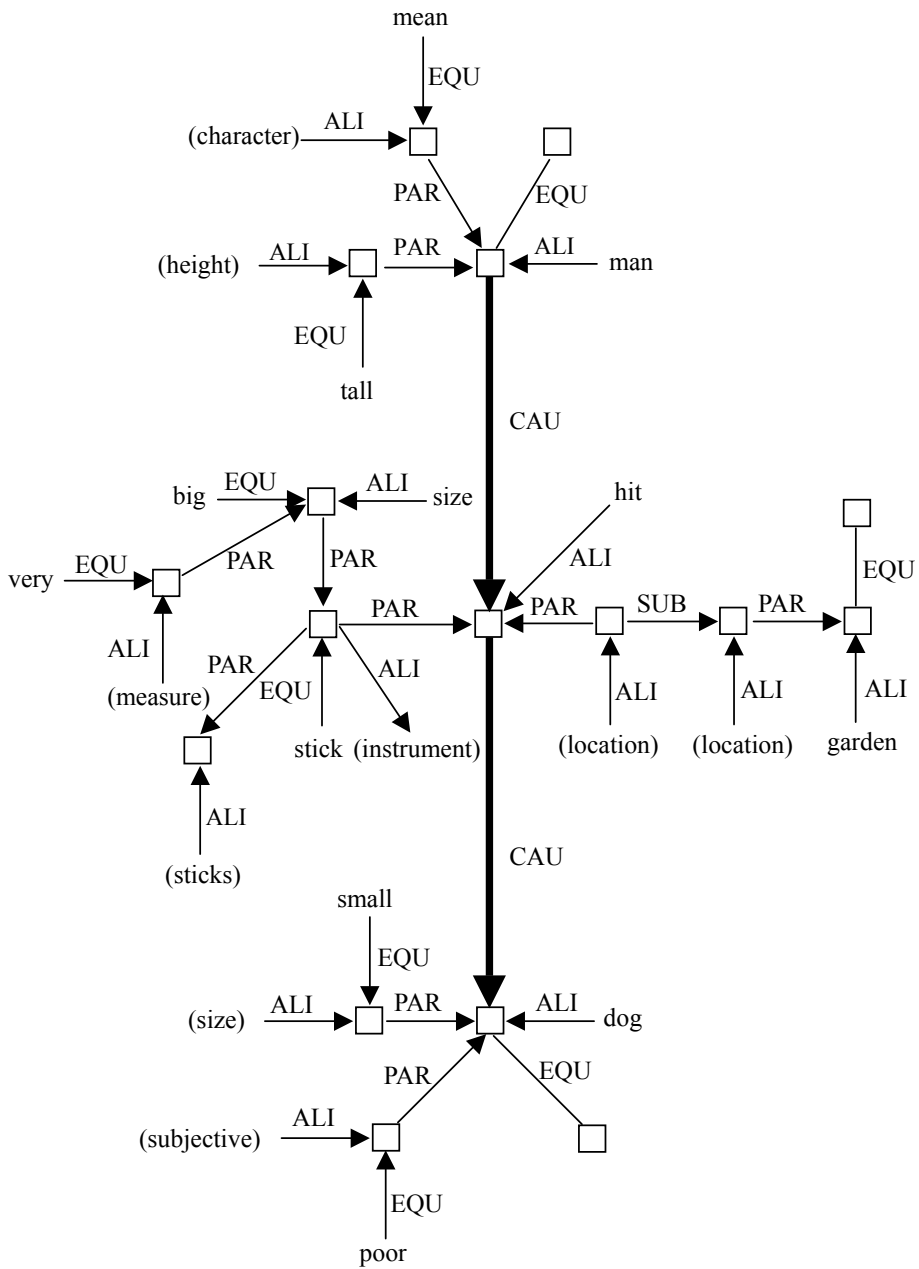


Figure 6.4 Sentence graph for the extended example sentence.
Words between parentheses do not occur.

6.4 Uttering a Sentence Graph with Reference Words

In language many times reference words are used. The most common word is the determiner “the”, that refers to something in the context of the conversation. In “man hits dog”, this reference has been omitted, but in “the man hits a dog” we have reference to some determined man, whereas the article “a” does not refer to a particular dog. Words like “this”, “that”, “these”, “those”, “who”, “which” and pronouns like “he”, “she”, “him”, “her”, “us”, etc. serve to avoid repetition of already known or mentioned constituent words in a discussion.

In Chapter 5 we considered the example sentence “The volcano, that lies in Alaska, 130 kilometers from Anchorage, erupted in 1992.” This sentence could also be uttered as “The volcano, that erupted in 1992, lies in Alaska, 130 kilometers from Anchorage.” In both cases the word “that” refers to the volcano and is used for economic reasons.

The basic structure of the sentence graph is

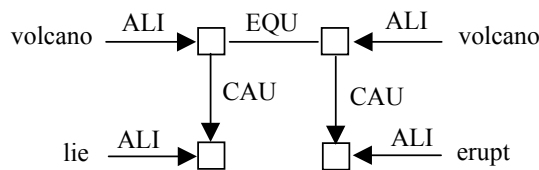


Figure 6.5 A subject for two verbs.

In uttering one could cope with the occurrence of two verbs by uttering two sentences, “The volcano lies, etc.” and “The (same) volcano erupted, etc.”. However, the repetition of “The volcano” can be avoided by referring to the once mentioned volcano by means of the word “that”. Due to the symmetry of the figure we can choose either of the two verbs for the main utterance path and utter the other verb in the subsentence containing “that”, on a side line so to say.

In a sentence like “He hit the dog, that hated him”, we have multiple use of reference. “He”, “the”, “that” and “him” all have a reference function. “He” refers to some man, “the” to a particular dog, “that” to that dog and “him” to “he”. In the sentence graph these references should be recognizable. As the reference is described by the EQU-link, we have a graph like

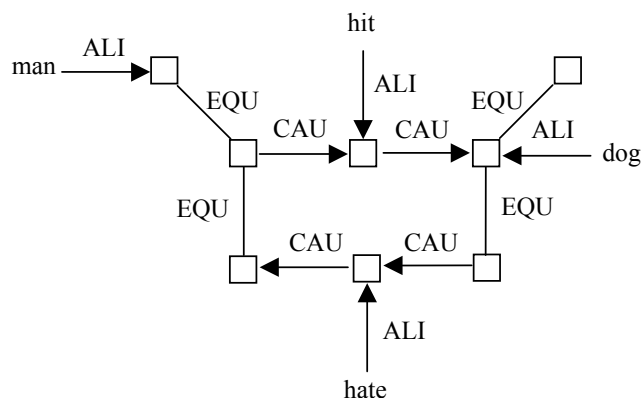


Figure 6.6 Sentence graph for “He hit the dog, that hated him”.

Note that we could have contracted two of the EQU-links, the two vertical ones, but then the reference words would not have been so clear. We essentially have a graph for two sentences; “He hit the dog” and “The dog hated him”, where we omit a discussion of the tenses.

Also note that the subgraph $\text{man} \xrightarrow{\text{ALI}} \square \text{---} \text{EQU} \text{---} \square$ is given as the word graph for “he”, which is considered to have the meaning “something like a man, to whom is referred”.

If a sentence graph contains more verbs, we see that we need not have a tree structure. Before uttering, reference links can be introduced so that the main SVO-structures are completely represented with tokens for S, V and O, here for “he hit dog” and “dog hate him”. Now the utterer can choose between cutting the graph by deleting EQU-links, while transferring all information attached to one token also to the other, or leaving the graph intact. In the first case more sentences have to be uttered, with repetition of words. In the second case reference words can be used. Once the graph has been replaced by more graphs, containing only one verb, the occurring phrases can be uttered according to rules as discussed in Section 6.3.

6.5 Uttering Graphs Containing Frames

In this section we discuss the uttering of sentences with sentence graphs that contain frames. At the same time some aspects of representing tense will be discussed. We consider an example mentioned by Radford [Radford, 1988], that reads “He might

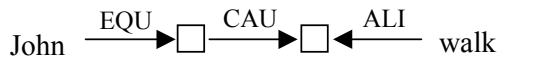
have been writing a letter”. We place “writing a letter” by “walking”. The things we want to investigate can also be discussed for the sentence “John may have been walking”.

We intend to gradually increase the complexity of the sentence in order to see how frames come into play. Both English and Chinese sentences are considered.

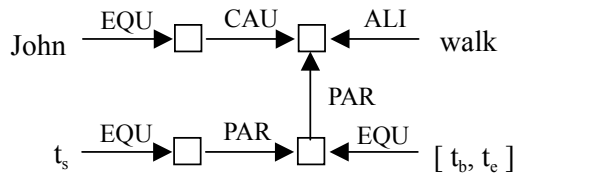
First we consider the very simple sentence

John walks.
John san4bu4.

The sentence graph is



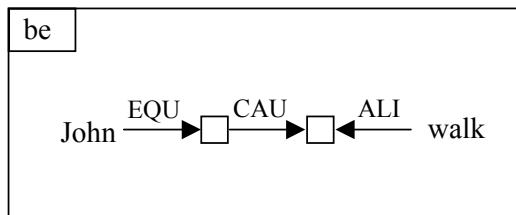
The tense, present, can be expressed by relating the time at which the act, described by the verb, is taking place to the time t_s at which the sentence is spoken. The present tense is characterized by $t_s \in [t_b, t_e]$, where the time interval $[t_b, t_e]$ denotes the time from the beginning of the act, denoted by t_b to the end of the act, denoted by t_e . In the sentence graph this leads to a graph of the following form:



Now we consider the sentence

John is walking.
John zai4 san4bu4.

The change that has taken place is the use of the auxiliary verb “be”. In the sentence graph “be” is expressed by a frame. The graph now looks like

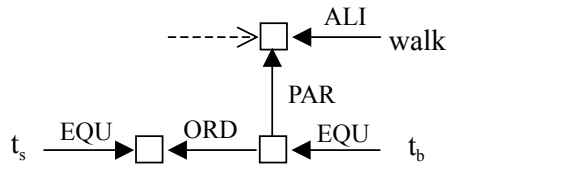


Note that there is hardly any difference in meaning. The “be” frame is put around the sentence graph of “John walk”. In Chinese “zai4” can be seen as an adverb, expressing time, namely “now”, “John now walk” is the literal translation of the Chinese sentence.

As a third sentence we consider the imperfect past tense. The sentence reads

John walked.
John san4bu4 le.

In the sentence graph of the first sentence only the time description changes. As the walking act has taken place in the past, but has not yet been ended for sure, we describe this by



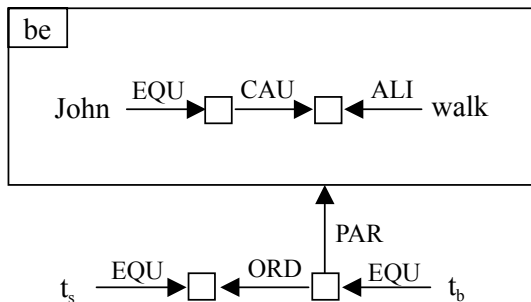
where only the essential part of the graph has been given. The ORD-relation expresses that the act began before the time of speaking.

In Chinese the use of the word “le” is quite typical. It is in one group with words like “ma”, see Section 5.2.2.

The fourth sentence is

John was walking (when I swam yesterday).
(Zuo2tian1 wo3 you2yong3 shi2) John zai4 san4bu4.

The sentence graph is identical with that of “John is walking”, but for the description of the tense aspects by “...shi2”. This is described by the graph construction used in the third sentence, to describe the imperfect tense, now attached to the “be”-frame.

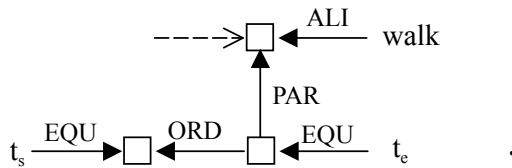


The literal translation of the Chinese sentence is “John in the past now walk”.

The fifth sentence is

John	has	walked.	
John	yi3jing4	san4bu4	le.

The perfect past tense is posing some problem, in English. The problem is the use of the auxiliary verb “have”. There are three merological relation types in the ontology of knowledge graphs; the FPAR-relation, describing properties, the SUB-relation, describing parts and the PAR-relation, describing attributes. We have seen, in Chapter 4, that “have” can be interpreted as “be with”. All three merological relationships are expressed in English by use of the words “with” or “of”. If the interpretation of “have” as “be with” is taken to hold universally, then in “John has walked”, there is the problem of identifying the part of the sentence graph corresponding to the word “with”. One way of looking at the use of “have” is that “John”, after having completed the walking obtains this as a property. However, in Chinese the word “yi3jing1” expresses the completion of the walking act. This word acts as an adverb of time. So the sentence graph looks like



The only change that has taken place, with respect to the sentence graph of “John walked” is that t_b has been replaced by t_e . This very concisely describes that the act has indeed been completed. It looks that the use of “have” in English to describe the perfect past tense must be seen as a special linguistic development. The Chinese way of expressing the perfect past tense is more consistent with the way other tenses are expressed. The knowledge graph representation is more closely following the Chinese language, at this point.

The sixth sentence reads

John	has	been	walking	(when	I	swam	yesterday).	
(Zuo2tian1	wo3	you2yong3	shi2)	John	yi3jing1	zai4	san4bu4	le.

Again the only change, with respect to the graph for “John was walking”, is the

replacement of t_b by t_c , and consequently in Chinese the use of the word “yi3 jing1” for expressing the completion.

Finally we come to the seventh sentence

John may have been walking (when I swam yesterday).
 (Zuo2tian1 wo3 you2yong3 shi2) John ke3neng2 yi3jing1
 zai4 san4bu4 le.

The auxiliary verb “may” is expressed by a POS-frame, put around the whole sentence graph for the sentence “John has been walking”. In Chinese the word “ke3 neng2” is seen as an adverb. In fact, one might say “John possibly has been walking”. The choice for the POS-frame can be defended also by the fact that yet another way to express the sentence graph is “It is possible that John has been walking”. Note the use of the two reference words “it” and “that”.

The peculiar use of the verb “have” in English for describing the perfect past tense should also be considered against the fact that in rather related languages like Dutch or German “has been” is expressed by “is geweest” respectively “ist gewesen”. So, in those languages in stead of “have” the auxiliary verb “be” is used. This feature is completely avoided in Chinese.

6.6 Uttering Quantification

Let us recall the way existential quantifier and universal quantifier were expressed in knowledge graph theory. In Section 3.2 we said that the existential quantifier is expressed by a distinct knowledge graph. In a very simple case we may have a graph like $\square \xleftarrow{\text{ALI}} x$, which can be uttered as “something like x”. Anything stated about x can be described by a sentence graph containing x. If this statement is P(x) then the sentence graph describes P(x), an open formula in logic. If x is instantiated, for example by a, then the graph $a \xrightarrow{\text{EQU}} \square \xleftarrow{\text{ALI}} x$ can be uttered as $\exists x$, there exists an x, namely a. Combining the graph for P(x) and the instantiation we obtain a knowledge graph that can be uttered as $\exists x P(x)$, a closed formula in logic, or as “there is an x for which P(x) holds” in natural language.

The universal quantification poses some delicate and interesting problems. In the beginning of the second phase of the knowledge graph project words like all, any,

every and each were considered to be expressible by the SKO-loop, see Figure 3.7.

However, like for the fifteen different words in Chinese for the preposition “in”, in principle for the four mentioned words there should be four different word graphs. Investigating the word “dou1” in the context of uttering a given sentence graph in Chinese, various subtle differences for the words used in universal quantification came forward. In the next section we will first discuss these differences before dealing with the word “dou1”.

6.6.1 All, any, each and every

Let us begin with giving the description of these four words as found, e.g., in the Oxford Pocket Dictionary [Allen, 1984]. We read

- all** : 1. whole amount
2. all persons concerned, everything
3. adv., entirely, quite
- any** : 1. one, no matter which, of several
2. some, no matter how much or many or of what
- each** : every one of two or more persons or things, regarded separately
- every** : 1. each single
2. all possible.

Before discussing these entries in the dictionary, let us consider the Chinese translation possibilities. Words used in Chinese in universal quantification are

- dou1** : *adv.* all
- suo3you3 de** : all
- ren4yi4** : arbitrary
- ge4ge4** : 1. each, every
2. one by one, separately
- mei3ge4** : every, each, per.

The translations are as found in A Modern Chinese-English Dictionary [Ce, 1988]. One interesting remark is still to be made first. The logical statement $\forall x P(x)$ is

expressed in Chinese as “ren4yi4 x P(x)” or as “sou3you3 x P(x)”. In English we can say “For all x P(x)” and “For any x P(x)”. The Chinese word corresponding to “for” is “dui4”. The association with “dui4” is “concentrating on”.

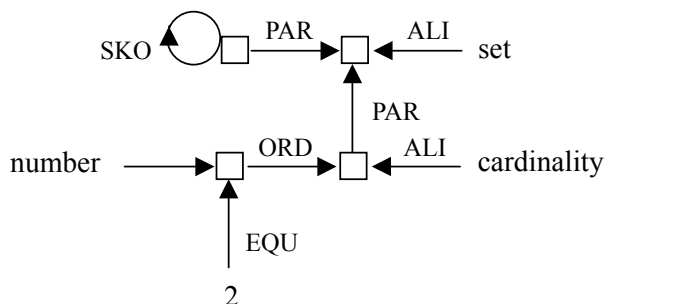
This last remark hints at a very important aspect, namely that a single element of a set is focused upon. In Chinese the statement “dui4 ren4yi4 x P(x)” is slightly preferred over “dui4 sou3you3 x P(x)”. The SKO-link was introduced with meaning “informationally dependent on”. A SKO-loop then can be read as “something informationally dependent only on itself”. This can be interpreted as “something arbitrarily considered”. Hence the word graph consisting of one token and a single SKO-loop can be named ANY. We take this as our starting point for finding word graphs for “each”, “every” and “all”.

The single element aspect comes forward in “every” with the meaning “each single”. In “each”, it comes forward in “regarded separately”. Here, however, there is the extra remark that “every one of *two or more* things” is part of the meaning of “each”. The entry for “every” also mentions “all possible”, meaning that there may be more elements. So whereas “any” focuses on the single element, “every” focuses on the single element *as part of a set*. This means that we can take the word graph



for the word “every”. Note the occurrence of the word graph for “any” in this graph, that can therefore be uttered as “any of (a) set”, which is synonym with “every”.

Now we can extend this graph to obtain the word graph for “each”:



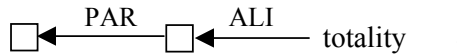
Note the occurrence of the word graph for “every” and the extension that can be uttered as “with cardinality greater than 2”.

In the meanings given for the Chinese words the single element aspect comes forward

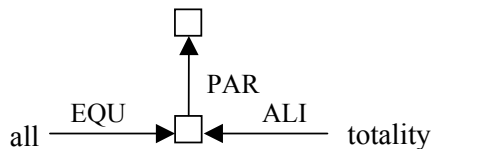
in “ge4ge4”, via the phrase “one by one”, and in “mei3ge4”, via “per”. The given entries suggest that the words “ge4ge4” and “mei3ge4” are closer in meaning than is expressed by the given word graphs. However, also in English the words “each” and “every” are not always distinguished very precisely, as can be seen from phrases like “each and every”. In mathematics we can speak of “every element of a set of one element”, but “each element of a set of one element” is in a subtle way less precise.

We are left with the word “all”. There were three entries given. The focus in all three meanings is on the *totality*. This brings us back to Kant’s ontology. Under the heading “Quantity” we found “unity”, “plurality” and “totality”. The first two express the element aspect and the set aspect. The third notion expresses the concept of “whole”. In language we see that we can speak of “all butter” as well as of “all dogs”. In both cases the totality aspect prevails. In a way the word “all” clearly differs from the three other words. This means that the use of the SKO-loop for describing the word “all” is put in doubt.

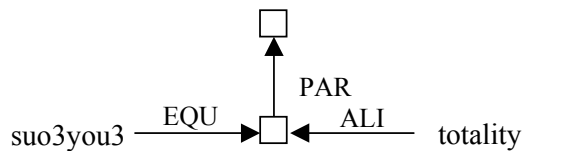
Consider expressions like “half the butter” or “almost all butter” or “hardly any butter”. What is described in these expressions are instantiations of the totality. For this reason we could consider the following graph:



Thus “totality” is seen as an attribute of something, that can assume different values, one of which is “all”. We then have the word graph



As “all butter” is expressed in Chinese as “suo3you3 de huang2you2” but not as “dou1 huang2you2”, this graph could also be given as the word graph for “suo3you3” in the form



The word “de” then corresponds to the PAR-link, as we have already seen before. We

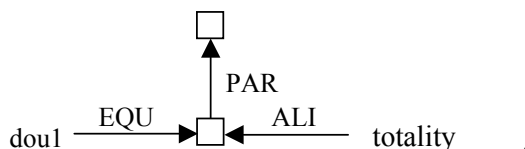
will discuss “dou1” in the next section. Note that “dou1” cannot be used for quantifying mass words.

6.6.2 Uttering the word “dou1”

As we will see in Section 6.7 the various ways to utter a sentence graph are controlled by the rules of the grammar that is assumed to be valid for the language. In Chapter 5 we discussed a grammar derived from the syntactic word graphs. This presupposes that the type of word is known. However, it turns out that for certain words, like the Chinese word “dou1”, there is disagreement on the type of that word. In a paper of Cheng [Cheng, 1995] on “dou1” quantification a discussion about the possible occurrence in a sentence can be found.

In this chapter we focus on uttering a sentence graph. This means that we choose to focus on sentence graphs containing the word “dou1”. In principle the possible utterings of such a sentence graph give all possible occurrences of the word “dou1”. However, we do not have rules for uttering such a sentence graph yet. We therefore studied the sentence graphs of several of the sentences mentioned by Cheng, in order to investigate the possibilities for uttering.

Note that this approach differs methodologically from that used by Cheng and others. They investigate certain sentences and partition them into the class of grammatically correct and the class of grammatically incorrect sentences. Then a discussion is held about similarity of sentences in the sense that “dou1” can or cannot occur in the same way as other words do. Cheng comes to the conclusion that “dou1” is a “nonmovable adverb”. However, in Section 6.6.1 “dou1” was seen as having a word graph like



This is an *adword*, because of the PAR-link. But to what type of word does it attach? As we discussed in Section 6.6.1 the totality either describes an aspect of a set or of an object. In both cases we do NOT have attachment to a verb. An essential remark in Cheng’s paper can be found in her example (2) that reads (she does not give numbers 1, 2, 3 or 4)

Tamen	dou	hen	xihuan	wo.
They	all	very	like	I

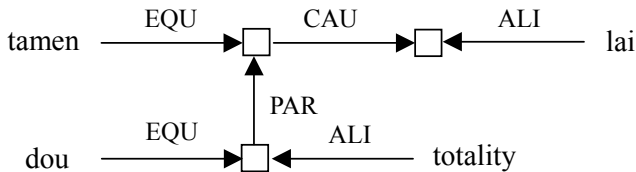
She remarks that “dou(1)” here quantifies a noun phrase NP to its left and that *the NP must have plural interpretation*. But that means, mathematically, that the NP must describe a set. We decided to focus on this aspect and investigate how various sentence patterns allowed the uttering of “dou” for the other example sentences in Section 2.1 of her paper.

Before doing this let us remark that statements like “tamen”, they, or “neixie xuesheng”, those students, have a plural interpretation. If the word “tamen” is used, in principle *all* the people described are meant. The same holds for “neixie xuesheng”. Any statement about “those students” in principle includes all of them. We may therefore consider the example sentences without the word “dou” to see whether the meaning of the sentence is essentially changed. If not the word “dou” is only used to express emphasis.

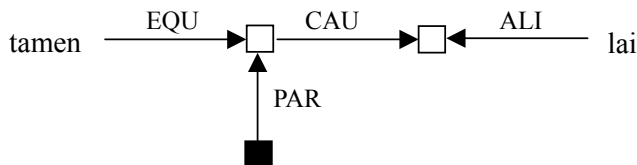
We now go through the example sentences.

- (1) tamen dou lai-le
 they all come-ASP
 “They all come.”
 (ASP, for aspect, is referring to the function of the word “le”)

The sentence graph, apart from the tense, is



The word graph for “dou” has been indicated here completely. In the following examples we will just link a black token to the word with plural interpretation. For this example sentence we would give the sentence graph



We see that “dou” is an adword of “tamen” and *not* an adword of “lai”. This speaks against the assumption that “dou” is an adverb. Note that the sentence “tamen lai-le” has essentially the same meaning. The word “dou” could have been omitted, and therewith the black token in the sentence graph.

Example sentence 2 involves a transitive verb.

- (2) tamen dou hen xihuan wo
 they all very like I
 “They all like me very much”.

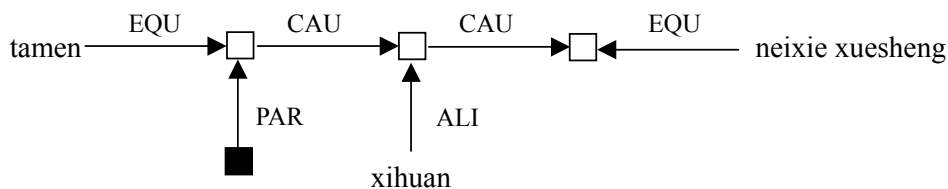
There is no essential difference with example sentence (1) as the word “dou” quantifies “tamen”, the only word with plural interpretation, and moreover it mainly functions as giving emphasis. It could have been left out here.

We now also consider this simple SVO-pattern in a sentence where both subject and object have plural interpretation.

- (3) tamen dou xihuan neixie xuesheng
 they all like those student.

Here both “tamen” and “neixie xuesheng” have plural interpretation. Yet the word “dou” can only be used once, namely quantifying the subject “tamen”.

The sentence graph is:



Again “dou” is used for emphasis. However, it cannot be an adword for “neixie xuesheng”. So, as an utterance rule, we can say that “dou” can only be used, to express emphasis, for a subject with plural interpretation.

Suppose we would like to say: “they like all those students”. In Chinese we might say “tamen xihuan suo you de neixie xuesheng”, but this is considered to be as good as simply “tamen xihuan neixie xuesheng”. This again shows that words with plural

interpretation do not have to be combined with words expressing universal quantification.

Next to the emphasis function, “dou” seems to have a reference function similar to that of determiners and pronouns.

- (4) zhexie xuesheng wo dou xihuan
 these student I all like
 “I like all of these students”.

Here “dou” is used as a reference word. The word “zhexie” already describes the plurality aspect. Replacing “dou” by “tamen” we have a similar sentence

- (5) zhexie xuesheng wo xihuan tamen
 these student I like they
 “These students I like them”,

where the only difference is that the referring word “tamen” now follows the verb.

Also in the case of an embedded sentence we encounter the reference function of “dou”. Cheng gives the sentence

- (6) neixie xuesheng wo xiangxin Lisi dou xihuan
 those students I believe Lisi all xihuan
 “All those students I believe Lisi likes them”.

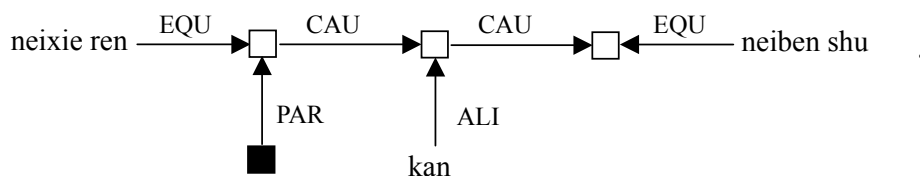
Already from the third line in which, in the translation, “them” is used for the reference, the reference function is clear. Also, from the translation, the direct adword function with respect to “those students” is evident.

Three other patterns will be considered.

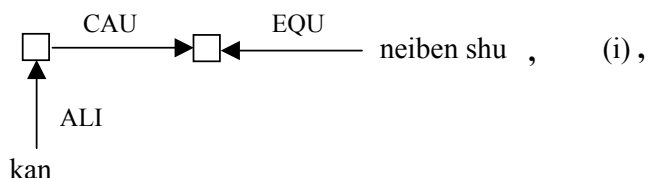
First there is the pattern of combination of “dou” and negation. Let us consider the example sentence

- (7) neixie ren \checkmark_i meiyou \checkmark_j kan-guo neiben shu,
 those person not read-ASP that book

where “dou” can be inserted on two places. We consider the sentence graph in both cases, without attention to the tense aspects.



This graph gives the basic structure without the NEG-frame: those people read that book. The NEG-frame can now be inserted in two ways, either around



or around the whole graph (j).

The meaning is then respectively

(i) All of those people did not read that book.

(j) Not all of those people read that book.

In the latter case in Chinese the subject “neixie ren” is taken out of the frame, and again “dou” has a reference function.

Finally there are the BA-pattern and the BEI-pattern.

A sentence in SVO-form may be uttered in SOV-form, but then the auxiliary word BA has to be used.

- (8) a. neixie xuesheng dou ba neiben shu mai-le
 those student all BA that book sell-ASP
 “All those students sold that book.”

Here the subject has plural interpretation and “neixie xuesheng” can be followed by “dou”, as emphasis.

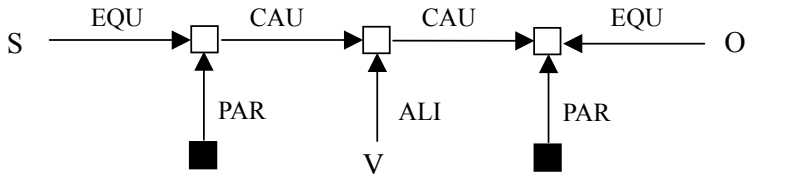
In the sentence:

- (8) b. zhangsan ba neixie shu dou mai-le
 zhangsan BA those book all sell-ASP
 “Zhangsan sold all those books”,

the object can be quantified by “dou”, by uttering “dou” after “neixie shu”.

In case both S and O have plural interpretation, again only one can be emphasized by “dou”.

In abstract sense we consider the following sentence graph:



We have assumed that both S and O have a plural interpretation and that there are two adwords “dou” attached to them. The uttering of this sentence graph is “All S V all O”. However, the essential meaning could also be uttered in “S V O”, because of the plural aspect of both S and O.

In Chinese we can say

“S dou BA O V”.

This is the uttering rule for “dou” in case of a sentence with BA-pattern.

Another, often used, pattern is the BEI-pattern, usually used for expressing the completed tense. Again Cheng gives two example sentences.

- (9) a. neixie xiaohai dou bei Lisi qifu-guo
 those children all BEI Lisi bully-ASP
 “Those children were bullied by Lisi.”

Here the uttering of “dou” is directly after “neixie xiaohai”, that has plural interpretation, and “dou” can occur only here. When uttered after “Lisi”, “dou” would have a reference function, but uttering it here is not considered.

- (9) b. zhangsan bei zhexie laoshi dou ma-le
 zhangsan BEI these teacher all scold-ASP
 “Zhangsan has been scolded by all these teachers.”

Again the uttering of “dou” is directly after “zhexie laoshi”, that has plural interpretation, and can occur only here.

Let us consider the abstract situation again, in which both S and O have plural interpretation.

The BEI-pattern turns the sentence “S V O” into the sentence “O BEI S V”. Uttering “dou” is now only possible in a sentence with one of S and O. In case both have plural interpretation both “O dou BEI S V” and “O BEI S dou V” are allowed. In

neixie xiaohai bei zhexie laoshi ma-le
those children BEI these teacher scold-ASP

we can utter “dou”, as emphasis, after “neixie xiaohai” or after “zhexie laoshi”.

From the knowledge graph point of view we see that “*dou*” is considered to be an adword attached to a noun with plural interpretation. It has two functions, one as quantifier, although that is not absolutely necessary. In that case the word puts emphasis on the noun. The other function is that of a reference word. It is therefore remarkable that in Cheng’s paper “*dou*” is considered to be an adverb. To put it in the words of Cheng, we think that “*dou*” is a head taking a noun as complement. In Section 3.2.1 Cheng states that “*dou*” falls within the class of *nonmovable adverbs*, like “*yijing*”, already. She gives example sentences

(30) a. Zhangsan yijing hui jia-le
Zhangsan already return home ASP
“Zhangsan has already return home.”

b. *Yijing zhangsan hui jia-le.

Sentence b is not correct. “*Yijing*” cannot “move” from after the subject to the front of the sentence. In the example

(20) a. jintian wo bu shufu
today I not comfortable
“Today I don’t feel well.”

b. wo jintian bu shufu

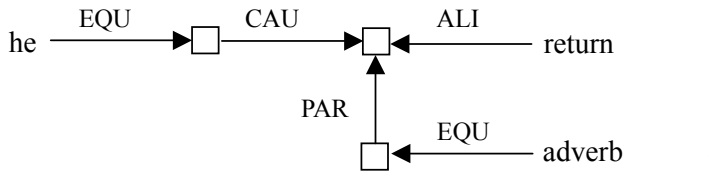
we see that “*jintian*” can be moved to the front. We quote:

“*Dou* is not a time adverb or attitude adverb, and it cannot appear before the subject. It is thus not a movable object. *Dou* falls within the class of nonmovable adverbs like

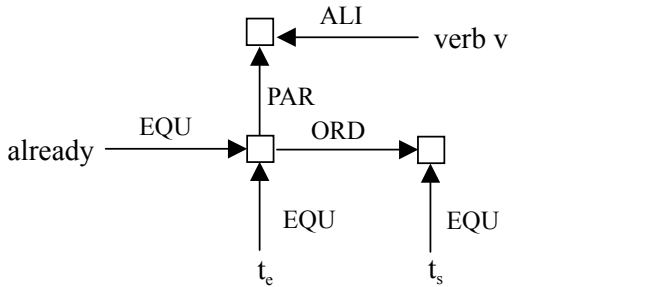
yijing ‘already’.”

This conclusion is mainly based on grounds concerning the *distribution* of adverbs in uttering. However, let us consider the two time adverbs “already” and “today”. One is nonmovable and the other one is. From the point of view of knowledge graph theory the difference should come forward in the sentence graph. We consider the two sentences “he returned today” and “he returned already”. The adverbs attach to the verb, by definition.

So in both sentences we have the basic structure

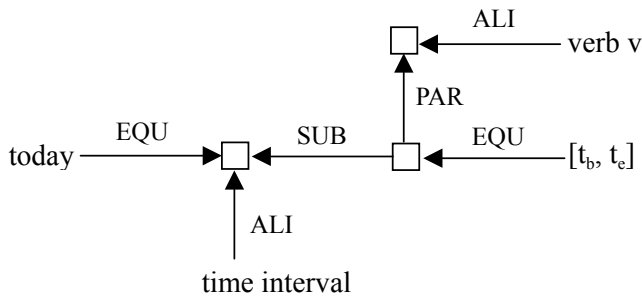


The difference must lie in the word graphs, the semantics, of the two adverbs. For “already” we have:



The act, described by the verb v , has finished at t_e , before the uttering time t_s of the sentence.

For “today” the graph is:



The difference between “already” and “today” is considerable, when we focus on the explicit description of the time aspects. There is a considerable semantic difference, although both graphs attach to the verb *v*, i.e., both words are adverbs.

This brings forward an important aspect of uttering. Although we are dealing with two time adverbs here, the structure of the sentence graph seems to have profound influence on uttering, in Chinese. So also for “dou” we may expect that its meaning plays an important role, i.e., the way it occurs in the sentence graph does, as “the structure is the meaning”. That “dou” has the distributional properties of a nonmovable adverb like “yijing” seems to be no reason to consider it to be a nonmovable adverb, like Cheng does.

6.7 Uttering and Grammar

Having considered the uttering of very specific types of words, we now want to start a more general discussion on uttering a sentence graph.

Let there be a set of utterings for a sentence graph. Each uttering should be grammatically correct. It must have a parse tree based on the grammar assumed to be valid. So there are as many parse trees for the one given sentence graph as there are utterings. A specific parse tree reflects an utterance path. We have seen that a sentence can be seen to consist of chunks. These chunks are on the one hand parts of the parse tree and on the other hand parts of the sentence graph. The following figure schematically describes the relationships between parse tree, utterance path, chunks and sentence graph.

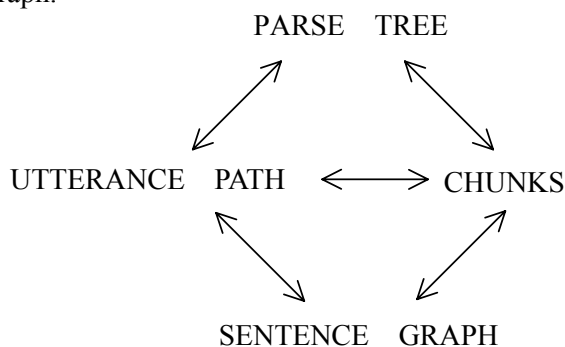
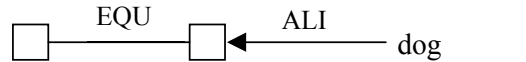


Figure 6.7 Relationships between parse tree, utterance path, chunks and sentence graph.

Our goal is to describe how a sentence graph can be uttered in a grammatically correct way. For this we have to indicate which words, or word groups, i.e., phrases, may follow each other. This typically is ruled by the syntax of a language and therefore the possible juxtaposition of two words is ruled by the production rules of the grammar.

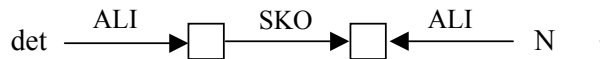
6.7.1 An introductory example

We will choose, as an example, the grammar for English derived in Chapter 5 from the way syntactic word graphs can combine. Let us, again, consider the uttering of the words “the” and “dog”, a determiner and a noun. In a sentence graph we might find the subgraph



where the EQU-link is the word graph for “the”.

The syntactic graph of a determiner in Figure 5.6 is

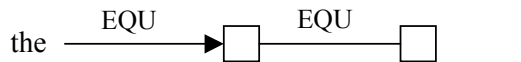


Let us now consider the production rule

$$6. \quad N \rightarrow \text{det } N,$$

expressing that a noun can be replaced by a determiner followed by the noun. The important aspect is that the rule does NOT state $6. N \rightarrow N \text{ det}$.

For this reason the subgraph can be uttered as “the dog”, but not as “dog the”. This is coming forward from the syntactic graph, from the direction of the SKO-arc. The word “the” might be given the semantic word graph:



Rule 6: $N \rightarrow \text{det } N$ gives us a possible uttering rule for a determiner attached to a noun. Whenever occurring in a sentence graph the allowed order can be indicated by an arc from the determiner to the noun. Note that this arc has no label, it merely expresses the possible order of the words in juxtaposition. Deleting the links of the sentence graph gives thus a graph containing only such “uttering arcs”, which might therefore be called an “uttering graph”.

Before systematically investigating the 18 production rules of our chosen grammar, we want to investigate rule 5:

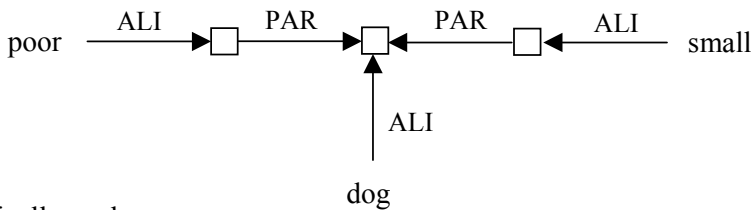
$$5. N \rightarrow \text{adj } N.$$

Here too an ordering for uttering an adjective and a noun is forced. Suppose we have the words “small” and “dog”, then we can say “small dog” and NOT “dog small”.

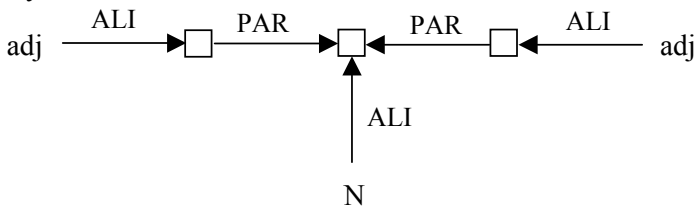
A complication comes in when rule 5 is applied twice:

$$N \rightarrow \text{adj } N \rightarrow \text{adj adj } N.$$

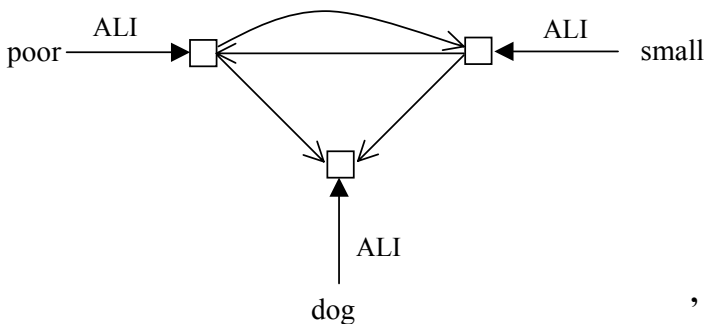
This implies that two adjectives can be uttered in juxtaposition before uttering the noun. Let us consider the words “poor”, “small” and “dog”. We may utter “poor small dog” as well as “small poor dog”. In a sentence graph we might find the subgraph:



Syntactically we have:



The uttering arcs are corresponding to the PAR-arcs towards the noun. The possibility of generating a second adjective by rule 5 implies that for the two adjectives there should be uttering arcs from one adjective to the other. The part of the uttering graph corresponding to the considered subgraph is now



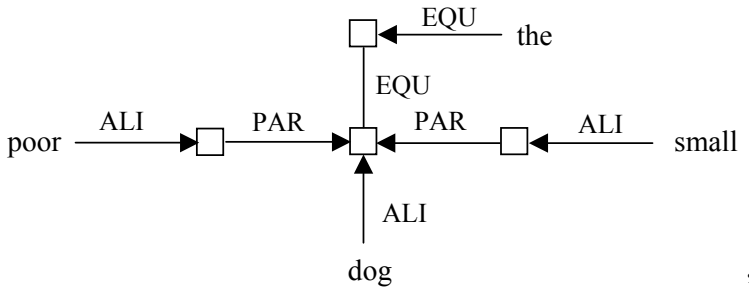
where we have maintained the typifying ALI-arcs for reasons of clarity.

Rules 5 and 6 can be combined, but only in the following order:

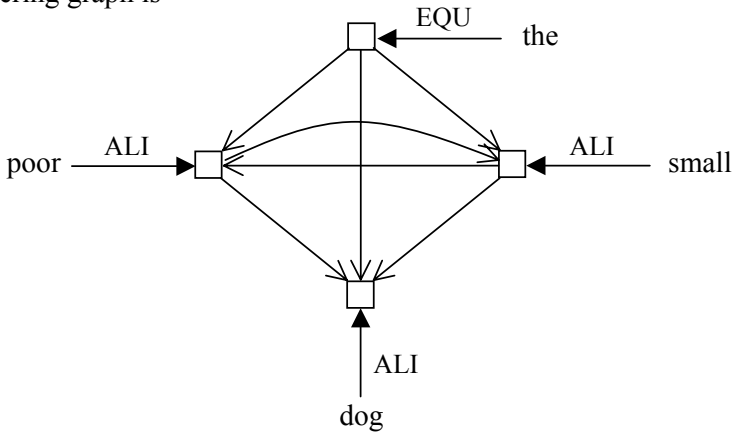
$$N \rightarrow \text{det } N \rightarrow \text{det adj } N.$$

This implies that there can be an uttering arc from the determiner to the adjective, but not the other way around.

Let us now consider the phrase “the poor small dog”, and let us give both the part of the sentence graph and its uttering graph. The semantic graph is



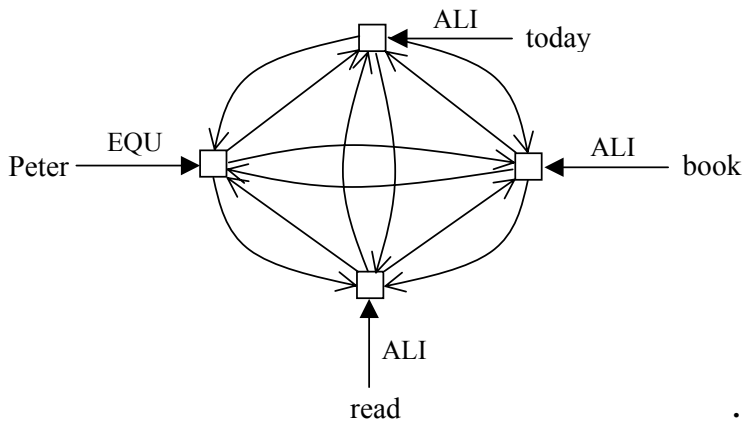
and its uttering graph is



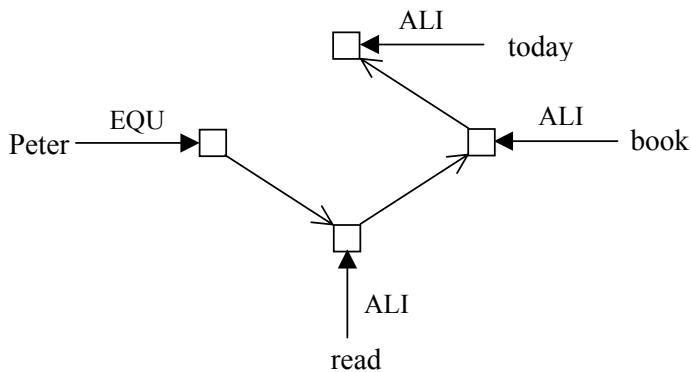
The arc from the determiner to the noun has been discussed. The arcs from the determiner to the two adjectives stem from the combination possibility for rules 5 and 6.

To finish this introductory example let us see how the semantic graph can be uttered. All four words have to be uttered. As there are only outgoing arcs for the determiner in the uttering graph, “the” must be uttered first. As there are only incoming arcs for the noun, “dog” must be uttered last. The two adjectives “poor” and “small” can be

uttered between “the” and “dog” in both possible orderings. Note that the two “uttering paths” in the uttering graph are Hamilton paths as all vertices are contained in the paths. In fact the number of possible uttering paths is equal to the number of Hamilton paths in the uttering graph. The uttering graph for the sentence graph given in Figure 6.2, for the Serbocroatian grammar, has 24 Hamilton paths and would look like



For the imaginary language X, only uttering was possible. The uttering graph looks like



and has one Hamilton path.

6.7.2 Uttering rules from production rules

Let us recall the grammar described in Section 5.3. We formed the following English grammar rules:

1. $S \rightarrow NP VP$
2. $NP \rightarrow PN$
3. $NP \rightarrow N$
4. $N \rightarrow N N$
5. $N \rightarrow \text{adj } N$
6. $N \rightarrow \text{det } N$
7. $AP \rightarrow \text{prep } N$
8. $N \rightarrow \text{num } N$
9. $N \rightarrow PN N$
10. $AP \rightarrow V N$
11. $VP \rightarrow V$
12. $VP \rightarrow V N$
13. $VP \rightarrow V PN$
14. $V \rightarrow V V$
15. $V \rightarrow \text{adv } V$
16. $V \rightarrow V \text{adv}$
17. $V \rightarrow V AP$
18. $\text{adj} \rightarrow \text{adv adj}$.

Note that this is just one of many grammars that may be considered. Our reasoning will be restricted by this choice.

The rules 2, 3 and 11 concern simple instantiation and therefore do not infer a condition on the order of uttering words or phrases. This leaves 15 rules to be considered.

6.7.2.1 Rules involving word types only

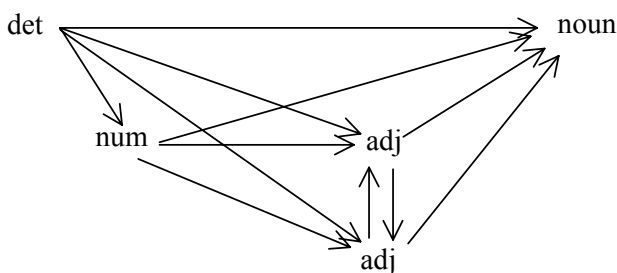
We start by considering the rules in which a noun N is involved, and no phrases. These rules are rules 4, 5, 6, 8, 9. It was already pointed out that there is a difference between the role of grammar rules in traditional parsing and the role they play in uttering a sentence graph. Given a sentence, the problem is to find a parse tree and the

order of applying rules is to be determined. However, when applying rules in arbitrary order we might generate phrases or sentences that are not correct. So only certain orderings of rules are possible. Rule 4. $N \rightarrow N N$ generates a juxtaposition of two nouns. The second noun cannot be taken to generate for example an adjective, by rule 5, that would then stand between the two nouns. We can say “severe thunder storm” but not “thunder severe storm”. An important other example is the combination of rule 5 and rule 6, as “the small dog” is a possible uttering, but “small the dog” is not. We will use our own knowledge to decide on the possibility of utterings.

Let us now consider the phrase

“The three mean tall men”.

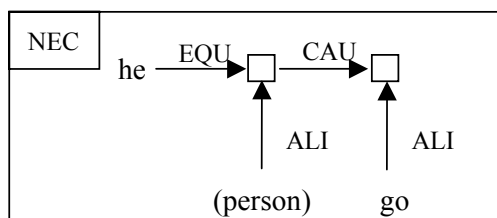
We know that the determiner “the” has to be uttered first, so the generation of this phrase should start with rule 6. Then the numeral must be uttered, which forces us to use rule 8. The two adjectives can now be generated by applying rule 5 twice. Their order is irrelevant. This now means that for uttering a part of a sentence graph in which a noun occurs, there are uttering orderings that can be indicated as:



These arcs will be called uttering arcs, *u-arcs*. Rule 9, in which a PN is generated before the noun is similar to rule 6. The schema describes the uttering rules for parts of the sentence in which the five rules must be used when parsing the uttered phrase. Rule 18 puts an adverb in front of an adjective, as e.g. in “very big”. For the uttering ordering expressed in the schema, this means that adv should be added and with arcs to adj, while from det and num there should be arcs to adv. Note that there cannot be an u-arc from adv to N. Also note that rule 18 can be repeated as we can say “very very big”. Remaining are rules 14, 15, and 16 involving V. Rules 15 and 16 describe that an adverb attached to a verb can be generated in both orderings. However, given a sentence graph, both cannot be used. For example we may say “he just came”, but not

“he came just”, while we can say “he worked hard”, but not “he hard worked”. So, in principle for uttering a verb-adverb attachment we need more information and cannot give a general uttering rule here.

Rule 14. $V \rightarrow V V$ is particularly interesting because of what we discussed in Section 6.5. The first V of $V V$ usually is an auxiliary verb, which in the sentence graph is represented by a frame. The sentence “He must go” has sentence graph



The frame expresses the auxiliary verb “must”. We have to start uttering with the PN “he”, because English is an SVO-language, but then immediately the auxiliary verb, the frame, must be uttered. Then the content of the frame must be uttered, which in this case is just the word “go”.

6.7.2.2 Rules involving phrases

We are left with rules 1, 7, 10, 12, 13 and 17. Let us begin with the rules that involve the adverbial phrase AP. Rule 7 shows that such a phrase may start with a preposition. We might therefore speak of prepositional phrase too, but prefer to use the more general notation AP.

In a sentence graph an AP starting with a preposition can be determined by that preposition. Note that the preposition, a link word, was one of the chunk indicators in Chapter 5.

Rule 7: $AP \rightarrow \text{prep } N$ generates a preposition before noun. This noun could be extended to a noun phrase, as considered in Section 6.7.2.1, but this means that there should then be u-arcs from the preposition to all the word types that can be uttered before the noun in the noun phrase. Rule 10 generates the order $V N$ as an AP. Rule 17 puts an AP behind the verb V . There are two conclusions here. An uttering, like “drinking coffee” may occur as an AP bringing two verbs in juxtaposition. This is just

a consequence of the fact that we took rule 10 into our grammar. Rule 17 restricts the position, in an uttering, of an AP. The AP should be uttered after the verb has been uttered. Repetition of rule 17 is possible and, here, two APs can be uttered after each other, in any order.

The verb phrase starts with a verb, as is determined by rules 12 and 13. Rule 12: $VP \rightarrow V N$ shows that there should be a u-arc from the V to the noun N, and to all that can be generated before that noun. Finally, rule 1 reflects the SVO-language and we have already discussed how to determine the NP. Determine the verb and, by the incoming CAU-arc, determine the noun in the NP, then add a u-arc from the end of the noun phrase (which is the noun) to the verb.

These are the u-arcs consistent with the considered grammar. We will now apply our findings to the extended example of Section 6.3.

6.7.3 Uttering paths for the extended example

We have discussed our representation of the uttering graph. We delete all links between tokens but maintain the EQU-arcs and ALI-arcs in order to indicate the words. For frame words like “with” and “in” we use auxiliary tokens to avoid too complicated graphs.

There are two ways to utter “the mean tall man” and “the small poor dog” parts. The APs can be uttered in only one way but in two possible orderings. The $2 \times 2 \times 2 = 8$ possible utterings correspond to the 8 Hamilton paths of the uttering graph. The 8 possible sentences are

1. The mean tall man hit in the garden with a very big stick the poor small dog.
2. The tall mean man hit in the garden with a very big stick the poor small dog.
3. The mean tall man hit in the garden with a very big stick the small poor dog.
4. The tall mean man hit in the garden with a very big stick the small poor dog.
5. The mean tall man hit with a very big stick in the garden the poor small dog.
6. The tall mean man hit with a very big stick in the garden the poor small dog.

7. The mean tall man hit with a very big stick in the garden the small poor dog.

8. The tall mean man hit with a very big stick in the garden the small poor dog.

Sentences 5, 6, 7, 8 are as 1, 2, 3, 4, but with interchange of the APs “in the garden” and “with a very big stick”.

Figure 6.8 gives the uttering graph of the sentence graph. The reader may read off the given 8 sentences by following the 8 different Hamilton paths.

Note that the APs could be uttered after “the poor small dog”, allowing the use of commas. This would again increase the number of possible utterings.

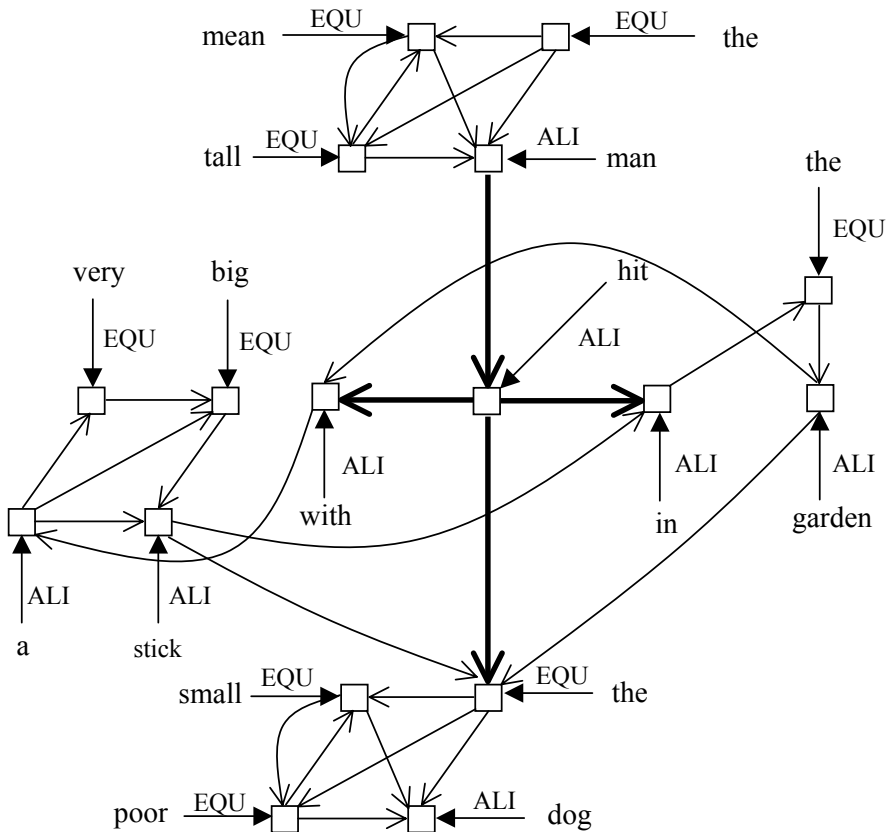


Figure 6.8 Uttering graph for the extended example.

Chapter 7

Information Extraction

7.1 Introduction

Over the last decade there has been a growing interest in developing systems for *information extraction (IE)*. In a broad view, information extraction refers to any process that creates structured representation of selected information drawn from one or more texts. Usually this process involves the identification of instances of a class of *events* or relationships and the extraction of the relevant arguments or relationships in a natural language text. The output of the extraction process, although varying in every case, is finally transformed to the content of some type of database.

An enormous amount of information exists only in natural language form. The idea of reducing the information in a document to a tabular structure goes back to the early days of NLP applications [DeJong, 1979, Schank & Abelson, 1977, Sager, 1987]. However, the specific notion of information extraction was relatively new in the series of *Message Understanding Conferences (MUCs)*. As a core of language technology, IE systems represent the need and ability to manipulate and analyze information automatically, by integrating a variety of natural language processing technologies. IE technology has not yet reached the market, but it was thought to be of great

significance to information end-user industries of all kinds as, for example, finance companies, banks, publishers and other document-dependent managing agencies.

To be a trend of “understanding” by extracting information from texts, IE technology should not be confused with the more mature technology of Information Retrieval (IR) that selects a relevant subset of documents from a large volume set by query. IE extracts information from the actual text of a document. Any application of IE is usually preceded by an IR phase. Information Extraction is a more limited task than “full text understanding”. In full text understanding, we expect the representation of *all* the information in a text in an explicit fashion. In information extraction, as a more focused and well-defined task, we restrict to the semantic range of the output: the relations we will represent, and the allowable fillers for each slot of a relation. For example, in the domain of terrorism, as the task given in the MUC-4 evaluation (1991), an IE system would extract the date, location, perpetrators, victims, targets and type of attack (bombing, arson, etc.). Since many phrases and even entire sentences can be ignored if they are not relevant to the domain, the IE process is computationally less expensive than in-depth natural language processing. In the last decade, IE has achieved notable success [MUC-3, MUC-4, MUC-5, MUC-6, MUC-7, Grishman & Sundheim, 1996, Cowie & Lehnert, 1996].

It should be noted that IE is not a wholly isolated information technology. For example, *MT* (*Machine Translation*) and IE are just two ways of producing information in applications and can be combined in different ways. One could translate a document and then extract information from the result or change the order of these procedures. Moreover, a simpler MT system might only be adequate to translate the contents of templates that resulted in an IE process. This means that the product of an IE system, i.e. the filled template, can be managed either as a compressed text itself, or as a form of database (with the fillers of the template slots corresponding to database fields).

7.2 The State of the Art of IE

Generally, the process of an IE system includes two major parts: (1) Extraction of the individual “facts” from the text through local text analysis, and (2) Integration of these facts to generate new facts through inference. Finally the pertinent facts are

represented and translated into the required output format.

According to the terminology established by the MUC, a *scenario* is specific to particular events or relations to be extracted, and a *template* refers to the final tabular output format of the IE process. Recent research on IE was stimulated in large part by the MUC evaluations. Five separate component tasks, which illustrate the main functional capabilities of current IE systems, were specified by recent MUC evaluation (MUC-7).

- (1) *Name Entity recognition* requires the recognition and classification of named entities such as organizations, persons, locations, dates and monetary amounts.
- (2) *Coreference resolution* requires the identification of expressions in the text that refer to the same object, set or activity. These include variant forms of name expression, definite noun phrases and their antecedents and pronouns and their antecedents.
- (3) *Template Element filling* requires the filling of small scale templates (slot-filler structures) for specified classes of entities in the text, such as organizations, persons, certain artifacts, and their locations, with slots such as name (plus name variants), description as supplied in the text, and subtype.
- (4) *Template Relation filling* requires filling a two-slot template representing a binary relation with pointers to template elements standing in the relation.
- (5) *Scenario Template filling* requires the detection of relations between template elements as participants in a particular type of event, or scenario, and the construction of an object-oriented structure recording the entities and various details of the relation [Humphreys, 2000].

Practically, IE development relies on recent advances in empirical NLP techniques. Many relatively independent modules within some general knowledge-based AI program have achieved significant success for a range of linguistic tasks, such as word sense tagging, syntactic parsing, sentence alignment and so on. Currently the most successful systems use a finite automatic approach, with patterns being derived from training data and corpora, or specified by computational linguistics. Recent research [Church *et al.*, 1996] has also shown that a number of quite independent

modules of analysis by learning and statistical methods can be built up independently from data, rather than coming from either intuition or some dependence on other parts of a linguistic theory.

There have been IE systems developed in groups stimulated by the MUC, such as POETIC [Mellish *et al.*, 1992], MITRE [Aberdeen *et al.*, 1995], FASTUS [Appelt *et al.*, 1995], SRA [Krupka, 1995], UMASS [Fisher *et al.*, 1995], LASIE [Gaizauskas *et al.*, 1995], NYU Proteus system [Yangarber & Grishman, 1998]), with alternative features to the IE task by applying NLP techniques. Many developers of IE systems have opted for robust shallow processing approaches that do not employ a general framework for “knowledge representation”. In other words, there may even be an attempt, without building a meaning representation of the overall text, nor representing and using world and domain knowledge in a general way to help in resolving ambiguities of attachment, word sense, quantifier scope, coreference, and so on. Such shallow approaches typically rely on collecting a large number of lexically triggered patterns for partial filling templates, domain-specific heuristics for merging partially filled templates to yield a final, maximally filled template, as exemplified in the systems of FASTUS [Appelt *et al.*, 1995] and the SRA and MITRE MUC-6 systems [Krupka, 1995, Aberdeen *et al.*, 1995]. However, there have been attempts to derive a richer meaning representation of the text with less task- and template-specific approaches, such as the discourse model and intermediate representation used in the design of the LASIE system [Gaizauskas *et al.*, 1995]. Such approaches were motivated by the belief that high level of precision in the IE task will not be achieved without attempting a deeper understanding of at least parts of the text. It was claimed that the MUC-6 evaluation showed that such an approach, with richer meaning representation, overall performed not worse than the shallow processing approaches [Cowie & Wilks, 2000].

A typical discussion about the NLP techniques used for IE is that of Hobbs [Hobbs, 1993]. According to Hobbs’ paper, the functionalities shared by most systems alternatively include the following [Cowie & Wilks, 2000]:

1. a Text Zoner, which turns a text into a set of segments.
2. a Preprocessor, which turns a text into a sequence of sentences.

3. a Filter, which turns a sequence of sentences into a smaller set of sentences by filtering out irrelevant ones.
4. a Preparser, which takes a sequence of lexical items and tries to identify reliably determinable small-scale structures.
5. a Parser, which takes a set of lexical items (words and phrases) and gives a set of parse-tree fragments as output.
6. a Fragment Combiner, which attempts to combine parse-tree or logical-form fragments into a structure of the same type for the whole sentence.
7. a Semantic Interpreter, which generates semantic structures or logical forms from parse-tree fragments.
8. a Lexical Disambiguator, which indexes lexical items to one and only one lexical sense, or can be viewed as reducing the ambiguity of the predicates in the logical form fragments.
9. a Coreference Resolver, which identifies different descriptions of the same entity in different parts of a text.
10. a Template Generator, which fills the IE templates from the semantic structures.

It should be noted that there exist disputes on the practice of IE with the above organization. For example, module 8 could be performed early on lexical items, or later on semantic structures. Within the process under module 5, some people use a syntactic parser but the majority uses some form of corpus-derived finite-state patterns to represent the lexical sequences, which process would be called “semantic parsing” [Cowie *et al.*, 1993].

For an IE task, defining templates is difficult, which involves the selection of the information elements required, and the definition of their relationships. The definition consists of two parts: a syntactic description of the structure of the template (often given in a standard form known as BNF-Backus Naur Form), and a written description of the rules on filling the templates and instructions on determining the content of the slots. The actual structure of the templates used has varied from the flat record structure of MUC-4 to a more complex object oriented definition used for

MUC-5 and MUC-6. For example, a person object might contain name, title, age and an employer slot, which is a pointer to an organization object. Such newer object style templates make it easier to handle multiple entities which share one slot, as they group together the information related to each entity in the corresponding object. However, the readability in printed form suffers a lot, as much of it consists of pointers.

Although many IE systems have been proved effective for information extraction on limited domains, there are difficulties in construction of a large number of domain-specific patterns. Manual creation of patterns is time consuming and error prone, even for a small application domain. In the following parts, a new IE approach is presented, which is using a domain-independent model described by knowledge graphs. This method is to Extract information by Knowledge Graphs (KGExtract) from the natural language texts.

7.3 Overview of the Approach

The development of language engineering applications, information extraction (IE) in particular, has demonstrated a need for the full range of NLP and AI techniques, from syntactic part-of-speech tagging through to knowledge representation and reasoning. The task of information extraction can be seen as a problem of semantic matching between a user-defined template and a piece of information written in natural language. To this purpose, the structural parsing oriented semantic processing will be applied to IE, and we will show that such a new IE technique has considerable advantages in comparison to traditional approaches to information extraction from the texts. For example, the method presented here, which is to encode the input pieces of information and the filled template into knowledge graphs, is a kind of graphic representation and it is domain-independent.

With respect to the advantages of the approach, our main points are:

- a simple (but semantically rigorous) model;
- the possibility of semantic checks guided by the model;
- a domain-independent representation;
- an automatic pattern acquisition;

Let us now discuss an approach that is a process consisting of the following phases:

Lexicon and Morphology. In the procedure of extracting information from NL texts, the precise duties of the lexical and morphological processing depend on the language that is being processed. For example, Chinese, without orthographically distinguished word boundaries, will require that some word segmentation procedure will be applied, but English can skip this procedure.

The most important problem faced in this phase is to handle the proper names in a text. Because most extraction tasks require the recognition of persons, companies, government organizations, locations, etc.

In addition to name identification, this phase must tag word types to words. We have discussed 8 word types in English in Chapter 5, such as noun, verb, adjective, pronoun, numeral, preposition, adverb and determiner. For each word type, we have created its syntactic word graph that represents the syntactic function of a word type. In Chinese we also chose 8 word types, but we replaced the “determiner” type by the “classifier” type.

Semantic chunk tagging. The role of this phase is to split the different sentences of the text into semantic chunks according to the chunk indicators, such as pairs of commas and/or period signs, auxiliary verbs, reference words, prepositions, “jumps”, etc. Here no more details about chunk indicators will be repeated since we have discussed the problem in Chapter 5.

Partial Structural Parsing. Some IE systems do not have any separate phase of syntactic analysis. Others attempt to build a complete parse of a sentence. Most systems fall in between, and build a series of parse fragments. In general, they only build structures about which they can be quite certain, either from syntactic or from semantic evidence. In our approach, a partial structural parsing method will be applied to every sentence of the input to build a series of semantic chunk knowledge graphs. We will then combine these knowledge graphs of chunks to derive the information that is to be extracted. The partial structural parsing can make the patterns, that will be mentioned in the next phase, to be created easily.

Domain-independent pattern creation. In order to extract information from texts, we have to have patterns representing entities and events occurring in the texts. Many

IE systems use a pattern-matching approach, but the set of patterns has to be created for each target task or target domain. If we are using a “pattern-matching” method, most work will probably be focused on the development of the set of patterns. However, for different domains changes will also be needed to the semantic hierarchy, to the set of inference rules, and to the rules for creating the output templates.

To summarize, other methods are domain-dependent, but our approach is domain-independent. This is the most important improvement of our approach in comparison with other systems.

Pattern Merging. This is the main part of our approach. The procedure of merging patterns is actually to integrate two semantic chunk graphs into a bigger one. It can be repeated until the number of semantic chunk graphs becomes 1.

Template Generation. Once a text has been fully processed and a domain-independent representation has been derived, this representation can be used to generate template structures.

7.4 Description of KG-extraction

This section addresses the theoretical issues related to the design and use of such a method for information extraction. We present some basic principles, and we illustrate a preliminary proposal for a model developed according to such principles.

Definition 7.1 A *KG-extraction* is the mapping of unstructured natural language texts onto predefined, structured representations, or templates, which, when filled, represent an extract of key information from the original text, with special regard to a model based on knowledge graph theory.

7.4.1 Partial structural parsing

It is very important to realize that the role of parsing in an information extraction system is not to perform full text understanding, but to perform parsing on the relevant parts of the text. The shallow parsing techniques tend to be imprecise, although efficient and transportable, whereas the full parsing approaches tend to be

very precise but not robust and efficient.

Our approach is to shift from a full structural parsing (which has been described in Chapter 5) to partial structural parsing. The partial structural parsing has the role not only of identifying syntactic structure but also of making the extracted information syntax independent “regularizing” or “standardizing” by constructing the semantic chunk graph. For more detail about regularizing, we refer to the example in Section 7.4.2.

Definition 7.2 *Partial structural parsing* is the mapping of a sentence that is in the input text onto a set of semantic chunk graphs of this sentence.

The goal of partial structural parsing is creating the scenario patterns of information to be extracted, not obtaining the full sentence graph. It is performed along almost the same phases as structural parsing, that is the mapping of a sentence on a semantic sentence graph, but for the last phase of structural parsing, which combines the various bigger semantic chunk graphs into a sentence graph.

7.4.2 An example of representing patterns with knowledge graphs:

KG-Structure

In order to extract the information from text, we have to have patterns representing entities of interest in the application domain (e.g., company takeovers, management successions) and relations between such entities in the texts.

In many other current extraction systems most of the text analysis is performed by matching text against a set of patterns. If the pattern matches a segment of the text, the segment of the text is assigned a label, and one or possibly more associated features. These patterns are domain specific. For the example of “executive succession”, there will be such patterns as:

<person> retires as <position>, <person> is succeeded by <person>.

In another text about “joint venture”, there will be the following pattern:

<company> forms joint venture with <company>.

In general, each application of extraction will be related to a different scenario. Most

work will probably be focused on the development of the set of patterns, because it is difficult and inconvenient to operate directly on the patterns. These patterns are varying as the system turns from domain to domain.

The approach described in this section is to represent such patterns by knowledge graphs. This provides a graphical representation of patterns (i.e., knowledge graphs), which is called KG-Structure. One can then operate conveniently on the knowledge graphs. The KG-Structure, which is domain-independent, aims at easing the burden of pattern creation.

In particular, different clause forms, such as active and passive forms, relative clause, reduced relatives, etc., are mapped onto essentially the same semantic structure (i.e. a semantic chunk graph). This regularization simplifies the scenario pattern creation.

For example, we do not need separate patterns for

Cars that are manufactured by GM.

... GM, which manufactures cars ...

... cars, which are manufactured by GM ...

... cars manufactured by GM ...

GM is to manufacture cars.

Cars are to be manufactured by GM.

GM is a car manufacturer.

etc.

Although these various clauses have different syntactic structure, they have the same meaning. This is why there is only one pattern represented by a KG-Structure that is to be constructed as follows.

7.4.3 Named entity recognition

Named entities (like organization, person, location, and position) are very important in the information extraction task. It is necessary to recognize the proper names in the

text and express them with a corresponding KG-structure.

Example

In the sentence “I want to go to San Francisco”, we recognize “San Francisco” as a place name, its representation by a KG-structure is as follows:



Other named entities can be represented with a similar structure, and we do not list them one by one.

7.4.4 Automatic pattern acquisition

In this section, we will propose an almost automatic method to acquire patterns useful for IE from the text input. This method does not require humans to design complicated patterns. The basic idea involves the following:

- Semantic chunk taggings are performed for each sentence in the text.
- Semantic chunks are extracted to be the source of the patterns, which are represented with a KG-structure.
- The source KG-structures of the tagged sentence are integrated on basis of their similarity.
- Inferencing and merging are performed by knowledge graph operation.
- Templates are then filled through matching a KG-definition with a KG-structure.

One of the strengths of this approach is that the KG-structure, being domain-independent, based on partial structural parsing, supports the generation of the templates or summaries in a language different from that of the input texts.

At the initial stage, each tagged sentence is regarded to be a pattern consisting only of semantic chunks and named entities. The merging can be done by combining the most similar pair of patterns into one pattern.

There are several methods to define similarity of structures that can be found in the literature. In this chapter we will not use any specific similarity measure.

7.4.5 Inference and merging

In many situations, partial information about an event may be spread over several sentences; this information needs to be combined before a template can be generated. In other cases, some of the information is only implicit, and needs to be made explicit through an inference process.

7.4.6 Generating templates

An IE system does not require full generation capabilities from the intermediate representation (the knowledge graph), and the task will be well-specified by a limited “domain model” rather than a full unrestricted “world model”. This makes a generation feasible for IE, because it will not involve finding solutions to all the problems of such a KG-structure.

Definition 7.3 A *KG-definition* is a knowledge graph that expresses the semantics of a template relevant to the information extracted.

Once a text has been fully processed and a KG-structure pattern, of those aspects of it required for the IE task, has been created, this pattern can be used to fill template structures. These template structures will include pointers to the entries (each entry is a knowledge graph expression of a location slot.) in the lexicon, which forms the KG-definition of the template. The KG-definitions of the templates are pre-defined and stored in the lexicon. Once a KG-definition of a template matches with a KG-structure of a pattern, a slot in the template will be filled up.

7.4.7 A worked out example

We will extract the information from the following text:

“George Gorrnick, 40 years old, president of the famous hotdog

manufacturer Hupplewhite, was appointed CEO of Lafarge Corporation, one of the leading construction material companies in North America. He will be succeeded by Mr. John.”

Input slots that stand for the information that people want to extract are:

EVENT

PERSON

AGE

OLD POSITION

NEW POSITION

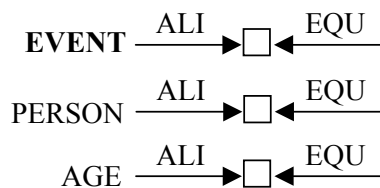
NEW COMPANY

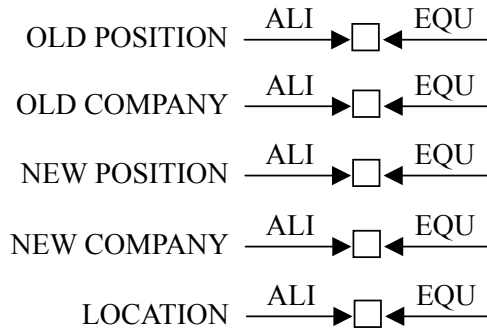
LOCATION.

As a result, we will fill each slot, and the output, that should be obtained, is shown as:

EVENT	appointment
PERSON	George Gorrnick
AGE	40
OLD POSITION	president
OLD COMPANY	Hupplewhite
NEW POSITION	CEO
NEW COMPANY	Lafarge Corporation
LOCATION	North America

The set of knowledge graphs corresponding to each slot is the following:





There are two sentences in our text, and we chunk each sentence step by step according to chunk indicators. There are 4 types of indicator, which have been mentioned in Chapter 5, that are used in this example. Besides them, we introduce a new indicator, which is the named entity (like organization, person, location, position), because they are very important in the information extraction task. Totally, we obtain the following 5 indicators:

- Indicator 0: comma or period signs.
- Indicator 1: auxiliary verbs, such as “will” in the second sentence.
- Indicator 2: reference words, such as “he” and “one”.
- Indicator 3: prepositions, such as “of”, “in”, as well as “by”.
- Indicator 4: names and numbers, such as “CEO” and “George Gorrnick”.

Easiest is Indicator 0. We get the chunks:

- 1 : George Gorrnick
- 2 : 40 years old
- 3 : president of the famous hotdog manufacturer Hupplewhite
- 4 : was appointed CEO of Lafarge Corporation
- 5 : one of the leading construction material companies in North America
- 6 : He will be succeeded by Mr. John
- 7 : effective October 1.

Note: Oct. 1 was abbreviated. The computer can replace Oct. 1 by October 1.

Indicator 3 is easy too. The prepositions cut the sentence just before the preposition. Just try to speak the sentence with natural pauses to see why we did this. We now find:

- 31 : president
- 32 : of the famous hotdog manufacturer Hupplewhite
- 41 : was appointed CEO
- 42 : of Lafarge Corporation
- 51 : one
- 52 : of the leading construction material companies
- 53 : in North America
- 61 : He will be succeeded
- 62 : by Mr. John,

next to chunks 1, 2 and 7.

Indicator 1 is about auxiliary verb forms and these, like prepositions, cut before the form. So

- 611 : He
- 612 : will be succeeded.

Note that “was appointed” has an auxiliary verb, but the sentence was already cut before “was” by the comma indicator.

Indicator 2 concerns “one” and “He”, but these chunks already stand alone in chunk 51 and chunk 611. So far we got:

- 1 : George Gorrnick
- 2 : 40 years old
- 31 : president
- 32 : of the famous hotdog manufacturer Hupplewhite
- 41 : was appointed CEO

- 42 : of Lafarge Corporation
- 51 : one
- 52 : of the leading construction material companies
- 53 : in North America
- 611 : He
- 612 : will be succeeded
- 62 : by Mr. John
- 7 : effective October 1.

We do not chunk up further in view of what we did so far. In particular “of”, “in” and “by” are linking **two** slots, indicating a relational template “in North America” is a chunk with one slot filled in, which may make it easier to find out the value of the other slot.

We want an **automatic** extraction procedure, to be followed by a computer. But a computer **cannot** make the jumps we make when we say “Now we make the semantic chunk graphs”. This is precisely the difficulty in artificial intelligence. We have to give very detailed instructions to go ahead in the information extraction process.

Now back to the names and numbers. We see CEO (Chief Executive Officer) as name, but it is the name of a position and so is “president”. “Officer” and “president” are both positions. The computer must know that and must know that CEO is short for Chief Executive Officer. When we prescribe/give slots like POSITION, then we must have a huge list of “values” for this slot and if “president” is not on that list, the computer cannot make semantic chunk graph 3. That is why it might be easier to generate names of slots ourselves. Suppose some lexicon gives: president: officer of a company, then we would introduce the slot OFFICER, and also for CEO we would do that. If the lexicon gives: president: position in a company, then we would generate the slot POSITION for “president” and OFFICER for “CEO”. Only when the computer knows that POSITION and OFFICER are similar, there is the possibility of reduction to just one slot, say POSITION. So this has its disadvantages too.

Assume that the computer knows all the prescribed slot names for the candidate words, so “president” is a word that can fill the slot POSITION. Because we prescribe the slots the lexicon of the computer may determine which words can fill one of the slots

EVENT, **PERSON**, **AGE**, **POSITION**, **COMPANY**, **LOCATION**. The computer may know for example:

EVENT : appointment, succession (recognition of verb forms, leading is not passing this test)

PERSON : He, Mr.

AGE : old (e.g. because the lexicon says something like “have age” for “old”)

POSITION : president, CEO

COMPANY : manufacturer, corporation, company

LOCATION :

So for these words an interpretation as slot fillers is assumed to be directly possible. The computer may look up “appointment” and “succession” as values of **EVENT**, as these are to be described by nouns. There is one **EVENT** per sentence, so that has been settled (as only thing so far).

What other preliminary action can the computer take? The names like George, Gorrnick, Hupplewhite, Lafarge, John, North America, October and the numbers 40 and 1 are supposed to be recognized as **NAME** and **NUMBER** respectively, but these are **not** among the given slots. What kind of names are they? The computer might find:

George : name of **PERSON**

Gorrnick : name of **PERSON** or name of **COMPANY** or name of **LOCATION**

Hupplewhite : name of **PERSON** or name of **COMPANY** or name of **LOCATION**

Lafarge : name of **PERSON** or name of **COMPANY**

North America : name of **CONTINENT**

John : name of **PERSON**

October : name of **MONTH**

40 : value of NUMBER

1 : value of NUMBER.

Before going over to the artificial intelligence part, let us remove adjectives: “famous”, “leading” and “effective”. Why? We are, given the slots, only interested in nouns, and more in particular in names and values. We can also replace by slot names where possible.

Having done all this preparation we now have the following:

1 : PERSON: George | PERSON, COMPANY or LOCATION: Gorrnick

2 : NUMBER: 40 | years | AGE: old

31 : POSITION: president

32 : of the hotdog | COMPANY: manufacturer |
PERSON,COMPANY or LOCATION: Hupplewhite

41 : EVENT: appointment | POSITION: CEO

42 : of PERSON or COMPANY: Lafarge | COMPANY: corporation

51 : one

52 : of the construction material | COMPANY: companies

53 : in | CONTINENT: North America

611 : PERSON: He

612 : EVENT: succession

62 : by | PERSON: Mr. | PERSON: John

7 : MONTH: October | NUMBER: 1.

From this we have to extract the desired information. That is, the computer has to and here is where the reasoning gets tougher for getting the semantic chunk graphs.

CHUNK 1 There are two names consecutive, one for a PERSON and one for PERSON, COMPANY or LOCATION. The computer should know that it has to conclude PERSON: George Gorrnick.

CHUNK 2 NUMBER: 40 and SET: years stand consecutive, so “40 years”. This is

followed by AGE: old which has a measure, so “40 years” must be the value of that measure. Conclusion AGE: 40 years.

CHUNK 31 POSITION: president. Here the main problem arises. OLD or NEW POSITION? The computer must choose OLD POSITION because of the place in the sentence. A position is attributed to a person and “president” follows “George Gorrnick”, so OLD POSITION: president.

CHUNK 32 hotdog is FOOD, and Hupplewhite is the name of a PERSON, COMPANY or LOCATION. So we extract COMPANY: Hupplewhite, as the other noun occurring in this chunk is of type COMPANY: manufacturer. The link implied by “of” is to president, but that means that this is the OLD COMPANY: Hupplewhite.

CHUNK 41 EVENT: appointment POSITION: CEO. See later.

CHUNK 42 of COMPANY or PERSON: Lafarge COMPANY: corporation. There is no problem here, it must be COMPANY: Lafarge.

CHUNK 51 one. This reference word still has to be dealt with, if necessary. The place in the sentence suggests reference to COMPANY: Lafarge.

CHUNK 52 of the construction material COMPANY: companies. This chunk does not contain information relevant to the given slots.

CHUNK 53 in CONTINENT: North America. Expansion of CONTINENT gives LOCATION, so LOCATION: North America is found.

CHUNK 611 PERSON: He. The reference must be to a person mentioned in the first sentence. The only person is George Gorrnick. Hupplewhite and Lafarge turned out to be companies.

CHUNK 612 EVENT: succession. See later.

CHUNK 62 by PERSON: Mr. PERSON: John. As “Mr.” is not a name the computer should combine to: by PERSON: John or Mr. John.

CHUNK 7 MONTH: October is a TIME-concept which is not one of the slots. So the computer should forget about this chunk.

As output we so far have for sentence 1:

EVENT	Appointment
PERSON	George Gorrnick
AGE	40 years
OLD POSITION	president
OLD COMPANY	Hupplewhite
NEW POSITION	
NEW COMPANY	
LOCATION	

We used the chunks 1,2, 31, 32 and 41 partly. For sentence 2 we so far have:

EVENT	Succession
PERSON	
AGE	
OLD POSITION	
OLD COMPANY	
NEW POSITION	
NEW COMPANY	
LOCATION	

Age and location are not mentioned at all in sentence 2, but COMPANY and PERSON and POSITION do occur. This has to be decided by solving the OLD/NEW problem.

The chunks 41 and 612 are the vital ones. The computer has to know what appoint and succeed mean.

The lexicon might give “appoint” = “give POSITION to”. The only position mentioned in chunk 41 is CEO. This must therefore, implied by “give”, be the NEW POSITION. From chunk 42 then follows that Lafarge is the NEW COMPANY and

the first template is filled after filling in the location: “North America”.

The lexicon might give “succeed” = “get POSITION of”. The preposition “by” leads to the proper choice. PERSON: John gets the NEW POSITION. This is implied by “gets”. The position is that of “He”, who is George Gorrnick, so it is president and of NEW COMPANY: Hupplewhite. For OLD POSITION and OLD COMPANY nothing is found.

7.4.8 Chunk graphs for the example

There is a difference between the status of the slot “EVENT” and the status of the other slots like “PERSON”, “AGE”, etc. The event is given by the whole text. To describe the event we choose to give nouns derived from the verbs used in the sentences. Thus we obtain “appointment” from “appointed” and “succession” from “succeeded”. For filling the other slots we should now discuss the role of semantic chunk graphs, as these form the essential parts of structural parsing. We will describe four phases illustrating the discussion given sofar.

The first phase gives the word graphs of the words occurring in the two sentences. We discuss the construction of these word graphs from an imaginary lexicon. The information included in the lexicon might be as follows:

1. George : Name of a male person.
 Gorrnick : No information in the lexicon.
2. 40 : Number.
 year : Measure of time interval.
 old : Of high age.
3. president (1) Leader of a state, (2) First officer of a company.
 of : Preposition, used for describing a property, part or attribute.
 the : Determiner.
 famous : Having fame.

-
- hotdog : Kind of sausage.
- manufacturer : (1) Factory, (2) Kind of company.
- Hupplewhite : No information in the lexicon.
4. was : Form of the verb “be”.
- appoint : Give a position to.
- CEO : Shorthand for Chief Executive Officer.
- of : Preposition, used for describing a property, part or attribute.
- Lafarge : No information in the lexicon.
- corporation : Kind of a company.
5. one : (1) Number, (2) Pronoun, referring to an element of a set.
- of : Preposition, used for describing a property, part or attribute.
- the : Determiner.
- leading : Adjective, built from the verb “lead”.
- construction : (1) Building (2) The act of building.
- material : Matter.
- company : Synonym of “firm”.
- In : Preposition, used for describing that something is part of something else.
- North America : Name of a continent.
6. he : Pronoun, referring to a male person.
- will : Form of the auxiliary verb “will”, used to express acts in the future.
- be : Auxiliary verb, used to express a situation.
- succeed : Get the position of.
- by : Preposition, used for describing an actor or a cause of a verb.


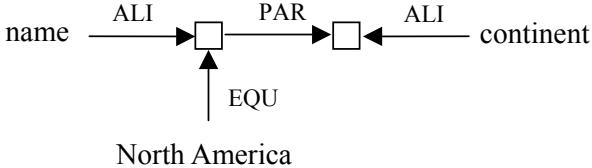
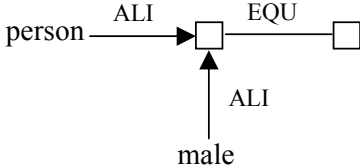
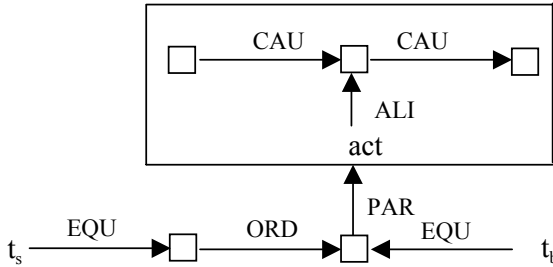

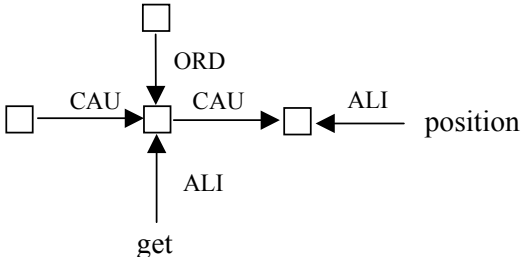

- Mr. : Address form for a male person.
- John : Name of a male person.
- 7. effective : (1) Causing effect, (2) Starting.
- October : Name of a month.
- 1 : Number.

The word graphs for these words can now be constructed

1.	George:		
	Gorrick:		
2.	40:		
	year:		
	old:		
3.	president:	(1)	
		(2)	

	of:	(1) □ $\xrightarrow{\text{FPAR}}$ □ , (2) □ $\xrightarrow{\text{SUB}}$ □ , (3) □ $\xrightarrow{\text{PAR}}$ □
	the:	□ $\xrightarrow{\text{EQU}}$ □
	famous:	fame $\xrightarrow{\text{ALI}}$ □ $\xrightarrow{\text{PAR}}$ □
	hotdog:	hotdog $\xrightarrow{\text{ALI}}$ □ $\xleftarrow{\text{FPAR}}$ □ $\xleftarrow{\text{ALI}}$ sausage
	manufacturer:	<p>(1) □ $\xleftarrow{\text{ALI}}$ factory , (2) □ $\xleftarrow{\text{ALI}}$ manufacturer</p> <p>□ $\xrightarrow{\text{CAU}}$ □ $\xrightarrow{\text{CAU}}$ □ $\xrightarrow{\text{CAU}}$ □</p> <p>□ $\xleftarrow{\text{ALI}}$ company $\xrightarrow{\text{FPAR}}$ □ $\xrightarrow{\text{CAU}}$ □</p>
	Huplewhite:	
4.	was:	<div style="border: 1px solid black; width: 150px; height: 50px; margin: 0 auto; padding: 5px;">be</div> <p>□ $\xrightarrow{\text{EQU}}$ □ $\xleftarrow{\text{ORD}}$ □ $\xleftarrow{\text{EQU}}$ □ $\xrightarrow{\text{PAR}}$ □</p> <p>t_s $\xrightarrow{\text{EQU}}$ □ $\xleftarrow{\text{ORD}}$ □ $\xleftarrow{\text{EQU}}$ t_b</p>
	appoint:	<p>□ $\xrightarrow{\text{CAU}}$ □ $\xrightarrow{\text{ORD}}$ □ $\xrightarrow{\text{CAU}}$ □ $\xleftarrow{\text{ALI}}$ position</p> <p>□ $\xrightarrow{\text{ALI}}$ give $\xrightarrow{\text{CAU}}$ □</p>

	<p>CEO:</p>	
	<p>of:</p>	<p>(1) $\square \xrightarrow{\text{FPAR}} \square$, (2) $\square \xrightarrow{\text{SUB}} \square$, (3) $\square \xrightarrow{\text{PAR}} \square$</p>
<p>5.</p>	<p>one:</p>	<p>(1) number $\xrightarrow{\text{ALI}}$ \square $\xleftarrow{\text{EQU}}$ 1 ,</p> <p>(2)</p>
	<p>of:</p>	<p>(1) $\square \xrightarrow{\text{FPAR}} \square$, (2) $\square \xrightarrow{\text{SUB}} \square$, (3) $\square \xrightarrow{\text{PAR}} \square$</p>
	<p>the:</p>	<p>$\square \xrightarrow{\text{EQU}} \square$</p>
	<p>leading:</p>	<p>leading $\xrightarrow{\text{ALI}}$ $\square \xrightarrow{\text{PAR}}$ \square</p>
	<p>construction:</p>	<p>(1) $\square \xleftarrow{\text{ALI}}$ building , (2) $\square \xrightarrow{\text{CAU}}$ $\square \xrightarrow{\text{CAU}}$ \square</p> <p style="text-align: center;">↑ build</p>
	<p>material:</p>	<p>$\square \xleftarrow{\text{ALI}}$ matter</p>
	<p>company:</p>	<p>company $\xrightarrow{\text{ALI}}$ $\square \xrightarrow{\text{EQU}}$ $\square \xleftarrow{\text{ALI}}$ firm</p>

	in:	
	North America:	
6.	he:	
	will:	
	be:	
	succeed:	
	by:	

	Mr. :	
	John:	
7.	effective:	<p>(1) </p> <p>(2) </p>
	October:	
	1:	

These word graphs contain only little relevant information. There are two persons: “George” and “John”. “Age” is mentioned in “old”, but not specified of whom. “Position” and “Company” occur here and there, also without specification.

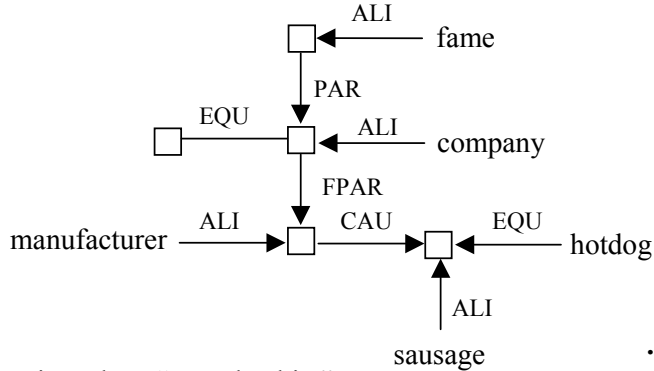
The second phase is to build chunk graphs from these word graphs. Note that we use *partial* structural parsing. The information that is to be extracted may be found from the chunks 1, 2, 31, 32, 41, 42, 51, 52, 53, 611, 612, 62, 7. Only if necessary, we combine these chunk graphs into graphs for larger chunks. If possible, we want to avoid complete structural parsing.

Chunk 1 : We only have at our disposal the word graph for “George”.

Chunk 2 : The three word graphs cannot yet be combined.

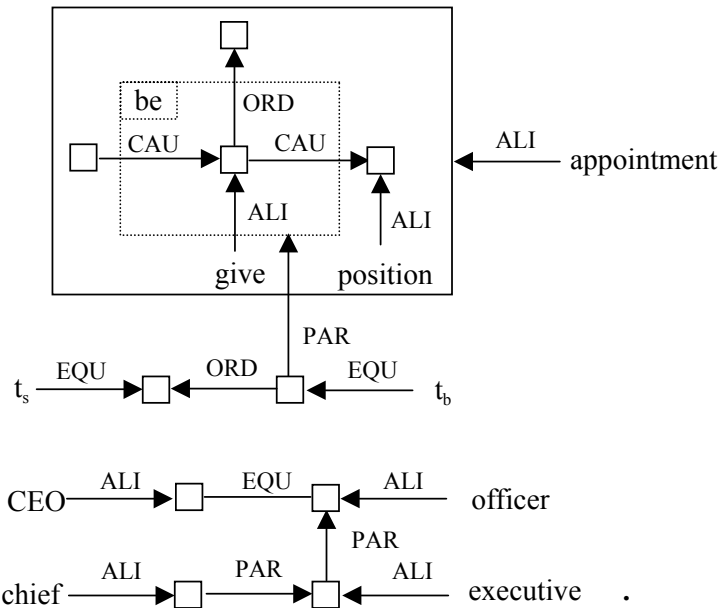
Chunk 31 : As this chunk has only one word, the chunk graph is just the word graph for “president”. We choose alternative (2).

Chunk 32 : We choose alternative (2) for “manufacturer”. Using the same methods as in Chapter 5, choosing alternative (1) for “of”, we obtained

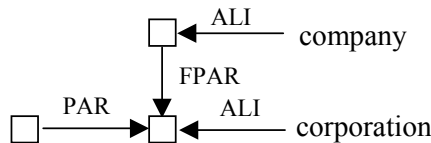


We cannot introduce “Hupplewhite” yet.

Chunk 41 :



Chunk 42 :

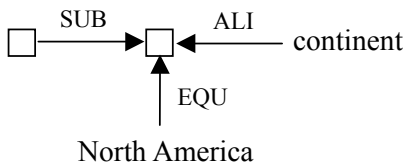


We cannot introduce “Lafarge” yet.

Chunk 51 : As this chunk has only one word the chunk graph is just the word graph for “one”.

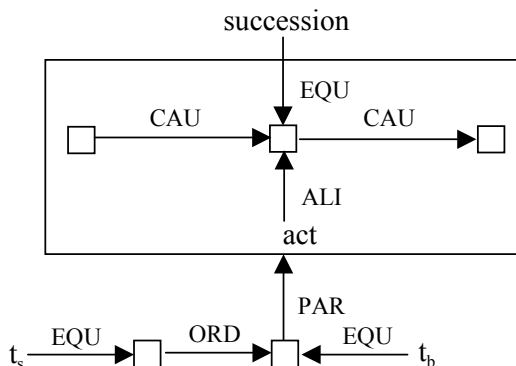
Chunk 52 : Without background knowledge, the word graphs cannot be combined.

Chunk 53 :



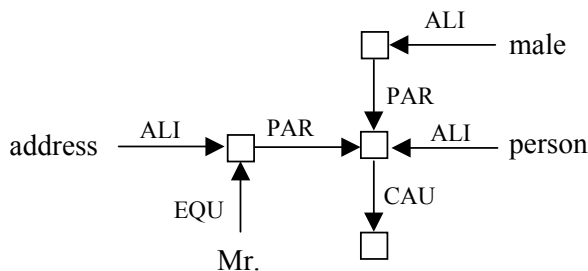
Chunk 611 : This chunk has only one word again.

Chunk 612 :

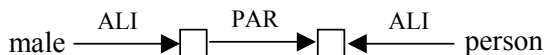


Note that the “act” is “be succeeded” and that this verb was already processed to fill the slot **EVENT**. Compare with the act “was appointed” in the first sentence.

Chunk 62 :



Note that “Mr.” and “John” can be combined if we assume that the fact that both word graphs contain the subgraph

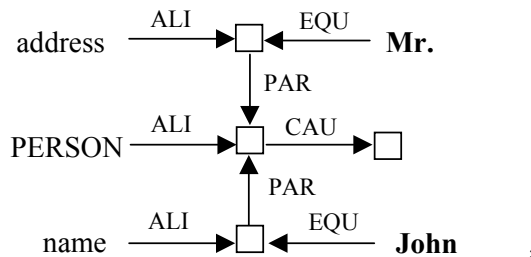


justifies this.

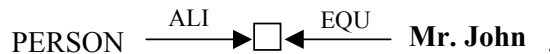
This is an example of similarity of two word graphs.

Chunk 7 : There is no possibility to combine the three word graphs.

Remarks: Due to the fact that various names did not have a word graph, the filling of slots is still not very well possible. Only Chunk 62 gives information when “Mr.” and “John” are combined. Then the chunk graph for “by Mr. John” is



where we now wrote person in capitals as this is one of the slots. From the chunk graph we now read off “Mr. John” as filler of the slot PERSON, we may replace the graph by

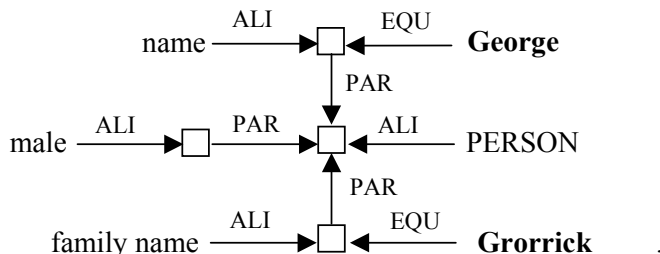


The third phase introduces reasoning by expansion of concepts. This holds both for the names of the slots and for the words occurring in the chunk graphs. As an example we consider the slot LOCATION and the word “continent” in Chunk 53. Any word graph for LOCATION may contain several instantiations or associations, without mentioning “continent”. Likewise the word graph for “continent” may not contain the concept “location”. However, this is rather unlikely. Describing a continent will involve mentioning its location.

To illustrate how important the expansion process is for obtaining our extraction goal, how much background knowledge is needed, we will now discuss the construction of chunk graphs in detail.

Chunk 1 : The word “Grorrick” was not encountered in the lexicon. Yet it has to be represented in relation with “George” as both words belong to the same chunk. What we need is *relevant* background information about “George”. It is a name in English, in fact it is a first name. Persons have both a first name and a family name. This is what makes it plausible that “Grorrick” is

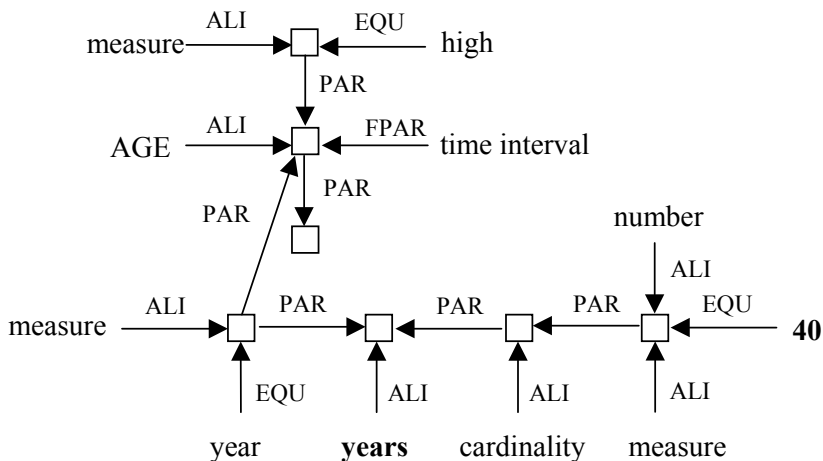
a family name. This information should be available to the computer. Note that it might be possible for the computer to expand the concept “name” to obtain this information. If not, the computer has no way to handle the word “Gorrick”. The chunk graph becomes:



and we have found the filler for the slot PERSON in the first sentence.

Chunk 2 : The relevant background information in this case is that “old” says something about a time interval. “40” stands before years (plural) and therefore relevant background information is that “years” is a set. If expansion shows that “40” can be the value of the cardinality of a set we can combine “40” and “years”. Expansion of “age” in the word graph of “old” may yield that it is a time interval.

Now we can combine into:



This rather complicated chunk graph contains AGE. The filler of AGE may be chosen from this graph by noting that the words “40” and “years” occur in the text. Other words are due to the construction of the word graphs (like “high”) or due to the expansion process (like “cardinality”).

Chunk 3 : The subchunks 31 and 32 each pose a special problem.

In chunk 31 only “president” is mentioned. The word graph contains the concept “officer”.

The slots OLD POSITION and NEW POSITION contain POSITION and a list of possible positions might **not** include “president” but may include “officer”. On the other hand expansion of “president”, by expansion of “officer”, may lead to the conclusion that “president” is a position.

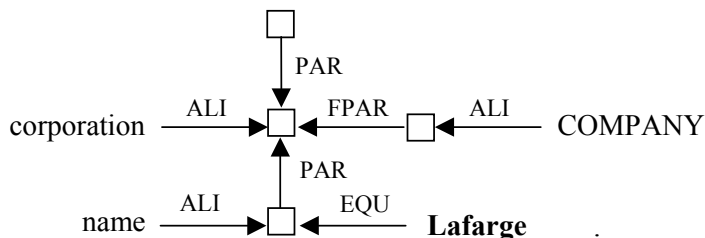
In both ways the link between POSITION and “president” can be established. What remains is the problem with OLD and NEW, as we already discussed just before we considered building chunk graphs. Solving that problem involves using the given text and not just expanding words of the chunk graph.

In chunk 32 the word “Hupplewhite” poses the problem. Being a word in the middle of a sentence beginning with the capital H suggests that “Hupplewhite” is a name. This also uses the given text. Therefore we should, in principle, not process this word in this third phase. However, we will discuss it here. The fact that the word follows the word “manufacturer” implies that it is the name of that “manufacturer”.

The chunk graph for 32, constructed so far ties the name up with COMPANY, and therefore we have found another potential filler. However, also COMPANY only occurs in the slot names OLD COMPANY and NEW COMPANY, so that we have the same problem as for OLD POSITION and NEW POSITION again.

Chunk 41 : The two subchunks can be combined due to the fact that expansion of “officer” gives that it is a “position”. From the combined graph we read off that CEO is a filler of POSITION.

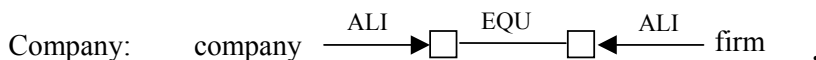
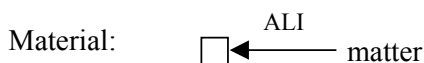
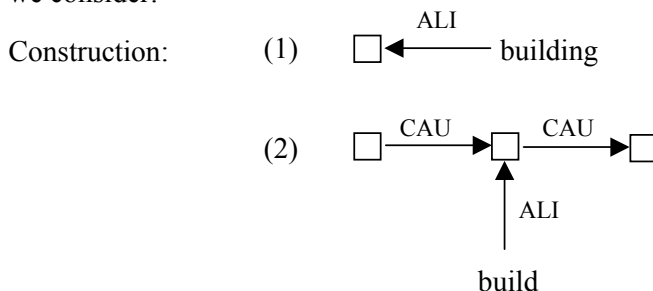
Chunk 42 : “Lafarge”, like “Hupplewhite”, must be a name and stands right before “corporation”, and as “corporation” is of type COMPANY we find another filler of OLD COMPANY or NEW COMPANY. The chunk graph looks like



The two chunk graphs could be combined by remarking that, in chunk graph 42, the PAR-link, that represents “of”, has a token that should occur in chunk graph 41. The word order suggests that this is “CEO”. For the extraction of knowledge, in the form of slot fillers, this combining is not absolutely necessary. Note that the subchunks 41 and 42 already gave the answer.

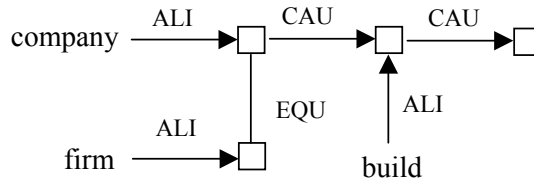
Chunk 5 : Chunk 51 must be interpreted as a pronoun, because “one” is used and not “1”, so we have to choose word graph (2).

Chunk 52 poses the main problem, coming from the phrase “leading”, as an adjective may be combined with “construction” as a noun. However it is to be combined with “companies”. How can a computer interpret the three consecutive nouns “construction”, “material” and “companies”? The basic idea is to use expansion of the, small, word graphs given. Suppose we consider:

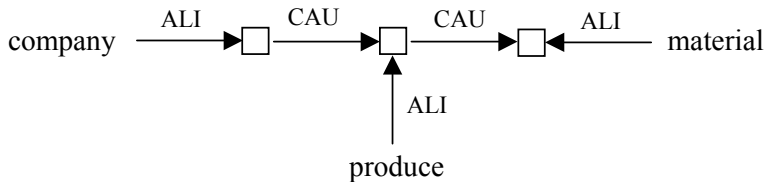


We have to find proper expansion. Let us start by saying that a “company”

does something, i. e., there is a CAU-arc going out from its token. This suggests that for construction we use the second word graph and then we can already construct

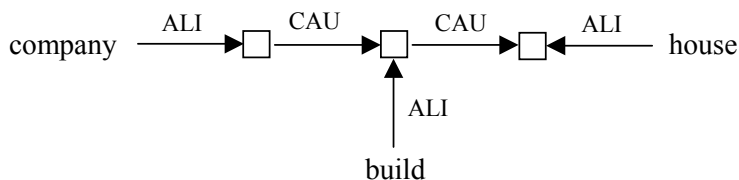


The word “material” or “matter”, because of its standing on the right of “construction”, must be expanded to link up with “building” as an instrument. However, it can also be linked with “companies” if we expand “companies” as entities producing something. This would lead to a graph like



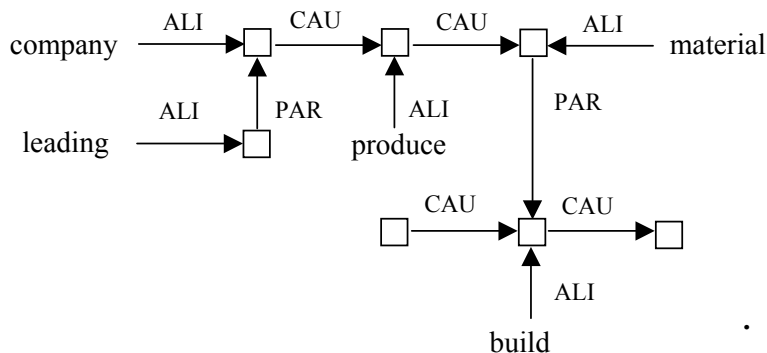
Note now that without the word “material” we would read “construction companies” and the first linking of graphs would be the only one. The sentence might have had the phrase “house construction companies”. That phrase indicates that the companies construct houses. The computer has to know how to deal with a sequence of nouns. We might instruct it in the sense that the last noun is the essential one. This would then mean that the adjective “leading” is to be attached to “companies”.

Then the relation with the forelast noun should be established. A “material company” asks for an interpretation of “material”. Is this a noun or an adjective? The lexicon only gave the noun interpretation. But then we can link by the expansion that companies produce, leading to the second graph. If “construction” is the forelast noun the first graph would result: the company constructs. The first noun in “house construction company” would be interpreted as the unspecified token, which would lead to the, correct, graph



The first noun in “construction material company” has to be linked to the graph constructed so far for “material company”, and, again, linking with the forelast word, “material” is searched for. We could use both word graphs for construction semantically. “Building material” can be both material of which a building consists (*after* the building) and material used for building (*during* the building). So, basically, there is very subtle ambiguity here. In practice, we would prefer to use the second interpretation, so material used, as instrument, during the building process.

As a result, we have



We have given this discussion, because of the interesting problem of constructing a chunk graph here. For the goal of information extraction it does not give any answer in the form of a slot filler.

Chunk 53 yields a slot filler as the expansion of “continent” may lead to the information that it is a LOCATION. Let us recall that for the **EVENT** “appointment” slot names were specified, of which LOCATION was one. Finding a filler for this slot is enough to conclude that we have found the required filler as chunk 53 belongs to the first sentence. More detailed expansion can lead to the information that it is the location of “companies” and, via “one”, the location of “Lafarge corporation”. But such a detailed analysis is not necessary. This is an example of the usefulness of *partial* structural parsing.

Chunk 6 : We now have to investigate the second template and find fillers for the slots.

The sentence is rather short indeed. We already used the word “succeeded” to fill the slot **EVENT** with “succession”. Next to that “Mr. John” was localized as a **PERSON**. So from the sentence part “He will be succeeded by Mr. John” we can only process chunk 611, which is the pronoun “He”. But this pronoun refers to a person, George Gorrnick, mentioned in the first sentence. The implications of this cannot be found by expansion within Chunk 6.

Chunk 7 : Chunk 7 only contains data referring to **TIME**, but this was not chosen to be a slot name. So we can refrain from processing this chunk.

The second sentence so far has only led to fillers for the slots **EVENT** and **POSITION**.

We have not been able to fill all the slots of the two templates corresponding to the two sentences. We definitely need some extra reasoning.

In the fourth phase we do not expand word graphs with lexical information, but, as remarked before, now the context information is used to decide upon fillers. We will not do this in detail, but will only mention what can be decided in this phase for this example.

For the first template, “appointment”, in principle enough information was found to fill the slots **OLD POSITION**, **NEW POSITION**, **OLD COMPANY** and **NEW COMPANY**. However, it was still to be decided which name should fill which slot. We have given a reasoning at the end of Section 7.4.7. to find

OLD POSITION	president
NEW POSITION	CEO
OLD COMPANY	Hupplewhite
NEW COMPANY	Lafarge

This completes the first template. For the second template **OLD POSITION** and **OLD COMPANY** cannot be determined. The pronoun “He” plays the vital role in determining the fillers for the slots **NEW POSITION** and **NEW COMPANY** of “Mr. John”. They are found by the fact that the word “succeed” is interpreted as “get the

position of”, where the free token in the pronoun “He” is identified with the only person mentioned in the first sentence, who is George Gorrnick. Therefore we get NEW POSITION: president and NEW COMPANY: Hupplewhite. The slot LOCATION is still to be filled. Due to the pronoun “He”, referring to George Gorrnick, we can only conclude that the succession took place at the manufacturer Hupplewhite. However, this company might be a company in South America. For the “appointment” a location is mentioned, but the LOCATION slot of the “succession” has to remain open.

Concluding, we see that there are four phases, that each can provide fillers for the chosen slots.

- The first phase, just the construction of word graphs, hardly gave any filler.
- The second phase, the construction of chunk graphs, gave some possibility to attach names to slots.

However, the important phases are:

- The third phase, in which expansion of word graphs gave the opportunity to link potential fillers to slots.
- The fourth phase, in which context information was used, in principle formed by both sentences, turned out to be of vital importance to decide on the proper choice of fillers.

All four phases should have their place in any automatic information extraction procedure, on the basis of KGExtract.

7.4.9 Discussion

Let us consider the 10 functionalities mentioned by Hobbs.

1. A Text Zoner. Clearly our parsing by chunks turns a text into a set of segments in much more detail.
2. A Preprocessor. Also this is covered by parsing by chunks.
3. A Filter. We consider the whole text without filtering. Filters, based on the types

of slots could easily be added.

4. A Preparser. The chunks are the small-scale structures that people are looking for.
5. A Parser. One of the new features of our approach is that the traditional parse trees get a much less important role to play.
6. A Fragment Combiner. The formulation of Hobbs stresses the traditional representation forms of parse-tree or logical-form fragment. Both are replaced by knowledge graphs.
7. A Semantic Interpreter. The traditional approach is to start with syntactic aspects. As we have discussed in the previous chapter, the essence of the knowledge graph approach is the semantic aspects.
8. A Lexical Disambiguator. In our analysis of the example disambiguation took place by taking into account the other parts of the sentence. Consider the discussion about Chunk 32, Hupplewhite could be the name of a PERSON, a COMPANY or a LOCATION. As seen before there is the word COMPANY, the interpretation as name of a COMPANY is most likely. Disambiguation is context dependent.
9. A Coreference Resolver. This too is a typical AI-problem, that was solved by taking into account the context. See the discussion about Chunk 611.
10. A Template Generator. We get the filled in templates as knowledge graph structures.

The hardest problems seem to be those encountered in 8. Disambiguation and 9. Coreference Resolving. Although our main goal in this chapter is to show the usefulness of the idea of partial structural parsing in the field of Information Extraction, the problems we hit upon deserve some further discussion.

We already saw in Chapter 5 that background knowledge is decisive for obtaining a sentence graph with structural parsing. Let us end with the thesis that intelligence, and therefore also artificial intelligence, heavily depends on the use of background knowledge.

A word graph is considered to be without limits essentially. A concept and its nearest

neighbors form a subgraph of the mind graph that can be called foreground knowledge. The subgraph of the mind graph arising after deletion of the concept token can be called background knowledge of that concept, see Chapter 3. In Section 4.3.1 we pointed out that expansion of concepts plays an important role in thinking. Given a concept the number of associations with that concept will in first instance be limited. A person does not have his whole mind graph at his disposal immediately. However, by considering the concepts in the associations, i.e. in the word graph of the concept, and replacing these concepts by their meaning, i.e. their word graphs, the word graph of the original concept can be “expanded” and a larger word graph is obtained. In principle this can go on indefinitely until the whole mind graph is obtained, i.e. a word graph corresponding to all knowledge available to that mind.

For a computer approach, that is simulating this process, we have at our disposal the word graph lexicon. The smaller this lexicon, the fewer the associations the computer has and the less expansion can take place. Like for human beings, the computer’s abilities to think, i.e. link somethings, are highly dependent on its information. The more information is contained in the lexicon of word graphs, i.e. the larger these are, the higher the probability that by expansion relevant linking of concepts takes place. There is, however, a second source of information, namely the context in which the concept is considered.

If, like in our example, two sentences are given, for extracting information from the second sentence the computer has the information contained in the first sentence at its disposal too. Next to its internal information, contained in the lexicon, there is the external information contained in the *context*. In a way the context also expands the knowledge of the computer. This becomes even clearer when we consider a dialogue. The description of a dialogue by means of knowledge graphs can be as follows. Speaker A says something and a sentence graph is made for this. The answer of speaker B is likewise transformed into a sentence graph, that is joined with the first graph. Every time new information is exchanged the graph representing what has been said sofar, in each of the minds of the speakers A and B, is expanded. This expansion is also due to context, now coming from the dialogue partner and not from the foregoing text.

So there are two forms of expansion available to the computer. One is due to

combination of word graphs from its lexicon, the other is due to context processing. The development of an automated information extraction procedure, based on this idea of expansion, is challenging.

Bibliography

[Aberdeen *et al.*, 1995] J. Aberdeen *et al.*, MITRE: *Description of the Alembic system used for MUC-6*. In: Proc. Sixth Message Understanding Conf., Columbia, MD, Morgan Kaufmann, 1995.

[Abney, 1991] S. P. Abney, *Parsing by chunks*. In: Principle-Based Parsing (Eds. R. Berwick, S. Abney and C. Tenny), Kluwer Academic Publishers, 1991.

[Allen, 1984] R. E. Allen, *The Pocket OXFORD Dictionary*, Seventh Edition, Oxford University Press, New York, 1984.

[Allen, 1987] J. Allen, *Natural Language Understanding*. The Benjamin/Cummings Publishing Company, Inc., California, 1987.

[Appelt *et al.*, 1995] D. Appelt *et al.*, *SRI International FASTUS system: MUC-6 test results and analysis*. In: Proc. Sixth Message Understanding Conf., Columbia, MD, Morgan Kaufmann, 1995.

[Bakker, 1987] R. R. Bakker, *Knowledge Graphs: representation and structuring of scientific knowledge*. Ph.D. thesis, University of Twente, Enschede, The Netherlands, ISBN 90-9001963-4, 1987.

[Bates, 1978] M. Bates, *The theory and practice of augmented transition network grammars*. In: Natural Language Communication with Computers (Ed. L. Bolc), Springer Verlag, Berlin, 191-259, 1978.

[Berg, 1993] H. van den Berg, *Knowledge Graphs and Logic: One of Two Kinds*. Ph.D. thesis, University of Twente, Enschede, The Netherlands, ISBN 90-9006360-9,

1993.

[Blok, 1997] H. E. Blok, *Knowledge Graph Framing and Exploitation*. Master's thesis, University of Twente, Faculty of Mathematical Sciences, Enschede, The Netherlands, 1997.

[Ce, 1988] C. Ce, *A Modern Chinese-English Dictionary*, Foreign Language Teaching and Research Press, 1988.

[Charniak, 1996] E. Charniak, *Tree-bank grammars*. In: Proc. of the Thirteenth National Conf. on Artificial Intelligence, Portlang, OR, 1031-1036, 1996.

[Cheng, 1995] L. L. Cheng, On Dou-Quantification, *Journal of East Asian Linguistics*, 4: 197-234, 1995.

[Chierchia & McConnell-Ginet, 1990] G. Chierchia and S. McConnell-Ginet, *Meaning and Grammar: An Introduction to Semantics*. MIT Press, Cambridge, 1990.

[Church *et al.*, 1996] K. Church, S. Young and G. Bloothcroft, *Corpus-based Methods in Language and Speech*. Dordrecht, Kluwer Academic, 1996.

[Collins, 1997] M. J. Collins, *Three generative, lexicalised models for statistical parsing*. In: Proc. of the 35th Annual Meeting of the Association for Computational Linguistics, 66-23, 1997.

[Cowie *et al.*, 1993] J. Cowie *et al.*, *The Diderot information extraction system*. In: Proc. of the first Conf. of the Pacific Association for Computational Linguistics, Vancouver, 1993.

[Cowie & Lehnert, 1996] J. Cowie and W. Lehnert, *Information Extraction*. In: Special NLP Issue of the Comm (Ed. Y. Wilks), ACM, 1996.

[Cowie & Wilks, 2000] J. Cowie and Y. Wilks, *Information extraction*. In: Handbook of Natural Language Processing (Eds. R. Dale, H. Moisl and H. Somers), New York: Marcel Dekker, 2000.

[DeJong, 1979] G. F. deJong, Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3: 251-273, 1979.

[de Vries, 1989] P. H. de Vries, *Representation of Science Texts in Knowledge Graphs*, Ph.D. thesis, University of Groningen, Groningen, The Netherlands, 1989.

[de Vries Robbé, 1987] P. F. de Vries Robbé, *Medische Besluitvorming: Een Aanzet to Formele Geneeskunde*, Ph.D. thesis, Department of Medicine, University of Groningen, Groningen, The Netherlands, 1987.

- [Fisher *et al.*, 1995] D. Fisher, S. Soderland, J. McCarthy, F. Feng and W. Lehnert, *Description of the UMass Systems as Used for MUC-6*. In: Proc. of the 6th Message Understanding Conf., Columbia, MD, 1996.
- [Gaizauskas *et al.*, 1995] T. Gaizauskas *et al.*, *Description of the LaSIE system as used for MUC-6*. In: Proc. Sixth Message Understanding Conf., Columbia, MD, Morgan Kaufmann, 1995.
- [Gardenfors, 1987] P. Gardenfors, *Generalized Quantifiers: Linguistic and Logical Approaches*. Volume 31 of Studies in Linguistic and Philosophy, D. Reidel Publishing Company, Dordrecht, 1987.
- [Graham, 1979] N. Graham, *Artificial Intelligence making machines "think"*. TAB BOOKS, U.S.A., 1979.
- [Grimes, 1975] J. Grimes, *Network Grammars*. Summer Institute of Linguistics Publications in Linguistic and Related Fields, no. 45, 1975.
- [Grishman & Sundheim, 1996] R. Grishman and B. Sundheim, *Message Understanding Conference – 6: A brief history*. In: Proc. 16th International Conf. On Computational Linguistics, Copenhagen, 1996.
- [Harary, 1972] F. Harary, *Graph Theory*. Addison-Wesley, Reading, 1972.
- [Harris, 1968] Z. S. Harris, *Mathematical Structures of Language*. Wiley-Interscience, New York, 1968.
- [Harris, 1982] Z. S. Harris, *A Grammar of English on Mathematical Principles*. John Wiley & Sons, New York, 1982.
- [Harris, 1985] M. D. Harris. *Introduction to Natural Language Processing*. A Prentice-Hall Company, Reston, Virginia, 1985.
- [Hobbs, 1993] J. R. Hobbs, *The generic information extraction system*. In: Proc. of the Fifth Message Understanding Conf., Morgan Kaufmann, 87-91, 1993.
- [Hoede & Willems, 1989] C. Hoede and M. Willems, *Knowledge Graphs and Natural Language*. Memorandum no.811, University of Twente, Enschede, The Netherlands, 1989.
- [Hoede & Li, 1996] C. Hoede and X. Li, *Word Graphs: The First Set*. In: Conceptual Structures: Knowledge Representation as Interlingua, Aux. Proc. of the Fourth International Conf. on Conceptual Structures (Eds. P. W. Eklund, G. Ellis and G. Mann), Bondi Beach, Sydney, Australia, 81-93, 1996.

- [Hoede & Liu, 1998] C. Hoede and X. Liu, *Word Graphs: The Second Set*. In: *Conceptual Structures: Theory, Tools and Applications*, Proc. of the 6th International Conf. on Conceptual Structures (Eds. M-L. Mugnier and M. Chein), Montpellier, Springer Lecture Notes in Artificial Intelligence no.1453, 375-389, 1998.
- [Hoede *et al.*, 2000] C. Hoede, X. Li, X. Liu and L. Zhang, *Knowledge Graph Analysis of Some Particular Problems in The Semantics of Chinese*, Memorandum no. 1516, Faculty of Mathematical Sciences, University of Twente, Enschede, The Netherlands, ISSN 0169-2690, 2000.
- [Hoede & Zhang, 2001a] C. Hoede and L. Zhang, *Word Graphs: The Third Set*. In: *Conceptual Structures: Broadening the Base*, Proc. of the 9th International Conf. on Conceptual Structures (Eds. H. S. Delugach and G. Stumme), CA, USA, Lecture Notes in Artificial Intelligence no.2120, 15-28, 2001.
- [Hoede & Zhang, 2001b] C. Hoede and L. Zhang, *Structural Parsing*. In: *Conceptual Structures: Extracting and Representing Semantics*, Aux. Proc. of the 9th International Conf. on Conceptual Structures (Ed. G. W. Mineau), CA, USA, 75-88, 2001.
- [Holland & Johansson, 1982] K. Holland and S. Johansson, *Word Frequencies in British and American English*. In: *Publication of the Norwegian Computing Center for the Humanities*, Bergen, Norway, 43-53, 1982.
- [Humphreys, 2000] Humphreys, *Two Applications of information extraction to biological science journal articles: enzyme interactions and protein structures*. In: *Proc. of the Pacific Symposium on Biocomputing (PSB-2000)*, Hawaii, 505-516, 2000.
- [Kaplan, 1975] R. M. Kaplan, *On process models for sentence analysis*. In: *Explorations in Cognition* (Eds. D. A. Norman and D. E. Rummelhart), Freeman, San Francisco, 117-135, 1975.
- [Kuno, 1965] S. Kuno, *The predictive analyser and a path elimination technique*. *Communications of the ACM*, 8: 687-698, 1965.
- [Krupka, 1995] G. Krupka, *SRA: description of the SRA system as used for MUC-6*. In: *Proc. Sixth Message Understanding Conf.*, Columbia, MD, Morgan Kaufmann, 1995.
- [Lehnert *et al.*, 1993] W. Lehnert *et al.*, *UMass/Hughes: Description of the CIRCUS system used for MUC-5*. In: *Proc. fifth Message Understanding Conf.*, Morgan Kaufmann, 1993.
- [Litman, 1996] D. J. Litman, *Cue phrase classification using machine learning*.

Journal of Artificial Intelligence Research, 5: 53-95, 1996.

[Liu, 2002] X. D. Liu, *The Chemistry of Chinese Language*, Ph.D. thesis, University of Twente, Enschede, The Netherlands, ISBN 90-3651834-2, 2002.

[Manning & Carpenter, 1997] C. D. Manning and B. Carpenter, *Three generative, lexicalised models for statistical parsing*. In: Proc. of the Fifth International Workshop on Parsing Technologies, 147-158, 1997.

[Mellish *et al.*, 1992] C. Mellish *et al.*, *The TIC message analyser*. Technical Report CSPR 225, University of Sussex, Sussex, England, 1992.

[Mikheev *et al.*, 1998] A. Mikheev, C. Grover and M. Moens, *Description of the LTG System Used for MUC-7*. Proc. of the seventh Message Understanding Conf., Washington, D.C., 1998.

[MUC-3, 1991] *Proc. of the Third Message Understanding Conf.*, Columbia, MD, Morgan Kaufmann, May 1991.

[MUC-4, 1992] *Proc. of the Fourth Message Understanding Conf.*, Columbia, MD, Morgan Kaufmann, 1992.

[MUC-5, 1993] *Proc. of the Fifth Message Understanding Conf.*, Morgan Kaufmann, August 1993.

[MUC-6, 1995] *Proc. of the Sixth Message Understanding Conf.*, Columbia, MD, Morgan Kaufmann, 1995.

[MUC-7, 1998] *Proc. of the Seventh Message Understanding Conf.*, Washington, D.C., Morgan Kaufmann, 1998.

[Peirce, 1885] C. S. Peirce, On the Algebra of Logic. *American Journal of Mathematics*, 7: 180-202, 1885.

[Pereira & Shabes, 1992] F. Pereira and Y. Shabes, *Inside-outside reestimation from partially bracketed Corpora*. In: Proc. of the 30th Annual Meeting of the Association for Computational Linguistics, Newark, Delaware, 128-135, 1992.

[Petrick, 1965] S. R. Petrick, *A Recognition Procedure for Transformational Grammars*. Ph.D. Thesis, MIT Cambridge, Mass., 1965.

[Petrick, 1966] S. R. Petrick, *A Program for Transformational Syntactic Analysis*. Airforce Cambridge Research Laboratories Report AFCRL-66-698, Cambridge, Mass., 1966.

[Rabiner, 1989] L. R. Rabiner, *A tutorial on hidden Markov models and selected*

- applications in speech recognition*. In: Proc. of the IEEE, 77/2: 257-286, 1989.
- [Radford, 1988] A. Radford, *Transformational Grammar: A First Course*. Cambridge University Press, Cambridge, 1988.
- [Ratnaparkhi, 1999] A. Ratnaparkhi, *Learning to parse natural language acquisition*. In: Advances in Computers (Eds. M. Yovits and M. Rubinoff), Academic Press, New York, 15:181-237, 1999.
- [Reidsma, 2001] D. Reidsma, *Juggling word graphs: A method for modeling the meaning of sentence using extended knowledge graphs*. Master's thesis, University of Twente, Faculty of Computer Science, Enschede, The Netherlands, 2001.
- [Rich, 1983] E. Rich, *Artificial Intelligence*. McGraw-Hill Inc., U.S.A., 1983.
- [Riesbeck, 1974] C. K. Riesbeck, *Computational Understanding: Analysis of Sentences and Context*. Ph.D. Thesis, Department of Computer Science, Stanford University, 1974.
- [Riesbeck, 1975a] C. K. Riesbeck, *Computational Understanding*. In: Proc. of the Workshop on Theoretical Issues in Natural Language Processing (Eds. R. C. Schank and B. L. Nash-Webber), Bolt, Beranek and Newman, Cambridge, Mass., 15-20, 1975a.
- [Riesbeck, 1975b] C. K. Riesbeck, *Computational Analysis*. In: Conceptual Information Processing (Ed. R. C. Schank), North-Holland, Amsterdam, 83-156, 1975b.
- [Sager *et al.*, 1987] N. Sager, C. Friedman and M. Lyman, *Medical language processing: computer management of narrative data*. Addison Wesley, 1987.
- [Schank, 1972] R. C. Schank, Conceptual Dependency: A Theory of Natural Language Understanding. *Cognitive Psychology*, 3/4: 552-630, 1972.
- [Schank, 1975] R. C. Schank, *Conceptual Information Processing*. North-Holland, Amsterdam and American Elsevier, New York, 1975.
- [Schank & Abelson, 1977] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdales, NJ, 1977.
- [Smit, 1991] H. J. Smit, *Consistency and Robustness of Knowledge Graphs*. Ph.D. thesis, University of Twente, Enschede, The Netherlands, 1991.
- [Sowa, 1994] J. F. Sowa, *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Draft of a book scheduled to be published by the PWS

Publishing Company, Boston Massachusetts, 99-11, 1994.

[Stevens & Rumelhart, 1975] A. L. Stevens and D. E. Rumelhart, *Errors in reading: analysis using an augmented transition network model of grammar*. In: Explorations in Cognition (Eds. D. A. Norman and D. E. Rumelhart), Freeman, San Francisco, 136-155, 1975.

[Thorne *et al.*, 1968] J. P. Thorne, P. Brately and H. Dewar, *The syntactic analysis of English by machine*. In: Machine Intelligence 3 (Ed. D. Michie), Edinburgh University Press, Edinburgh, 281-309, 1968.

[Waltz, 1978] D. L. Waltz, An English language question answering system for a large relational data base. *Communications of the ACM*, 21/7: 526-539, 1978.

[Willems, 1993] M. Willems, *Chemistry of Language*. Ph.D. thesis, University of Twente, Enschede, The Netherlands, ISBN 90-9005672-6, 1993.

[Winograd, 1972] T. Winograd, *Understanding Natural Language*. Edinburgh: Edinburgh University Press, 1972.

[Woods, 1970] W. A. Woods, Transition network grammars for natural language analysis. *Communications of the ACM*, 13: 591-606, 1970.

[Woods *et al.*, 1972] W. A. Woods, R. M. Kaplan and B. Nash-Webber, *The Lunar Sciences Natural Language Information System*. BBN Report no.2378, Bolt, Beranek and Newman, Cambridge, Mass., Also available as publication N72-28984 of the US National Technical Information Service, 1972.

[Woods, 1977] W. A. Woods, *Lunar rocks in natural English: explorations in natural language question-answering*. In: Linguistic Structures Processing (Ed. A. Zampolli), North-Holland, Amsterdam, 521-568, 1977.

[Yangarber & Grishman, 1998] R. Yangarber and R. Grishman, *NYU: Description of the Proteus/PET System as used for MUC-7 ST*. In: Proc. of the Seventh Message Understanding Conf., Washington, D.C., 1998.

[Yao, 1995] T. S. Yao, *Natural Language Processing*. Tsinghua University Press, Beijing, ISBN 7-302-01911-8, 1995.

[Zhu, 1984] D. X. Zhu, *Grammar Tutorial*. Commerce Printing House, Beijing, 1984.

Appendix I

Word Graphs: The First Set

In this appendix a summary is given of the paper of Hoede and Li [Hoede & Li, 1996], in which word graphs are discussed for a set of prepositions, nouns and verbs.

I.1 Introduction

In building a lexicon of word graphs we should start with the simple subgraphs of the mindgraph. For this reason we start the study of prepositions. Prepositions form the glue for the more complex words. Before dealing with them systematically we have to discuss a few other things.

The meaning of a word is its word graph. The word graph to be included in a lexicon depends on the maker of the lexicon, like is the case with normal lexical. There is no boundary for the word graph. The more associations a mind considers should be made in order to have the proper picture of a concept, the larger the word graph will be.

The word graph that is actually taken up into a lexicon depends entirely on what the maker of the lexicon wants to express. Although one of the major goals of knowledge graph theory is to deal with linguistic problems, in particular with pragmatics, i.e. the

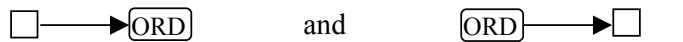
handling of background information, in this appendix we will focus on minimal word graphs. By this we mean word graphs that express the essence of a word, its most basic meaning. Whenever words are used in other meanings there should be other word graphs, describing those meanings, in which the basic meaning is recognizable as a subgraph.

I.2 Word Graphs for Some Simple Prepositions

We consider one simple arc of type ORD. For our purpose we consider the *total graph* representation of a graph in which arcs are also represented by vertices, and type-less arcs represent the structure of the graph. Figure I.1 gives the ORD-link and its total graph representation.



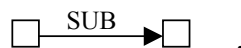
Figure I.1(a) has subgraphs like



Simple as these graphs are, they express what we would describe by the words FROM and TO. Herewith we have found the first two of our lexicon of word graphs of representations in a very basic form.

Prepositions turn out to be directly linked with the elements of the ontology of knowledge graphs. Like the arcs of the mind graph they glue (some)things together and may be considered the elementary particles of language.

As another example, that gives us the opportunity to introduce the Chinese language, we consider the word IN. Its word graph is taken to be



From a Chinese dictionary [Ge, 1978] we read sixteen different meanings for IN, given by sixteen different character combinations. These character combinations express different connotations of the word IN and should therefore be representable by different word graphs, according to our theory. The exercise was carried through to

find the answer to the question “what is the meaning of this word?” by examining direct translation of the characters. Note that word graphs depend on the maker(s) of the lexicon. If a reader does not agree with the proposed word graph, he/she gives another meaning to the word. One may expect, however, that especially the very simple words will show high similarity between proposed word graphs.

An important preliminary remark to be made is about merology. This is a notoriously difficult and much debated field. For expressing “part of”-relationships quite a few proposals have been made. In our ontology we have proposed SUB, PAR and FPAR as types of relations that are of merological nature. They express, essentially, set inclusion, attribution by the mind and relationship due to framing by the mind. The corresponding closest verbal descriptions of these relationships are by “part of”, “attribute of” and “property of”. The fact that in all three cases the word OF is used shows the source of the discussion in merology. OF is a homonym. The three types are extremely close for the mind that has to bring the relationships “under words”. The word graphs of OF and WITH are given in Figure I.2, where we recognize this triple of types.



Figure I.2. Word graphs for the prepositions OF and WITH.

Figure I.3 shows one of the 16 Chinese words for IN. The word graph gives the annotation that we are considering areas.

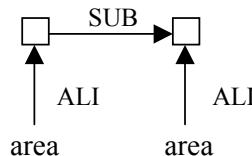


Figure I.3. Word graph for ZAI4...SHANG4.

Here we see a dominant arc of type SUB. In Section 4.3.3 we discussed logic words of the second kind. This is a standard example. One of the basic relation types stands

central in the word graph.

In the analysis of the Chinese words for IN the SUB-relationship has in several cases not been chosen. Instead the PAR-relationship seemed to describe the meaning better. For the list of sixteen word graphs, we refer to the original paper [Hoede & Li, 1995], we do not repeat it here.

I.3 Word Graphs for Other Prepositions

In this section the chosen prepositions are mentioned in the grammar of Quirk *et al.* [Quirk *et al.*, 1972]. Here we classify the prepositions into six parts, because we want to comment on the various groups. We only list the words that we chose in each group, for the word graphs of these words, in detail we refer to the original paper.

The first group consists of the simplest graphs that can be made out of our ontology elements, like we discussed in Section I.2. The list of words in this group is:

FROM	TO	OF	IN	WITH	LIKE	BY	BECAUSE	FOR
------	----	----	----	------	------	----	---------	-----

By gradually adding more structure other prepositions arise. Ordering in space as seen from the point of view of a speaker can be longitudinal or transversal and in the second case horizontal or vertical. The following group is used for describing spatial relations mainly.

BEFORE	AFTER	OVER	UNDER	BY	AT
ON	ABOVE	BELOW	BESIDE	IN FRONT OF	BEHIND

Take IN FRONT OF as an example, we give the word graph of it in Figure I.4.

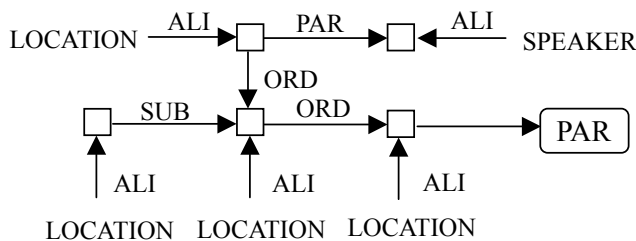


Figure I.4. Word graph for IN FRONT OF

In the example IN FRONT OF (Chinese: ZAI4...QIAN2 MIAN4, literally AT...FRONT FACE.) we note the tendency in English to break up the graph into smaller parts. IN and OF have already been described by word graphs, which can be recognized in Figure I.4. The location of the speaker has been given as reference point. The prepositions in the next group are often used in more abstract discussions as well. Hence the lack of reference to location etc.

BETWEEN	PAST	AMIDST	AMONGST
---------	------	--------	---------

The remaining prepositions roughly can be divided into three groups. A rather large group consists of more complex prepositions in the sense that these can also be expressed in terms of prepositions belonging to the above that we have described. The two other groups involve negation and universal quantification respectively.

- BEYOND ≡ AT LOCATION PAST
- SINCE ≡ AT TIMES AFTER TIME VALUE
- UNTIL ≡ AT TIMES BEFORE TIME VALUE
- UPTO ≡ AT VALUES ORDERED TO VALUE
- ACROSS ≡ AT LOCATION PAST
- THROUGH ≡ MOVING AT LOCATION
- UP ≡ THROUGH LOCATIONS (following vertical ordering)
- DOWN ≡ THROUGH LOCATIONS (against vertical ordering)
- ALONG ≡ THROUGH LOCATIONS (following path).

The word graphs of these prepositions are already that large that they can be brought under words by covering the graphs with smaller word graphs.

- OFF ≡ NOT ON
- OUT OF ≡ NOT IN
- WITHOUT ≡ NOT WITH
- AGAINST ≡ NOT FOR.

This small list shows the use of negation, that can be expressed by the NEG-frame.

There are quite a few prepositions expressing NOT INCLUDING: EXCEPT FOR, WITH THE EXCEPTION OF, APART FROM, EXCEPTING, EXCEPT, BUT, BUT FOR (Chinese: CHU2 WAI4, literally REMOVE OUTSIDE).

Some of these are synonyms. These prepositions have word graphs with strong similarity to the word graph of WITHOUT.

- ALL OVER
- ALL ALONG
- ALL AROUND
- THROUGH OUT ≡ AT ALL LOCATIONS.

These four prepositions show universal quantification for which we can use the SKO-loop. The word graph for AROUND is essentially AMIDST with focus on locations.

With the group IN SPITE OF, DESPITE, FOR ALL, WITH ALL (Chinese: JIN3 GUAN3, literally OVER MANAGE) and the group WITH REGARD TO, WITH REFERENCE TO, AS TO, AS FOR (Chinese: GUAN1 YU2..., literally CLOSE AT), with $\boxed{\text{PAR}} \longrightarrow \square \xrightarrow{\text{EQU}} \square$ as word graph, we conclude our discussion of the word graphs for prepositions.

I.4 Word Graphs for Verbs and Nouns

Verbs like BE or EXIST occur frequently in natural language text. The word like BE is so basic that it deserves to be described by the most basic frame, which is the empty frame of Figure I.5.

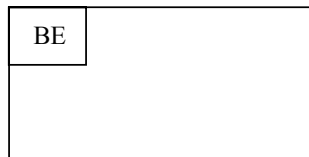


Figure I.5. Word graph for BE.

Note that we have chosen not to use the FPAR-relation but its alternative, which is naming a rectangle with the associated word in the corner.

The merological relation types FPAR, SUB and PAR are at the basis of the other very basic word HAVE, which is described by the word graphs in Figure I.5.

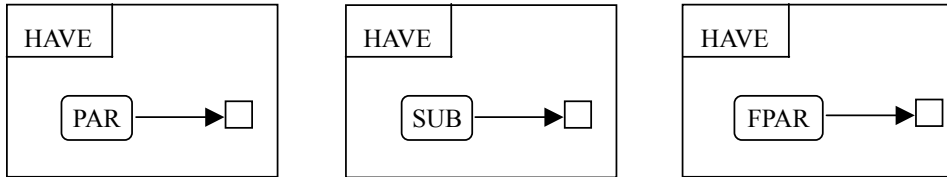


Figure I.6. Word graphs for HAVE.

These graphs can be read as “be something with”.

Verbs like CAN and MUST can be expressed by POS and NEC frames, i.e. by rectangles with names CAN or MUST, or by means of an attribute POSSIBILITY with value POSSIBLE or an attribute NECESSITY with value NECESSARY.

Normal verbs are represented by including two causal links, in case of transitive verbs, or one causal link, in case of intransitive verbs, see Figure I.6.

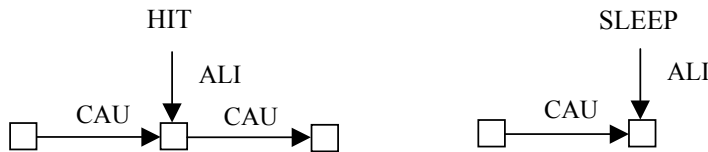


Figure I.7. Word graphs for HIT and SLEEP.

There are, at least, three different forms, in which a noun may be represented.

The simplest form is the one, in which only the directed ALI-link is used. Figure I.8 is to be read as “something like a dog”.



Figure I.8. Word graph for DOG.

We can expand the concept of DOG to a more complex word graph, by replacing the token with a frame. In general, a word may be related to a large frame, in which there is a more complex word graph to explain the meaning of the word, e.g. by expanding words occurring in the definition, just like we did for the noun “APPOINTMENT” in Section 7.4.8.

If a word is an instantiation of another word, like “dog Pluto”, we give the word graph as in Figure I.9. The directed EQU-link is used between a word and a token to evaluate or instantiate the token. Figure I.9 is to be read as “something like a dog equal to Pluto”.

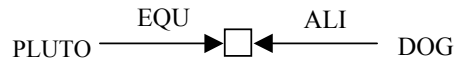


Figure I.9. Word graph for DOG PLUTO.

If a word is a subtype of another word, like “hotdog”, that we mentioned in Section 7.4.8, we give the word graph as in Figure I.10. The directed FPAR-link is used between two tokens. Figure I.10 is to be read as “hotdog is a kind of sausage”.



Figure I.10. Word graph for HOTDOG ISA SAUSAGE.

Here we consider word graphs for nouns in the context of the *type hierarchy*, that is very important in a practical NLP systems, such as the IE system.

If a word is the synonym of another word, like “company and firm” that we mentioned in Section 7.4.8, we give the word graph for it as Figure I.11. The EQU-link is used between two tokens. Figure I.11 is to be read as “company is synonym with firm”.

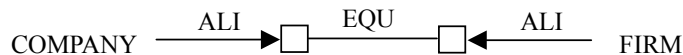


Figure I.11. Word graph for COMPANY SYNONYM WITH FIRM.

For practical applications in terms of knowledge graph theory, we will of course need larger word graphs as most of these problems have to be explained by background knowledge and background knowledge is only obtained in larger versions of the word graphs. In their simplest form all nouns and verbs are thus, in principle, described by word graphs.

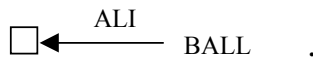
Appendix II

Word Graphs: The Second Set

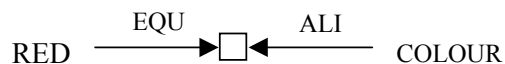
In continuation of Appendix I on word graphs for a set of prepositions, in this appendix, a summary is given of the paper of Hoede and Liu [Hoede & Liu, 1998], who studied word graphs for adjectives, adverbs and Chinese classifier words. It is argued that these three classes of words belong to a general class of words that may be called *adwords*. These words express the fact that certain graphs may be brought into connection with graphs that describe the important classes of nouns and verbs.

II.1 Introduction

Suppose we have the word combination RED BALL. The word BALL can be represented, according to our formalism, by

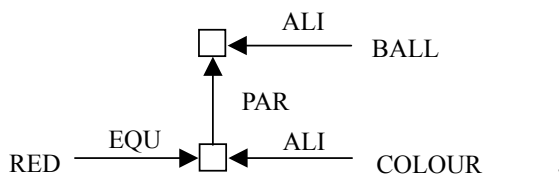


Now RED is a word attributed to the word BALL in the sense that its word graph is linked to the word graph of BALL. For RED we would take the following word graph into the lexicon:



analogous to the graph for DOG PLUTO. Now we say “red is the colour of the ball”

and the word graphs are to be linked in a way that we use the word OF for. As we see colour as an exterior attribute of ball we choose the PAR-relationship to represent RED BALL by



It is in this way that RED is a word linked to the word BALL. It is usually called an adjective. Note that without the word RED we would still have COLOURED BALL. Also note that we do not say RED COLOUR but do say THE COLOUR RED. It should be clarifying that HAVE was seen as BE WITH, BE being represented by the empty frame and WITH having word graph $\boxed{\text{PAR}} \longrightarrow \square$. So the graph might also be brought under words by “(the) ball has colour red”. Note that people may differ in opinion on expressing RED BALL. This is just the view of Hoede and Liu.

Quite a few adjectives are similar to RED and can be seen as adwords linked by the PAR- relationship. However, there are other ways how word graphs can be linked to the word graph of a noun and a verb. One particular way is by the FPAR-relationship that links the constituents of a definition to the defined concept.

Suppose, for example, that a stone is defined as “a structure of molecules” (which would not be precise enough, but lexica contain scores of unprecise definitions). If the type of molecules is denoted in the graph, by means of an EQU-arc, as silicon, we may speak of a SILICON STONE and SILICON is now functioning as an adjective. This type of adjective is of another nature than the adjective RED, the difference being expressed by the way of linking, one time by a PAR-relationship and the other time by an FPAR-relationship. The essential linguistic phenomenon is that word graphs are linked to other word graphs. As this can be down in various way, to nouns and to verbs, it is more natural to speak of adwords. We will discuss adjectives and adverbs from this point of view. The interesting phenomenon of classifiers in Chinese is closely related to our way of viewing adwords. In Chinese it is not well-spoken to say “a spear”. One should say “a stick spear”, where stick is a word expressing a classifying aspect of spear. A HORSE is not YI MA but should be expressed as YI PI MA, where PI is the classifier, the meaning of which seems to have been lost in the course of time. Yet the adword PI should be expressed in proper speaking. There are

more than 400 of these classifiers. We will discuss several of them from the same point of view as before and give word graphs for them.

II.2 Adwords

In this appendix we cannot give an extensive treatment of the grammatical aspects of adjectives, adverbs and classifiers. We will restrict ourselves to some major subclasses of these adwords. The book of Quirk *et al.* [Quirk *et al.*, 1972] was used for reference, more specifically its Chapter 5.

We will not stress syntactic problems. A certain knowledge graph is brought under words by expressing by words certain of its subgraphs. The way this is done differs from language to language. In an extremely simple example we have that RED BALL in English is uttered as BALLON ROUGE in French. Any knowledge graph admits an *utterance path*. Usually there are several utterance paths, i.e. ways of uttering the words, having word graphs that cover the knowledge graph, in a linear order. See also Chapter 6 of this thesis. As our example RED BALL or BALL WITH COLOUR RED shows, there are ways of bringing a knowledge graph under words that are more precise than others. In natural language use often the less precise descriptions, like RED BALL, are given. That RED is a colour and that colour can be attributed to a BALL is background knowledge for these concepts RED and BALL, that enables the short but incomplete utterance path. A lexicon may contain a word graph for RED that includes the colour concept, but a word graph for BALL that does not mention the possibility of attribution by means of a PAR-relationship. A machine may then not be able to create a connected knowledge graph for RED BALL, unless it is instructed to interpret the syntactic fact that both words are uttered together as justifying the linking of their word graphs by some arc. For a more elaborate discussion of the interplay of semantics and syntax we refer to Chapter 5 of this thesis.

II.2.1 Adjectives

As word graphs are supposed to grasp the semantics of a word, we focus on the paragraphs 5.37 to 5.41 in Quirk *et al.*, in which they give a semantic subclassification of adjectives.

They make the distinction stative/dynamic, gradable/non-gradable and inherent/non-inherent. Their Table 5:2 looks as follows:

stative	gradable	inherent	
+	+	+	BLACK (coat)
-	+	+	BRAVE (man)
+	-	+	BRITISH (citizen)
+	+	-	NEW (friend)

Adjectives are characteristically stative, most are gradable and most are inherent. The normal adjective type is all three, like BLACK. Quirk *et al.* give the imperative as a way to distinguish stative from dynamic adjectives.

One can say BE CAREFUL but not BE TALL or BE BLACK, BE BRITISH respectively BE NEW. One can say BE BRAVE, explaining the minus sign in the first column. BLACKER, BRAVER, and NEWER are gradings but BRITISHER is not possible, explaining the minus sign in the second column. Inherent adjectives characterize the referent of the noun directly. They consider BLACK, TRUE and BRITISH to be inherent adjectives.

Before undertaking discussions we should reproduce their premodification examples on page 925 in which combinations of adwords are mentioned.

determiners etc.	general	age	colour	participle	provenance	noun	denominal	head
THE	HECTIC						SOCIAL	LIFE
THE	EXTRAVAGANT					LONDON	SOCIAL	LIFE
A			GREY	CRUMBLING	GOTHIC	CHURCH		TOWER
A	SMALL		GREEN	CARVED		JADE		IDOL
SOME	INTRICATE	OLD		INTERLOCKING	CHINESE			DESIGNS

It is clear that here proposals have been made for what Quirk *et al.* call *semantic sets*. This is exactly what will be done, but the basis of the proposal will be the types of relationships a noun, or verb, can have with other words.

II.2.1.1 The FPAR-adwords

We use the FPAR-relationship to represent the definitional contents of a concept. Any word used in the definition might be called an FPAR-adword. However, usually some restrictions are made. If the definition contains a preposition like OF, this word is not considered an adword. Another remark that should be made is that a definition may contain the concept of colour. In that case the adjective, say GREY, is considered to be inherent. The definition of an elephant may contain the statement that it is a grey animal. In BLACK COAT, however, the adjective cannot be considered to be inherent, as was assumed in the table given. The point is that colour is attributed subjectively, a red ball in green light looks black. Even in white light objects may have different colours for a colour blind person. For this reason the PAR-relationship is used and RED is considered to be a PAR-adword and colour to be non-inherent in general.

Similarly BRAVE and BRITISH are disputable as inherent adjectives, for different reason. Braveness is present according to a judgement. One is considered to be brave by others. BRAVE can be seen as an instantiation of judgement. Other adjectives of this type are KIND and UGLY. BRITISH does not describe an inherent aspect either. Anything that is part of Britain, seen as a frame, can be called BRITISH. In this way the frame name determines the adjective for its constituents. BRITISH may therefore be called an inverse FPAR-adword. Examples of this type of adjectives are LAMB in LAMB MEAT and CITY in CITY COUNCIL or CHURCH in CHURCH TOWER.

It is concluded that the restriction inherent/non-inherent had better be replaced by the distinction FPAR/NONFPAR. For material objects a typical FPAR-adword expresses the sort of material. So in JADE IDOL, JADE describes the material and is a typical FPAR-adword.

If STEAM is per definition HOT WATER VAPOUR then HOT is an FPAR-adword as is WATER, instantiating relative temperature and material of vapour respectively. Note that we speak of relative temperature as HOT is not a temperature, like FAST is

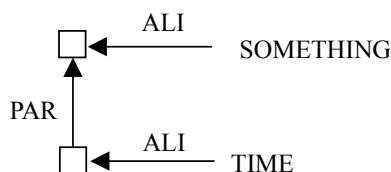
not a velocity.

Within a frame concepts may occur that allow a measure, like temperature or length. In those cases the corresponding FPAR-adwords are gradable, like in hot, hotter or long, longer. Usually these adwords do not indicate absolute temperature or length but relative temperature and relative length. In “a two meter man”, the precise length TWO METER may be interpreted as an FPAR-adword too.

II.2.1.2 The PAR-adwords

We use the PAR-relationship to represent exterior attribution. Judgements on a concept are typical exterior attributions. BRAVE was already classified as a PAR-adword. BEAUTIFUL is another good example.

An important class of words concerns space and time aspects. A ball may exist at some location at some moment in time. Its space-time coordinates will be seen as determined from the outside, i.e. by exterior attribution, and therefore represented by a PAR-relationship. Adjectives like EARLY and SUDDEN belong to this class and also OLD and NEW. The link to the main concept, or head as Quirk *et al.* call it, is by a PAR-link that connects the time aspect to the concept in the following way

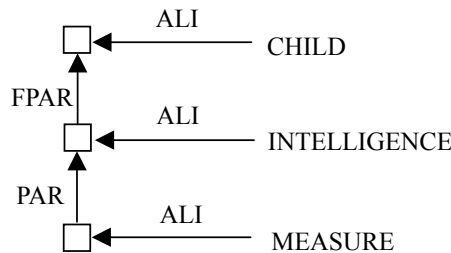


The word graphs for the adjectives embed the time aspect in a more elaborate word graph in order to express the various meanings. In these graphs the time of the speech act may play an important role, like in the description of tense, a theme that we discussed in Section 6.5. A speech act may occur at time t_0 , whereas the intended description concerns something at time t_1 , often determined by the discourse of which the speech act is part. In a historical account one may read “the former king was tyrannical but the new king was a very kind person”. The adjective FORMER refers to a time before a certain time t_2 , when the king was replaced, the adjective NEW refers to a time after time t_2 . The fact that this took place before t_0 is apparent from the past tense coming forward in WAS.

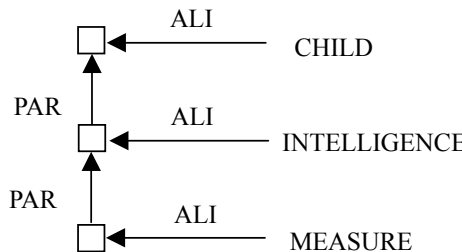
The word graph for FORMER and NEW will contain an ORD-relationship. In EARLY reference to a time interval will have to be made, expressing that the first part of the time interval is meant. This also holds for OLD or ANCIENT. Differences between the three adjectives must come forward in the word graphs, but if these are left out by the speaker, in his choice of words, he might speak of “in the early days”, “in the old days” or “in the ancient days” to express the same thing. The more elaborate word graphs may become quite large. This situation is similar for dictionaries, where bad dictionaries give short definitions, and good dictionaries try to give more precise definitions.

Definitions may contain concepts susceptible to objective measure. We mentioned temperature or relative temperature in Section II.2.1.1 and HOT as an FPAR-adword. Concepts may also be susceptible to subjective measure. In INTELLIGENT child, UTTER fool and LOUD noise we have examples of adwords that express subjective measurement. Having intelligence, which might be part of the frame CHILD, does not justify to speak of an INTELLIGENT CHILD. For this some measure should be used which is externally attributed to INTELLIGENCE.

So we have



Depending on the outcome of the measurement the word INTELLIGENT or STUPID may be appropriate. In this description, with INTELLIGENCE as part of the frame, INTELLIGENT is an FPAR-adword. However, if intelligence is seen as something attributed from the outside it is a PAR-adword and we have

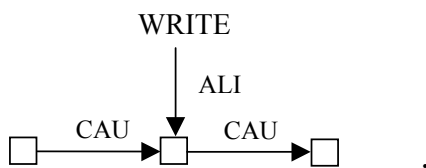


Somebody is held to be a fool and any measurement is subjective. UTTER refers to the extreme end of an imaginary scale. LOUD is clearly also used for a subjective measurement and therefore classified as a PAR-adword. Even if measured, in decibels, it is still a subjective choice to say that certain numbers of decibels correspond to LOUD. Yet if NOISE is defined by a frame including such a measurement, LOUD should be classified as an FPAR-adword.

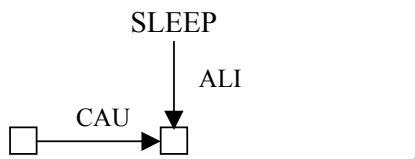
The discussion should have shown that the classification that is proposed depends on the explicit choice of the frames in defining concepts.

II.2.1.3 The CAU-adwords

In the representation of language by knowledge graphs verbs are represented by a token and CAU-relationships. Transitive verbs like WRITE are represented as



whereas intransitive verbs like SLEEP are represented as



Consider the leftmost token. It represents “something writing” respectively “something sleeping”. Thus WRITING and SLEEPING are adjectives of that something and are, for obvious reasons, classified as CAU-adwords.

The rightmost token in the first knowledge graph might represent a letter and can be described as “written letter”. So the adjective WRITTEN can also be classified as a CAU-adword. Again we should note, like for NEW, that the word graph should contain information referring to the time aspects. WINNING team, BOILING water and MARRIED couple give examples of CAU-adwords. The last example is somewhat tricky. Marrying usually involves two persons and “A marries B” and “B marries A”. Then “A and B have got married”, or both are “in the state of marriage”.

This brings us to a special class of adjectives, that describe *states*.

Suppose something is subject to a series of changes. At any time it is then in a certain state. In HAPPY girl the adjective describes a state that the noun is in, “girl being happy” is doing the same. The verb BE was represented by the empty frame. HAVE, which was defined as BE WITH, CAN and MUST are verbs that likewise correspond to the frame relationships that were distinguished. When they are used, they usually express states, focusing on the process rather than on the subject or object related to the process by a CAU-relationship. Yet the adjectives describing states are discussed in this section, as we stress the relationship with verbs.

States are exemplified by adjectives like ABLAZE, ASLEEP or ALIVE, where descriptions of processes stand central. Having talent or having a disease is expressed by TALENTED respectively DISEASED. The form suggests a verb. However, A BOY WITH TALENT is a way of wording that seems better than TALENTED BOY. The adjectives in ESCAPED PRISONER or RETIRED EMPLOYEE correspond to intransitive verbs ESCAPE and RETIRE. Like for objects of a transitive verb the –ED ending is used, but now to express a state after the process, of escaping respectively retiring, has finished. Strictly speaking the two adjectives are not CAU-adwords. THE PRISONER WHO IS ESCAPED respectively THE EMPLOYEE WHO IS RETIRED seem better ways of wording than THE ESCAPED PRISONER respectively THE RETIRED EMPLOYEE.

The check on the distinction stative/dynamic, by trying out the imperative, like in BE CAREFUL, corresponds to checking whether the adjective can be seen as describing a state.

It should be noted that predicative use of an adjective, expresses a BE-frame. THE CAR IS HEAVY instead of THE HEAVY CAR expresses the “being heavy” of the car explicitly. In Turkish DOKTORDADIR literally means DOCTOR AT BE. The very frequently used agglutination DIR expresses the BE-frame. Most of the a-adjectives, like ABLAZE, are predicative only. THE HOUSE IS ABLAZE can be said, THE ABLAZE HOUSE cannot be said. In THE CAR IS HEAVY the use of the adjective HEAVY is predicative. The analogy is suggested by the word IS. However, in this case this word IS stems e.g. from A TRUCK IS (DEFINED AS) A HEAVY CAR which is shortcutted to A TRUCK IS HEAVY or even THE CAR IS HEAVY. This

explains the predicative use of an adjective that is essentially an FPAR-adword. HEAVY is pushed into the role of a state describer like ABLAZE.

II.2.1.4 The ALI-adwords

The ALI-relationship between two concepts expresses the likeness of the concepts. This relationship may be seen as *primus inter pares* as the process of concept creation seems to depend heavily on the becoming aware of likeness. Prototype definitions express what has been seen as common properties of a set of somethings.

Adjectives that are expressing that a concept looks *like* another concept may have specific endings. DISASTROUS expresses a similarity with a disaster and INDUSTRIOUS expresses a similarity with certain aspects of industry. Other endings are -ISH as in FOOLISH or -LIKE as in CHILDLIKE, for example in combination with BEHAVIOUR.

A special category of adjectives is ALI-adwords that are themselves nouns. THUNDER in THUNDER NOISE or TRAITOR in TRAITOR KNIGHT expresses that the noise is like heard when thunder occurs or that the knight acts like a traitor, respectively. Especially for this category of adjectives, nouns acting as adjectives, it becomes clear that the classification that we give in different types of adwords makes sense.

II.2.2 Adverbs

Nouns and verbs are describing concepts basically in the same way. Whereas words of the first type do not necessarily include CAU-relationships in their definition, words of the second type do. But for this difference a verb may be seen as a special type of noun. This means that there is no basic difference in the way other words may act as adwords of verbs in comparison to the way other words act as adwords of nouns. This is also underlined by the possibility of substantiation of verbs. Compare TO PLAY, PLAYING and A PLAY.

We will discuss only a few examples.

Time and location of the act expressed by the verb are natural aspects to consider in

relation to the verb, far more so than for nouns. Many adverbs refer to these aspects and are classified as PAR-adwords. We mention OFTEN, OUTSIDE, BRIEFLY, EVER as examples. The explicit word graphs for these words may become quite large as rather complex aspects are expressed. Judgements and measurements are two aspects that are also often expressed by PAR-adwords. WELL, QUITE, EXTREMELY, ENOUGH, MUCH, ALMOST are reflecting judgements or measurements. A judgement may be interpreted as a subjective measurement, hence the treatment of these two aspects at the same time. CAU-adwords refer to influences or, in most case, to consequences of the acts described by the verbs. Hence adverbs like AMAZINGLY or SURPRISINGLY may be mentioned. ALI-adwords include CLOCKWISE, or any of the many adverbs with ending –WISE, but also words like TOO or AS are used as adverbs and clearly should be another subclass of these adwords. FPAR-adwords are somewhat rare. In “the country is deteriorating economically” the adverb ECONOMICALLY indicates in what respect the country deteriorates. The country’s economy must be seen as frame part and for this reason ECONOMICALLY may be classified as an FPAR-adword.

Two remarks are to be made still. Firstly, there is a set of words, mentioned by Quirk *et al.*, that are sometimes used as adverbs, but actually refer to the use of logic in language. NEVERTHELESS, HOWEVER, THOUGH, YET, SO, ELSE are such words. Hoede and Liu prefer to include them in a special third list of words graphs with NO and PROBABLY, to mention some other potential adverbs. Chapter 4 of this thesis focuses on such logic words.

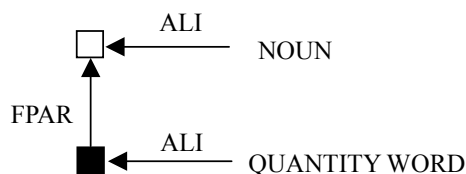
Secondly, we would like to comment on an example of Quirk *et al.*; A FAR MORE EASILY INTELLIGIBLE EXPLANATION, showing compilation of adwords. In traditional discussion we would have to decide whether we are dealing with adjectives or adverbs. The great advantage of our theory is that this discussion is avoided. Instead one may have the discussion about the classification that is be given to each of the adwords. However, once a knowledge graph, that expresses the text, is made out of the word graph, the way these word graphs glue together, by which type of arc, immediately gives the answer in that discussion. An important aspect of this example is that the sheer possibility of compiling adwords in language is an argument for the modeling of language in terms of knowledge graphs, built from word graphs.

II.2.3 Classifiers in the Chinese Language

In Chinese, a special class of words is formed by the *quantity words*, as Chinese prefer to call them, which are also called *classifiers*.

II.2.3.1 FPAR-classifiers

One of the most frequently used quantity words is GE, used in combination with a word like DONG XI, THING. For A THING in Chinese we should say, YI GE DONG XI. There are many nouns for which the quantity word GE is used. As this example already shows the quantity word may be seen as a word naming a subframe of a frame carrying the name of the noun. Hence here the relationship between noun and quantity word is an FPAR-relationship, the quantity word is an FPAR-adword and, in terms of knowledge graphs, we get the following graph.



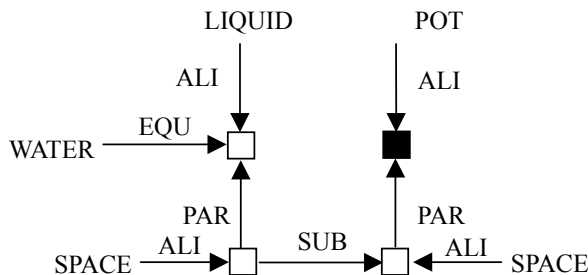
Here, ■ indicates the something described by the quantity word.

Although a word like GE is not felt to have a meaning of its own by Chinese, it should refer to something felt as an essential property of the following noun. Many quantity words express the property of the noun that it describes a unit. Another example YI FENG XIN for A LETTER, shows an adword FENG, that in the combination is not felt to have a meaning. However, there is a verb FENG for folding and it is clear that XIN (LETTER) is described in a way that expresses the property that letters usually consist of folded paper.

For A WELL we have YI YAN JING, where the quantity word YAN has some meaning, namely HOLE. If a well is the configuration of water coming from a hole in the ground, the hole property of a well is clearly mentioned as property not to be deleted in the description. Quantity words like these should perhaps better be called *property words* or classifiers.

II.2.3.2 Other classifiers

Next to the FPAR-relationship to a noun of a property word, there are many other classifiers that do have a separate meaning, but are used in combination with a noun to express a certain feature. These quantity words can be divided into 13 subclasses. We will not discuss this in more detail now and only mention a typical example: YI HU SHUI, A POT OF WATER. In English too a class description of water is necessary. HU, or POT, is an adword for SHUI, or WATER. A pot typically contains a liquid, as a container it has a specific shape and is made of some material like China, metal or class. The important feature is that of containing a liquid. In knowledge graph representation we have in first instance



The graph can be extended to include the other features of POT like shape or material, but the containment feature gives the direct link to the noun.

In terms of adwords we cannot say that POT is a PAR-adword, the linking to WATER is more complex, in fact our example is a clear example of a relationship in knowledge graph theory. Two concepts are part of one graph and it is this graph that characterizes the relationship between the two concepts. If we have to baptize this relationship we would choose the word CONTAINER or CONTAINMENT. So HU is an adword of SHUI linked to it by a relationship of complex nature, that is however representable by the basic types of arcs.

There are quite a few adwords of the type CONTAINER, PING or BOTTLE is just one other example.

For the list of word graphs for the other 12 classifiers, see the original paper [Hoede & Liu, 1998].

Index

- adwordial phrase 84
- adwords 84, 130, 201
- argument transition network 9
- artificial intelligence 1, 106
- attribute 24

- causal relation 22
- chunk IX
- chunk indicators 4
- compositionality principle 34
- computational linguistics 2
- concept 3, 19
- concept identification 36
- concept integration 37
- conceptual dependency 16
- context 181
- coreference resolution 145

- dependency relation 25
- distribution 132

- equal 23
- event 143
- existential graph 29

- functional 25
- frame 26

- general 43
- grammar 7

- IE 144
- informational dependency 25
- inherent 51
- integrated 16

- joining sentences 3

- KG-definition 154
- KG-extraction 150
- knowledge graphs VII
- knowledge representation 1

-
- logic words 4
 - logic words of the first kind 45
 - logic words of the second kind 48

 - mappings 25
 - mind graph VIII, 28
 - MUCs 143
 - MT 144

 - Name Entity recognition 145
 - natural language processing 2
 - natural language understanding 2
 - NECPAR 26
 - NEGPAR 26
 - nonmovable adverbs 131
 - non-syntactic 16

 - ontology 27
 - ordering 25
 - other logic words 44

 - parsing 7
 - partial structural parsing 151
 - phrase marker 110
 - POSPAR 26
 - property sets 24
 - pure logic words 44

 - relation integration 37
 - relationship 19

 - scenario 145
 - semantic chunks 17
 - semantic sentence graph 17
 - semantic word graphs IX, 17, 71
 - sentence graph VIII
 - structural parsing IX, 4, 17, 69
 - syntactic sentence graphs 17
 - syntactic word graphs IX, 17, 71

 - template 145
 - token 20
 - totality 124
 - transformational grammar 9
 - type hierarchy 21

 - u-arc 139
 - utterance paths IX, 4, 17, 108

 - value 23

 - word types 72

Summary

In Chapter 1, Section 1.3, an outline of the contents of this thesis is given. In this summary we focus on the background of the problems considered and on future research items.

This thesis mainly addresses the following points.

- Although there are many ontologies for knowledge representation, till now no one can be called universal, because it can replace the others. This is why the *knowledge graph theory* was put forward and has been developed gradually. Given two ontologies O_1 and O_2 , we may say that O_1 is superior to O_2 , if the elements of O_2 can be expressed in terms of the elements of O_1 and not the other way around. As knowledge is represented in terms of language, the first goal must be to express language in terms of the knowledge graph ontology. Chapter 2 gives a short account of parsing natural language for those readers that are not familiar with that subject. Chapter 3 focuses on the knowledge graph ontology and makes comparisons with a few other well-known ontologies. A comparison with the several hundreds of ontologies known poses a challenging problem.
- Human language is a kind of special symbol system, in which a *word* is a basic component. Each word not only has specific meanings, its *semantic* aspects, but also has a structural function as it plays a role with respect to other words, its *syntactic* aspects. We propose how to express words by *word graphs* both semantically and syntactically. However, in the knowledge graph project the focus

first was on the semantic word graphs. A series of papers on making word graphs, both for English and Chinese has been published. The first set of words consisted of prepositions, verbs and nouns, the second set consisted of adjectives, adverbs and Chinese classifiers. The third set consists of the group of so-called logic words and is described in Chapter 4. A point of further research is to see whether some more sets are to be included for English and Chinese, and to see whether some special word classes exist in other languages. The appendices I and II give a survey of the first two sets mentioned.

- Parsing is an essential part of Natural Language Processing (NLP). In the context of knowledge graph theory, parsing is more special than traditional parsing and is called *structural parsing*, which aims at transforming an input sentence into a *sentence graph*, i.e. giving a meaning to this sentence. We argue that the structural parsing developed in Chapter 5 of this thesis could be used both in English and in Chinese. For mapping a sentence on a sentence graph, both syntactic and semantic information is needed. As one of the goals is to develop translation systems, with the main steps: structural parsing, transformation of the sentence graph and uttering the sentence graph in the target language, an important point of further research is to develop computer programs for structural parsing.
- Given a sentence graph there are usually several ways how such a graph can be brought under words, i.e. can be *uttered*. The sentences arising from these ways of uttering consist of words occurring in the sentence graph in a specific order. Languages differ in the way the words occurring in the sentence are ordered. This is studied in Chapter 6. The development of computer programs for uttering a sentence graph, as well as for the transformation of sentence graphs from one language into another, should also be part of a project on automatic translation based on knowledge graph theory.
- *Applications* of knowledge graph theory are a challenge, especially in NLP. Based on the theory developed in this thesis, we developed a method for carrying out Information Extraction (IE). Again, the implementation of this method poses an interesting challenge. This in particular with respect to the artificial intelligence aspects that have come forward in our explicit treatment, by hand, of an example, that forms the contents of Chapter 7.

Curriculum Vitae

The author of this thesis was born on March 10, 1964 in Xian, Shaanxi Province in the northwest of China.

From 1971 until 1983 she attended primary and grammar school in her home town. From September 1983 to July 1987, she studied computer software at the Northwest University. In July 1987, she was appointed to study Artificial Intelligence under the supervision of Prof. Zhou Guodong. In the course of her studies she concentrated on *knowledge representation*, and her master thesis was titled “An Expert System Tool Based on Conceptual Structure”. Here the author investigated conceptual graph theory and its applications in building an expert system.

After obtaining the master’s degree, she worked as a teacher at the Department of Computer Science of the Northwest University on July 1990. On May 1996 she won the first prize of the “Teaching Competition” held by the same university. In January 1998 she started as a Ph.D. student under the supervision of Prof. Dr. Cornelis Hoede from the University of Twente and Prof. Dr. Li Xueliang from the Northwestern Polytechnical University. In 1999, she was appointed associate professor.