A Robust Unification-Based Parser for Chinese Natural Language Processing

CHAN Shuen-ti Roy

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Computer Science and Engineering

August 2001

Acknowledgement

First of all, I would like to thank Prof. Lee Kin Hong, my supervisor, for his guidance and patience. My research could not have been done reasonably without his insightful advice. For the past two years, he gave me encouragement, support, and guidance on my papers (published in ICCLC 2000 and ICCPOL 2001) and my thesis. It would not be possible to have my papers published without him. Moreover, Prof Lee supported me financially to the conferences held in Chicago and Seoul. My knowledge has widely broadened due to such remarkable experience.

Secondly, my great gratitude goes to Prof. Gu Yang of the Department of Modern Languages and Intercultural Studies and Dr. Moon Yiu Sang of my department, who marked my term papers and thesis. My research is very much related to Chinese and English linguistics. Before joining the department I was weakly trained in that area and thanks to the guidance of Prof Gu, I have become quite competent in general linguistics. Prof Gu is also a very good partner to talk to. She is very kind to me and always gives me courage to go further in research. Dr Moon knows much about the limitation of natural language processing and is a very funny person. He gave me numerous suggestions and I learnt a lot from him.

Thirdly, my gratitude also goes to the Department of Computer Science &

Engineering, CUHK. She provides very good equipment and a quiet office environment for high quality research.

Fourthly, I'd like to thank my fellow colleagues, both in CUHK and HKU, who helped me solving a lot of computer and life problems. What's more, our lively talks enlightened me to the art of succeeding in scientific research. Special thanks to all boys and girls in Rooms 1013, 913 and 1005, particularly (in alphabetical order) Willis Chan Wai To, Cheung Yin Ling, Vincent Cheung Wing Hang, Matthew Leung Ka Hing, Raymond Chung Lui Ming, Napoleon Lee Shing Yan, Tommy Tang Wai Kwan, Data Tsung Hin Chung, Yan Men Hin and Ham Wong Siu Ham. They gave me a happy and wonderful postgraduate life. My teachers and ex-colleagues in HKU also gave me much spiritual support. Cheers to them.

Finally, I must thank my family for taking care of me, feeding me and enjoying life with me. My study would be a mess if my family were not giving me wholehearted support. Thank you very much.

Abstract

In natural language processing (NLP), there are two major streams of methods used. One is unification-based grammar (UBG), or generally referred as deep information processing. Another one is corpus-based processing, or shallow information processing. UBG performs deep linguistic analysis through information-rich grammar and lexicon. Corpus-based processing learns linguistic generalizations in a corpus with the help of statistical models. Researches have noted that a NLP system without deep information processing cannot handle sophisticated task.

However, there are some problems with UBG, when it is put into practical use. First, by theory a unification model cannot handle irregular input. It assumes that any word appears in the input text can be found in the lexicon. If not, the parser fails every time new words appear. However, in real practice we will face many words that do not appear in the lexicon.

The second issue is that the unification-based grammar formalisms are originally developed particularly for English and other Indo-European languages. They do not share the same problems faced by Chinese. For instance, segmentation of sentences into words is a vital part of Chinese language processing and it directly affects the design of parser. Previous works in this area often treat segmentation and parsing two

independent issues. We suggest that, however, the parser must be able to deal with any correct (multiple correct segmentations are possible) segmentation output. Also, the notion of Chinese sentence is confusing enough, as it cannot be determined exclusively with syntactic terms. If we simply feed the text between two full stops to the parser, we may not get a correct output. This means we must find method to choose a suitable parsing unit.

In this thesis a robust UBG parser for Chinese NLP, SERUP (Statistically Enhanced Robust Unification Parser), will be presented. It can be viewed as hybrid of both UBG and statistical processing. The ultimate of it is to give correct parsing result for all grammatically correct Chinese sentences. It consists of a UBG backbone, with a robust statistics-driven automatic preprocessing procedure that segments and standardizes raw texts to allow the parser to be used in practical applications. Apart from phrasal rules, the parser also contains morphological rules so that multiple segmentation results problem discussed above can be relieved. It also tries to identify novel word compound in the input texts before parsing takes place with statistical method, such that the parser will not fail easily. Furthermore, we clarify the concept of "Chinese sentence", which is important for discourse reconstruction from discrete clauses. This is of particular importance to Chinese, a language that linguists consider without clear notion of "sentence".

Results show that SERUP is a promising practical-theoretic model for Chinese language understanding.

論文題目: 一個用於漢語理解的穩健的合一文法爲本之語法分析器

作者　　: 陳旋第

學校　　: 香港中文大學

學系　　: 計算機學與工程學系

修讀學位: 哲學碩士

論文撮要:

在自然語言處理 (Natural Language Processing) 這個課題上， 一般會用到兩種技術。 其一乃以合一文法 (Unification-Based Grammar) 爲本的『深層信息處理』(deep information processing)。 其二乃以統計學原理爲本的語料庫 (corpus) 處理法， 或稱『表層信息處理』 (shallow information processing)。 合一文法把滿載語言信息的詞彙 (lexicon) 跟著語法約束而結合成句並作出分析； 語料庫處理法則利用統計學模型來找出語料庫中的語言泛化 (linguistic generalization)。 隨著多年的研究， 計算語言學家意識到一個缺乏深層信息處理模組的自然語言理解系統是不可能勝任高等作業的。

然則合一文法不能直接放在實際應用上。 首先，根據理論基礎合一文法假定所有在文本中出現的詞彙均可在詞彙庫中找到，否則合一 (unification) 將會失敗。可是現實並非如此。 我們往往會遇到詞彙庫裏沒有的新詞。

其次合一文法理論起源於英語及其他印歐語言，因著這些語言和漢語有著差異，以致有些言語處理手法對漢語而言並不適用。 舉例說詞的切分問題乃漢語處理的最大難題， 其解決方法將直接影響到語法分析器的設計。 傳統的研究通常把詞的切分 (lexical segmentation) 和語法分析 (parsing) 視作兩個互不相干的獨立項目， 我們則強調它們是不可分割的， 任何正確的切分結果 (漢語中容許多重

切分) 都應該能被語法分析器所接納及作出準確分析。 另外， 句子 (sentence) 對漢語而言是個令人混淆的概念。 我們不能用西方的句法名詞來定義漢語的句子。 假若我們隨便將兩個句號之間的文字當作一句子並把它投進語法分析器中，結果肯定不正確。 這暗示著我們一定要找到辦法來定義語法分析器的最小分析單位。

本篇論文將展示一個可用於漢語理解的穩健的 (robust) 合一文法爲本之語法分析器 SERUP。 它結合了合一文法和統計學處理模組。 其最終目標爲給所有合符文法的漢語文本給予正確的分析。 它的骨架是以合一文法爲本的語法規則，配合由統計學驅動的自動預處理程式 (preprocessing procedure) 去進行詞的切分和文本統一化，使語法分析器可用於一般應用系統。 除了短語規則 (phrasal rules)， 我們更引進了詞形態 (morphology)、 詞組合規則來應付多重詞切分的問題。 我們亦嘗試在文本中找出新詞，並在語法分析進行前將其組合，以提高語法分析器的穩健性。 再者，我們將闡明『漢語句子』 (Chinese sentence) 的概念。 此概念對漢語篇章 (discourse) 重組關係至大。

研究結果展示出 SERUP 作爲一個理論兼實踐並重的漢語理解模型是頗爲成功的。

Abbreviation

Symbols	Meaning
→	Can be rewritten as (Rewrite symbol)
<	Linear precedence symbol
=	Path unification operator

Linguistics terms	
A	Adjective
Adv	Adverb
AP	Adjectival phrase
H	Head
N	Noun
NP	Noun phrase
P	Preposition
POS	Part Of Speech
PP	Prepositional phrase
S	Sentence
V	Verb
VP	Verb phrase

1. Introduction

1.1. The nature of natural language processing

Information is the key of the 21st century. As we are approaching an era that information dominates, more and more people expect the emergence of machines that approach human performance in the linguistics tasks of reading, writing, hearing, and speaking. If so, human can communicate with computers through a totally natural interface, as if two people are speaking to each other. No doubt this is a very enormous goal and is easy enough for common people to see the potential difficulties. I will start by addressing the core of this challenge, i.e. the nature of languages.

Language science, being a complicated field of natural science, is studied in several different academic disciplines. Each discipline defines its own set of problems and methods for addressing them. Sometimes the assumptions and conclusions made in one discipline contradict those in another. To name a few, for instance, a general linguist deals with human languages as a sign system, and attempts to develop theories explaining general universal regularities of language [1]. He uses language models and intuitions about well-formedness and meaning. A psychologist, on the other hand, studies the cognitive processes of human including language production and perception. He considers questions such as how people identify the appropriate structure of a sentence and when they decide on the appropriate meaning for utterances. Experiments and statistical analysis are the major tools to draw correlations and conclusions. A philosopher considers how meanings are conveyed through words and how objects are identified in the world. Logic and formal

mathematical models provide him a structured mechanism to fit the world of languages into some universal principles.

Having understood the subtleties of languages, it is understandable that computational models are needed for the computer to understand language in a more efficient and coherent manner. There are two main streams in the evolution of such models.

On the one hand, computational linguists see the need of a model for exploring the nature of linguistic communication. They develop computational theories of language using the notions of algorithms and data structures from computer science, and also the theories of set and logic in mathematics to simulate the human counterpart. They believe that by observing the performances of the computer programs in real situations, they can be incrementally improved until deep understanding is acquired. On the other hand, engineers see the need of natural language interface and applications in real life. They need language models to store and retrieve linguistic knowledge, but they care about how good the models work rather than how they reflect the way humans process language.

There has been much dispute between these scientific and engineering views. Computational linguists believe that natural language is so complex that an ad hoc approach without a well-specified underlying theory will not be successful. Engineers claim that the present state of knowledge about natural language processing is not adequate enough to draw conclusion that it is feasible to build a cognitively correct model. Any less scientific model that appears to work should not be neglected [2].

A more recent view, which is also the foundation of this thesis, is to incorporate the

advantages of both views to come up to a model that serves engineering purpose yet allows enrichment and expansion in its infrastructure. It will be explained in details in due course.

1.2. Applications of natural language processing

Natural language processing research is nearly as old as the age of computer. Its contents have been continuously upgrading alongside the development of computer technology that is getting more and more sophisticated and diverse.

The applications of natural language processing up to now can be viewed as the by-products of three periods of computational linguistics research.

The first period was dated from 1950s to 1966. In 1949, Andrew Booth of the University of London and Richard H. Richens started the world first ever **machine translation** project. The motivation behind this was highly historical. At that time, just 5 years after the World War II, there was an urgent need in translating Russian scientific and military articles to English. Qualified human technical translators are hard to find, expensive, and slow (translating somewhere around 4 to 6 pages per day, on the average) [3]. The ultimate goal of machine translation, no matter it can be achieved or not, is to achieve high-speed high-quality translation of arbitrary text through computer programs. This first attempt led to many such researches all around the globe. Despite a major setback in the late 1960s, machine translation however is still the most popular topic of all natural language processing researches.

Then came a long retrospective period: from 1966 to early1980s. In 1966, the US

National Academy of Sciences issued the notorious ALPAC report that nearly put an end to all machine translation researches. It said that there were already too many human translators, that many of the texts being translated by MT were not needed, and that as long as the machine translation output had to be edited it was too expensive [4].

Such disaster, however, gave rise to a long burst of innovations and discoveries in linguistics and computational linguistics. Chomsky's [5] inspiring work of transformational generative grammar gave rise to a number of influential grammatical theories. Parsing techniques were greatly improved. Computational semantic models were also suggested. All these advancements led to a burst of text-based applications, such as **question-answering systems** and **story understanding systems**.

In a question-answering system, the user is allowed to inquire a certain fact using natural language query; a story understanding system is similar to the type of reading comprehension tests used in schools and provides a very rich method for evaluating the depth of understanding the system is able to achieve [2].

The Internet era: started from late 1980s brought new dimensions to the field. Since the birth of the Internet, the rate and directions of information exchange have been skyrocketing. It is obvious that human effort alone is not enough to make full use of the piles of information, which are mainly expressed in natural languages. A number of practical and high-marketing-value applications became available, e.g. **automatic text categorization**, **automatic text summarization**, **information retrieval**, **automatic text correction**, etc.

A system with automatic text categorization categorizes and makes indexes for a large number of arbitrary articles. An automatic text summarization system summarizes any arbitrary article, usually in the form of knowledge frame or event frame. Information retrieval generally refers to the automatic extraction of knowledge with natural language query (another form of question-answering system). Nowadays it is mostly refer to the extraction of relevant documents in the World Wide Web. An automatic text correction system is usually embedded into word processing software that locates spelling and grammatical errors automatically and provides possible solutions.

Apart from these text-based applications, we also see a number of dialogue-based applications, such as **speech question-answering system**, **speech-controlled user interface**, **speech translation system**, etc.

A speech question-answering system differs from its text-based counterpart only in the medium of query. The human user is allowed to make a query by speech, so there is no need to type. Speech-controlled user interface allows human user to use speech commands to control the computer rather than pointing and clicking the mouse here and there. Speech translation system takes human speech in source language as input, translates it and generates synthesized speech in target language.

These applications take advantages of the relatively well-established speech recognition technology and statistical language processing, however, the core of them is still the same techniques used in text-based natural language processing.

1.3. Purpose of study

In natural language processing (NLP), there are two major streams of methods used. One is unification-based grammar (UBG), or generally referred as deep information processing. Another one is corpus-based processing, or shallow information processing. UBG has information-rich grammar and lexicon and is widely believed to be the most suitable linguistic theory for structural analysis that is crucial to a number of NLP tasks that require strong language understanding like machine translation. Corpus-based processing learns linguistic generalizations in a corpus with the help of statistical models. Researches elsewhere have noted that a NLP system without deep information processing cannot handle sophisticated task [6]. However, UBG is only a theory, it is not yet ready for practical use. First, a UBG parser fails to produce output every time new words appear. In real practice we will definitely face many words that do not appear in the lexicon.

The second issue is that UBG are originally developed for English and other Indo-European languages, but not for Chinese. For the model to work for Chinese, we have to deal with the characteristics of it, for example, segmentation problem and vague definition of sentence.

In this thesis a robust UBG parser for Chinese NLP, SERUP (Statistically Enhanced Robust Unification Parser), will be presented. We want to take advantages of the power of UBG, but at the same time using statistical method to improve the performance of it, particularly in allowing more arbitrary texts to get correct parsing result.

We will restate our goal in a more detail fashion in chapter 3.

1.4. Organization of this thesis

Chapter 2 presents the organization and the two main streams in natural language processing—unification- based grammar approach and corpus-based approach.

Chapter 3 discusses the difficulties in Chinese language processing and its related works. Discussion about these works is divided into two sections, one for unification-based approach, and the other for corpus-based approach. Apart from a summary, pros and cons will also be mentioned for each piece of work. Chapter 3.6 is a restatement of our goal of this thesis in detail.

Chapter 4 gives an overview of our parser—SERUP (Statistical Enhanced Robust Unification-based Parser). This chapter, as well as the subsequent chapters, is mostly based on the two technical papers we submitted to the ICCLC2000 (International Conference on Chinese Language Computing [47], ICCPOL2001 (International Conference on Computer Processing of Oriental Languages) [55].

The details of each step of SERUP are given in chapter 5 to chapter 7. Chapter 5 talks about the automatic preprocessing module, which is the first step of the parser. Chapter 6 is about the second step—grammar construction. We will lay out the foundation and implementation criteria in this chapter. Chapter 7 deals with structural disambiguation using semantic collocation method during parsing.

Chapter 8 gives the implementation detail, performance and evaluation of SERUP. It

includes the overall accuracy and runtime metrics of the parser as well as the accuracies of every part of the system mentioned in chapter 5 to 7.

In Chapter 9 we give a short summary of the thesis, then discuss some of the future works that can further enhance the performance of SERUP.

2. Organization and methods in natural language processing

2.1. Organization of natural language processing system

The organization of a natural language processing system, whether it is dedicated to information retrieval or question-answering, is the same in the sense that its processes must be a subset of a classical (based on deep language analysis) machine translation system. Why is it so? An intuitive answer is that translation (human translation, not machine one) is probably the most complicated language processing routine in human brain. It calls for source language knowledge as well as target language knowledge, which are the accumulation of linguistic, social, psychological and historical knowledge. Machine translation system merely imitates the human cognitive process (whether the imitation is correct or not) and so the complexities are the same. For instance, information retrieval requires strong source language reasoning but does not care about regeneration of target language.

Figure 2.1 shows the system architecture of a classical machine translation system based on text understanding. Furthermore, following the flow of control, realization of the response will not get satisfactory result, if the processes on the left hand side, i.e. the source language analysis, produces incorrect representation. In other words, failure in earlier stages will be carried to later ones. It is clear that parsing is the most crucial phase of all.

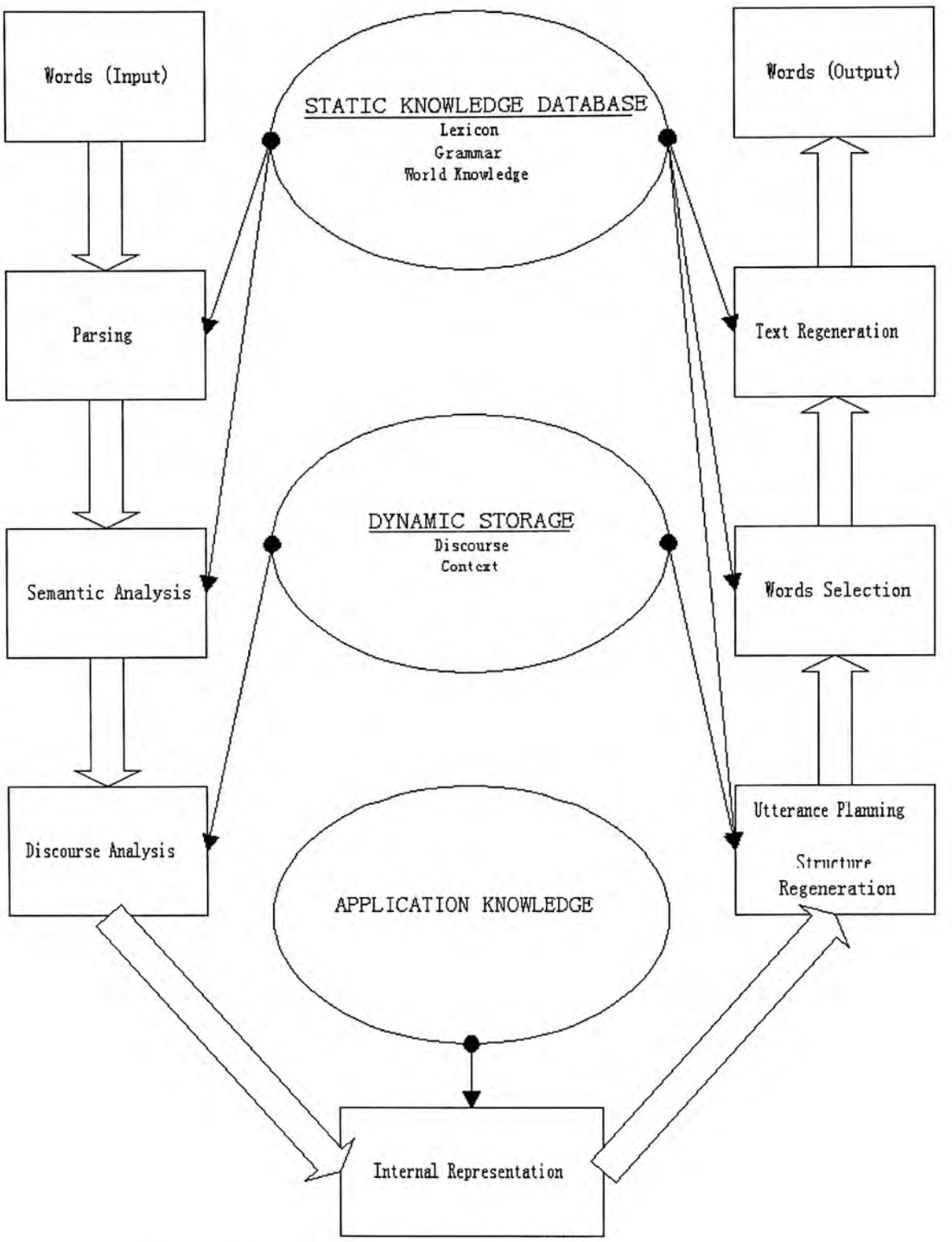


Figure 2.1. Architecture of a classical machine translation system.

2.2. Methods employed

In the early days, natural language processing research was mainly about syntactic formalisms and parsing. The notion of context-free grammar (CFG) was introduced by Chomsky [5] and has been studied extensively since in linguistics and computer science. Many parsing algorithms originally for analyzing programming languages were adapted to fit the parsing of natural language. Recursive transition networks (RTNs) introduced in Woods [7] and definite clause grammar (DCG) introduced in Pereira and Warren [8] are the other two popular simple grammatical formalisms.

However these grammar formalisms were proven inadequate to deal with the complicated natural language grammar and new theories and methods have been proposed. Here the two major trends in current computational linguistics, i.e. unification-based grammar processing and corpus-based processing, will be discussed.

2.3. Unification-based grammar processing

Decades of research in computational linguistics, formal linguistics and natural language processing resulted in the formation of a particular approach to encoding linguistic information, both syntactic and semantic, which has been called "unification-based" or more recently "constraint-based grammar formalisms" [9, 10]. The fundamental operation upon them is the unification operation, which is based on the use of feature structures. Linguistic information is stored in both grammar and lexicon. Through unification, information is shared and combined to form a bigger structure. Constraints are added to guide correct unification. As a result surface

sentence can be analyzed in a clear and structural manner.

A feature structure is a set of Features (Attributes) and Values. It contains at most one value for each feature [11]. Consider the following feature matrix or attribute-value matrix (AVM):

(1)

$$\begin{bmatrix} \text{CAT} & \text{N} \\ \text{PER} & 1 \\ \text{NUM} & \text{SING} \\ \text{CASE} & \end{bmatrix}$$

It denotes a common first person singular noun. The left hand column labels "CAT", "PER", "NUM" and "CASE" are generally known as features or attributes. "N", "1" and "SING" are known as the values of the features or attributes. In such notation, each feature (or attribute) takes either one value or no value. For instance, this feature matrix contains no value for the attribute CASE, as indicated by a blank. Another way to represent the feature matrix is to use a Directed Acyclic Graph (DAG), which is borrowed from graph theory in discrete mathematics (Figure 2.2).

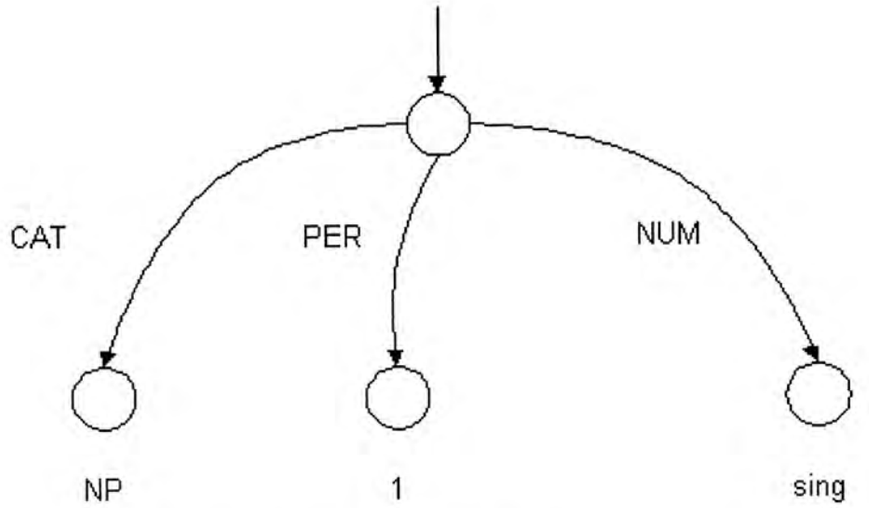


Figure 2.2. Directed Acyclic Graph

We have exactly the same structure as the feature matrix, but it works better in visualization of the feature structure. Most UBG formalisms allow the feature value to be either an atomic symbol (character or number) or another feature structure, just like Figure 2.3.

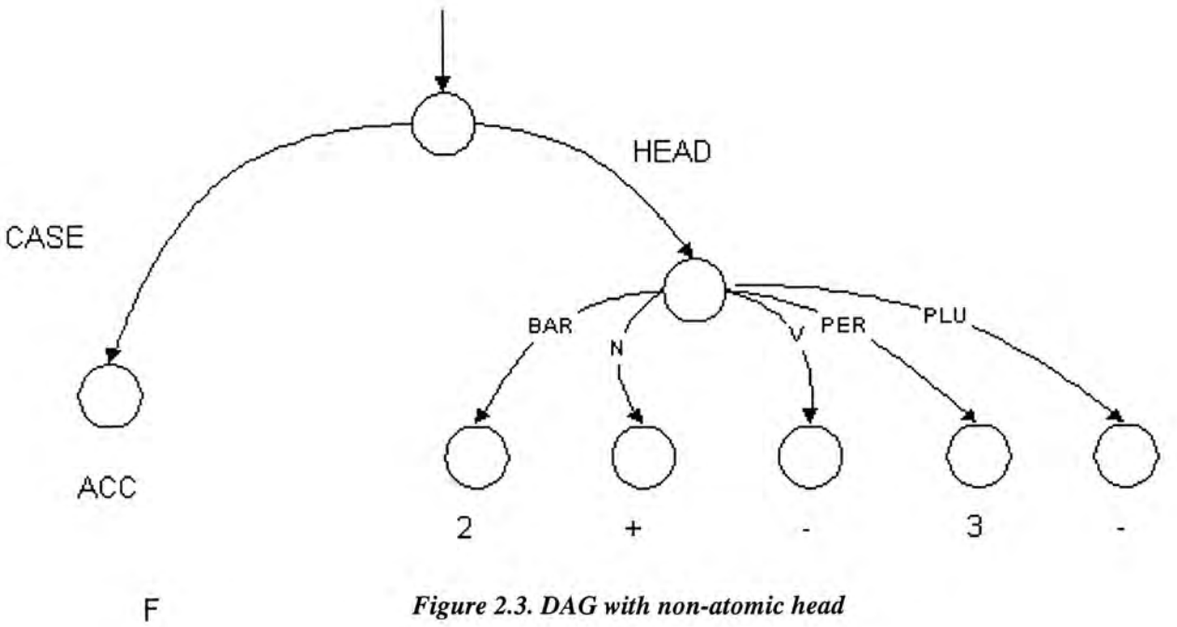
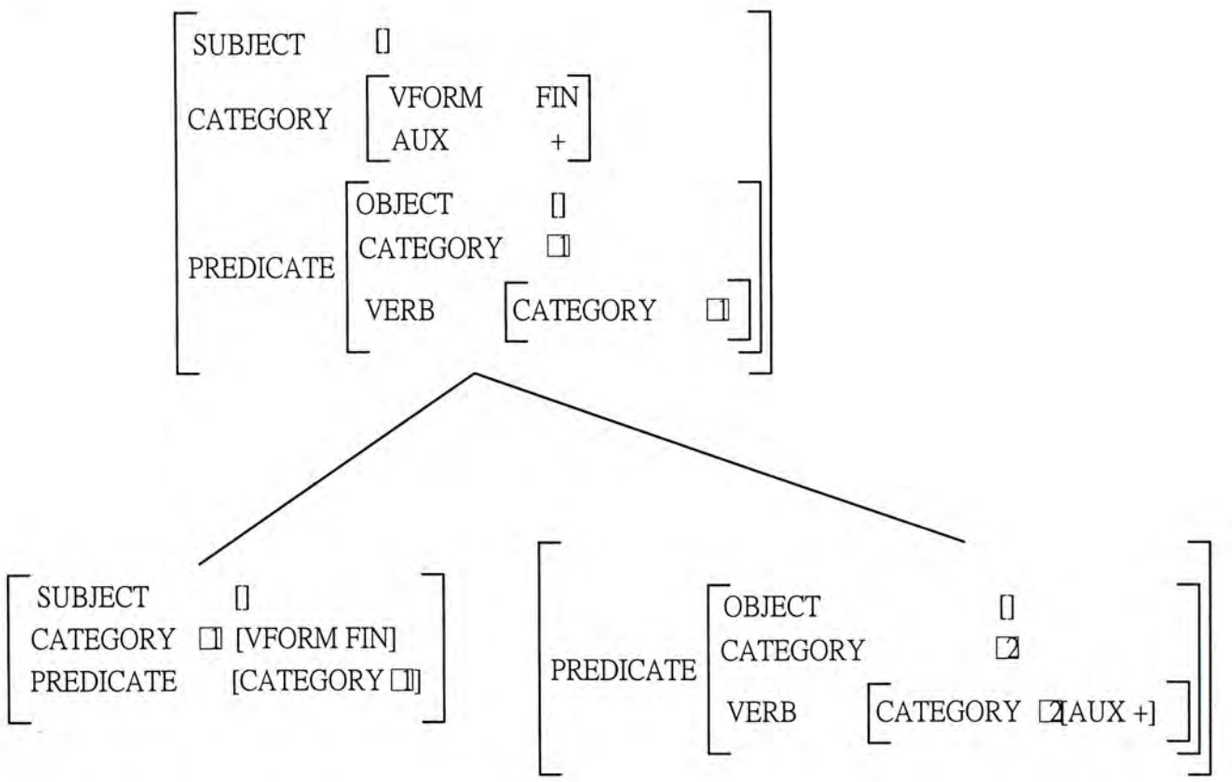


Figure 2.3. DAG with non-atomic head

Here HEAD dominates five other attribute-value pairs, they are [BAR 2], [N +], [V -], [PER 3] and [PLU -]. The advantage of such nested feature structures allow features to be grouped together, so that they can be referenced as a complete unit rather than discrete individuals. This allows a large piece of information to be shared during unification in a structured manner.

Before the emergence of unification-based grammar formalism, people believed that agreement, percolation, etc., could be performed by just transformational rules. In another words, the phrase-structure rules would generate the tree and then the transformations would copy features from one place to another [11]. The modern view, however, is that the tree can be realized as a big feature structure and each tree node represents a piece of information to be unified to the root node feature structure. (2) shows a typical scenario of feature structures unifying up the tree.

(2)



An important characteristic of feature structures is that they can be reentrant. A reentrant feature structure is one in which two features in the structure share one common value. In (2) we see two square boxes marked 1 and 2. They represent the common feature structure shared by the attribute CATEGORY. The unification process forced the squares boxes to unify and it finally leads to the root node structure. Detail foundation and operation of unification can be found in [9, 10].

The most influential unification-based grammar formalisms include Generalized Phrase Structure Grammar (GPSG) [12] and Head-driven Phrase Structure Grammar (HPSG) [13, 14]. These formalisms behave as a kind syntactic theory or syntactic-semantic theory (in the case of HPSG). According to each particular theory there is different treatment of constraint and the manner of information sharing, but they share a number of common advantages. First, they are declarative. That means the lexical information defines legal sentences or phrases. The change in lexical information does not affect the underlying processing procedures and results. Second, they are independent of parsing algorithms. That means improvement of parsers does not require a modification in the grammar and lexicon. Third, partial parse is allowed. At any moment in the unification processes, the accumulated information shows the partial result up to that moment even if the input is ungrammatical. That also implies that we can dig out the possible grammatical errors in a wrong input by checking each node in the feature structure tree.

All in all unification-based grammar formalisms provide a scientific, powerful and logical way for linguistic analysis. The output of these grammar formalisms is in the form feature structure (the same form as the lexicon). The information stored in it can

be readily used by applications, such as information retrieval and machine translation.

Here we will give a more detailed description of GPSG and HPSG. Please note that throughout this thesis the term "feature" is interchangeable with "attribute". There is no difference between them.

2.3.1. Generalized Phase Structure Grammar (GPSG)

Around 1980, Gerald Gazdar of the University of Sussex proposed the GPSG [12] formalism, aiming to restrain the power of transformational grammar, yet providing an account of linguistic structure universal to all languages.

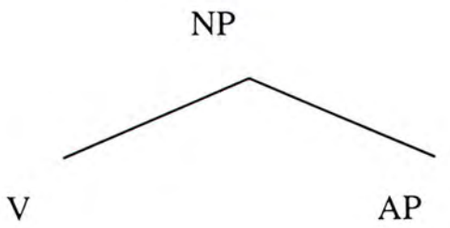
Key characteristics of GPSG are:

1. Context-free based X-Bar syntax
2. Grammar rules are written in the form of Immediate Dominance rules and Linear Precedence statements (ID/LP)
3. Restricted feature/value system as informational domain
4. Use both simply-valued features (e.g. NUMBER, CASE) and complex-valued features (e.g. SLASH, REFL)
5. Succeeded in dealing with the subtleties of coordination and long-distance dependencies

What is significant about the adoption of X-Bar theory in GPSG is that it dictates which kind of phrase structure rules GPSG can possess. The X-Bar theory says that constituents above the lexical level are to be seen as projections of lexical categories

[15]. This means that the category of a phrase must match that of its head. For example, a phrase built round a noun is automatically a noun phrase. As a result, subtrees such as (3) (described by the rule (3a)) are excluded.

(3)



(3a) NP → V AP

The ID/LP format for grammar is also unique to GPSG. A familiar phrase structure like:

(4) S → NP VP

is expressed in two rules (4a) and (5) in GPSG.

(4a) S → NP, VP (Immediate dominance rule (ID-rule))

(5) NP < NP (Linear precedence statement (LP-statement))

(4a) is known as ID-rule because it tells us S can immediately dominate NP and VP. (5) is known as LP-statement because it tells us that the NP precedes the VP when they are sisters to each other. GPSG believes that languages in the world share more or less the same ID rules, i.e. the constituents of forming phrases and sentences are the same, while the only difference is their word order, as stated by the LP-statements.

Features sharing principles form the most crucial part of the whole GPSG theory. They are the Head Feature Convention (HFC), the Foot Feature Principle (FFP), the Feature Co-occurrence Restriction (FCR), the Control Agreement Principle (CAP) and the Feature Specification Default (FSD). Luckily the principles' names are already clear enough to understand, so we will not explain them one by one. Anyway these many feature principles restrict the occurrence and the flow of features in the tree built according to the ID/LP format grammar. For example, (6) is a GPSG rule

(6) S → X^2, H[-SUBJ]

that legitimizes the structure with S being the mother node of X^2 (X can be any category) and H[-SUBJ] (H means the "head". The head of S is a VP.). The features on each node in the tree have to be unified with the features on the corresponding node in the rule. At any time, the feature principles must be strictly obeyed during tree formation. Otherwise the resulting tree will be illegal. How this is done? First we transform (6) back to its feature structure representation (7).

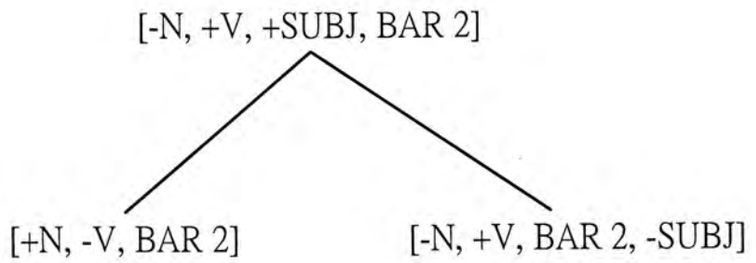
(7) [-N, +V, BAR 2, +SUBJ] → [BAR 2], [-N, +V, BAR 2, -SUBJ]

The feature [+SUBJ] is obligatory for the symbol S as stated by the rule. X^2 can be any element with [BAR 2]. The symbol H stands for head in the rule, and by the Head Feature Convention (HFC), the rule head H must share the same head features[1] with

[1] In GPSG, features are divided into head and non-head. The HFC only governs the percolation of head features. N, V, BAR and SUBJ are all head features.

the terminal element of the rule, unless there are other constraints in the rule. Here the only outstanding feature is SUBJ, so the HFC will count it out during features unification. (8) illustrates the scene.

(8)



The tree is then checked against the LP-statements and the rest of feature principles to determine the validity of it.

The drawbacks of GPSG are that it makes use of a large number of rules, which affects parsing efficiency and theory clarity. Also, principles governing feature/value system are complicated, for instance there is no rule telling the sequence of when to apply which rule to check the well-formedness of trees. Moreover, semantics of words are separated from syntax. The expressive power is bounded by CFG as well. Barton [16] claims that parsing GPSG style syntax may lead to computational explosion, because of the large number of rules and complicated feature/value system. Many GPSG parsing algorithms had been proposed, and only few of them retain 100% of the original GPSG description [15].

Nevertheless, GPSG provides a straightforward way to express information and capture generalizations of human languages. As a result, it is considered as the best grammar formalism for describing a wide coverage grammar [15].

2.3.2. Head-driven Phrase Structure Grammar (HPSG)

Carl Pollard and his colleagues of Hewlett-Packard developed the HPSG formalism during a research into implementation of GPSG [13]. Based on GPSG and head grammars and inspired by the lexicalism movement, HPSG takes further advantage of the power of unification. HPSG allows only concatenation, replaces the metarules of GPSG completely with lexical rules, and removes many of the restrictions yielding finiteness of the informational domain. Pairings of strings are determined by a bottom-up rule application algorithm, which makes it a procedural formalism. Subcategorization and semantics are embedded in the lexicon, i.e. allowing parsing with both syntactic and semantic information at the same time, which greatly enhance the efficiency of the parsing process. In GPSG, the lexicon contains only limited information as most important constraints are left in the rules. HPSG instead encodes most constraints in the lexicon, thus avoiding the complicated feature principles. Moreover, HPSG lexicon has an inheritance-based organization, which makes it a more powerful theory.

There are some rules and principles in HPSG, but unlike GPSG, they are all in the form of feature structures. That means, unification is the only operation needed in HPSG. Whether a final feature structure is valid or not is simply meant by unifying all feature structures of the constituents together with the rules and principles.

Three major principles of HPSG are [13]:

(9) Head Feature Principle

$$[\text{DTRS}_{\text{headed-structure}}[\,]] \Rightarrow \begin{bmatrix} \text{SYN|LOC|HEAD} & \boxed{1} \\ \text{DTRS|HEAD-DTR|SYN|LOC|HEAD} & \boxed{1} \end{bmatrix}$$

(10) Subcategorization Principle

$$[\text{DTRS}_{\text{headed-structure}}[\,]] \Rightarrow \begin{bmatrix} \text{SYN|LOC|SUBCAT} & \boxed{2} \\ \text{DTRS} & \begin{bmatrix} \text{HEAD-DTR|SYN|LOC|SUBCAT} & \text{append}(\boxed{1}, \boxed{2}) \\ \text{COMP-DTRS} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

(11) Semantics Principle

$$[\text{DTRS}_{\text{headed-structure}}[\,]] \Rightarrow \begin{bmatrix} \text{SEM} & \begin{bmatrix} \text{CONT}_{\text{successviely-combine-semantics}}(\boxed{1}, \boxed{2}) \\ \text{INDICES}_{\text{collect-indices}}(\boxed{3}) \end{bmatrix} \\ \text{DTRS} & \boxed{3}\begin{bmatrix} \text{HEAD-DTR|SEM|CONT} & \boxed{1} \\ \text{COMP-DTRS} & \boxed{2} \end{bmatrix} \end{bmatrix}$$

Grammar rules in HPSG are considered a very partially specified phrasal sign which constitutes one of the options offered by the language in question for making big signs from little ones [13]. In fact, the rules in HPSG are schematized from many conventional phrase structure rules. Here is an example ((12a) is in the standard "rewrite" notation):

(12) Rule 1

$$\begin{bmatrix} \text{SYN|LOC|SUBCAT} < > \\ \text{DTRS} \begin{bmatrix} \text{HEAD-DTR|SYN|LOC|LEX} - \\ \text{COMP-DTRS} <[\ ]> \end{bmatrix} \end{bmatrix}$$

(12a) [SUBCAT < >] → H[LEX -], C

This says that any item with an empty SUBCAT list can consist of a non-lexical head (e.g. a phrase) and a complement. This covers not only an S consisting of an NP and a VP, but also an NP consisting of an article or an NP plus an N1. While in GPSG, rules having such underlying structure must be explicitly written down. Other rules in HPSG include inverted sentence, adjunct rules, etc.

This makes HPSG a lexical theory, while GPSG remains a syntactic theory. However, although HPSG differs from GPSG in many aspects, it is very greatly indebted to GPSG [15].

2.3.3. Common drawbacks of UBGs

The drawback of unification-based grammar formalism is that there is no efficient parsing algorithm to ensure high efficiency. Earley [17] proved that context-free grammar parsing is $O(n^3)$. This can be a huge number in the case of heavy ambiguities, as in the case of human languages. Many unification-based grammar formalisms have a context-free grammar backbone, so they also suffer from that. Furthermore, unification or feature-structure sharing is a very expensive operation. In fact, unification-based parsing is so costly that only very few commercial natural-language

processing systems are based of the technique. The German Verbmobil project is using XPSG (an extended version of HPSG) as the model of their deep information processing module [6]. Despite its analytical power, it has been criticized for its slow response.

Another critical problem is that UBG remains a theory more than a practical method. It assumes what you fed into the formalism must be correct. It also assumes that the lexicon contains all necessary information. So in other words, a UBG aims at "correct input correct output". It does not care about input that does not match its required format. This can be a problem in practical application, where we can find many non-literal signs, and probably a lot of novel word compounds not found in the lexicon.

2.4. Corpus-based processing

Statistical techniques were first used by the speech-recognition community to predict the next word in utterance on the basis of the words recognized so far (the Hidden-Markov Model). However, purely statistical techniques without linguistic understanding were proven vulnerable. As a result, researchers have been trying to combine traditional artificial intelligence-natural language processing techniques with statistical processing.

Generally statistical techniques can be concluded as fast and flexible. They are fast because only simple mathematical operations are involved, there is no costly feature-structure copying as in unification-based grammar processing; they are flexible because they can be applied to almost every job. Not only they can be used in

speech recognition, they can also be fitted into a number of linguistic tasks like parsing, disambiguation and machine translation. In recent years statistical processing has become the dominating stream in natural language processing.

The foundation of statistical processing is the probability theory and statistical models, like Neural Network (NN), Hopfield network, Hidden Markov Model (HMM), Bayesian network and genetic algorithm. For these models to work, we must have a corpus. The corpus is a large set of text the represents the set of linguistic information that the computer can look into. The text is usually tagged with linguistic information of the source language. Probabilities of the occurrence and co-occurrence of words and other kinds of relationship can then be trained and obtained from the corpus. These probabilities are useful to a number of linguistic tasks like parsing and disambiguation. Charniak [18] and Mann [19] give a full account of the foundations and application areas of statistical processing.

2.4.1. Drawback of corpus-based processing

Corpus-based processing suffers from the fact that no statistical models up to date can truly imitate the human brain. Even neural network is just a simplified model of what is believed to exist in human brain. This implies that statistical processing can never achieve perfect accuracy (in real practice it is even far from getting reasonable result) unless we found a computational model that can be compared to the human mind. Moreover, unlike unification-based grammar that performance can be improved by gradual refining the grammar and lexicon, a statistical model usually has an upper limit of accuracy.

Another drawback is that statistical processing depends on the corpus it works on. As the corpus represents the domain of linguistic knowledge known to the computer, it is clear that the quality and size of the corpus is crucial. But how to evaluate the corpus and whether a bigger corpus would yield better result are still under critical questioning.

3. Difficulties in Chinese language processing and its related works

3.1. A glance at the history

China was among the first few countries that started machine translation research in the 1950s. In the past 20 years, natural language processing has become a very popular research area in Mainland China and Taiwan. However, despite much effort in the period, hardly any result could be compared with the European countries' counterpart. At first people argued that the difference in manpower and time involved accounted for the difference. For instance, researchers working on European languages outnumbered those working on Chinese. Gradually, nevertheless, people realized that a big gap was formed mainly because the research results on European languages did not work on Chinese. It is the difference of languages that matters.

3.2. Difficulties in syntactic analysis of Chinese

No two languages on the earth are the same. Therefore, although the principles of natural language processing are shared among languages, the details of implementation and areas of emphasis can differ a lot. Chinese, a Sino-Tibetan language, is considered an isolating language (though not strictly isolating), and it has its own writing system. In addition, Chinese has many unique characteristics that make syntactic analysis, i.e. the important first step of language understanding, particularly difficult.

Languages exist on earth can be generally divided into three categories. They are isolating languages, inflectional languages, and agglutinating languages. Chinese, as mentioned above, is an isolating language. Examples of the other two categories are English[2] and Japanese respectively. I will explain the problems faced by Chinese with respect to English and Japanese below. The account is concluded from Yu [20]. The difficulties of Chinese syntactic analysis cannot be fully demonstrated without a detailed comparison between these three categories of languages.

3.2.1. Writing system of Chinese causes segmentation problem

Since ancient time Chinese texts have been written without space between characters. The Chinese writing system had a reformation in modern time, with paragraph, punctuations added to ease the reading task. However, modern Chinese is still written without spaces between characters in the same sentence. Ambiguities often arise because we do not know the boundary of a multi-syllabic word, or we simply do not know whether a character exists as a monosyllabic word or part of a multi-syllabic word. This is commonly known as segmentation problem. Below is a good example.

(13) 從前門口走過的人很多。 [21]

(Many people walking pass the front door.)

---

[2] Although modern English is virtually without inflection and so it is arguable whether English is a proper representative of the inflectional language family, it still possesses inflectional characteristics. As I do not have knowledge in another Indo-European language that is truly inflectional (e.g. Russian), I am forced to confine to English.

Scanning from left to right (normal Chinese reader's manner), we have two possible segmentations[3]:

(13a) *從前 門口 走過 的 人 很多。

(Long ago doorway walked pass people many)

(13b) 從 前門口 走過 的 人 很多。

(From front door walked pass people many)

Native reader would immediately cross out (13a), because it is semantically wrong. So the correct segmentation is (13b). However, the computer does not know it, unless there is a way to instruct the computer that "門口" (doorway) can not be followed by "走過" (to walk pass).

What's worst, sometimes there can be more than one correct segmentation, and the resulting sentence meanings differ. Look at this example.

(14) 太陽能使水變熱。[22]

(Solar energy heats up the water/The Sun can heat up water)

(14a) 太陽能 使 水 變熱。

(Solar energy makes water becomes hot)

---

[3] The number of possible segmentations, in fact, is more than two, because Chinese natives have a set of morphological rules to make compounds.

(14b) 太陽 能 使 水 變熱。

(The Sun can make water becomes hot)

Both (14a) and (14b) are interpretable segmentations, and even for a native it is impossible to declare which one should be taken as the correct interpretation unless discourse and context knowledge is available.

As a result, segmentation problem in Chinese induces much workload to the computer. It also makes language analysis difficult and error-prone. The fact that no perfect word segmentation can be guaranteed turns out to be a decisive reason why Chinese language processing lags behind the European counterpart.

English, for instance, does not have such segmentation problem because the writing system of English ensure there is a space between every two morphological compounds, or simply words. Japanese text, although written in the same manner as Chinese, i.e. written serially character by character without space, can virtually resolve all segmentation problems. Japanese texts are generally mixed with Chinese characters (Kanji) and Japanese characters (Hiragana & Katakana). They provide clear clues for segmentation. Also, there are particles in Japanese that distinguish the grammatical function or each constituent. This is not the case in Chinese.

3.2.2. Words serving multiple grammatical functions without inflection

We can find the part-of-speech (POS) of a word in English or Japanese dictionaries easily. Or we can tell it from the morphological characteristics of the words in

question. For example, the suffix "-ation" and "-ness" signify noun, almost all words with the first alphabet capitalized must be proper noun. In Japanese, "-ます" ending signifies verb, words ending with "-い" are mostly adjectives.

In inflectional and agglutinating languages, words change in form according to the word's part-of-speech. In English, nouns can be divided into singular and plural by judging the "-s" or "-es" attaching them; personal pronouns change according to their case (e.g. I, me, my). Japanese does not have morphological changes for nouns, but there are case markers (particles) to do the job. For example, "-が" signifies subject, "-は" signifies topic, "-を" signifies object, "-の" signifies adjunct, etc. Furthermore, there is a clear relationship between the part-of-speech and grammatical function in these languages. In Indo-European languages, one part-of-speech can only serve as one grammatical function. For example, in English a verb generally acts as predicate, noun acts as subject or object, adjective acts as noun modifier and adverb acts as adverbial. However, such relationship is blurred for Chinese. In Chinese, nouns, verbs and adjectives are considered multi-functional [23].

English verbs in gerund form can take subject and object positions; when they are in past participle form they may act as modifier. So arguably one can say English verbs are multi-functional. However, when an English verb takes up a function rather than predicate, it must go through morphological changes. On the other hand, Japanese verbs have different endings for different grammatical functions.

Unlike English and Japanese, however, Chinese verbs do not go through morphological change whatever grammatical slots they are taking. This applies to other part-of-speech in Chinese as well. Here is an example.

(15) 紅太陽從東方昇起。 (The red sun rises from the east.)

(16) 太陽漸漸紅起來。 (The sun is reddening gradually.)

(17) 紅是太陽的顏色。 (Red is the color of the sun.)

In (15), (16) and (17), "紅" (red) is an adjective, a verb and a noun respectively. However, in all three cases, it does not take any morphological change at all. Moreover, in (16), the adjective "紅" (red) serves as predicate, in (17), it serves as the subject.

Summing up the points in this section, we see that during parsing both inflectional and agglutinating languages, the part-of-speech of words are already available. It means these two types of languages have a firm basis for syntactic analysis. Chinese, on the other hand, is very difficult to parse because of multi-functional words and the lack of morphological changes.

3.2.3. Word order of Chinese

As there is a lack of morphological change in form, and there are no particles to distinguish different grammatical functions, the word order becomes important in sentence formation in Chinese. Chinese was commonly recognized as a SVO (Subject-Verb-Object) language, while English and Japanese are SVO and SOV (Subject-Object-Verb) languages respectively. However, for the expression "I have finished homework" there can be three equally common Chinese equivalent expressions.

(18) 我做完了家課。 (SVO)

(19) 家課我做完了。 (OSV)

(19) 我家課做完了。 (SOV)

The flexibility of word order and the lack of morphological changes in Chinese thus bring difficulty in parsing and rule description. Particularly the later fact implies that purely syntactic knowledge cannot distinguish SOV and OSV.

3.2.4. The Chinese grammatical word

Xuci ("虛詞") (grammatical word) are of particular importance in Chinese due to the fact that Chinese lacks morphological changes. They are used extensively to identify the grammatical functions of a word or phrase. For example, "的" is used to transform predicate to nominal.

(20) 喝茶 (To drink tea/Predicative)

(21) 喝的茶 (Tea we drink/Nominal)

(22) 喝茶的 (The one who drinks tea/Nominal)

"的" is also used extensively as possessive or modifying particles, for example:

(23) 我的父親 (My father)

(24) 我的書包 (My bag)

However, the problem in Chinese is that whether to use particles is indeed very flexible. For example for (23) and (24) above we delete the particle "的" but still

yielding the same meaning, as in (25) and (26).

(25) 我父親 (My father)

(26) 我書包 (My bag)

The aspect markers "著" "了" "過", adverb "將" and time adverb "將來" are used to indicate tense in Chinese. However, in many occasions they can be omitted. For example the clause "我吃飯" (I eat/ate rice) can be the answer of (27) – (30).

(27) 昨天晚上你做什麼? (What did you do yesterday night?)

(28) 明天下午你做什麼? (What will you do tomorrow afternoon?)

(29) 剛才你做什麼? (What were you doing just before?)

(30) 你在做什麼? (What are you doing?)

That means the clause "我吃飯" (I am eating/ate/will eat/have eaten rice) can represent past, future, past perfect and progressive tense. Without knowing the context where the utterance is made, it would be impossible to determine the tense.

"被" (bei4) and "把" (ba2) sentences are often considered classical sentence structures in Chinese [24], however in real they can be omitted as well. For example, (19) can be viewed as (31) with "把" deleted.

(31) 我把家課做完了。 (I have finished the homework.)

The omission of the originally available syntactic clues given by xuci introduces extra difficulty in parsing. Moreover, many xuci have root words counterpart, so it is not easy to distinguish when a word is grammatical or root. For instance, the characters

"在", "給", "跟" are particles as well as verbs.

In English and Japanese, particles cannot be omitted, such that parsing can be performed with a reliable syntactic clue.

3.3. Related works

Serious research in Chinese computational linguistics had a rather late start. Despite the early research in machine translation in Mainland China back in the 1950s, theoretic research was rarely seen until the 1980s. Nowadays researchers of Chinese NLP mostly come from Mainland China, Taiwan, Singapore and the United States.

As influenced by the Western experience, Chinese NLP researches are also divided into deep and shallow information processing.

3.3.1. Unification grammar processing approach

3.3.1.1. Logic-based parsing

Chen [25] uses Prolog to design a logic-based Chinese parser that performs segmentation and syntactic parsing at the same time. In order to deal with the maximal freedom of empty categories in Chinese, principles in government and binding [26], e.g. C-Command and subjacency conditions are embedded implicitly in the integrated segmentation-parsing model to decide which constituents are moved and/or deleted. Grammar rules are in PATR-II [9] style.

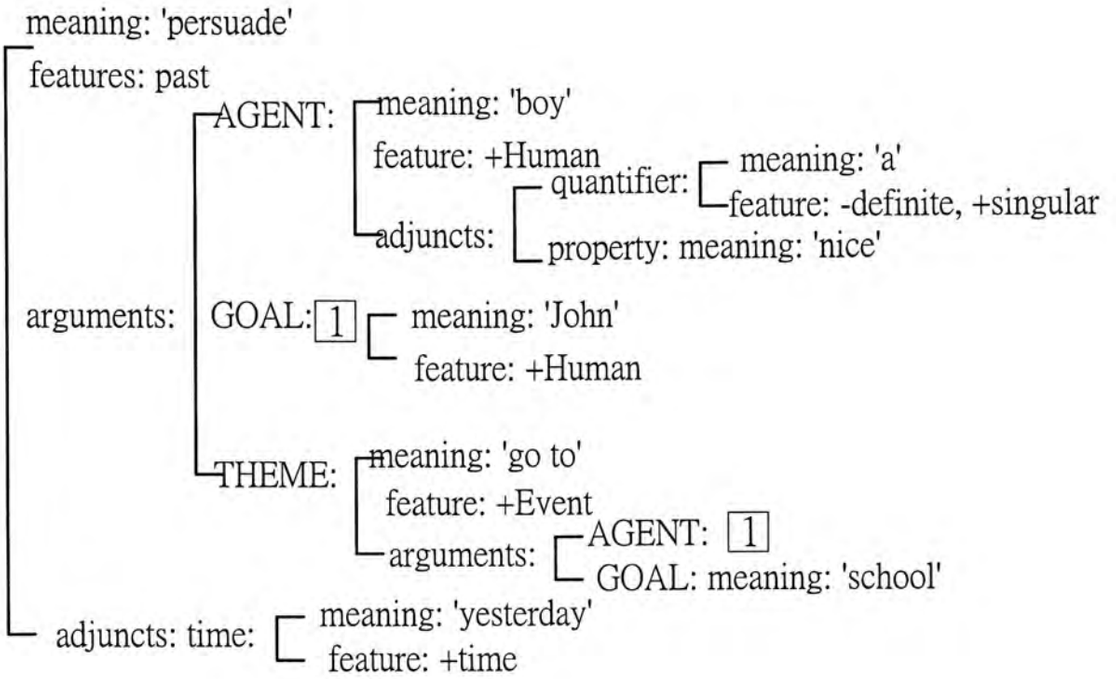
Chen's paper is the first of its kind in the Chinese NLP literature. Its contribution is

more on the theoretical side than the engineering side. For example, logic-based segmentation is slow and cannot be compared with statistical segmentation. Logic-based parsing is usually too slow for practical use. The paper successfully points out the significance of using unification-based approach in Chinese parsing, but it does not provide a clear description of the model in use.

3.3.1.2. Information-based Case Grammar

Since 1988 Academia Sinica of Taiwan has been seriously involved in devising the Information-based Case Grammar (ICG), a new conceptual representation for Chinese parsing [27]. ICG is a lexical-based grammatical formalism that combines principles in case grammar [28], GPSG and HPSG. It has a context-free backbone, grammar rules in GPSG's ID/LP style, and features percolation principles in GPSG and HPSG. What's new in ICG is that syntactic and semantic constraints on grammatical phrasal patterns are encoded in thematic structures. The feature structure of a potential phrasal head denotes partial information for defining the set of legal and grammatical phrases. It also provides enough information to identify the thematic roles for arguments and adjuncts. As a result, parsing and thematic analysis are achieved simultaneously in ICG. For example, the sentence "A nice boy persuaded John to go to school yesterday." has the feature structure represented by (32).

(32)



It can be seen that parsing in ICG is the identification of thematic roles. The semantic features would be unified during parsing while the syntactic features are no more than constraints guiding appropriate unification [27].

ICG looks good as formalism for Chinese in that it points out the fact the Chinese is a weakly marked language with little inflection and syntactic-only representations would cause tremendous ambiguities. So semantic information is indispensable in parsing. In ICG, semantic filtering and thematic roles identification depend on selectional restrictions, a technique works alongside with semantic network in artificial intelligence (AI). However, as we will further discuss in chapter 6, that selectional restrictions never guarantee closure and thus semantic filtering using it is not adequate, particularly in formalism like ICG that aims at representing Chinese language in all domains.

Moreover, papers and reports of ICG do not talk about how ICG interacts with segmentation, and its use in larger context. Furthermore, ICG does not deal with the vagueness of Chinese sentences. This means although ICG looks a clean formalism, it does not attempt to handle some long-standing linguistic problems.

3.3.1.3. HPSG

From 1991-1994, Lee et al. [29-31] published a series of papers describing the use of HPSG in Chinese parsing. They use HPSG to solve nominalization problem in Chinese, to verify segmentation results and to employ it as the core of their Chinese-English MAchine Translation (CEMAT) project. Unlike the ICG, they do not give a clear description of how they fit Chinese into the HPSG framework. Moreover, their solution to the nominalization problem is rather ad hoc. Instead of fitting the problem into their HPSG interpretation with a clear explanation of the new features added and its interaction with the principles of HPSG, they simply describe how the feature will affect the parsing algorithm. Discussion of how they handle Chinese syntax with HPSG is also missing. It is hard to convince the reader that their work is successful.

3.3.2. Corpus-based processing approach

3.3.2.1. Statistical segmentation

Chiang et al. [32], Sproat and Shih [33], and many others have employed statistical method to Chinese word segmentation. The core of their methods is the same. Score is given to every segmentation result. The one with highest score will be taken as the final segmentation. Sproat and Shih [33] even attempt to identify Chinese names and

transliteration in the text. All statistical segmentation researches report accuracies of around 95% in closed test and around 80% in open test.

3.3.2.2. Probabilistic parsing

Yao & Lua [34] attempt to parse Chinese with a probabilistic context-free grammar (PCFG) parser. In PCFG, grammar rules are associated with statistic probabilities that are obtained through a training stage. During sentence parsing, grammar rules specify the structures allowable in the language, while probabilities specify the distributional regularities of sentence structures in the language.

The PCFG parsing procedure is composed of two parts. First, grammar parsing to find out all possible parses which fit grammar rules. Second, searching the parse tree space formed by the above step to find out the best parse that has the maximum parse probability.

Given the highly ambiguous nature of Chinese grammar, a PCFG parser that works only with syntactic information (to be exact, part-of-speech information) is deemed to fail. Moreover, it will always favor rules with higher probabilities. That means rules with low probabilities will not be chosen when they are competing with high-probability rules. This results in low accuracy of the parser.

3.3.2.3. Example-based parsing

Example-based parsing is a technique directly inherited from example-based translation. It has been found that over 60% of the sentences in domain-specific texts follow almost the same patterns [35]. When the human translator going over the

corpus of representative texts, he/she should be able to identify some sentences that can be used as examples in the system. This means that no rules are needed to translate these sentences as the method of pattern matching can be applied to produce the translated text. When the text is translated more by examples and less by rules, then the translation would be syntactically highly accurate.

The same technique can be employed over source language analysis and target language generation. Liu & Zhou [36] and Wu [37] use example matching with linguistic heuristics to replace the traditional syntactic analysis in machine translation. In their implementation, source sentence are ranked according to the similarity measures between itself and each example in the database.

Example-based parsing suffers the same as PCFG in that high-probability examples are always chosen in favor of low-probability one. Although statistical parsing enjoys a much faster speed and coverage than unification-based grammar parsing, the accuracy is not guaranteed.

3.4. Restatement of goal

As described in chapter 2.2.1, in natural language processing (NLP), the rational unification-based grammar (UBG) processing approach allows deep information processing, and is widely believed to be the most suitable linguistic representation for a number of NLP tasks that require strong language understanding like machine translation. However, there are some unsolved issues. First, by theory a unification model cannot handle irregular input. It assumes that any word appears in the input text can be found in the lexicon. If not, the parser fails every time new words appear.

However, in real practice we will face many words that do not appear in the lexicon See (33):

(33) 邵逸夫 於 10/10/2001 捐建 了 此 樓。

(Sir Run Run Shaw donated to the completion of this building on 10/10/2001.)

The first compound is a Chinese human name. The second underlined compound is a date in Arabic number notation. A dictionary (or lexicon) must not have such entry, because arbitrary numbers, date and time are derivable from a basic set of elements. The third underlined compound is a typical disyllabic verb in Chinese. The number of such verbs is infinite, and a dictionary will have most such verbs unstated. This means that if a unification-based grammar parser is to be used in real application, we must have means to deal with the unknown word problem.

The second issue is that the unification-based grammar formalisms are originally developed particularly for English and other Indo-European languages. They do not share the same problems faced by Chinese which was discussed in chapter 3.2. For instance, segmentation of sentences into words is a vital part of Chinese language processing and it directly affects the design of parser. Previous works in this area often treat segmentation and parsing two independent issues. We suggest that, however, the parser must be able to deal with any correct (we have seen that multiple correct segmentations are possible) segmentation output. Also, the notion of Chinese sentence is confusing enough, as it cannot be determined exclusively with syntactic terms [38]. If we simply feed the text between two full stops to the parser, we may not get a correct output. This means we must find method to choose a suitable parsing unit.

In this thesis a robust UBG parser for Chinese NLP will be presented. It can be viewed as hybrid of both UBG and statistical processing. The ultimate of it is to give correct parsing result for all grammatically correct Chinese sentences. It consists of a UBG backbone, with a robust statistics-driven automatic preprocessing procedure that segments and standardizes raw texts to allow the parser to be used in practical applications. Apart from phrasal rules, the parser also contains morphological rules so that multiple segmentation results problem discussed above can be relieved. It also tries to identify novel word compound in the input texts before parsing takes place with statistical method, such that the parser will not fail easily. Furthermore, we clarify the concept of "Chinese sentence", which is important for discourse reconstruction from discrete clauses. This is of particular importance to Chinese, a language that linguists consider without clear notion of "sentence".

The parser proposed in this thesis links segmentation, parsing, semantic disambiguation and discourse reconstruction as one unit. Its contribution to the engineering community is that it describes a way for utilizing and improving the quality of a unification-based grammar parser with statistical manipulation in practical application. Its contribution to the linguistic (i.e. scientific) community is that it incorporates recent advancements in Chinese linguistics into its design.

We believe that hybrid language processing will become the dominating method in NLP research. For a sophisticated NLP application we need a unification-based grammar formalism to store linguistic knowledge, but we also need corpus-based method to mediate the problems remained. Our parser is the first of such kind in Chinese NLP and will become a springboard for more sophisticated formalisms and

applications.

4. SERUP: Statistical-Enhanced Robust Unification Parser

In chapter 3 we have talked about the difficulties of Chinese language processing, the methods used and what we want to achieve. As a unification-based grammar (UBG) parser concerns only about the well-formedness of the input but not the format of the input, it will fail every time there is something unknown to the parser. This include numerical values, date and time compounds in Arabic notation, and words that do not appear in the dictionary. So what does it imply is that to improve the robustness of the UBG parser is to try handling the problems before parsing is performed, apart from the necessary grammar rules refinement.

The job includes segmentation of lexical tokens (an unavoidable process in Chinese), segmentation of smallest parsing unit (the deal with the vagueness of Chinese sentence), conversion or removal of strange tokens and identification of new words, etc. To deal with the problem of novel word compounds, we will also present a simple but effective word identification algorithm. Moreover, we will argue the traditional viewpoint that "sentence as parsing unit". Latest linguistic researches have shown that the syntactic notion is not compatible to Chinese. We will describe a mechanism that allows reconstruction of discourse from discrete clauses. In addition, we will go through grammar construction with UBG in our implementation, which is the core of later stages' semantic and discourse processing. Our system works in accord to the flowchart (Figure 4.1).

The input source text first undergoes segmentation and new words identification. By

then the input text should contain only words that can be recognized by the parser. In another words the parser should be able to directly parse or form morphological compounds with the segmentation results. All words that are not combinable with morphological rules should have been identified and combined up to this point.

The next step is part-of-speech (POS) tagging, which is needed to identify the smallest parsing unit. The smallest parsing unit in our framework is defined as a verb phrase or a verb phrase with its subject. The reason for this will be explained in detail in chapter 5. Our POS-tagger uses hidden Markov model (HMM) so there is a need of training corpus.

After separating the input text into relevant parsing units, they can be sent to the unification-based parser. Currently we are using PC-PATRII as the unification engine. The reason for choosing it will be explained in chapter 8. For PC-PATRII to work, we need to provide two files, one for lexicon, one for the grammar. The parsing output will be an analyzed structure of the input in the form of feature structure.

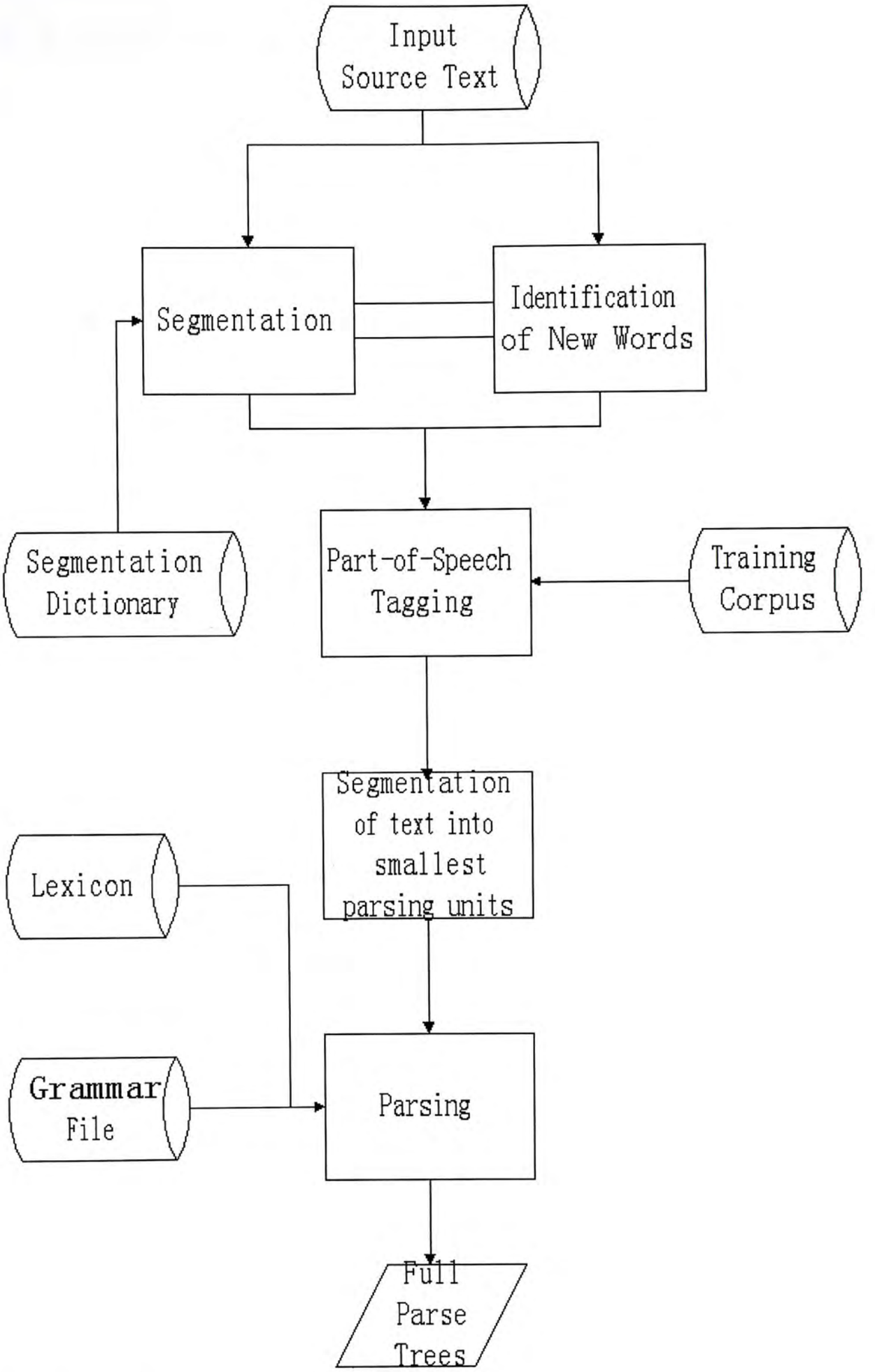


Figure 4.1 System flowchart of SERUP

5. Step One: automatic preprocessing

Automatic preprocessing is the first and the most vital part of SERUP. Only a successful preprocessing module can ensure the most number of sentences getting a correct parser in the unification-based grammar parser.

The procedures include segmentation of lexical tokens, date, time and numerical conversion, identification of new words, and segmentation of smallest parsing units.

5.1. Segmentation of lexical tokens

Segmentation of lexical tokens is an unavoidable process for languages without space delimiters between words or word compounds. For example, Chinese and Japanese do not have spaces at all. In chapter 3.2.1 we have discussed in detail what is the problem and how the problem of segmentation is crucial to Chinese language processing. Some people suggested that segmentation might not be an essential process. Instead of segmenting the text, we only need to search through the sentence and locate keywords. However this is wrong. If we do not know the boundary of each constituent in the sentence, we cannot guarantee the keywords located are correct. In other words, we cannot tell whether a character sequence in the sentence should be a constituent in it or not until we have decided the boundaries of the remaining constituents. So segmentation is an inevitable process.

Chinese segmentation has received generous attention for over two decades and there are also quite a number of free segmentation services available on the Internet. Methods used in segmentation include heuristic-based (or rule-based) and

statistical-based. Chapter 3.3.2.1 has described effort in statistical segmentation. All these segmentation algorithms claim to obtain an accuracy of more than 95% on average, depending on the context. We argue that such figures do not bear much significance, if the segmented texts cannot be tagged/parsed correctly in subsequent stages. For example, the sentence

(34) 政府勸告市民切勿大量買入紅籌股。

(The Government advised her citizen not to buy in red stock in large amount.)

can be segmented as

(34a) 政府 勸告 市民 切勿 大 量 買 入 紅籌股。

(34b) 政府 勸告 市民 切勿 大量 買入 紅籌股。

If '大量' (large amount) and '買入' (to buy in) are in the segmentation dictionary, things will be way easier for the parser to get a correct interpretation. If not, the subsequent stages in the parser must be able to tell '大' (big) is the modifier of the noun '量' (amount), while '入' (in) is the complement of '買' (to buy), otherwise the parser is by no means robust. This implies morphological rules that combine characters into words are essential in the UBG parser.

The segmentation dictionary maintains a huge list of items that are used as the basis of segmentation. As the segmentation dictionary itself is independent of the segmentation program, a slight change in the dictionary can affect the resulting output. The key issue here is that what entry should we put in the dictionary. We can, of

course, just put terms appear in a general purpose Chinese words dictionary[4] (or Ci2dan3 詞典) to the segmentation dictionary. However many non-derivable technical terms (e.g. chemical terms and law terms) will not get correct segmentation by then. So in real our segmentation dictionary contains terms not just in general purpose dictionary, but also from technical dictionaries. Moreover, we are continuously adding human names, corporation names and geographical locations to the dictionary to deal with text from daily newspaper. Furthermore, we have also put a number of frequently appeared phrasal expressions in the dictionary. These expressions can be broken down into items locatable in the dictionaries mentioned above, and during segmentation they will have higher preferences over discrete words because of the maximum-match (MM) heuristic used. So in the segmentation dictionary we also have chunks. Chunk segmentation is a simple but effective way to achieve faster segmentation, and also reducing the likelihood of misinterpretation in later stages of processing. Table 5.1 shows some example of chunk entries.

Chunk entry	Part-of-speech	Constituent breakdown
一次過撥款	Verb	Adverb + Verb
已繳部分股款的股本	Noun	Adverb + Noun Phrase
以私人協約形式售賣	Verb	Prep Phrase + Verb
在正式交易時間以外的交易	Noun Phrase	Prep Phrase + Noun

Table 5.1. Examples of chunk entry in segmentation dictionary

SERUP makes use and improves a rule-based segmentation program by Lee et al. [40]. It differs from traditional segmentation programs that it does not screen a sentence from the first character or from the last character. Instead it looks for the monosyllabic

[4] Opposite to a word dictionary is a character dictionary that only lists the characters one by one.

morphemes and numeral in the sentence as the basis of segmentation. Heuristics are applied to the segments separated by these points. The flow of segmentation is shown in Figure 5.1. Heuristics used include classifier rule (CR), behind classifier rule (BCR), modified behind classifier rule (MBCR), dictionary classifier rule (DCR), single rule (SR), modified single rule (MSR) and numeral rule (NR).

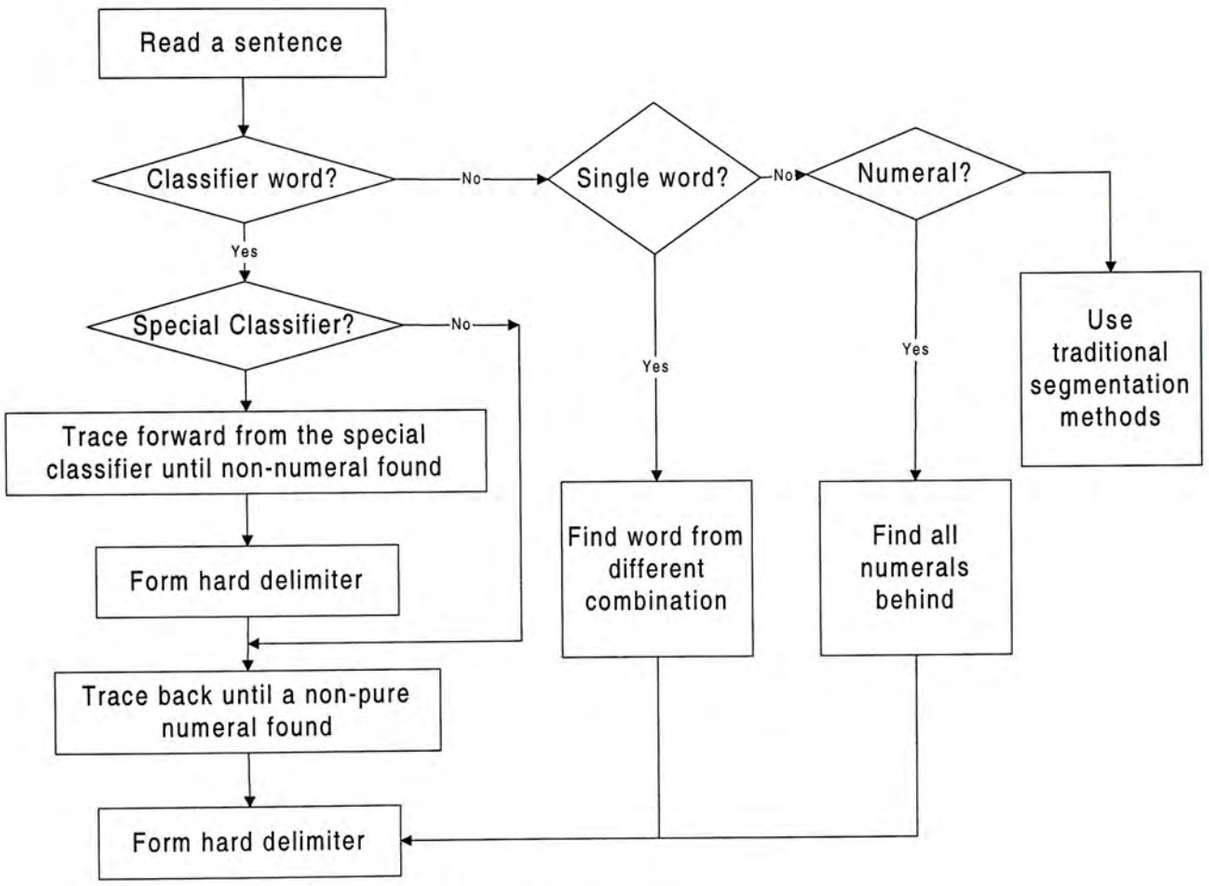


Figure 5.1 Flowchart of the segmentation process

The last words here are about the output of the segmentation result. The segmentation program tries to add hard delimiter for all terms appear in segmentation dictionary. But how about the terms not found in dictionary? In this case, those terms will appear as a sequence of space-separated characters, e.g.

(35) 中國 健力寶 集團 的 股價 上升 了。

(The stock price of Chinese Jianlibao Group has risen.)

If the organization name"健力寶" is not found in the segmentation dictionary, it will be segmented as three individual characters "健", "力" and "寶". So the output will be

(35a) 中國 健 力 寶 集團 的 股價 上升 了。

For (36), if the disyllabic verb"跳投" (to jump shoot) is not found, the output will be (36a).

(36) 喬登 又 跳投 得手。 (Michael Jordan scored again from jump shoot.)

(36a) 喬登 又 跳 投 得手。

SERUP will then depend on the new word identification algorithm to help grouping the characters into words.

5.2. Conversion of date, time and numerals

In real texts, there exist two writing systems for expressing date, time and numeric

compounds. Some use Chinese characters (e.g. 一時三十分), some use Arabic characters (e.g. 1:30, 13:30), and some use Arabic characters plus 'AM/PM' (e.g. 1:30am). Before passing anything to the tagger/parser, we have to convert these three systems into one universal representation. In SERUP, all time compounds are converted to the standard 24-hour notation. Date compounds, such as '五月二十四日', '一九九八年六月十五', '17-05-2000', '05-17-2000' and '2000/05/17' will all be converted to the 'yyyy-mm-dd' format. Numerical compounds will be converted to Arabic figures for easy computation. For example '四百零一' will be converted to '401'. The reason for such conversion is that calculation of numerals is often common in discourse, and that is a necessary part if we want to construct a full discourse structure. For example, in

(37) 藍籌股三十三檔成份股中：十六檔股票上漲，十六檔股票下跌，一檔持平。
(In the thirty-three composition stocks in Hong Kong Blue Stock, sixteen of them rose, sixteen of them dropped, one of them held position.)

The statement is semantically correct only when the sum of the numbers in each coordinating clauses equals to that in the prepositional phrase that sets the topic. Therefore number conversion is necessary for the computation.

5.3. Identification of new words

In chapters 1.3 and 3.4, we have stated that theoretical UBG formalism assumes all the words in the input sentence are in and well defined in the lexicon. However, there is definitely not the case in real situation, because no dictionary can list out all possible words allowed in a language. A single unidentified word is enough to fail a

whole sentence in a UBG parser. Therefore, there is a need to identify new words in raw texts and give them senses and locate their grammatical functions (part-of-speech). Attempts have been made to integrate segmentation with UBG parsing [25][41], but none of them tackle the issue of new words that do not exist in the dictionary. It is sure that a pure unification model is not enough (at least in the state-of-the-art) for every part of natural language processing. A possible solution or mediation is to seek help from statistical processing. As usual, it does not guarantee high accuracy, but it always give fast response time and is crucial in practical use where time is still a bit more important than quality. Although we have mentioned before that the SERUP has a number of morphological rules, in order to achieve maximum speed, we'd like to have some statistically determinable compounds to be grouped first, thus reducing the workload of the parser. We will mention this point again in due course.

There are two major kinds of new words in Chinese texts: proper nouns (e.g. human names, organization names, etc.) and multi-syllabic (but mainly disyllabic) words. Examples of proper nouns are "朱元璋" (Chinese name, Emperor of Ming Dynasty), "克林頓" (transliterated name of "Clinton"), "香港政府" (The Hong Kong Government) and "深井" (A village in Hong Kong). Examples of multi-syllabic words are "跳投" (disyllabic verb: to jump shoot), "茶廠" (disyllabic noun: a tea factory), "免提電話" (multi-syllabic noun: hand-free phone). We will describe how we handle these two kinds of words in the subchapters follow.

5.3.1. Proper nouns – Chinese names

The commonest proper nouns in text are human names. Luckily, Chinese names (as

well as Japanese and Korean names) are the easiest to identify because their surnames are seldom used in forming words with other characters. Surnames like "黃" (Wang2), "陳" (Chen2) and "張" (Zhang1) can exist as word-forming characters. For example, "黃" (wang2)[5] is also an adjective representing the color "yellow", as in "黃色" (yellow); "陳" (chen2) is an adjective meaning "old" and "torn", as in "陳舊" (old and torn); "張" (zhang1) is also a verb meaning "to open" as in "張開" (to open). In such cases, the surname characters will not be segmented as single characters. As a result, they will not be considered surname and will not take part in the name-forming algorithm given later.

Thus we have sufficient clues to locate the boundary of possible names in the text. Problem remains in locating given names, however. How do we know how many characters a particular given name will have? A Chinese given name usually has one or two characters[6]. We have devised a simple algorithm (algorithm 5.1) to make decision. This simple pattern-matching algorithm has been incorporated in the segmentation program mentioned in chapter 5.1.

---

[5] The Pinyin used here are in lower case letters to distinguish themselves from their surname counterparts.

[6] Some new born babies in Mainland China have three characters given names, because the parents are aware of the fact the too many people are sharing the same names. However we are not going to deal with this in SERUP.

```
ALGORITHM 5.1: Deciding the length of given name

1. For the character sequence {C0,S,C1,C2,C3,…,Cn} where
S is the surname[7], and C0,C1,…,Cn are single characters,
2.  If C0 can form word with S and/or characters behind
3.    Form a hard delimiter from C0 to Cm, where m≦n[8]
4.    Quit
5.  Else S is confirmed to be a boundary
6.    If C3 form word with characters behind
7.      The sequence {S,C1,C2} is taken as a valid name
8.      Quit
9.    If C2 form word with characters behind
10.   The sequence {S,C1} is taken as a valid name
11.   Quit
12. If C1, C2, C3 remains as single characters and C3 is
13. not a function word, e.g. "的", "在", "往", etc.
14.   If Prob(C2 appears 2nd pos of given name) > Threshold
15.     The sequence {S,C1,C2} is taken as a valid name
16.     Quit
17.   Else the sequence {S,C1} is taken as a valid name
18.    Quit

End of algorithm
```

Algorithm 5.1. Deciding the length of a given name

The algorithm always takes a character sequence as a valid Chinese name, if the characters do not form words with other characters. Lines 6-11 locate single-character or double-character given names without controversies. It is hard to make decision

[7] The surname can be of one of two characters.

[8] "m" is an integer denoting the position of the character in the character string.

when there are three or more single characters appear after the surname. In this case, we need the help of statistics. We obtained the probabilities of characters being the second character of given names in our name corpus[9]. If the probability is higher than a threshold (determined by experiment), the character sequence {S,C1,C2} will be recognized as a valid name. The reason behind is that we observe that the last character of a given name is a deciding factor of identifying a given name. For example, the characters "國" and "榮" appear very often either as the first or second position of given names, e.g. "國榮", "國輝", "國棟", "衛國", "有國", "志榮", "錦榮", "俊榮", "榮俊", etc. Some characters seldom appear as the last character of a given name, e.g. "系", "代", "自", etc., but they can appear in a given name if they are followed by characters like "國" and "榮". For instance, "系國", "代國" and "自榮" look more likely to be possible given names than "國系". "國代" and "榮自".

Chinese name identification is not absolutely obvious, though. In real text, sometimes only the given name appears, but not the surname. What's worse, in Mainland China many parents like to name their children a single character so that the whole name appears as a standard entry in a dictionary. For example, "舒暢" and "路途", etc. When the segmentation program meets these terms, they will be recognized as standard dictionary term only. In fact, these two problems are so monumental and we will leave them as future work.

Other proper nouns include organization names, transliterated names and geographical locations. As they are hard to detect during segmentation, they are left out to the next subchapter.

---

[9] Our name corpus is the Telephone Yellow Page 2000 published by HK Telecom.

5.3.2. Other proper nouns and multi-syllabic words

5.3.2.1. Organization names, geographical locations and transliterated names

For organization names, we look for the common keywords like 集團 (group), 公司 (company), etc. The algorithm is as follows.

```
ALGORITHM 5.2: Locating organization names

  1. Search for keywords like "集團", "公司", etc.
  2. If the sequence {W,C1,C2,…,Cn,K} is found, where W is
  a word, followed by a space, i.e. segmented, and C1, C2,…,
  Cn are space separated characters, and K is the keyword
  3.    Group C1-Cn and K as a valid organization name
  4.    Quit

End of algorithm
```

Algorithm 5.2. Locating organization names in sentence.

A problem here is that the character sequence {C1,…, Cn} may actually contain other new words in it, and so it is not safe to assume the sequence must be the name of the organization. In fact, in SERUP locating organization names is not performed until locating disyllabic words is finished. In case the organization name has a standard entry counterpart in the segmentation dictionary, e.g. "忠誠" (honest) in "忠誠銀行" (Honest Bank), we believe that the best way is to make "忠誠銀行" (Honest Bank) an entry. Otherwise it is not a fault to treat "忠誠銀行" (Honest Bank) as an "Adjective +

Noun" noun phrase with "銀行" (bank) as head. It should be noted as well that if the organization keywords are missing, identification will be a difficult job very much like the case when surname is missing in Chinese name identification.

For geographical locations, as long as keywords exist (e.g. "村" village, "城" castle, "市" city, "鎮" town, "國" country), we can use Algorithm 5.2 to do the job but this time substituting the relevant keywords. For the case where keywords are missing is again outside the scope of this thesis.

The issue of transliterated names is much more complicated that we are not going to deal with it in this thesis. Rather there is a section in chapter 9 future work talking about it.

5.3.2.2. Multi-syllabic words

In real texts, there exist many multi-syllabic words that are not included in the dictionary. In the following sentences [42] the italic underlined word compound would probably be left unidentified:

(38) 中國 *女壘* 獲 香港 企業 支持。 (Disyllabic noun)

(Chinese female baseball team gains support from Hong Kong enterprise.)

(39) 邵逸夫 *捐建* 了 此 樓。 (Disyllabic verb)

(Sir Run Run Shaw donated to the completion of this building.)

(40) 中國 *健力寶* 集團 公開 *招股*。 (Multi-syllabic noun)

(Chinese Jianlibao Group openly asked for stock capital.)

The algorithm we used in SERUP to identify multi-syllabic word is very much indebted to Wu and Jiang [42] paper presented in the Second Chinese Language Processing Workshop in the 38th Annual Meeting of the Association of Computational Linguistics (ACL). Their intuition is that if there is a sequence of single characters after the completion of word segmentation, morphological derivation (in our case this is done in the parser) and Chinese names identification, it is very likely to be a new word. They give such an example:

(41) 維初男隊中鋒范建毅居然冷射得手。[42]

(Weichu men team's center forward Fan Jianyi snatched a goal with an unexpected shot.)

Say, the terms "維初" (an organization name) and "冷射" (to shoot suddenly) are not in the dictionary. After segmentation, we will have (41a), keeping in mind that unfound words will appear as sequence of space-separated characters.

(41a) 維 初 男隊 中鋒 范建毅 居然 冷 射 得手。

What they want to do is do find such sequence and group the characters into words. However, Not every single-character sequence can form a word in Chinese. Consider

(42) 她回了國才會來看你。[42]

(She will not visit you unless she has returned to the country.)

After segmentation we may get either (42a) or (42b):

(42a) 她 回 了 國 才會 來 看 你。

(42b) 她 回 了 國 才 會 來 看 你。

(42a) contains a sequence of 4 characters, while (42b) even contains one of 9 characters. However, in both cases, the sequences are definitely are a word, because each character in the sequence is a free morpheme, that means they can exist alone without combining other characters to form a word. So the existence of single-character sequence is not the only criteria of locating new words. Another important criteria is that each character should have low likelihood to be free morpheme.

5.3.2.2.1. The algorithm in detail

In Wu and Jiang's implementation, the Independent Word Probability (IWP) of each Chinese character, which is simply the probability of a character being a free morpheme, is defined by the following equation

$$IWP(c) = Occurrence(Word(c)) / Occurrence(c),$$

where Occurrence(Word(c)) is the count of character c appearing as free morpheme in a text corpus, and Occurrence(c) is the count of character c in the same corpus. The IWPs of characters can be easily found by scanning through the text corpus and collect statistics at the same time.

The parsed corpus in their research contains 5000 sentences. Although the corpus is not big enough to contain every Chinese character, what the algorithm actually need in real is just a corpus that contains all the frequently used free morphemes.

Once the IWPs are obtained, the IWP of a single-character sequence (with at least 2 characters) is simply the joint probability of the IWP of each character in the single-character sequence. The result is then compared to a threshold T determined by experiment. It IWP is smaller than T, the single-character sequence will be taken as a new word. If it exceeds the predefined threshold it is very unlikely to be a new string. The algorithm is summarized in Algorithm 5.3.

```
ALGORITHM 5.3: New word identification by Wu & Jiang [42]

  1. Perform segmentation on input sentence
  2. Identify Chinese names and derivational morphological
  compounds
  3. For each single-character sequence S = {C1,C2,…,Cn},
  where C1,…,Cn are space-separated characters
  4.  Calculate IWP(S)
  5.  If IWP(S) < Threshold
  6.    S is taken as a new word
  7.    Quit
  8.  Else Quit

End of algorithm
```

Algorithm 5.3. New word identification by Wu & Jiang [42].

The algorithm is very intuitive. However, there are three flaws in it. First, it does not deal with the case when there is more than one sub-sequence with IWP lower than threshold in the same sequence. For example,

(43) 我 昨 午 買 了 手 天 威 船 業 的 股票。

(Yesterday I bought one unit of stock of Tianwei Ship Service.)

There is a 10-character sequence{我,昨,午,買,了,手,天,威,船,業} in the sentence. The IWP of this sequence no doubt will be higher than threshold and so it will not be recognized as a word. However, it is also obviously that the sub-sequence{天,威,船,業} may have lower enough IWP to form a word. But in the algorithm, it does not state whether sub-sequence should be taken into account or not. Particularly like (43) the new word identification process will stop upon knowing the 10-character sequence cannot form a word. So the new words may not be found as expected.

Again in (43), suppose the IWP of each character we have allows {手,天,威,船,業} to have an overall score lower than threshold, should we take it as a word? The answer here is no, but the algorithm is likely to take it anyway because the character "手" (hand) has a higher tendency to appear in disyllabic words in the corpus (e.g. "左手" left hand, "右手" right hand, "助手" assistant, "幫手" helper) than being a single character. So probably we have

$$IWP(\{手,天,威,船,業\}) < IWP(\{天,威,船,業\}) < Threshold.$$

The algorithm will take the sequence will lowest score, in this case, a wrong decision.

So the second flaw of the algorithm is that it cannot handle the case when the sequence of possible word-forming characters is in the middle of the single character strings.

The third flaw is that, a bound noun appearing as the first character of the single character sequence maybe the head noun of the compound formed with the previous word. For example

(44) 婚姻 法 監 保 婚姻 。

(Marriage Law monitors and protects marriage.)

"法" should be the head noun of the compound "婚姻法" (Marriage Law), but not the first character of the unlikely new word "法監". However, in real all "法", "監" and "保" are bound morphemes, i.e. their IWPs are very low. So "法監" and even "法監保" will have high enough IWPs to be chosen as new word instead of "婚姻法" (Marriage Law). This implies that any sequence of bound morphemes would be combined in the algorithm. Massive over-generation will result under this flaw and we have to find a new strategy to minimize such problem.

We help solving such situation by collecting two more important statistical results, they are the probabilities of a bound morpheme occupying the first position and the second position of a disyllabic word respectively. After locating all single character sequence that score higher than threshold (up to this point is the same as Wu and Jiang's original algorithm), we calculate the product of the relevant positional probabilities. By these probabilities, the system will be able to tell whether a character is likely to appear in a particularly position of the new word.

Another problem arises here is that how do we give score to a new word with more than two characters. The statistics we have concern only about the first and second positions, but not the positions thereafter. Our claim is that it is not significant to get the positional probabilities after the second one. Chinese vocabulary is dominated by disyllabic words, which constitute more than one half of all. Statistics has shown support to our insight. Of the 85135 words in Wu and Jiang's dictionary, 9217 of them are monosyllabic, 47778 are disyllabic, 17094 are tri-syllabic, and the rest has four or more characters. Moreover, in the 17094 tri-syllabic words, over 90% of them are derivational nouns with the last characters (3rd position) being the morphological head nouns. In addition, most disyllabic nouns also have their head located in the last character [43], and this further suggests that the probability of a character appearing in the third position of a word is exchangeable with that in the second position. Table 5.2 shows the internal structure of some common words with two, three and four characters respectively (Key: Det = Determiner, N = Noun, V = Verb, A = Adjective).

Two characters	Three characters	Four characters
Det + N (雙塔)	N + N + N (金字塔)	V + N + V + N (議事論事)
V + V (玩笑)	V + V + N (拘留所)	
N + N (場所)	V + V + V (開玩笑)	
A + A (亮麗)		
A + N (麗人)		

Table 5.2. Internal structures of some multi-syllabic words

It can be seen that any character that appears in the last position of a three-character word can appear in the last position of a disyllabic word as well. Therefore it is safe to assume that the probability of a character appearing in the third position of a word is the same with that in the second position. Another reason for this manipulation is that we want to eliminate sparse data. As the summation of the probabilities of a character occupying the 1^{st} to the n^{th} position of words must be one, if we count the occurrence of the third and the rest positions as well, we will create a lot of entries with very low probabilities. These entries do not contribute anything in the real calculation because they are never selected. So, in an attempt to allow more characters to join competition of new word formation, we utilize the two most significant probabilities only.

The case of four characters is treated differently. As most four-character words are in the form of {V+N+V+N}, which means they can be further decomposed into two {V+N} units, the probability of a character occupying the third position should be taken as the probability of the first position, and that of the fourth position should be taken as the probability of the second position. In fact, four-character words are very rare in real because almost all four-character words are well-established idioms and the dictionary should have abundant entries of them. Moreover, the fact that most four-character words are in the form of {V+N+V+N} allows the parser to recognize them without much ambiguity. A possible reading of theses words may then be serial verb construction (SVC). This will be covered again in the next chapter about grammar construction.

Now back to the discussion of (44). "法" mostly occurs in the second position of a verb, and the last position of a compound noun. The score of "法監" is given by the

following equation:

$$\text{Prob}(\text{“法監”} = \text{new word}) = \text{Prob}(\text{“法”} = 1^{st} \text{ position}) * \text{Prob}(\text{“監”} = 2^{nd} \text{ position})$$

Both probabilities are low and thus prob(“法監” = new word) will have a very low score and will not be taken as a word.

Algorithm 5.4 shows our modified version of Wu and Jiang's original algorithm. Please note again lines 10-14 how we assign positional probabilities to word having more than two characters. One more thing to note is that we collect the probabilities of a character being a bound morpheme (BMP) rather than the IWP. The reason for this is totally for the ease of understanding. With IWP, we pick the sequence with lowest score, but with BMP we pick the highest score. We believe this is a more natural notation.

ALGORITHM 5.4: Modified new word identification in SERUP

1. Perform segmentation on input sentence
2. Identify Chinese names
3. For each single-character sequence S = {C1,C2,…,Cn}, where C1,…,Cn are space-separated characters
4. For each sub-sequence S1 with 2 or more characters, i.e. S1 = {Cs1,Cs2,…,Csn}
5. Calculate BMP(S1)
6. If BMP(S1) > Threshold
7. Store S1 in list L
8. For each sub-sequence Ssub in L
9. For each character Cm in Ssub
10. If length(Ssub) is odd /* assume length(Ssub) <= 5 */
11. Prob(Cm in 3rd pos) = Prob(Cm in 2nd pos)
12. Else /* length is even */
13. Prob(Cm in 3rd pos) = Prob(Cm in 1st pos)
14. Prob(Cm in 4th pos) = Prob(Cm in 2nd pos)
15. Joint probability Jc *= Prob(Cm in current pos)
16. Store Jc in list L1
17. Find M = max(L1)
18. Take Ssub as a new word with Prob(Ssub) = M
19. Quit

End of algorithm

Algorithm 5.4. Modified new word identification in SERUP

The biggest difference between Wu and Jiang's original algorithm and ours is that theirs leads to over-generation easily, because it groups every single character sequence once the joint probability is over threshold. Ours seldom over-generate, but may suffer from under-generation. We perform second screening by checking the characters tend to form new words in particular positions. It is hard to say which approach is better because the original algorithm is a bit more robust than ours, while ours have higher accuracies. However, it is true that it is easier to add rules in the parser to deal with those under-generated cases rather than devising method to prune away over-generated result. In addition, it is a more reasonable analysis.

After identifying new words, we have to do two more things. The first one is to assign part-of-speech to them, the second one is to create information-rich lexical entry for them. These two steps are necessary in SERUP because without part-of-speech and other linguistic information the new word is useless to the UBG parser.

5.3.2.2.2. Assigning part-of-speech

Part-of-speech (POS) of words can be assigned by matching the word-internal structure and the POS sequence of the newly found word. For example, a disyllabic Chinese verb can have the structure V-V, V-N, N-V, or Adv-V. Therefore, if the newly found word has a POS sequence that matches one of the above, it can be recognized as a verb. However, this is not a good method, nor a scientific method that looks into the detail of Chinese linguistics at all. Firstly, most Chinese characters can have more than two different POSs. In chapter 3.2.2 we show that there is no inflectional change at all when a Chinese character/word take up different grammatical functions.

Therefore we have no ground to assign a particular POS to a character without getting more linguistic information, not to say to determine the POS of a multi-syllabic word.

At the same time when we examine the word formation of Chinese, we can see a character with a particular POS usually takes up a particular position a word according to the word's POS. Packard [43] further suggests that the POS of the word governs the internal POS sequence of a word. Table 5.3 shows some examples.

Character	POS in word	Examples
口	Noun	口號, 口氣, 口徑, 口袋, 口述, 口服
科	Noun	科技, 科長, 科舉, 科目
呼	Verb	呼吸, 呼聲, 呼叫, 呼喊, 呼喚
黑	Adjective	黑色, 黑煙, 黑板, 黑夜, 黑馬, 黑心
射	Verb	折射, 發射, 輻射, 投射, 影射
失	Verb	得失, 報失, 迷失, 損失

Table 5.3. Examples of characters in i^{th} position having the same POS in words

For instance, "口" (mouth) in line one of the table is a noun throughout the examples behind. "呼" (to call) is a verb, "黑" (black) is an adjective, etc. In dictionary, however, "失" (to lose) can be both verb and noun. Therefore it seems reasonable to make the hypothesis that "a candidate new string for a new word is likely to have a given POS if the component characters of this word have appeared in the corresponding positions of many existing words with this POS" [42].

Currently we are using the method given in the same paper by Wu & Jiang [42]. To represent the likelihood of a character to appear in a given position of a word with a given POS and a given length, probabilities of the following form are collected for all

characters in the dictionary:

Prob(***Character*** appears as (***Category***, ***Position***, ***Length***))

where ***Category*** is the POS of a word, ***Position*** represents the position of the ***Character*** in the word, and ***Length*** is the length of the word. So, in this notation, the probability of the character "□" (mouth) appearing as the first character in a three-character noun is given by Prob("□" appears as (N,1,3)). Further assumptions made are that ***Category*** is limited to the open class categories only, i.e. noun, verb and adjective. New words formed can only fall into the open classes; also, ***Length*** is assumed to be 2-4. As a result, we have 27 probabilities for each character. The probabilities can be obtained just by counting the number of occurrences of a character in a given position of words with given POS and length, then divided by the total number of occurrences of this character in the dictionary headwords. For example, the equation

Prob(C appears as (N,1,2)) = Occurrence(C appears as (N,1,2)) / Occurrence(C)

gives the probability of a character C appearing in the first position of a disyllabic noun. Table 5.4 shows the probabilities of the character "射" (to shoot)[10].

[10] The data here is different from that in Wu & Jiang's paper because our dictionaries are not the same.

Positional probability of "射" (to shoot)	Value (round off to 3 after d.p.)
Noun,1,2	0.138
Noun,2,2	0.069
Verb,1,2	0.241
Verb,2,2	0.483

Table 5.4. Positional probabilities of "射" (to shoot).

According to the statistics, "射" (to shoot) tends to appear as the second character of a disyllabic verb.

Then we can determine the POS of the newly found words with these statistics. The probabilities are a word being a noun, a verb or an adjective are the joint probabilities of the Prob(***Character*** appears as (***Category***, ***Position***, ***Length***)) of the component characters of the word. After that we compare the probabilities against an experimentally determined threshold (in Wu & Jiang it is 0.75). The word will be assigned a certain category if that joint probability is higher than the threshold. In the case where more than one probability are higher the threshold, more than one category are assigned to the word, as if a common multiple functional entry in a dictionary. When there is no probability higher than the threshold, by default the word will be treated as a noun, because by proportion most new words are noun. For example, "維初" in (41) passes the BMP test, but it does not pass the POS test. By default it is given the category noun.

5.3.2.2.3. Creating lexical entry for new word

One issue unique in SERUP is that after assigning POS to new word, we also need to

create a information-rich lexical entry for it so that the UBG parser can utilize. As we are not yet in the chapter talking about grammatical construction in SERUP, we only describe in general how this is achieved. Our approach is straightforward. We copy the feature structure of the head noun/verb/adjective in accordance to the Headedness Principle [43] to the feature structure of the new word. The Headedness Principle says that for a disyllabic noun, the right-hand side noun would be the head; and for a disyllabic verb, the left-hand side verb would be the head. For adjective, however, it does not matter whether to choose the left-hand side or the right-hand side adjective. We will cover it again in chapter 6.

The new word identification in SERUP, although utilizing some useful linguistic information, better result is possible if we also incorporate semantic information, but not just syntactic one, into processing. We will talk about such possibility in chapter 9 future work.

5.4. Defining smallest parsing unit

5.4.1. The Chinese sentence

In practice, source texts are mostly written in paragraphs, but parsers work on sentential level (future parsers may work on discourse level, but sentential processing is always the basis). As a result, we must find a way to break paragraphs into smaller parsing units. The most intuitive way is to break each paragraph into sentences. For language with verb inflection (e.g. Indo-European languages, Japanese, Korean, etc.), it is very easy to determine sentence boundary. It is either specified by a full stop or by sentence final verb endings. For Chinese, however, the term "sentence" is

confusing enough, as it cannot be determined exclusively with syntactic terms [38]. Important evidence is that Chinese is considered a topic-prominent language [24]. Moreover, "topic chain" has been considered an important structural notion in Chinese discourse and sometimes as a syntactic category as well [38]. Tsao defines a topic chain as "a stretch of discourse composed of one or more comment clauses sharing a common topic, which heads the chain." For example, (45) is a topic chain [38].

(45) 他 來 看 你 了 ， (ZA[11]) 還 帶來 幾個 朋友 來 呢。

(He came to see you (and he) brought some friends with him.)

It can thought as a combination of two sentences, (45a) and (45b) linked together. Note that after linking, the subject of (45b) is deleted and reappears in the form of zero anaphor (ZA).

(45a) 他 來 看 你 了 。

(He came to see you.)

(45b) 他 還 帶來 幾個 朋友 來 呢。

(He brought some friends with him.)

Clearly (45) is a valid sentence in Chinese, but this sentence can hardly be described by the notion of syntax in Western grammar. It is not only an unit with "complete meaning", it also represents a discourse. So topic is indeed a discourse notion that

[11] ZA stands for Zero Anaphora.

implies that the study of topic would be meaningless if it was confined within the domain of a sentence. Current UBG theories can only deal with syntax and semantics, but not discourse. That means when we are fitting Chinese into UBG, we cannot assume the standard input to the parser must be a sentence. Literature in Chinese language processing never talks about the treatment of Chinese sentence. Researches in UBG always assume the input is a simple sentence (simple sentence structure, no subordination) or well-formed coordinate sentence. However, we would not be able to move forward if we do not tackle the problems brought by topic chain and discourse.

Chu [38] further suggests that the definition of sentence in Chinese should be "one or more clauses that are related by formal devices identifiable by overt signals." These signals include topic chain, conjunction, adverb, verb form, mode of presentation, clause order, end of discourse, zero anaphor, clause order and sentence-final particles. So it is also clear that reasonable sentence boundaries[12] are not determinable before parsing. Given the fact that state-of-the-art computational linguistic theory cannot tackle discourse during parsing, what we should do is to try breaking the paragraphs into elements, and have them reconstructed after parsing. Syntactic and semantic information is made available after parsing, and only from that time can we start regrouping.

5.4.2. Breaking down the paragraphs

Now back to problem of paragraph segmentation. If we come across a Chinese

---

[12] Chu [] conducted a survey to 60 native Mandarin speakers and different sets of sentence boundaries are recognized. He concluded that many people are not aware of the overt signals that determine the boundary of a Chinese sentence.

sentence like (46), how should we feed it into the parser?

(46) 我聽說某公司的營運總裁收購了一間小店，只是不停蝕本，它的大老板很不喜歡。

(I heard that the operation manager of a certain company bought a small shop that never stops debiting money. Its boss was not happy about that.)

(46a) 我聽說某公司的營運總裁收購了一間小店，

(I heard that the operation manager of a certain company bought a small shop)

(46b) 只是不停蝕本，

(The shop never stops debiting money.)

(46c) 它的大老板很不喜歡。

(Its boss was not happy about that.)

It is clear to native speakers that (46b) and (46c) are closely linked to (46a), and the segments form one complete sentence, rather than two, or even three independent sentences.

In SERUP, we define the smallest parsing unit to be a clause (verb phrase) and a clause with its subject (in this sense, it is equal to the syntactic notion of sentence in its original sense). This directly relates to the definition of a Chinese sentence. As a Chinese sentence is comprised of one or more clauses, it is very reasonable to choose a clause as the smallest unit, and it will be the basis of discourse reconstruction.

There are two straightforward methods to determine the clause boundary. The first one is to use full stop as delimiter. Full stops are used to signify the end of an utterance, but from chapter 5.4.1, we know the problems in defining a Chinese sentence and so using full stops as delimiters is meaningless. The second method, which is a more plausible one, is to use comma as delimiter.

Another two problems occurred. First, in Chinese, commas can be used to coordinate nouns, adjectives and even verbs. As a result we must find out all non-clause forming, comma separated constituents, and group them with clause-forming ones. Therefore, in

(47) 中國電信下跌了 2.1%，*香港電訊，匯豐控股與和記黃埔則小幅下跌。*
(China Telecom dropped 2.1%. Hong Kong Telecom, HSBC and Hutchison dropped slightly.)

The system treats it as two units, but not three units (the italic part is one single unit).

Second, a topic chain needs to be broken down into clauses. A topic chain in Chinese is usually linked by a topic in the form of ZA (zero anaphor) [38]. Each ZA-clause is fed into the system as separate parsing unit together with the punctuation. SERUP is able to tell such verb phrase is a sentence with head pronoun dropped, but not ordinary verb phrase. So the topic chain

(48) 昨日市場成交高點出現在早上，(ZA) 爲 16,046 點，(ZA) 但隨後即下跌。
(Yesterday the highest moment of stock exchange appeared in the morning. The index reached 16046, but dropped soon after.)

is handled as three separated units, with the subject of the two verb phrases as ZA.

Third, comma is also used in presenting vocative[13]. Therefore the sentence

(49) 電訊盈科，這隻股的股價終於急劇下跌。

(PCCW, its stock value finally drops rapidly.)

is handled as one unit.

Fourth, prepositional phrase should be attached to a clause. So the sentence

(50) 在過去一年，中國電子產業類股主導了股票市場的漲勢。

(In the past year, Chinese electronic industry stocks dominated the expansion of stock market.)

is handled as one unit.

5.4.3. Implementation

Segmentation of smallest parsing unit is done with using a POS-tagger in SERUP. A POS-tagger selects the most likely sequence of syntactic categories for the words in a sentence. A general method is to use statistical method is estimate the probability of a word having a certain POS. SERUP is using the trigram model in such estimation.

---

[13] A vocative is a nominal element added to a sentence or clause optionally, denoting the one or more people to whom it is addressed, and signaling the fact that it is addressed to them [9].

The trigram model uses the conditional probability of one category (or word) given the two preceding categories (or words), that is prob(Ci | Ci-2 Ci-1). Therefore, the probability of a string of words having POS Ci to Cn is given by the equation:

$$Prob(C1,\ldots,Cn) \sim= \Pi_{i=1,n} Prob(Ci|Ci\text{-}2\ Ci\text{-}1)$$

SERUP uses the standard Viterbi [44] algorithm on Markov models in implementing the tagger. As POS tagging has been written extensively in both Western and Chinese language processing literature [2, 18], we are not going to discuss the detail here.

The tagset we are using is modeled after the Tsinghua University's Grammatical Knowledge base of Contemporary Chinese project. They use 26 grammatical categories in their dictionary. We add 3 more tags for noun phrase, verb phrase, and prepositional phrase. So there is a total of 29 tags. The reason why we choose this relatively small tagset for SERUP is that it is easy to understand and follow. Moreover, our use of tagger is just to locate regular patterns in the text, but not as the basis of parsing. Therefore we do not need a very sophisticated tagger. In addition, our tagger is used particularly for the segmentation program in SERUP. Recall in chapter 5.1 that phrasal entries are put in the segmentation dictionary for chunk processing. This implies that the tagger must provide phrasal tags as well. Table 5.5 shows the SERUP tagset.

Tag (as used in SERUP)	Tag explanation
a	Adjective
b	Distinguisher
c	Connective
d	Adverb
e	Exclamation
f	Directive
g	Free morpheme
h	Pre-linking
I	Idiom
j	Abbreviation
k	Post-linking
l	Slang
m	Numeral
n	Noun
o	Onomatopoeia
p	Preposition
q	Classifier
r	Pronoun
s	Locative
t	Time
u	Particle
v	Verb
w	Punctuation
x	Bound morpheme
y	Emotion
z	Stative adjective
A	Noun Phrase (NP)
B	Verb Phrase (NP)
C	Prepositional Phrase (PP)

Table 5.5. SERUP tagset.

After segmentation has finished, we tagged the results using the tagger. Boundaries of verb phrases can be located by careful examination of the pattern of tags. Principles laid out in chapter 5.4.2 are strictly followed in the process. For example, in the sequence "N,N-V-N。", the first 'N' will not be segmented; in the sequence "N-V-N,N-V-N。", the middle 'N,N' will not be segmented as an coordinative noun, because it should be immediately followed by a full stop if so, thus yielding two "N-V-N" segments. Table 5.6 shows some of the rules in deciding clause boundaries. It should be noted that, however, the method employed works best only with contemporary written Chinese (particularly formal documents and functional articles), in which the writing style and the use of punctuation is easier to follow. These rules can be realized through regular expressions. In SERUP, segmentation of clause is written in PERL, an extraction language with excellent regular expression support. We will talk about it again in chapter 8: implementation.

Tag patterns	Explanation
any tag + Directive + Comma + any tag	Prepositional phrase, not segmented
Noun + Verb + Noun + Punctuation + Noun + Punctuation	Possible coordinate object noun phrase One segment
Verb + Noun + Verb + Punctuation 拖累 港股 急挫，	Verb phrase with verb complement, segmented
Noun + Preposition + Noun + Comma + Noun + Verb + Noun + Punctuation 他 以 米，酒 來 維持 生命。	Possible coordinate noun complement in prepositional phrase

Table 5.6. Examples of rules determining clause boundary.

6. Step Two: grammar construction

6.1. Criteria in choosing a UBG model

The choice of parser for our NLP research is a Unification-Based Grammar (UBG) one. In chapters 2.2 and 2.3 we have discussed about the power and advantages of using UBG in grammar description, and a short summary for each influential UBG theory. For example, the declarative nature of UBG makes it flexible and independent of the parsing algorithm. Moreover, troublesome syntactic phenomenon like long-distance dependency and coordination can be solved easily with feature structures used in GPSG and HPSG. Different grammar models differ from each other in terms of expressiveness and the depth of linguistic representation, which also directly affect their computational efficiencies and human efforts in building the necessary grammar rules and complex lexicon.

For SERUP we have developed a moderate-sized Chinese grammar following the representation and principles in GPSG and HPSG. Description of both famous formalisms can be found in chapter 2.3. In summary, GPSP provides a useful tool for modeling a wide coverage grammar, the use of special features such as SLASH even allow transformation of trees be represented by just feature structures. HPSG is similar to GPSG in many ways, but it has very clear principles in guiding unification, especially that all constraints are encoded in the lexicon, so the well-formedness of trees is only a matter of unification among lexicon according to several simple principles, but not some complex interactions between principles and the grammar. Also, HPSG performs semantic analysis at the same time. Semantic interpretation and

constraint of every word is encoded in the lexicon as well. During unification, semantic information of sentence constituents is fitted into place by first-order logic deduction, which is also encoded in the form of feature structure (the detail of how this is done is omitted here).

SERUP takes advantages of both formalisms. The syntax part is written in GPSG style, as it is probably the best tool in fast grammar modeling. The features adopted in SERUP also follow GPSG mainly. Some important features like subcategorization and slash are retained in our model. However, our lexicon and the feature structure sharing principles we are employing are greatly influenced by HPSG. For instance, there is no need to use GPSG's feature principles at all because those necessary linguistic constraints are encoded in our lexicon. Therefore SERUP is a lexically rich model (HPSG) with a context-free grammar backbone (GPSG).

Both GPSG and HPSG are linguistic theories that capitalize phrase-based analysis. It is natural to ask whether phrasal analysis is suitable in Chinese. Some Chinese linguists point out that phrases in Chinese behave like phrases, sentences and subordinate clause in English [23]. Moreover, the rules governing the formation of word compounds are mainly those governing the formation of phrases. Therefore it is a theoretically reasonable approach to analyze Chinese with phrase-based analysis.

6.2. The grammar in details

SERUP has lexicon modeled after the HPSG counterpart [13]. In HPSG, every feature structure is considered a sign, and a sign contains phonetics (PHON), syntactic (SYN), and semantic (SEM) values. The simplest sign has the following look (in the form of

attribute-value matrix (AVM)):

(51)

$$\begin{bmatrix} \text{PHON} \\ \text{SYN} \\ \text{SEM} \end{bmatrix}$$

The PHON attribute is assumed to be some kind of feature representation of the sign's sound content that serves as the basis for phonological and phonetic interpretation [13]. Because SERUP has nothing deal with phonetics, the PHON attribute will only store the surface string. The SYN feature contains the syntactic capabilities of the sign, while the SEM feature stores the semantic information of the sign. SERUP lexicon has the same basic structure. However, the features used inside the sign are closer to GPSG rather than HPSG.

6.2.1. The PHON feature

In HPSG and SERUP, the surface string is stored in the PHON feature. For example, the entry "學校" (school) will have a sign like this (for illustration the SYN and SEM attributes are omitted):

(52)

$$\begin{bmatrix} \text{PHON} & \textit{學校 (school)} \\ \text{SYN} & \\ \text{SEM} & \end{bmatrix}$$

Please note that the PHON attribute does not represent any constraint in unification,

although in HPSG it is stated clearly the PHON attribute would be employed in language analysis too, such as pronunciation change with syntax and prosody [45].

6.2.2. The SYN feature

All information governs the syntactic capabilities of a sign is stored in the SYN attribute. There are two major kinds of features: head features and non-head features (a portion of them are foot features). Head features are the features own by the head daughter of a mother node in the tree. For instance, the V node is the head daughter of the node VP (this is the direct consequence of the X-bar analysis as discussed in chapter 2.3.3.), and subsets of V's features are called the head features. The features of other non-head daughter can only be foot features.

Table 6.1 shows the head features belonging to the SYN attribute in SERUP. The first column shows the attribute, the second column tells where do the attributes exist, while the third column shows the values the attributes can take..

Attribute	Appear in which sign?	Range of value
CAT	All categories	N, V, A, Adv, etc.
BAR	All categories	W, G, -1, -0, 0, 1, 2
SUBCAT	Verb	1-n, where n is an integer
ASP	Verb	Chinese aspect markers
VCOMP	Verb	Chinese verb complements
LOC	Noun	+/-
PRO	Noun	+/-
CASE	Noun	NOM, ACC, DAT
BOUND	All categories	+/-
ROOT	All zero-level categories	+/-

Table 6.1. Head features under the SYN feature.

CAT signifies the category (part-of-speech) of the sign. The main categories we are adopting can be referred to those non-phrase tags in Table 5.7. The category of a sign determines which grammar rules a sign can choose. So for the same entry "學校" (school), the feature structure will be:

(53)

$$\begin{bmatrix} \text{PHON} & \textit{學校 (school)} \\ \text{SYN} & [\text{CAT} \quad \text{N}] \\ \text{SEM} & \end{bmatrix}$$

The BAR attribute tells which level the sign is in under the X-bar theory discussed in chapter 2.3.3.1. In GPSG, zero indicates the word level, two indicates the highest number for hierarchical level, so and maximal projections (phrases) will have this value for their feature. In SERUP, as we also deal with morphology and the formation

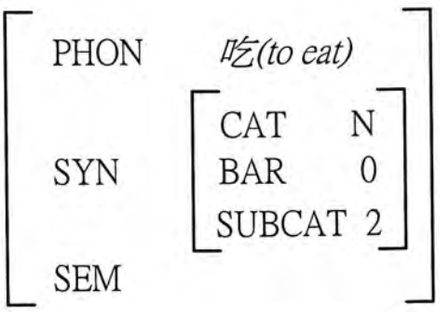
of words from characters (recall in chapter 5.3: new word identification that characters fail to form words will be given another chance in the parser through morphological rules), four new levels (-0, -1, W and G) are introduced. Chapter 6.2.10 will go further around this issue. (54) shows the same entry "學校" (school) with the BAR value filled.

(54)

$$\begin{bmatrix} \text{PHON} & \textit{學校 (school)} \\ \text{SYN} & \begin{bmatrix} \text{CAT} & \text{N} \\ \text{BAR} & 0 \end{bmatrix} \\ \text{SEM} & \end{bmatrix}$$

The SUBCAT feature stands for subcategorization, which tells the complement pattern of a verb. For example, the verb "吃" (to eat) takes a noun phrase object, while the verb "送" (to give) takes two noun phrases as direct object and indirect object. Clearly different grammar rules are needed for each type of complement pattern, and the SUBCAT is for such purpose. We are adopting GPSG implementation of the SUBCAT feature in SERUP. In GPSG, SUBCAT has a value of a positive integer greater than zero. When the parser sees a verb, those rules waiting for a verb will be selected. However, not all these rules are applicable because apart from category selection, there is also feature checking (or more accurately constraint checking). The SUBCAT value of the verb is then compared against the SUBCAT values stated by the grammar rules. Only those rules admitting verbs with that SUBCAT value will be chosen finally. This drastically reduces the number of structural ambiguities, although at the expense of a larger number of rules. The SUBCAT value only exists in verb. For example, the SYN attribute of the verb "吃" (to eat) will subsume (55).

(55)



In SERUP a transitive verb bears the subcategorization value 2. This accounts for the SUBCAT value in (55).

The ASP attribute almost corresponds to the VFORM attribute in HPSG. In Chinese, tenses are not realized by inflections of verb, but through the use of aspect markers. The aspect markers in Chinese are "著", "了", "過", "過了", "在-著", "完", etc. Similar to ASP, VCOMP is used to encode the Chinese verb complements such as "起來" (resultative complement), "得好" (manner complement) in "做得好" (well done).

The following three attributes in the list are noun attributes, i.e. they belong solely to nouns (no matter of the bar level). Nouns with [+LOC] means that the noun denotes a geographical location, such as "中國" (China) and "學校" (school), or an organization, such as "匯豐銀行" (Hong Kong Bank). All [+LOC] nouns can combine with the preposition "在" (in, at) to form prepositional phrases, e.g.

(56) 在中國 有很多窮人。 (In China, there are many poor people.)

Pronouns are different from common nouns in that they can be served as referential anaphora. Therefore it is necessary to distinguish them from common nouns. In

SERUP, both pronoun and noun have the category N, but pronoun has the [+PRO] attribute denoting itself a pronoun. For example, "我" (I, me), "你" (You) and "他" (he, him). bear the [+PRO] feature, but common nouns only have [-PRO].

The CASE attribute is originally used by GPSG to tell whether a noun is nominative or accusative. In English, pronouns take different morphological form under different cases. For example "I", "You", "We" are nominative (so that their CASE attribute is [CASE NOM]), but "Me", "You", "Us" are accusative ([CASE ACC]). In SERUP, apart from the nominative and accusative cases to distinguish subject and object, we also need to dative case ([CASE DAT]) to distinguish the dative noun in ditransitive verb phrases.

The feature BOUND and ROOT are solely used for morphological rules that will be discussed in chapter 6.6.

6.2.3. The SEM feature

In HPSG, semantic of a sentence is rendered at the same time as syntactic parsing. It follows the principle of situation semantics that regards individuals, properties, and relations as things in the word, but not conceptual objects that only exist in the minds of certain organism [13]. Whatever kind of semantic representation HPSG uses is not that significant (it only reflects how a model views the world, but this particular view does not change the integrity of the world), what's significant is that it demonstrates how syntactic-semantic parsing is possible.

SERUP concerns syntactic issue rather than semantic one, so we will not talk about

how semantic content is realized in it, although it is absolutely possible to fit the same semantic model of HPSG to SERUP.

In SERUP, however, the SEM attribute is used to encode the semantic information needed for structural disambiguation. The information includes the subject-predicate and predicate-object semantic collocation restriction. A more detail discussion will be given in next chapter.

6.2.4. Grammar rules and features principles

In the following sub-chapters we will describe how grammar rules are constructed in SERUP. For the purpose of illustration, we will only show some fundamental rules in the Chinese grammar. In fact, all grammar rules are constructed along the same principles. Our main references of Chinese grammar are Li and Thompson [24] and Zhu [46].

Before moving forward to the grammar rules, we have to first introduce some underlying feature principles. After seeing the complexity of feature structures in the preceding chapters, it seems clear that there must be principles to govern the distribution of features in a UBG, for example, given rule (57)

(57) S → NP VP

we only know the constraint of categories (we need a NP and a VP to complete the right hand side of the rule to form a S), but we do not know how the features are shared through unification. The guiding principles in SERUP to do the job are the

Head Feature Principle (HFP), the Foot Feature Principle (FFP), and the Subcategorization Principle (SP).

The Head Feature Principle (HFP) is directly inherited from GPSG and HPSG, which says that "a node and its head must share the same head features, where this is possible" [15]. For instance, VP is the head of S and so they share the same head features, unless otherwise stated in the grammar rule (it will be covered soon); N is the head of NP so they also share the same head features. Head features are indeed a subset of features belonging to a sign. In SERUP, examples of head features include CAT, BAR, SUBCAT, ASP and VCOMP. As a result, by rule (57) we immediately know that the head features of the VP on the right hand side must equal to that of the left-hand side S.

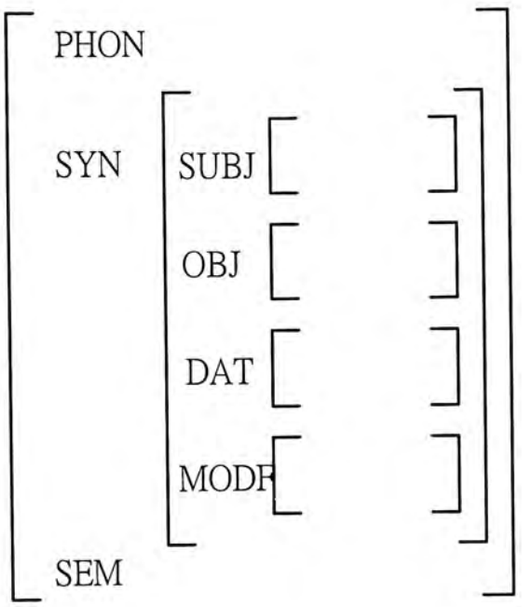
The Foot Feature Principle (FFP) says that any foot feature instantiated on a daughter in a local tree must also be instantiated on the mother in that tree and vice versa [15]. In other words, instantiated foot features on mother and daughters in a local tree must be identical. The best example of foot feature is the SLASH feature.

The Subcategorization Principle (SP) is originally from HPSG. Recall that in HPSG complements[14] to the head are encoded in the subcategorization list, and the list gets shorter and shorter (until it is empty) when complements arrive. In SERUP, the complements are however encoded in predefined slots named SUBJ, OBJ, DAT, MODF (modifier) and COMP (Verb complement is encoded in the feature VCOMP,

[14] Subject is not considered a complement in the Government Binding (GB) theory, but we do not distinguish it for simplicity in discussion.

so it is irrelevant to verb nodes). So the Subcategorization Principle in SERUP is rephrased as "complements to the head must fill up the grammatical slots where the SYN|SUBCAT feature states or as the grammar rule requires". Therefore if the SUBCAT value is 2 (i.e. transitive verb that takes 1 noun phrase object), the complement NP must be unified with OBJ in (58). The other slots are left empty.

(58)



6.2.5.Verb phrases

Verbs are the most classic and commonest predicates in Chinese. Moreover, they are the integral part of SERUP. From chapter 5.4 we know that verb phrases form its basic unit. Similar to the English counterpart, Chinese verbs can also appear in different syntactic constructs according to their type. Here we show some of the basic verb phrases [47].

6.2.5.1. Intransitive verb

Intransitive verbs do not take any object and can form verb phrase on their own. (59)

is probably is simplest rule of all.

(59) VP → H[1]

Here "H" refers to the head of the phrase, in this case it is a verb. So the "H" of noun phrase is a noun. The bracket and number after "H" refers to the subcategorization value of the verb. Apart from the features inherited from the rule (e.g. the SUBCAT value here), the rule also implies the sharing of feature structure between VP and H, by virtue of the Head Feature Principle (HFP). Verbs captured by (59) include "下雨" (to rain), "出場" (to come out of stage), etc. Examples of sentence having intransitive verbs are:

(60) 今天 VP[下雨[15] 了]。 (Today it rained.)

(61) 主角 終於 VP[出場 了]。 (The principle character is finally on the stage.)

6.2.5.2. Transitive verb

Rule (62) can capture all verbs that take only one noun phrase argument as object. Apart from HFC here, the object NP is also unified to the OBJ attribute in VP. Verbs captured by (62) include "吃" (to eat), "跳過" (to jump over), etc.

(62) VP → H[2] NP

Examples of sentence having transitive verbs are:

---

[15] Throughout this chapter, head words are underlined.

(63) 我 VP[吃過 午飯 了]。 (I have had lunch.)

(64) 跳遠 選手 VP[跳過 了 四米]。 (The long jump athlete jumped over 4 metres.)

6.2.5.3. Ditransitive verb

Some verbs in Chinese must take two noun phrases arguments as object to have complete semantic meaning, for example "還" (to return), "寄" (to send through mail), "送" (to give).

(65) VP → H[3] NP NP

Examples of sentence having ditransitive verbs are:

(66) 他 VP[送了 一本書 給我]。 (He gave me a book.)

(67) 我 VP[把書 還了 圖書館]。 (I returned the book back to the library.)

6.2.5.4. Verb taking sentential argument

Words qualified to take a sentential argument include "說" (to say), "希望" (to hope), "認爲" (to posit), etc.

(68) VP → H[4] S

Examples of sentence having verb taking sentential argument are:

(69) 媽媽 VP[說 S[我 一定 準時 畢業]]。

(Mum said I will graduate in time definitely.)

(70) 中國 VP[希望 S[可 在 年底 前 加入 世貿]]。

(China hopes to join the WTO by the end of this year.)

6.2.5.5. Auxiliary verbs

The set of Chinese auxiliary verbs include "應該" (should), "能" (able), "可以" (can), etc. An auxiliary verb must co-occur with a verb phrase, but not a noun phrase.

(71) VP[+AUX] → H[5] VP

The feature [+AUX] is instantiated on both VP and H.

Examples of sentence having auxiliary verbs are:

(72) 我 VP[應該 VP[做得到 這件事]]。

(I should be able to do this (thing).)

(73) 香港 的 經濟 明年 VP[可以 VP[復甦]] 嗎？

(Can Hong Kong's economy revive next year?)

6.2.6.Noun phrases

Noun phrase has long been recognized as the most troublesome in parsing. Complex noun phrases bring lots of structural ambiguities. In the case of Chinese, the case is even worse as words do not have inflections when taking different grammatical functions. When several nouns are serially linked together, it is not easy even for a

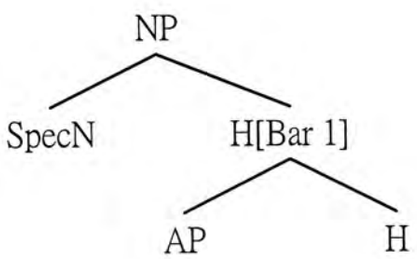
native Chinese to tell which noun modifies which and whether a word is a noun or a verb, not to say a computer program.

Here we will describe the SERUP rules to deal with both simple noun phrases and complex noun phrases. Complex noun phrases include noun phrase with relative clause and recursive noun phrase.

6.2.6.1. Simple noun phrase

(74) shows the structure of a common simple noun phrase in Chinese.

(74)



Where SpecN is noun specifier such as "這個" (this) and "那" (that); AP is the adjective phrase such as "非常美麗" (very beautiful), "有點頑固" (a bit stubborn). The following rules are able to describe (74):

(75) NP → SpecN H[BAR 1]

(76) N1 → AP H

In (75), the [BAR 1] after H forces the head noun to have BAR value 1. (76) allows recursive noun phrase as the H is also [BAR 1]. For instance a noun can be modified by many adjective phrases at the same time. Unless otherwise stated in the rule, the

head always share the same head features with higher-level constituent of the same type.

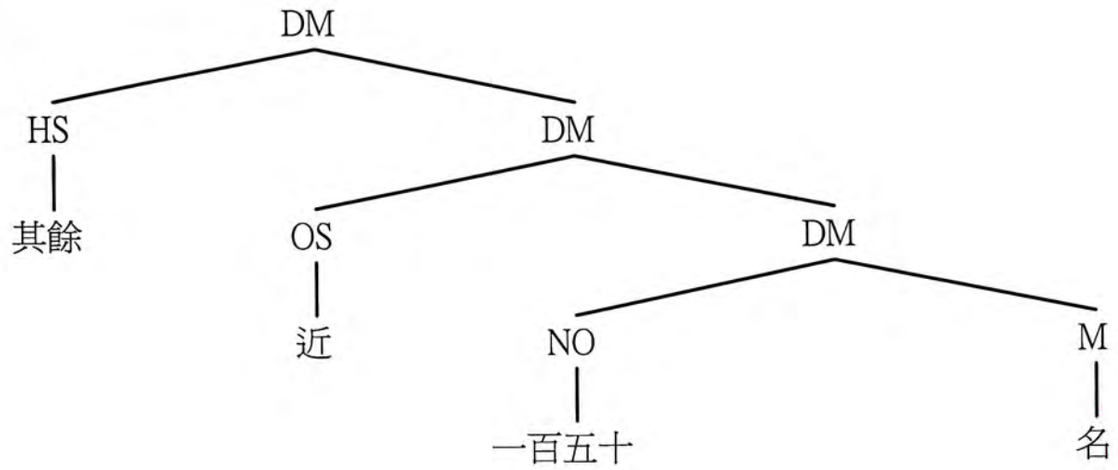
Examples of simple nouns phrases captured by (75) and (76) are:

(77) NP[這個 NP[先進 的 國家]] (This advanced country)

(78) NP[那裡 的 NP[人民]] (The people there)

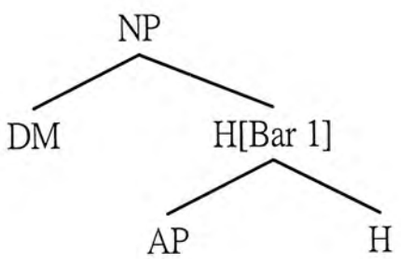
The SpecN in (75) turns out to be insufficient to capture those noun phrases with more complicated determinative-measure (DM) compounds, like "這三本" (translation: these three volumes), "其餘五件" (the other five pieces), etc.. Mo et al. [48] described a method of separating DM compounds parsing from the main syntactic parsing stage. Although their analysis of Chinese DM compounds is a good one, we do not agree on their separating the whole DM identification process from the main parsing stage. In SERUP, we only separate the identification of date, time and numeral (see chapter 5.2) from main parsing. The rest of DM compounds are handled at the same time with other phrases. Our reason is that DM compounds are highly structured, and unlike numerals, they can be derived easily from phrase structure rules. (79) shows Mo et. al. [48] analysis of the DM compound "其餘近一百五十名" (about 150 other people).

(79)



Substituting the SpecN node in (75) with DM gets (80), which gives a better coverage of simple noun phrases.

(80)



Here NO and M mean numeral and classifier. OS and HS refer to ordinal specific determiner and determiners that mean "the other members of the set". Rules (81) and (82) [48] are realization of (80).

(81) DM → {HS/OS} DM

(82) DM → NO M

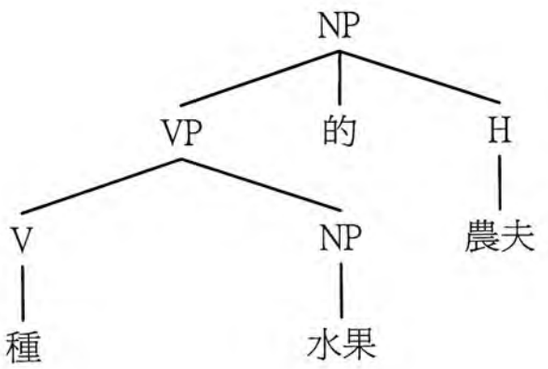
SERUP has similar rules, only the category names are different for terminology standardization. Specifying categories with different names do not affect any part of

parsing.

6.2.6.2. Complex noun phrase

Noun phrases with nominalized modifier can cause much trouble to a Chinese parser. The culprit again is what we have discussed in chapter 3.2.2, i.e. words taking different grammatical functions without inflections. For example, the noun phrase "種水果的農夫" (farmer(s) who grow fruits) has the structure [V + N + 的 + N] which is realized by the tree (83) and rule (84).

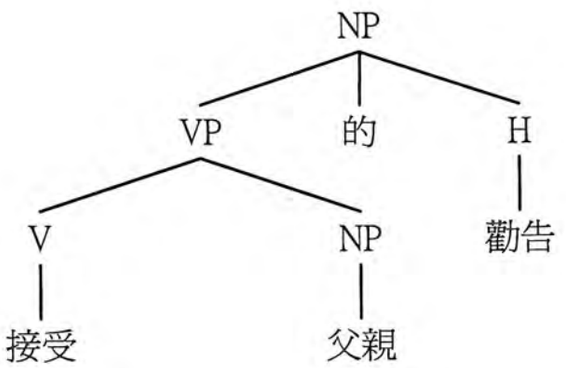
(83)



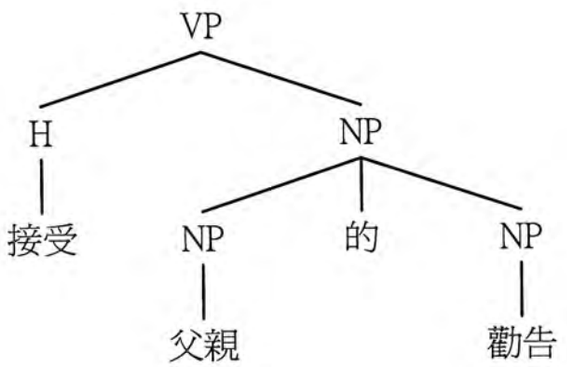
(84) NP → VP 的 H

And for the phrase "接受父親的勸告" (to accept father's advice/the advice of accepting father), its structure is also [V + N + 的 + N], but there are in fact two readings of the phrase (see trees (85) and (86)).

(85)



(86)

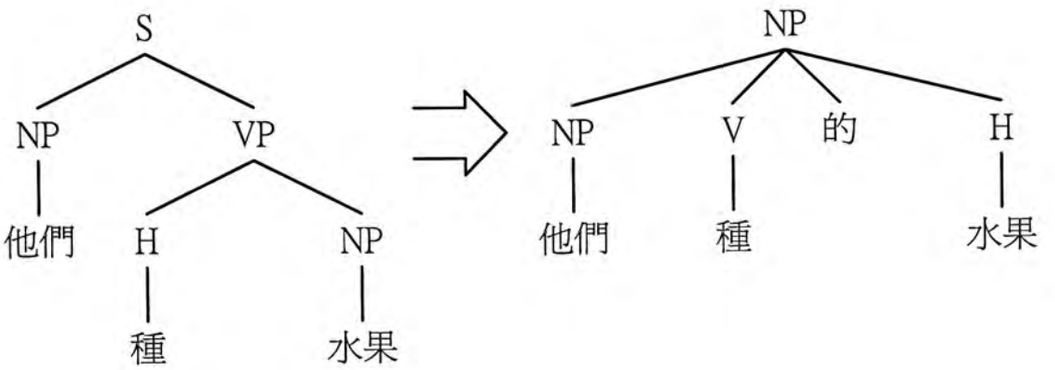


It can seen that just from syntactic structure we do not have a clue of which tree is the correct interpretation. If we also know the semantic collocation between the verb and the nouns, it is possible to cross out (85) as the wrong interpretation (although syntactically correct, it is rather awkward is Chinese). However, semantic checking is both difficult and expensive. In the next chapter we will discuss more about the disambiguation of Chinese verb phrase/noun phrase.

There are more types of nominalization in noun phrase. For example, in "他們種的水果" (the fruits they plant), the subject is present but the direct object is missing. Instead, the direct object is that participant to which the head noun "水果" (fruit)

refers. The noun phrase can be viewed as a transformation from a simple declarative sentence S (see (87)). Rule (88) captures this type of construct. However, if the verb is ditransitive, we need (89).

(87)



(88) NP → NP V[2] 的 H

(89) NP → NP V[3] NP 的 H

However, we must check the semantic collocation between the subject NP and the predicate verb, just like what we do to simple declarative sentence.

The following two noun phrases (90) and (91), captured by the same rule (92), definitely need semantic checking in telling whether the head noun is the subject or the object of the clause. In (90) the head noun "錢" (money) is the object while in (91) the head noun "人" (people) is the subject of the relative clause.

(90) $_{NP}$[今天 $_{NP}$[贏的錢]] (The money won today)

(91) $_{NP}$[今天 $_{NP}$[贏的人]] (The people who won today)

(92) NP → Adv[+TIME] V 的 H

The final type of complex noun phrase in our discussion is one that allows gaps to occur. For example, (93) is a syntactically completed noun phrase that can be captured by rule (89). (93a) – (93f) are the "gap version" of (93), and they all are possible to have the same meaning as (93).

(93) $_{NP}$[他賣我們的書] (The book he sold us)

(93a) $_{NP}$[他賣我們的] (missing direct object)

(93b) $_{NP}$[他賣的書] (missing indirect object)

(93c) $_{NP}$[他賣的] (missing both objects)

(93d) $_{NP}$[賣我們的書] (missing subject)

(93e) $_{NP}$[賣的書] (missing subject and indirect object)

(93f) $_{NP}$[賣的] (missing subject and both objects)

The SLASH feature borrowed from the same famous one in GPSG allows the context-free grammar to deal with unbounded dependencies (or long-distance dependencies) and gaps in language without performing transformation on trees. This is achieved by encoding the gap in the feature structure. Through the unification of feature structures, information of the gap is also percolated up the tree. The NULL feature is used to terminate the SLASH feature mentioned above from endlessly percolating down the tree.

How should we describe (93a) – (93f) with the SLASH feature? Observe that in (93) only the direct object, i.e. "書" (book) can move to become the topicalized object of the resulting sentence (see (94)), but the subject and indirect object cannot (see (95) and (96)). Therefore the missing subject and indirect object can only be treated as

missing constituents, or zero anaphora. They do not move elsewhere, they simply do not exist.

(94) NP[+TOPIC][書]是 NP[他賣我們的]。 (The book is what he sold to us.)

(95) * NP[+TOPIC][我們]是 NP[他賣的書]。 (We are the books he sold.)

(96) * NP[+TOPIC][他]是 NP[我們賣的書]。 (He is the book we sold.)

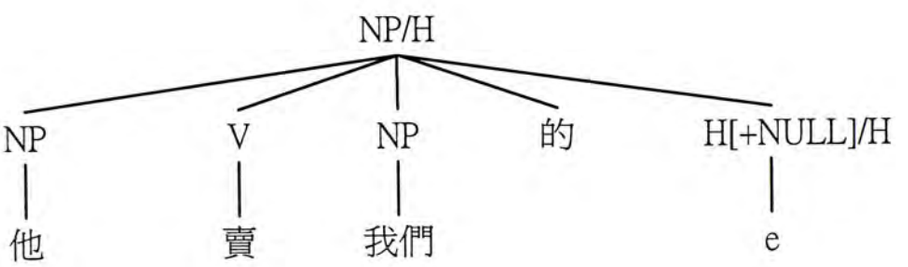
Therefore only (93a), (93c) and (93f) can take the SLASH feature on rule (89) to make (97). (97) is able to describe (93a), (93c) and (93f), as illustrated by trees (98), (99) and (100) respectively.

(89) NP → NP V[3] NP 的 H

⇩

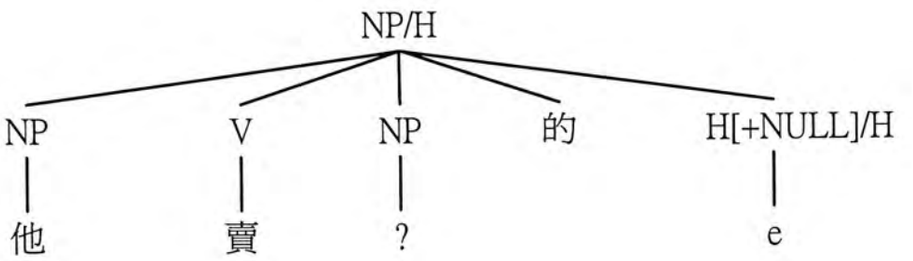
(97) NP/H → NP V[3] NP 的 H/H

(98)

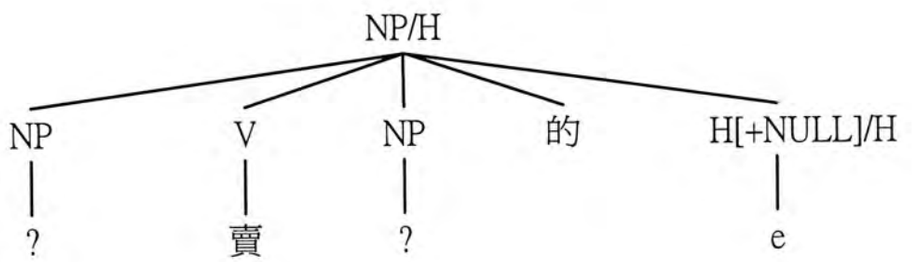


Note that [+NULL] is needed to terminate the SLASH propagation down the tree.

(99)



(100)



6.2.7. Prepositional phrases

Prepositional phrases are common in Chinese, not just only in occurrences, but also the fact that the preposition head can be omitted freely without changing the meaning of the phrase. For example, "球場上躺著兩個人" (On the pitch two people are lying.) actually contains a prepositional phrase "在球場上" (on the pitch), but the preposition "在" is not compulsory. Another example is "學生五月份開始留校溫習" (translation see below), which can mean either (101) or (102).

(101) 學生 PP[自從五月份]開始留校溫習。

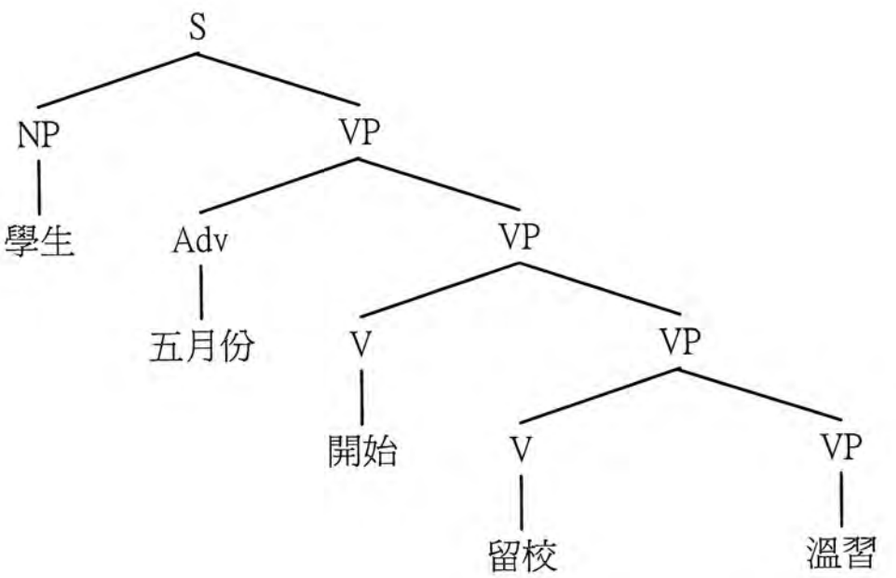
(Students started staying at school to study since May.)

(102) 學生 PP[由五月份]開始留校溫習。

(Students will start staying at school to study from May.)

Although the predicative meaning of (101) and (102) is the same, the tense indicated is different. That means the omission of head in the prepositional phrase can lead to ambiguous semantic readings. However, syntactically we will not be able to deduce the missing head, and so in the case of "學生五月份開始留校溫習" (In May the students start staying at school to study.), we can only render it as (103).

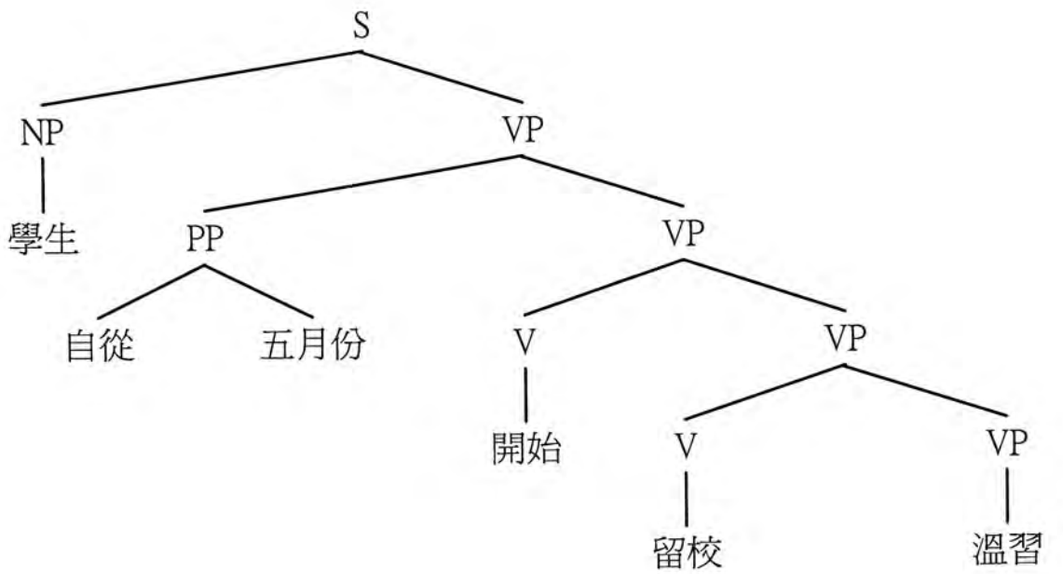
(103)



In the case where the preposition head is not missing, we can easily come up with tree (105) with rule (104).

(104) PP → H NP

(105)



6.2.8. "Ba2" and "Bei4" constructions

The *ba2*(把) and *bei4*(被) constructions are much discussed in the Chinese linguistic literature and have long been treated as special construct of Chinese. Both of them belong to the prepositional phrase family, but they are different in that they "bring" subject or object with them, leaving a gap.

The function of "把" (ba2) is to bring out the object. After "把" (ba2) no monosyllabic verb can appear. There must be at least aspect marker or complement attaching to it if so.

For a simple verb phrase rule like (62), we have the "把" (ba2) transformation of it, which results in rules (106) and (107).

(62) VP → H[2] NP

(106) VP → 把 NP VP/NP

(107) VP → H[SUBCAT 2, +ASP/+COMP] NP/NP

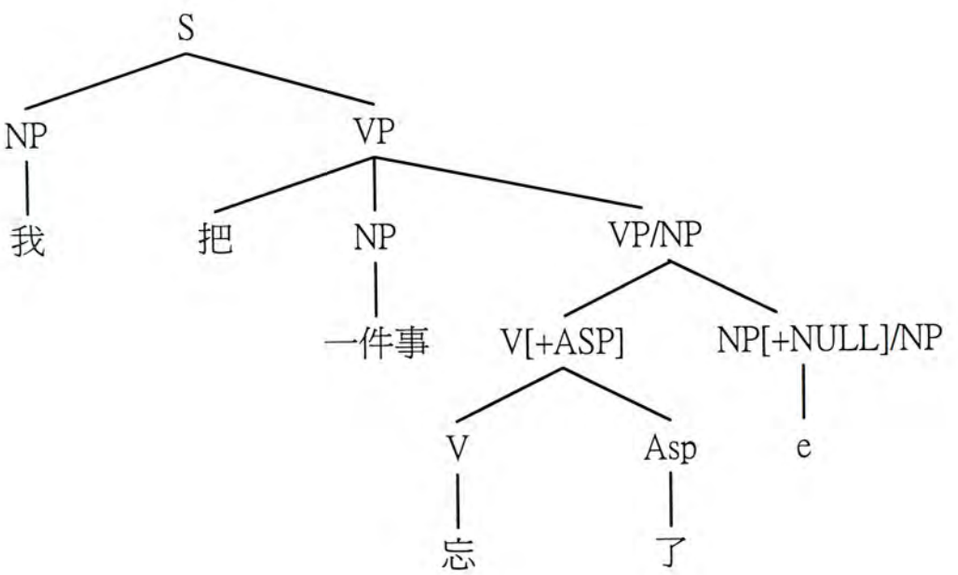
The following example will illustrate how these rules work.

(108) S[我 VP[忘了一件事]]。(I forgot one thing.)

(109) S[我 VP[把一件事忘了]]。

(108) is captured by (62), and (109) by (106) and (107). Tree (110) shows how the SLASH feature is propagated down the tree.

(110)



The function of "被" (bei4) is to bring out the subject, or the agent of action. Unlike

"把" (ba2), there is no strict requirement of the nature of the head verb, whether it is monosyllabic or not is not important.

The "被" (bei4) transformation can be captured by the same set of rules as in the "把" (ba2) construction, only that the character "把" (ba2) in the rule needs to be changed to "被" (bei4).

(62) VP → H[2] NP

(111) VP → 被 NP VP/NP

(112) VP → H[2] NP/NP

The following sentences illustrate the transformation. A realization of (115) is given by tree (117).

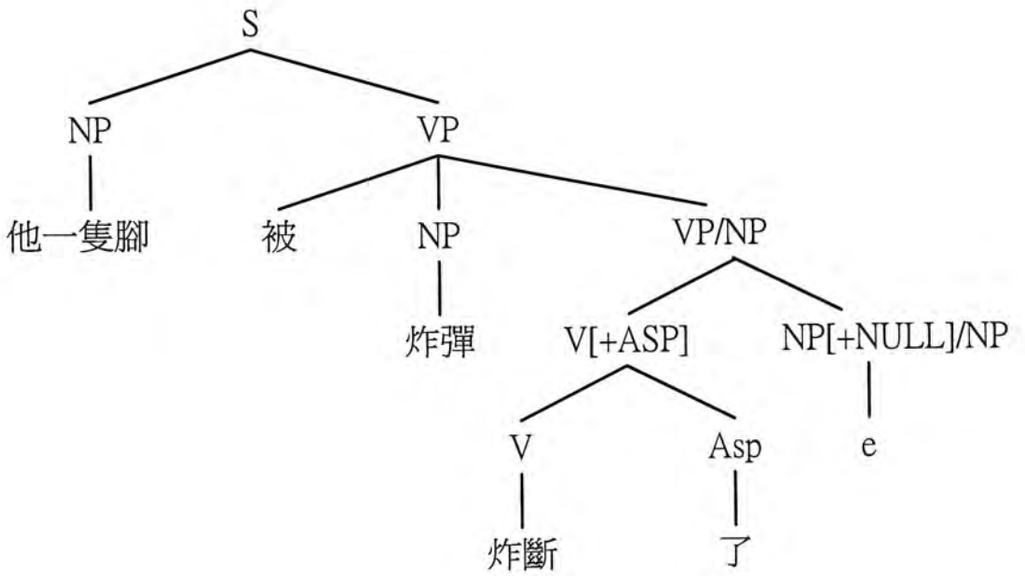
(113) S[NP[炸彈]VP[炸斷了他一隻腳]]。 (The bomb blew away his leg.)

(114) S[NP[遊人]VP[破壞了公園的景觀]]。 (Tourists upset the garden scenery.)

(115) S[NP[他一隻腳]VP[被炸彈炸斷了]]。 (His leg was blown away by a bomb.)

(116) S[NP[公園的景觀]VP[被遊人破壞了]]。(The garden scenery is upset by tourists.)

(117)

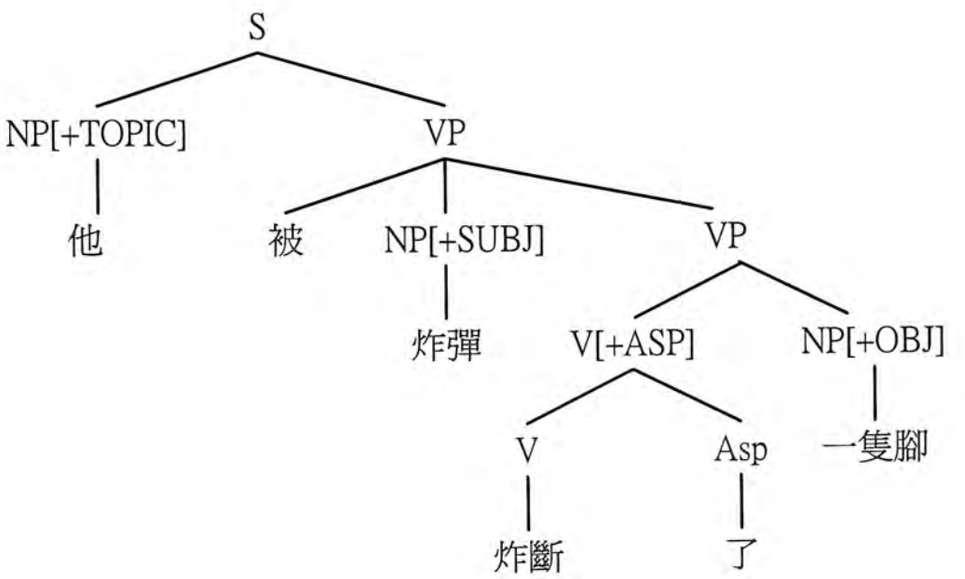


There are cases, however, that the SLASH feature is not needed in "被" (bei4) construction. For instance, (118) is also very common in Chinese.

(118) $s[_{NP}[他]_{VP}[被炸彈炸斷了一隻腳]]$。 (His leg was blown away by a bomb.)

The object "一隻腳" (one leg) is not missing from the VP. Instead, "他" (He) is put in the start of the sentence and become a topic. Thus (118) is a topic-comment construction (as realized by (119)).

(119)



In dealing with such construct, the rule has to be stated clearly which NP serves which post, otherwise we would have wrong interpretation.

6.2.9. The terminal node S

Recall that in SERUP the basic parsing unit is a clause or clause with the subject (so syntactically it is equal to English's sentence), we will see how the terminal node S is formed in SERUP.

The simplest rule to reach a S is (57), now restate as (120).

(120) S[SUBJ NP] → NP VP

The rule says that the NP is the subject of the sentence, unless the subject has already been assigned in the VP node as in "被" (bei4) construction. Again, we see the power of UBG. Only the sentence that fulfills all constraints set by the grammar will be

admitted as a correct one, and this can be done neatly through the power of unification.

Apart from (120), we also need to deal with other clauses that are segmented by the paragraph segmentation module earlier. They are all VP by nature but in fact they are clauses that are linked by zero-anaphora subject. How can they go up a level and become a S? In SERUP this is achieved by the use of punctuation. We have stated several times that punctuations are indeed very important syntactic clue to parsing regardless how ambigious and sporadic they can be.

The attribute PUNC is used to encode the punctuation appears in the input string. Another useful attribute is the CONN attribute that encodes conjunction, disjunction and connective appear in the input string. Rule (121) and (122) illustrate how punctuation and connectives are accounted for.

(121) S[PUNC α] → VP Punc[FORM α]

(122) S[CONN α] → Conn[FORM α] S[-CONN]

Two rules not only encoded the occurrence of punctuation and connective, they also captured the actual lexical form of them through the unification of "α". For example rule (121) forces the punctuation "α" to appear in the PUNC feature of the left hand side S. Furthermore, with rule (121), a clause can "rise" to a sentence if it is known that a punctuation follows it. Rule (122) captures a sentential connective such as "因爲" (because), "但是" (however) and "雖然" (although). In this rule, the right hand side S has the feature [-CONN] which means "no connective". It must be so because most of the time only one sentential connective is allowed in a sentence. In fact, connectives can

appear in a phrase to o and we can account for that easily by adding rules like (122) but working on phrasal level..

6.2.10. Summary of phrasal rules

The following table (table 6.2) gives a summary of the phrasal rules (as appear in the chapter 6.2 The leftmost column refers to the numbered line occur throughout the chapter. The middle columns shows the corresponding phrase structure rule while the rightmost column shows examples of the language data captured by the phrase structure rules.

Number in bracket	Phrase structure rule	Example
57/120	S → NP VP	今天下雨了。
59	VP → H[1]	下雨了。
62	VP → H[2] NP	吃過午飯了。
65	VP → H[3] NP NP	送了一本書給我。
68	VP → H[4] S	說我一定準時畢業。
71	VP[+AUX] → H[5] VP	應該做得到這件事。
75	NP → SpecN, H[BAR 1]	這個先進的國家
76	N1 → AP H	先進的國家
84	NP → VP 的 H	種水果的農夫
88	NP → NP V[2] 的 H	他們種的水果
89	NP → NP V[3] NP 的 H	他們送我的水果
92	NP → Adv[+TIME] V 的 H	今天贏的錢
97	NP/H → NP V[3] NP 的 H/H	他們送我的
104	PP → H NP	由五月份
106	VP → 把 NP VP/NP	把一件事忘了
111	VP → 被 NP VP/NP	被炸彈炸斷了
121	S[PUNC α] → VP Punc[FORM α]	今天下雨了。
122	S[CONN α] → Conn[FORM α] S[-CONN]	因爲今天下雨了。

Table 6.2. Summary of phrasal rules.

6.2.11. Morphological rules

The new word identification algorithm tries to dig out words not appear in the dictionary and give them corresponding feature structures. However, as the process is purely statistical, there would be many cases of wrong generation and

under-generation. We have discussed about how wrong generation can be reduced by utilizing more statistical information in chapter 5.3. For the case of under-generation, it is clear that we need lower level morphological rules in the grammar to capture them.

Packard [93] proposes an alternative X-bar morphology for Chinese, in contrast to Tang and Selkirk [93]. Tang's approach allows excessive over-generation, while Selkirk's approach does not allow some common morphological in Chinese to exist. Packard's approach in essence consists of assigning X-bar values to Chinese morpheme types (root word, bound root, word-forming affix and grammatical affix), and then combining these X-bar elements into words using word-formation rules that generate complex structures but are limited in their generating power [93].

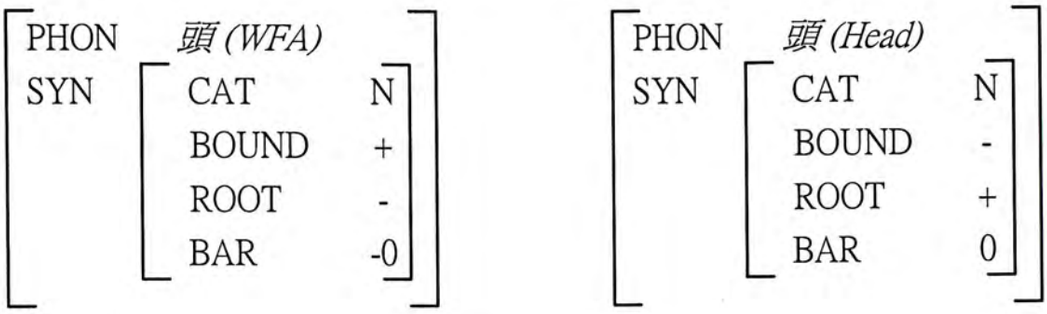
Table 6.2 shows the classification of morphemes in Packard's proposal. Root means the character itself has a meaning, in contrast to function word, which carries primarily grammatical, rather than lexical, meaning and which fulfill mainly syntactic and structural functions [1].

	X-bar designation	may stand alone	may be stems	may be heads	Examples
root words	X^{-0}	yes	yes	yes	人, 書, 房
bound roots	X^{-1}	no	yes	yes	員, 林, 蚣
word-forming affix	X^{w}	no	no	yes	頭, 兒, 子
grammatical affixes	G	no	no	no	了, 著, 過

Table 6.2. Morpheme classification.

That is why we need the features BOUND and ROOT. The feature BOUND is used to tell whether a character or word is a bound morpheme or not. A bound morpheme in Chinese is a character that must combine with another character to form a word. It takes Boolean value "+" (add sign) or "–" (minus sign). The feature ROOT tells whether a character is a root word or not. With these two features we can distinguish the first three types of morpheme in Table 6.2. The last one, grammatical affix is treated as the category post-linking in SERUP, so there is no need to use features to distinguish it. So the feature structure [-BOUND, +ROOT] represents free root words, [+BOUND, +ROOT] represents bound root words and [+BOUND, -ROOT] represents word-forming affixes. For example, the character "頭" (head) will have two lexical entries (see (123), please note that the SEM attribute is omitted here for simplicity), one for the "head" meaning, one for its capability of serving as word-forming affixes, as in word like "老頭" (old buddy).

(123)

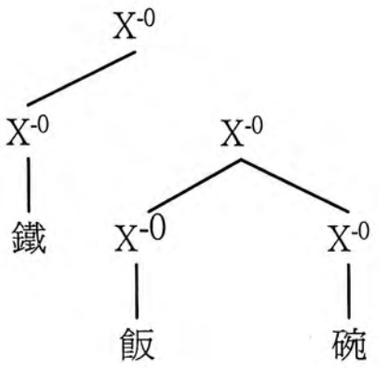


Packard further proposes nine context-free rules to capture possible Chinese word forms. They are shown in (124) – (132) [93].

Number	Morphological rules	Example
(124)	$X^{-0} \rightarrow X^{-0}\ X^{-0}$ (compound word)	冰山
(125)	$X^{-0} \rightarrow X^{-1}\ X^{-1}$ (bound root word)	木板
(126)	$X^{-0} \rightarrow X^{-0}\ X^{-1}$ (bound root word)	電腦
(127)	$X^{-0} \rightarrow X^{-1}\ X^{-0}$ (bound root word)	足球
(128)	$X^{-0} \rightarrow X^{-0}\ X^{W}$ (derived word)	人性
(129)	$X^{-0} \rightarrow X^{-1}\ X^{W}$ (derived word)	鼻子
(130)	$X^{-0} \rightarrow X^{W}\ X^{-0}$ (derived word)	可笑
(131)	$X^{-0} \rightarrow X^{W}\ X^{-1}$ (derived word)	非法
(132)	$X^{-0} \rightarrow X^{-0}\ X^{G}$ (grammatical word)	人們

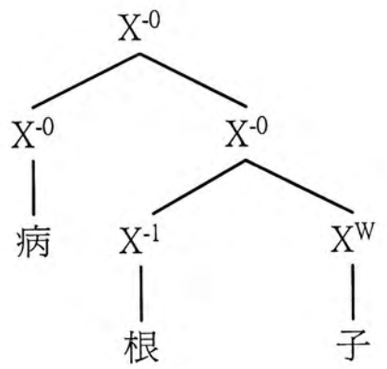
For example, the word "鐵飯碗" has the internal structure (133), which can be captured by (124) recursively.

(133)



The word "病根子" has the internal structure (134), which can be captured by (129) then (124).

(134)



Finally we need a rule to promote X^{-0} to X^{0}, as if it is a common word in the dictionary. (135) will do the final job.

(135) $X^{0} \rightarrow X^{-0}$

Questions certainly arise about the generative power of the morphological rules and the accuracy of determining the final category of the morphological compound. First of all, the rules given by Packard do over-generate. However, the number of new words appearing in input text is limited. Also the statistical new word identification module identifies a majority of new words found. Therefore the morphological rules

will be called very sporadically. The biggest reason why we need the statistical new word identification module is exactly the fact that morphological unification will lead to heavy ambiguities. By far Chinese morphological analysis has not yet discovered the conditions of word formation and so we cannot totally rely on morphological rules. On the other hand, handling new words with complex lexicon is a more structured approach and we should not overlook its contribution to the computational linguistic world.

Secondly, most of the morphological compounds will be assigned the correct category according to the Headedness Principle. Exceptions do exist, but those exceptions are usually long-established dictionary entries. New words mostly are created by contracting and concatenating old words that follow the Headedness Principle. Therefore the assignment of category should not be a big problem.

7. Step Three: resolving structural ambiguities

Structural ambiguities are common in natural language parsing, because human languages, unlike programming languages, are non-deterministic. In practical application structural ambiguities are hazard because they use up a lot of system resources and drastically lower system performance. The situation is even worse in unification-based grammar (UBG) parsing because feature structures created during unification, whether they are correct or not, will be used again during parsing. A general method to reduce structural ambiguities in UBG is to dismiss useless feature structures by introducing more constraints, such that incorrect feature structures would not be used again. However, this is not as straightforward as it suggests.

7.1. Sources of ambiguities

Given the high complexity of natural languages, and the virtually unbounded set of lexicon, it is very common to have ambiguities in daily life utterance. For example,

(136) Look at the first page of the book which is written by him.

There can be two different interpretations due to different prepositional phrase attachment. "He" may have written just "the first page of the book" or "the book". In fact, example (136) shows a classical example of structural ambiguity. Both interpretations are grammatically and semantically correct. Unless we know the context of the utterance, we cannot decide which interpretation should be taken.

Look at another example:

(137) Look at the pages of the book that are written by him.

The same prepositional phrase attachment ambiguity still exists, but it can be solved, because in English the subject-number agreement constraint must be met in a correct sentence. Therefore, we know that "the book" is not written by "him" because "the book" does not agree with "are".

In Chinese, structural ambiguity is very common partly because of the relatively free word order, and partly because of the vast semantic ambiguity. Recall in chapter 3.2.2 that Chinese words can take up multiple grammatical functions without any inflectional changes. In other words, there are few or even no syntactic clues that we can rely on (just like agreement in English or particles in Japanese) in solving structural ambiguities in Chinese.

For example, the following list of constructs ((138) – (157) just a subset of all possible ambiguous constructs) cause troublesome structural ambiguities where syntactic information alone may not be enough to solve. Unless we know the validity of the verb-object collocation, very often there is no way to disambiguate them [39].

(138) PV + NP (PV refers to those words that can be noun or verb)

(139) $_{VP}$[管理 酒店] (VP: To manage a hotel)

(140) $_{NP}$[管理 階級] (NP: Managerial staff)

(141) Preposition + NP + PV + NP

(142) 根據 香港 $_{NP}$[$_{VP}$[管理 酒店] 的 經驗] (According to the experience of Hong Kong's hotel management)

(143) 根據 香港 $_{NP}$[$_{NP}$[管理酒店] 的 經驗] (According to the experience of Hong Kong Management Hotel)

(144) PV + NP + 的 + NP

(145) $_{VP}$[攻擊 $_{NP}$[球員的身體]] (VP: To attack the player's body)

(146) $_{NP}$[$_{NP}$[攻擊球員] 的 表現] (NP: The performance of striker)

(147) NP + PV + PV + NP

(148) $_{NP}$[人民代表] 代表 人民 (People's representative represents people)

(149) *人民 $_{VP}$[代表 $_{NP}$[代表人民]][16]

(150) V + NP + 的 + NP

(151) $_{VP}$[吃 $_{NP}$[羊的肉]] (To eat lamb's meat)

(152) *$_{NP}$[$_{VP}$[吃羊] 的 肉] (Lamb eating meat)

(153) Preposition + V + NP +的 + NP

(154) 由 $_{NP}$[$_{VP}$[到達香港] 的 一刻] (From the moment arriving Hong Kong)

(155) *由 $_{VP}$[到達 $_{NP}$[香港的一刻]] (From arriving Hong Kong's moment)

(156) Topicalized object NP + Subject NP + V

(157) 這本書 $_{VP}$[書店 不賣] (This book is not sold in book stores)

---

[16] The asterisk is a standard notation in linguistics signifying an "illegal" expression to native speakers.

Of course, (138), (144) and (150) is a matter of NP or VP, and given other constituents in the same sentence it is possible to disambiguate them without consulting semantic information. On the other hand, such redundancies (for each wrong interpretation there will be a corresponding feature structure) can be viewed as hazardous to the already slow enough unification parser. In the constructs above, the NP-NP combination and V-NP verb-object selectional restrictions are particularly responsible for the majority of ambiguities arisen.

Solving structural ambiguity is not that straightforward, however. For the "V + NP1 的 NP2" construct (see (150)), three constraints are needed [49]:

Constraint 1: NP1 can serve as patient[17] of V.

Constraint 2: NP2 can serve as patient of V and when NP1 is the patient of V, NP2 is the agent of V

Constraint 3: There is possessive relationship between NP1 and NP2, where NP1 is the possessor and NP2 is the possessed object.

With these extra pieces of information it is possible to decide the syntactic structure allowed earlier. Each of the three examples below (158) – (160) has two possible readings, but one of them is not acceptable.

(158) 咬死了農夫的雞

(158r1) $_{VP}$[咬死了 $_{NP}$[農夫的雞]] (VP: Bit the farmer's chicken to death)

---

[17] Patient is the term denoting the "recipient of an action" in case grammar.

(158r2) *NP[VP[咬死了農夫] 的 雞] (NP: The chicken that bit the farmer to death)

(159) 咬死了狐狸的狗

(159r1) *VP[咬死了 NP[狐狸的狗]] (VP: Bit the fox's dog to death)

(159r2) NP[VP[咬死了狐狸] 的 狗] (NP: The dog that bit the fox to death)

(160) 賣掉了農夫的狗

(160r1) VP[賣掉了 NP[農夫的狗]] (VP: Sold the farmer's dog)

(160r2) *NP[VP[賣掉了農夫] 的 狗] (NP: The dog that sold the farmer)

(158r2) cannot fulfill constraint 2, so (158) can only be a verb phrase. (159r1) cannot fulfill constraint 3, so (159) can only be a noun phrase. Finally, (160r2) cannot fulfill constraints 1 and 2, so (160) can only be a verb phrase.

Some grammatical frameworks, such as case grammar, attempts to locate the thematic relationships between phrasal constituents as the basis of language analysis, and thus avoiding the ambiguities arisen from structural analysis. However, before the correct labeling of cases can be done, we need to identify the phrasal constituents and the relationships among the constituents, which in the case of Chinese, is often difficult by using only syntactic evidence.

7.2. The traditional practices: an illustration

In solving the ambiguities mentioned in chapter 7.1, word semantics is a crucial ingredient. With carefully crafted semantic information, we can cross out all those parses that contain non-allowable word compounds. For example, we can have (161)

but not (162) in Chinese, because "打敗" (to beat) can only take an animate object as argument.

(161) $_{VP}$[打敗了敵人] (Defeated the enemy)

(162) *$_{VP}$[打敗了木材] (Defeated the wood)

As a result, we know that (163) has really got two possible meanings (163r1) and (163r2), but (164) has only one (164r1), because "木材" (wood) cannot be "打敗" (beaten).

(163) 打敗了敵人的車隊

(163r1) $_{VP}$[打敗了 $_{NP}$[敵人的車隊]] (Defeated the enemy's automobile troop)

(163r2) $_{NP}$[$_{VP}$[打敗了敵人] 的 車隊] (The automobile troop that defeated the enemy)

(164) 打敗了木材的車隊

(164r1) $_{VP}$[打敗了 $_{NP}$[木材的車隊]] (Defeated the enemy's wood automobile troop)[18]

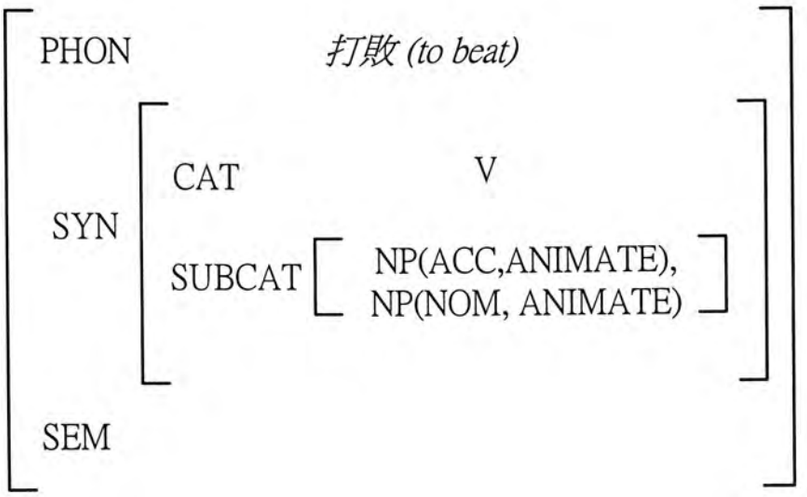
(164r2) *$_{NP}$[$_{VP}$[打敗了木材] 的 車隊] (The automobile troop that defeated the wood)

One approach to locate (164r2) as wrong analysis is to assign semantic classes to the nouns "敵人" (enemy) and "木材" (wood). For example, "敵人" (enemy) can be

[18] Although the expression is also strange, it is nonetheless acceptable in many contexts, which means cars that transport wood or cars belonging to wood company.

categorized as ANIMATE and "木材" (wood) as MATERIAL. Then in the lexicon we may have

(165)



It restricts "打敗" (to beat) to take only animate object. Thus "打敗木材" (to beat wood) is blocked.

7.3. Deficiency of current practices

The componential analysis of structural semantics that has been widely used and is still a core part of many existing parsers and NLP systems, however, did not work as well as they were believed in modern computational linguistics point of view.

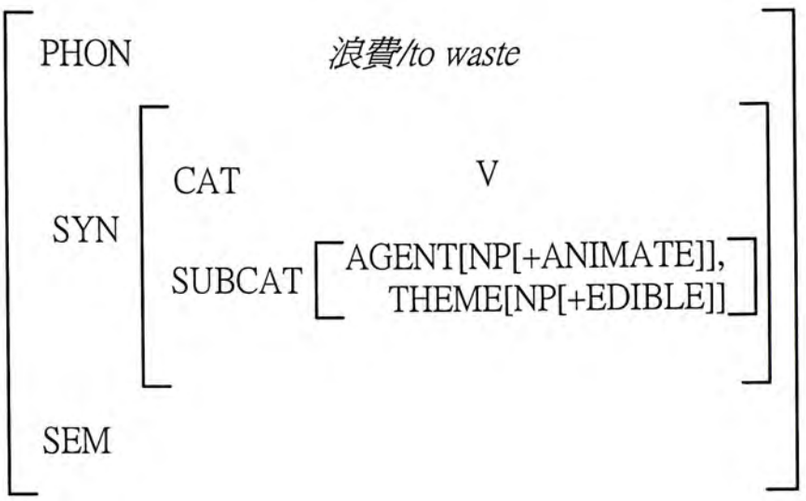
In structural semantics, classifications were made by careful analysis of the concept, the intrinsic property, the seme of a word, and how words having compatible semantic relation (e.g. antonymy, hypernymy, hypnoymy) can be grouped together as distinct semantic classes in the huge semantic features hierarchy. Different classification schemes have been suggested for nouns. For example, Pun [50] and Mo [51] provide several noun hierarchies. They share a common characteristics that the classification

of concrete nouns are detailed, but that of abstract nouns are ambiguous and unclear.

Upon careful investigation over componential analysis, we came to a conclusion that it is not the best semantic representation for the sake of accurate disambiguation. First of all, the foundation of componential analysis is not solid enough. It operates on the assumption that it is possible, even in semantics (not just in phonetics), to describe the whole lexicon of a language with limited inventory of universally valid features [1]. However, what we have seen so far is not comprehensive enough to confirm this assumption. It is hard to judge how many different semantic features are needed to distinguish one word from another. One may claim that deriving semantic features for physical object is easy, as supported by the good examples mentioned above, but too often one fails to do so with abstract nouns, particularly those related to emotion, empathy and feeling.

Semantic features alone are not helpful enough in crossing out invalid verb-object (V-NP) collocation, not to say performing semantic restriction required by case grammar. For example, in the conceptual structure [51] of Information-based Case Grammar for Chinese [27], each verb entry possesses subcategorization information. Assuming the verb 浪費 (to waste) to have feature structure like

(166)



and given a list of language evidence:

(167) $_{VP}$[浪費食物] (to waste food)

(168) $_{VP}$[浪費飲品] (to waste drinks)

(169) $_{VP}$[浪費紙張] (to waste paper)

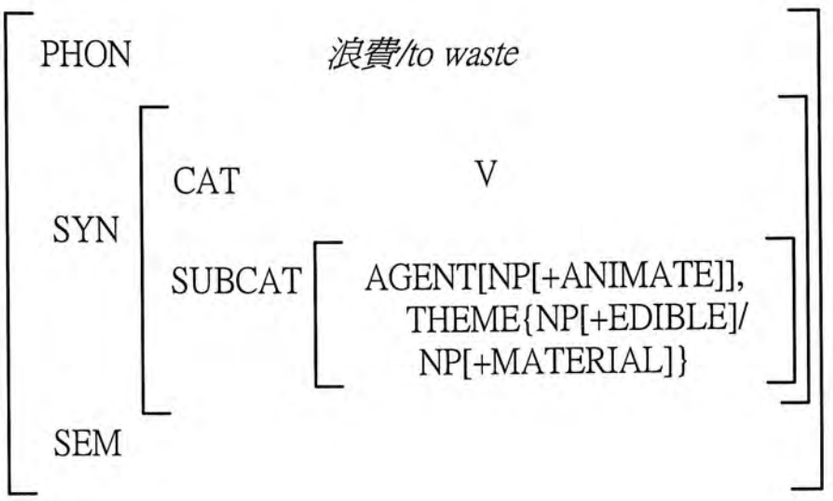
(170) *$_{VP}$[浪費人類] (to waste human)

(171) $_{VP}$[浪費金錢] (to waste money)

Phrases (167) and (168) can be nicely captured. However, if so (169) and (171) are left out[19]. The feature structure can be modified for such case:

[19] It is arguable whethe r紙張 (paper) and 金錢 (money) can be classified as [+EDIBLE], or more reasonably as [+CONSUMABLE]. Examples (167)-(169) & (171) can then be captured with one semantic feature. But (170) remains troublesome.

(172)

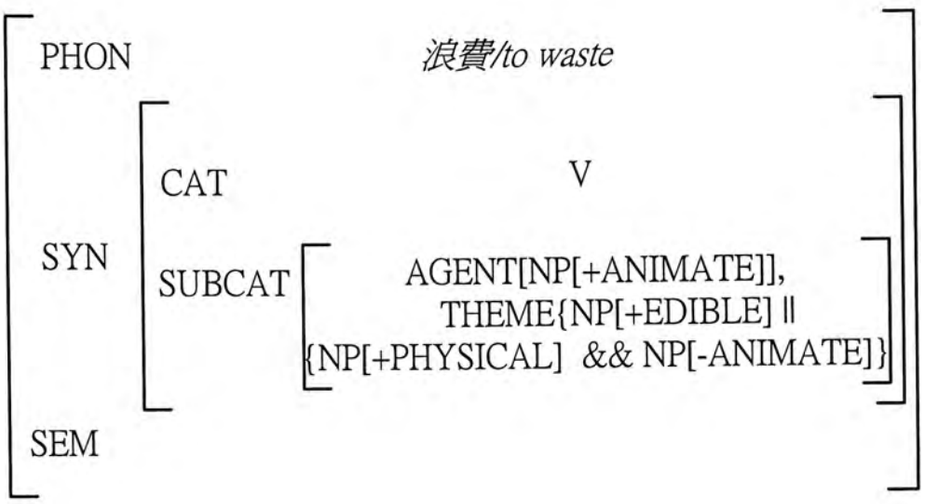


Here "紙張" (paper) can be [+PHYSICAL] as well, but note that [+PHYSICAL] also means that every semantic feature underlying [+PHYSICAL] in the semantic class hierarchy will be admitted as well. As a result (140), which is not permitted, is most probably considered correct under such scheme, because "人類" (human) is [+PHYSICAL]. In fact, all substances that bear a physical shape will be under [+PHYSICAL]. We will turn to this issue again in due course.

So it is clear enough that just one semantic feature alone in the verb entry can only capture a small subset of all possible constructions allowed by the language.

What if we allow more than one semantic feature, say [-ANIMATE], to the lexicon? Look at this:

(173)



Now (167)-(171) excluding (170) are accounted for correctly. But how about "人才" (talent, [+ANIMATE]) in "浪費人才" (to waste talent)? We also observe in Chinese that

(174) $_{VP}$[浪費心思] (to waste idea)

(175) $_{VP}$[浪費精神] (to waste energy)

(176) $_{VP}$[浪費時間] (to waste time)

(177) $_{VP}$[浪費青春] (to spoil youth)

are valid expressions. The question is: how many more semantic features do we need to distinctly classify the abstract object nouns in (174)-(177)? At this stage it may still look easy to do so. Let's elaborate on (176). "時間" (time) is most likely classified as a non-physical noun having [+TIME] semantic feature. In the same category we found "時份" (a point in time), "冬至" (winter solstice), "時段" (a period), etc. Unfortunately all these three words having [+TIME] feature do not form valid VP with "浪費" (to waste):

(178) *$_{VP}$[浪費時份] (to waste a point in time)[20]

(179) *$_{VP}$[浪費冬至] (to waste winter solstice)

(180) *$_{VP}$[浪費時段] (to waste a period)

Of the four [+TIME] nouns mentioned, only "時間" (time) collocates with "浪費" (to waste); however for the verb "渡過"(to pass), only "冬至"(winter solstice) can collocate with it[21].

What is worse is that such situation is common in Chinese. Let's look at another example. The nouns "缺點"(shortcoming) and "缺陷"(defect) belong to the same semantic class (their meanings are very similar), but they do not collocate with the same verb "改善"(to improve) and "修補"(to mend). See

(181) $_{VP}$[改善缺點] (to improve one's shortcoming)

(182) *$_{VP}$[改善缺陷] (to improve one's defect)

(183) *$_{VP}$[修補缺點] (to mend one's shortcoming)

(184) $_{VP}$[修補缺陷] (to mend one's defect)

Shortcoming can be improved, but not defect, which is innate; on the other hand, shortcoming cannot be mended. This implies that if the nouns are to be distinguished, a new semantic feature must be added. Otherwise there will be a massive over-generation of VPs. So we may add a feature, say [+INNATE], to "缺陷" (defect)

---

[20] It should be noted that, the definite reading of the object noun, e.g. "這個時份" (this point in time) can collocate with "浪費" (to waste). The same applies to (179) and (180).

[21] It is again arguable that the nouns in (174)-(177) can bear [+CONSUMABLE]. However, it is hard to judge whether the nouns in (178)-(180) can take the same feature as well.

and modify the feature structure of 改善 (improve) to take only [-INNATE] nouns[22]. However, whether 缺陷 (defect) should bear the [+INNATE] feature is questionable. The problem here is that we do not know which and how many more semantic features we need to handle every single case. First of all, there is no standard method to conduct componential analysis. Secondly, we have stated earlier that the set of semantic features is infinite, and their denotations are highly changeable in the real world. At the same time we cannot ensure there is no further over-fitting after adding more semantic features. As there is no universal standard in dividing seme, crossing of meaning must exist among semantic features, and thus leads to over-fitting (see Figure 7.1).

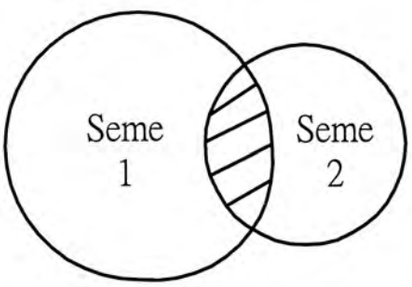


Figure 7.1. Some pairs of seme have intersection.

7.4. A new point of view: Wu (1999)

Wu [39] suggests in his book <<Chinese Computational Linguistics--analysis of relationships, semantic relationships and formality>> that the traditional semantic analysis does not help much in modern Chinese computational linguistics, particular in the sense that they often fail to distinguish the subtlety of each individual member

---

[22] The scenario is actually more complicated as "改善" (to improve) can collocate with "伙食" (meals), "環境" (environment), "條件" (condition), etc.

in a defined semantic group. A much better approach should be utilizing the "relational seme" (We will call it R-seme from now on) of each valid verb-object VP to be the basis of disambiguation. R-seme is defined as the horizontal (not vertical/hierarchic) relationship between two constituents (in this case the verb and its object), which can be derived from language data or dictionaries. We do not need to worry whether there is over fitting or not because R-seme works for a particular pair of language sample. In short, the difference between seme and R-seme is exactly the one between semantics and computational semantics.

The difference between seme and R-seme is that seme is structured and relates all nouns (or verbs, adjectives) in a big semantic hierarchy. R-seme arises only from real instances of language use. It does not aim to relate anything into a system. It only tells what kinds of collocations are allowed in a language. Thus it represents genuine language experience, and it does not allow any invalid semantic combination to happen.

For example, using the same example "改善缺點" (to improve ones' shortcomings) as above. Wu's principle is that, when two words can collocate with each other, there must be a common R-seme. In this case, the common R-seme of "改善" (to improve) and "缺點" (shortcoming) is "to improve/can be improved"[23]. In the unification process, feature structures of the words are checked. If a common R-seme is found, unification succeeds. "缺點" (shortcoming) may have many more R-seme, depending on the number of verbs that can collocate with it, e.g. "認識" (to recognize), "體諒"

[23] In real implementation, our R-seme is just "can be improved", because "to improve" is simply the complement.

(to understand), "隱藏" (to hide), etc. However, they do not interfere a successful unification with "改善" (to improve) nor over-fit as long as the common R-seme exists in the both lexical entries.

Wu supports his scheme by performing a careful empirical study on more than 2000 Chinese verbs and 3000 nouns. He discovered that the seme (not R-seme) of nouns are very complicated and fuzzy (exactly the same observation we have discussed in chapter 7.3) but seme of verbs are simple and straightforward. As a result disambiguation using noun semantic classes is ineffective and not fruitful. Therefore verbs should be emphasized instead of nouns in disambiguation. Here R-seme distinguishes as an appropriate candidate for verb-semantic-based disambiguation because it is defined according to the verbs semantics. In fact most disyllabic verbs possess only one R-seme (unlike monosyllabic verbs that can have several meanings). Only few of them possess two or more R-seme. Monosyllabic verbs are more complicated, but nonetheless verb semantics are simpler and more definite than nouns.

7.5. Improvement over Wu (1999)

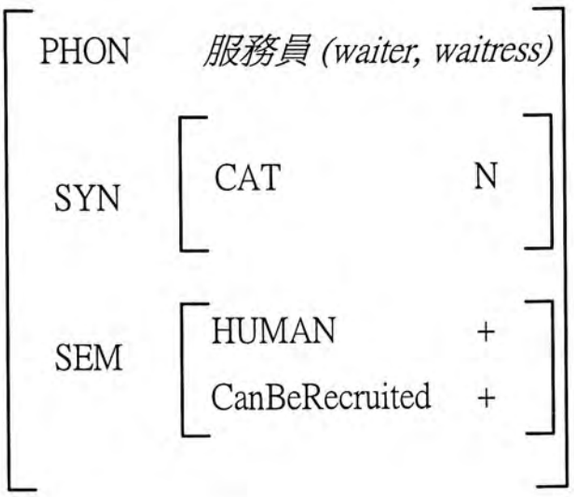
Although Wu's approach saves our parser from over-fitting, assigning R-seme to the lexicon can be troublesome, complicated and tedious, particularly for nouns that can collocate with many verbs. For example, out of the 3000 verbs under study, more than 480 of them can take nouns in the human class, like "學人" (scholar), "員工" (worker), "小朋友" (young buddy), etc. Under Wu's scheme, each of these nouns will bear more than 480 R-seme! The lexicon will become far too big and hence reduces processing speed.

In dealing with such problem, we use an intuitive yet effective method: to combine the structural approach with Wu's one based on collocation evidence in real language data. We discover that many non-abstract nouns are relatively easier (or even very easy) to describe in component analysis. For example, those nouns refer to human[24], animals, organization, and the morphological nouns created with these as semantic head. Verb-object collocation of them is stable, unlike the case of abstract nouns as illustrated in the examples mentioned above. So seme and R-seme each occupy one side of the balance, having just one scheme of semantic features for all verb-object VPs simply does not work well. Seme works fine for a subset of non-abstract nouns, while R-seme works where seme fails. In practice, the combination of the two principles leads to a faster yet reliable verb-object VP screening.

For example, the noun "服務員" ((this) waiter/waitress) will be assigned [+HUMAN] only, unless there is a verb that takes its meaning as an occupation. For instance, the verb "招聘" (to recruit) takes member of the class "occupation" as objects. So we add the relevant R-seme to "服務員" ((this) waiter/waitress).

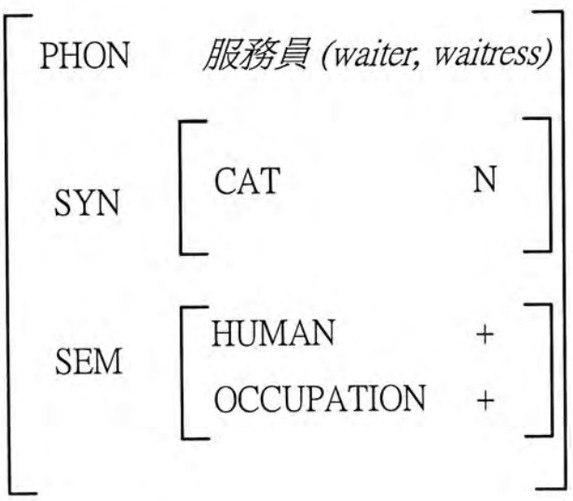
[24] To be exact, the nouns with [+HUMAN] must also be [+DEFINITE], i.e. definite nouns.

(185)



Or we can assign another big class [+OCCUPATION] to it, as in (186)

(186)

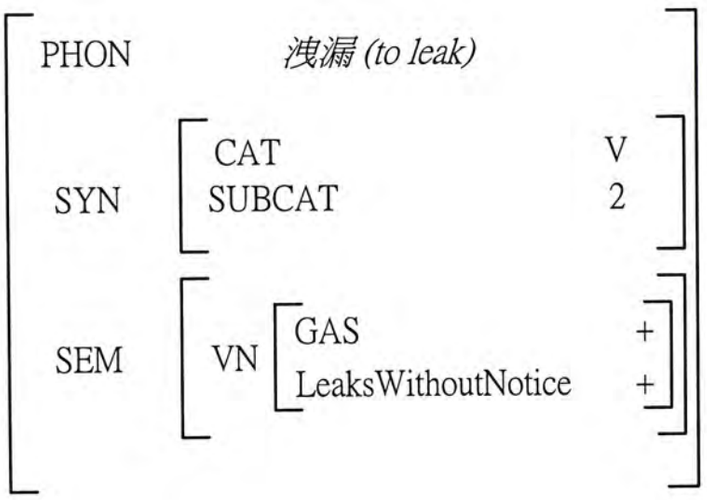


It seems easy to deal with verb like "招聘" (to recruit) which has a very narrow meaning, and very luckily the nouns it collocates with can also be classified without doubt. Of course, when there exist another verb that takes "服務員" ((this) waiter/waitress) as object, we can add the relevant R-seme to the lexical entry, or give another semantic class label to it if the nouns under the same label behave the same as it. What is different from the addition of noun semantic class semantic feature as in

chapter 7.3 is that R-seme is finite (at least, at any point in language development), but semantic features are not. In fact, this is the most important difference. R-seme can ensure complete closure at any point of language development but semantic features cannot. The example above does not illustrate the full power of R-seme, but we will see more of it in the following.

Take the verb "洩漏" (to leak) as another example. It takes members of the general class "gas" as its object, so it has [+GAS] feature. We also notice that many nouns can collocate with "洩漏" (to leak), like "幅射" (radiation) and "機密" (something confidential). In this case, we add a R-seme [+LeaksWithoutNotice] to the lexical entry as such:

(187)



Here "機密" (something confidential) is an abstract noun, so it is best to use R-seme. "幅射" (radiation) is not abstract, but it is difficult enough to find an appropriate semantic class for it. Therefore, we can also use R-seme to simplify the problem.

Sometimes, even though we are quite sure about the logic of physics, for example, we know that the verb "打爛" (to break) can apply to all physical objects that have

concrete shapes. While this is the truth in physics, in language we may take another point of view. For instance, the validity of (188) does not imply a universal truth that [+PHYSICAL] is a good enough semantic class to represent the objects take can collocate with "打爛" (to break), simply because we can find exception like (189) so easily. "紙張" (paper) can only be "撕爛" (to tear off) .

(188) 打爛牆壁　(to break the wall)

(189) *打爛紙張　(to break the paper)

A possible solution is to restrain [+PHYSICAL] to [+CONCRETE], and take away paper from the [+CONCRETE] class.

7.6. Conclusion on semantic features

There may be controversy over the choice of semantic classes in our approach, but one must note that controversies exist in all disciplines of linguistics (and so computational linguistics) as well. However, there is a principle for choosing distinguished semantic classes for disambiguation. First, the class must represent a reasonably large set of nouns. Second, all members in the same class should collocate with at least one same verb. Any candidate noun not fulfilling these two requirements should be taken out from its candidate semantic class. Just keep in mind that the use of semantic class labels is to mediate the ad-hoc nature of R-seme, so any class label that covers a reasonably large amount of nouns is fine. The relationship between semantic class and R-seme is shown in Figure 7.2. The more semantic classes we have, the fewer R-seme we need, and vice versa. Note that the no matter how many semantic classes are there we still need R-seme to fill the remaining portion.

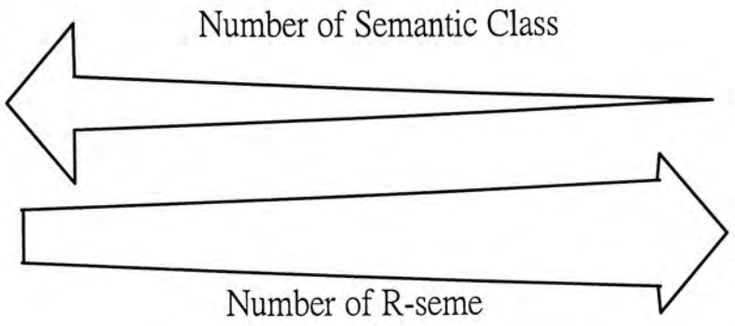


Figure 7.2. Relationship between semantic class and R-seme

Although in reality defining R-seme may not be as easy as illustrated, it is still far easier to be handled than traditional semantic classes. The reason is that in theory R-seme can describe all possible collocation relationships in ANY static point of language development, but semantic class cannot. We have proven that with a number of strong counterexamples in this chapter. In fact, if we are allowed to focus on a specific domain, R-seme will excel as the best semantic markers. In specific domain the collocation relationships are limited and can be predicted easily as well. For a bigger context however, a more extensive study of the combining use of R-seme and semantic class seems imminent.

8. Implementation, performance and evaluation

8.1. Implementation

The C and PERL programming languages are used extensively in the implementation of SERUP. The segmentation program is solely written in C, while the other modules in automatic preprocessing are written in PERL. The reason of choosing C is that it is a very fast programming language and in fact many state-of-the-art NLP researches[25] use C as the core programming language. PERL is a widely used script language in Unix that is well known for its extensive regular expression support. Nowadays there is Windows version available too.

The unification parser in SERUP is implemented using the PC-PATRII syntactic parser [56], a modern implementation in C language of the famous PATR-II formalism by Shieber [12]. PATR-II style grammar rules are easy to implement, readily expandable, and are easy to trace and maintain. The main reason why we adopted using PATR-II is that the program has a tremendously fast response time, which is commonly not the case in the traditional Prolog-based implementation.

However, PATR-II is only a linguistic tool rather than a linguistic theory. Only simple unification statements and a context-free backbone are allowed in the language. Therefore it is not straightforward to represent a grammar with PATR-II. In fact, in

[25] The German Verbmobil project uses mainly C, plus Prolog, Tcl/Tk, and Java.

order to fit our grammar to PATR-II, we need to do a lot of "translation" job. For example, the GPSG style grammar rule

(190) S[+SUBJ] → NP VP

needs to be rewritten like this:

(191)
```
Rule {Simple declarative Sentence}
     S → NP VP                                  /* line 1*/
       <S SYN> = <VP SYN>                       /* line 2*/
       <VP SYN SUBJ> = <NP>                     /* line 3*/
       <S SEM> = <VP SEM>                       /* line 4*/
       <VP SEM SUBJ> = <NP SEM SEME>  /* line 5*/
```

The feature percolation needs to be written down explicitly, which shows the clumsiness of PATR-II. For example, line 1 is for the context-free rule, and we can see it operates only on categorical level. Line 2 says that the SYN attribute of S and its head, i.e. VP, must be the same. Line 3 says the subject of the sentence is NP. If the subject has already been identified in VP (for example by the "被" (bei4) construct), unification will fail here and we immediately know that the NP is not a subject. Line 4 forces the unification of the SEM attribute. And finally line 5 checks whether the subject NP and the predicate VP have a valid semantic collocation.

On one hand, translating grammar rules to PATR-II is very troublesome; on the other hand it does allow the grammar writer to understand the real "flow" of feature value during the unification process. So debugging is straightforward and efficient.

Appendix I gives an example run of SERUP. It shows how the input text is processed in each phase of SERUP. Appendix II and III are some sample grammar rules and lexicon in PATR-II formalism that we have implemented.

8.2. Performance and evaluation

8.2.1. The test set

The test set consists of 5 economic articles extracted from our article database containing various articles of Hong Kong newspapers in a period of 3 months from April 2000 to July 2000. Each article contains 30 comma-separated segments, averaging 750 characters.

8.2.2. Segmentation of lexical tokens

The segmentation program in SERUP scores a very high accuracy. The segmentation dictionary[26] we are using has 132340 terms, where 33423 are proper names (human names, organization names and geographical location), 23234 are standard dictionary entries, 5342 are frequently used phrases, and 2342 are others.

The segmentation accuracies are collected by finding the number of wrong segmentation divided by the total number of segments. Table 8.1 shows the result. All the accuracies obtained are compared to human analysis.

---

[26] The segmentation dictionary differs from the UBG lexicon in that the entries in the former one are not identified with feature structure, but surface phonological forms only. This is for the sake of fast computation only. The UBG lexicon should have the number of entries as the segmentation dictionary.

Article set	Accuracy
Set 1	96.24%
Set 2	98.53%
Set 3	97.81%
Set 4	96.26%
Set 5	98.84%
Overall	97.54%

Table 8.1. Segmentation result.

It can be seen from the result that lexical segmentation is still a bottleneck although the overall accuracy of the segmentation program reaches 97%, which is a reasonable figure in practical application. An accuracy of 97% simple means that there is 3% of errors and in human language 3% of errors can mean a lot! That means we have to ensure a perfect segmentation.

The major source of error is ambiguities. In chapter 3.2.1 we have seen some classical examples of segmentation ambiguities. SERUP is currently using relative frequency count in solving ambiguities. For example, in (192) "這個" (this) and "個人" (individual) are candidate segments, but "這個" (this) has a higher frequency count than "個人" (individual) in corpus, so "這個" (this) is taken as the final segment, as in (192a).

(192) 這個人的品行欠佳。 (The conduct of this guy is not good.)

(192a) 這個 人 的 品行 欠佳[27]。

[27] If "這個人" (this guy) appears as a term in the lexicon, problem arisen due to (192) will not happen.

However, frequency count is just a simple heuristic. It does not guarantee correct answer, particularly when the frequency counts of the candidate segments are very close. For example in (193) "從中" (from) has a lower frequency count than "中學" (middle school), which leads to wrong segmentation ((193a)). Correct segmentation should be (193b).

(193) 總裁從中學到了不少金融知識。

(The president learnt many things about finance (from the incident)).

(193a) *總裁 從 中學 到了 不少 金融 知識。

(193a) 總裁 從中 學到了 不少 金融 知識。

8.2.3.New word identification

The performance of our new word identification algorithm is easy to compare with Wu and Jiang [42] original one, because their algorithm is well written in their paper. We randomly injected new words not exist in segmentation dictionary to the segmented articles, and then run the new word identification programs. Table 8.2 shows the result. The figures inside the bracket are the result of Wu and Jiang's original algorithm.

Article set	Number of new words	Number of correctly identified new word	Percentage of identification
Set 1	10	8(5)	80%(50%)
Set 2	15	10(6)	66.67%(40%)
Set 3	20	12(14)	60%(70%)
Set 4	30	24(20)	80%(66.67%)
Set 5	40	33(28)	82.5%(70%)

Table 8.2. New word identification algorithm performance.

Our algorithm generally gives a higher number of correct identification, unlike Wu and Jiang's original one which easily leads to over-generation. In fact, in set 3 their algorithm outperformed us. This is because set 3 contains a number of new organization names, and our algorithm does not work very well in this area. Our algorithm passes unsure cases to the parser, so it does not matter whether 100% accuracy can be obtained.

There are some cases that new words are correctly identified by Wu and Jiang's algorithm, but not ours. For example,

(194) 他們 都 長得 牛高馬大。 (They have become tall and big boys.)

Wu and Jiang's algorithm can capture "牛高馬大" (tall as cow, big as horse) because all four characters in it are very likely to appear as bound morphemes. Wu and Jiang always identify such possible bound morpheme sequence with great flexibility. However, our algorithm does not admit that because we also check whether the characters fit the positions they used to appear.

Our algorithm successfully capture the new word "細審" (screen carefully) in (195), rather than the illegal word "組細審" as identified by Wu and Jiang.

(195) 電視節目 要 先 經 節目 審查 組 細 審 後 方可 播出。
(TV shows must be carefully screened by the program screening board before broadcasting.)

This is because "組" (group) has a higher tendency to fit the head noun position of a word, "組細審" will be filtered out in the second round statistical checking in our algorithm. This shows an advantage of our algorithm over Wu and Jiang.

Both algorithms fail to identify "美的空調" (Mei3di4 air conditioner) in (196) because the grammatical word "的" appears as part of a new word.

(196) 大陸 居民 多 愛買 美的空調。 (Most Mainland residents like to buy Mei3di4 air conditioner.)

As both algorithms rely on free morpheme to determine the possible bound morpheme sequence, "美" (bound adjective) and "的" (free grammatical word) will not combine. Rather the parser would render it a modifier-head structure, with "美" (beautiful) filling the modifier slot of the head noun "空調" (air conditioner).

However, the new word identification algorithm has shortcomings. Currently our algorithm only employs syntactic knowledge and it is clear that semantic knowledge is also indispensable in determining the compatibility and meaning of new words.

Moreover, much more theoretic works are needed because new word identification may be best handled by the unification grammar. Li [41] proposes a character-based HPSG parser for Chinese, but it is only the very beginning. Lots of works are waiting to be done.

Furthermore, there is much more to be done in correctly identifying transliterated names. Sproat et al. [33] presents a finite-state machine approach to handle them. They treat Chinese names and transliterated names the same as common character sequence. Probabilities of a character appearing as part of a transliterated name (Or Chinese name) are obtained in corpus. Whether a character sequence is taken as a transliterated name or not depends on the product of the relevant probability of each character in the sequence, as well as the probabilities of being other possible compounds. Although such approach is clean and straightforward, they do not take into account the fact that a foreign syllable can be rendered as all Chinese characters that bear the same or very similar sound in either Cantonese or Mandarin. For example, "森柏斯" and "森派斯" both refer to the tennis player Pete Sampras. Or we can even write the name as "心拍絲", which has the same pronunciation as "森柏斯" but looks different in characters. Table 8.3 shows some examples of multiple transliterations.

Original name	Chinese transliteration
Beckham	碧咸, 貝克漢姆, 碧鑑
Gregory Peck	格哥利柏, 格利哥利柏, 佳哥利柏
Christopher	吉里斯托佛, 克利斯多弗
Jordan	喬登, 佐敦, 僑登, 佐丹

Table 8.3. Examples of multiple transliterations in Chinese

Usually such kind of subtle difference is hardly found in a corpus. As a result, many character sequences will score low even though their pronunciations are similar using Sproat et al. [33] algorithm. On the other hand if we try to list out all possible (impossible in real) transliterations in the segmentation dictionary it will easily explode in size. Therefore, it is clear in dealing with transliterated names we also need to take phonetic information into account. A possible solution is to employ technique in speech recognition. In such case similarity measures are given to each sequence. If the sequence is close enough in pronunciation to a name in segmentation dictionary, we will take that as a transliterated name. We are not going to divulge into the details here.

8.2.4. Parsing unit segmentation

Parsing unit segmentation relies on POS-tagging result. Table 8.4 shows the POS-tagging accuracy. Currently the overall accuracy is not very high because the training set is not big enough. We foresee a drastic improvement after enlarging the training set.

Training set size (article)	Tagging accuracy
30	84.6%
50	86.1%
100	92.5%

Table 8.4. POS-tagging accuracy.

Of course the accuracy of the POS-tagger directly affects the accuracy of clause segmentation discussed in chapter 5.4. Table 8.5 shows how the tagging accuracy

affects it.

Article set	POS-tagger accuracies	Clause Segmentation Accuracies
Set 1	84.6% / 86.1% / 92.5%	89.3% / 93.1% /96.6%
Set 2		87% / 91.6% /96.5%
Set 3		88.5% / 92.3% /96.7%
Set 4		90.3% / 97.2% /97.8%
Set 5		89.3% / 92.1% /96.2%

Table 8.5. Clause segmentation accuracy.

From the results it can be seen that low POS-tagger accuracies do not necessarily imply low clause segmentation accuracies. The reason is that some of the errors in POS-tagging do not participate in determining clause boundary. Recall in chapter 5.4 that we only make use of the several tags before and after the punctuation marks in the implementation. Therefore, errors appearing outside the required tags have no effect at all. In general, the higher the POS-tagger accuracy the higher is the clause segmentation accuracy.

In general, POS-tagging is rather reliable. With a good training corpus, POS-tagging can easily achieve a correctness percentage of higher than 97%, which seems enough for our parsing unit segmentation routine, as long as the errors do not appear in the checking patterns. In recent years, however, people have been arguing whether the concept of part-of-speech (and thus relating to POS-tagging) is relevant, particularly to the Chinese language. This foresees a major change in the paradigm of NLP. Abandoning POS means a total change in parsing and system implementation. This is too big and serious an issue to discuss here. In the meantime we will keep improving

the tagging accuracy by utilizing different statistical models.

Segmentation of parsing unit is not yet perfect. The first reason is due to the accumulative errors in the first two phases. We have run tests with the same five articles manually corrected after segmentation and tagging. The accuracy of the third phase reaches at least 99% after such human intervention. The second reason, which is the more crucial one, is that sentence patterns are infinitive. We do not know how many rules we need to cover all patterns. For example, in the sequence "N-V，V-V。" (as in "他說，不要前進。" (He says, do not go forward.)), the V-V turns out to the argument of the "V" in "N-V". Therefore the two comma separated segments should be treated as one unit, but in our current implementation it is divided as two units. This suggests that we need to know the argument structure of the verbs as well. A straightforward solution is to enrich the tag set. However, as we have pointed out, complicating the tag set will put burden on human annotators and the error rate will increase as well. An even more straightforward solution that can be applied in SERUP is to make use of the UBG lexicon during parsing unit segmentation. When we see a verb, we checked that against the lexical entry to tell its argument structure. This also suggests that syntactic evidence alone may not be enough for such "early discourse management". We can try guessing the most feasible parsing units with the help of statistics, but further research is much needed.

8.2.5. The grammar

Evaluating the grammar at an early stage is not very meaningful because the coverage is deemed to be low but later on the coverage will be drastically increased after constant revision of the grammar. Unfortunately we are forced to evaluate our

grammar early because of the amount of time available. Moreover, cross evaluation is not possible because no previous work concretely describe its grammar. Even if cross evaluation is possible, it would be a linguistic one rather than a computational one. Anyway we provide the corresponding statistics here such that a numerical evaluation of SERUP is possible. The grammar size currently is about 200 rules, with most of them verb phrase rules and noun phrases rules. Table 8.6 shows the coverage of our grammar for the same five sets of articles as used in the tests above.

Article set	Grammar Coverage
Set 1	72.3%
Set 2	71.0%
Set 3	68.2%
Set 4	65.8%
Set 5	69.2%

Table 8.6. Grammar coverage.

Currently our grammar coverage is not broad enough. In the coming future we will continue to strive for better coverage and ultimate accuracy of our parser. We believe that the overall accuracy can reach as high as 98% for domain-specific texts, and 90% for general domain texts.

For evaluating the performance of the disambiguation method we mentioned in chapter 7, we prepared two sets of lexicons of the same words, one with semantic features based on noun categorization in <<同義詞詞林>> (Chinese Thesaurus, commonly known as "Cilin") [52], probably the most cited work in Chinese computational linguistics [53], one with R-seme obtained from <<商務新詞典>> (The Commercial Press Dictionary of Words) [54]. 300 predicate-object phrases were

parsed using different sets of seme. The number of valid predicate-object combination, the number of correctly accepted VP, and the number of wrongly accepted VP (i.e. over-fitting) are counted. Results are shown in table 8.7.

	Number of Valid Combination	Number of Correctly Accepted VPs	Number of Wrongly Accepted VPs	Percentage of Correct Identification	Percentage of Over-fitting
Set1 (seme)	215	198	102	92.09%	34%
Set2 (R-seme + seme)		215	0	100%	0%

Table 8.7. Comparison of different sets of seme.

It can be seen that our approach scored 100% accuracy, while the one based on compositional analysis only scored around 83%. Traditional semantic classes fail to handle cases like:

(197) *發生 浪潮 (a trend occurs)/ 發生 變故 (something wrong occurs)

(198) *浪費 工業 (to waste industry)/ 浪費 食物 (to waste food)

(199) *遭遇 命運 (to encounter destiny)/ 遭遇 變故 (to encounter bad changes)

In (197)-(199) the verb phrases before the strokes are wrong but the second verb phrases after the strokes are correct. Semantic classes fail in (197) because "浪潮" (trend) is usually classified as the same class as "變故" (bad changes). It is a common case of over-fitting. (198) fails as "工業" (industry) and "食物" (food) belong to different classes. (199) is more or less the same as (197). However, these cases can be resolved with using R-seme, as there is no need to classify abstract noun and there is

no limit of how many R-seme a verb can possess.

What is significant here is not the perfect result but the fact that perfect accuracy is possible with the new approach. On the other hand, solving selectional restriction with noun seme can never achieve 100% accuracy in theory. In real experiment, 100% accuracy using seme only is not unreachable, depending on the VP we choose. In other words, there is an urgent need to find another way to perform unbiased evaluation. However, this is not straightforward at all. Although R-seme is finite, the numbers of verb and noun are not. How many verbs and nouns should we include in the experiment? How can we assure that they can represent the infinite set of verbs and nouns[28]? These problems remain to be conquered.

The second set of data we need is the average number of semantic features/R-seme needed in each lexical entry for semantic checking of the 300 predicate-object phrases mentioned.

Semantic feature set	Average number of features in lexicon
R-seme only	8.6
R-seme + seme	4.3

Table 8.8. Average number of semantic features in lexicon.

Table 8.8 shows that using a combination of R-seme and seme can drastically reduce the number of semantic features needed in semantic checking. However, it should be noted that the number of features needed is proportional to the number of

[28] We have discussed in chapter 7 that in theory R-seme can represent all possible collocation relationships in a static point of language development. Can we go farther?

predicate-objects phrases encountered.

8.3. Overall performance of SERUP

To test the overall performance of SERUP we prepared another five articles randomly selected from our economic articles (mainly from newspaper) corpus that were not used in the previous testing. The test was conducted on a PentiumII 350MHz personal computer, with 128Mb Ram. Each article contains 34 comma-separated strings on average, and a number of new words as well. The accuracy is given by the total number of correct parsed output divided by the total number of parsing units input to the parser. Table 8.9 shows the performance of SERUP for each of the five articles and the corresponding average total processing time in second.

Article	Accuracy	Total Processing Time (average of 10 runs)
Article 1	61.4%	5.46s
Article 2	64.6%	4.25s
Article 3	58.1%	5.37s
Article 4	55.2%	6.53s
Article 5	63.8%	6.1s
Overall	60.62%	5.54s

Table 8.9. Overall system accuracy and total processing time.

The overall accuracy is not very reasonable, mostly because of the accumulated errors from every stage and the low coverage of the grammar. Grammar coverage will increase accordingly when we encode more and more linguistic analysis into the unification-based grammar. The first few stages can enjoy a higher accuracy after

careful refinement in statistical manipulation. Both issues are important to the accuracy of the system. The results obtained are also test set related. It is possible to choose a test set that score very high in the experiment. Therefore it is not fair to say SERUP has poor performance. In fact, SERUP is the first of such model, and an overall performance of 60% is already a good result, keeping in mind that our grammar coverage is relatively low due to lack of time for implementation. We foresee in the near future SERUP overall performance can reach 80% (estimation only), when we have improved our grammar extensively.

The total processing time is still a bit slow for practical application, particularly to web-based applications. In fact, the PERL programs and the PC-PATRII are the bottleneck. We do not mean that the programs are poor, but further improvement has to be done. For example a tailor-made unification parser must be written to obtain maximum speed for the UBG we proposed. In the long run we are planning to develop a computational-unification-based grammar model to replace the separate modules described in this paper. We strongly believe that such a model will allow easy multiple-sentences parsing and will be a breakthrough in NLP. However, this requires a lot of theoretical researches, and unless there is a more commonly accepted analysis of Chinese, it is hard to evaluate such a model.

9. Conclusion

9.1. Summary of this thesis

Unification-based grammar (UBG) processing and corpus-based processing are the two dominating methods in natural language processing. UBG is widely believed to be the most suitable linguistic theory for structural analysis, while corpus-based processing solves a number of linguistic tasks by finding linguistic generalizations in a corpus with the help of statistical models. Researches have noted that a NLP system without deep information processing cannot handle sophisticated task. However, UBG as a theory assumes integrity of input and so it fails to produce output when new words appear. In real practice we will definitely face many words that do not appear in the lexicon. Another issue is that UBG are originally developed for English and other Indo-European languages, but not for Chinese. For the model to work for Chinese, we have to deal with the characteristics of it, for example, segmentation problem and vague definition of sentence. Furthermore, unification-based grammar parsing tends to create a lot of useless feature structures when ambiguities arose. We would like to reduce the number of structural ambiguities as well.

We have in this thesis discussed about the difficulties in Chinese language processing, the methods in language processing and our proposal of a statistical enhanced robust unification based grammar parser – SERUP. The statistical processing modules include lexical segmentation, new word identification and parsing unit segmentation. Details of each phase of processing have been discussed in chapter 5. The basis and details of the core unification-based grammar has been well discussed in chapter 6.

Structural ambiguities are common in natural language parsing. In unification-based grammar (UBG) parsing structural ambiguities are hazard because they use up a lot of system resources and drastically lower system performance. An improved method based on traditional effort has been proposed and well-justified in chapter 7.

The performance of lexical segmentation is around 97% accuracy, that of new word identification is around 75%, and that of parsing unit segmentation averages around 90%. An overall parser accuracy of 60.64% has been recorded with an under-developing grammar. This seemingly low performance does not reflect actual scene, however. First, the accuracy will be improved drastically when the grammar becomes more and more sophisticated. Second, as we are the first ones who evaluate the parser on such large structure, cross evaluation is unavailable. We cannot draw conclusion that SERUP is performing poorly. Third, and the most important point is that overall accuracy is text-dependent. No fair measurement can be done anyway. Nevertheless, such results are enough to reflect that SERUP as a practical-theoretic model is a successful experimental research in Chinese natural language processing.

9.2. Contribution of this thesis

Several new ideas have been discussed thoroughly in this thesis. Our major contribution can be summarized as the following:

1) We have presented a practical model for parsing Chinese text with unification-based grammar. This thesis is the first account of how unification-based grammar can be employed in a practical Chinese natural language processing application.

2) We have improved parsing coverage by pre-parsing new word identification module. Our algorithm gives more correct result than previous work.
3) We are the first one to deal with the notion of Chinese sentence and parsing unit segmentation. Chinese sentence is a much neglected but crucial issue in Chinese natural language processing. We have proposed a method to segment Chinese text into reasonable parsing units.
4) We have further reduced structural ambiguity during parsing by improving disambiguation using semantic features. We saw the deficiency of the traditional method of disambiguation using semantic classes only. We have defined a new way to handle the problem.

9.3. Future work

SERUP is the first attempt to discuss how a unification-based grammar parser can be improved to allow practical use of it. Many problems still exist, however, as discussed in the previous chapter. In fact, much future work can be done to further improve the performance of SERUP.

First of all, we can pass multiple correct lexical segmentation results (due to ambiguity) to the later modules. If so, we can handle all segmentation results and let the parser to decide which output is the best. This however induce burden to system performance. A possible solution is to introduce parallel parsing to speed up computation.

Second, the new word identification algorithm has to be revised. We will investigate how semantic knowledge can be utilized with syntactic knowledge in determining the

probability of new words. Furthermore, identification of transliterated names is to be done.

Third, we will take up the challenge of discourse reconstruction. This is certainly the most challenging of all. It is because in Chinese, overt syntactic/discourse markers can be omitted rather freely. Discovering covert markers usually requires strong checking of semantic coherence among the subject/head verb/object of clauses. What is significant in SERUP is that we utilize punctuation as well in parsing and all these overt markers (including coordination markers, sentence final particles and aspect markers) are well encoded in the feature structure. Therefore after successful parsing we can utilize these pieces of information in determining the boundary of discourse sentence.

For example, (200) is a short paragraph containing 7 comma/full stop separated sentences/clauses. After parsing we get segments (200a)-(200g).

(200) 美國科技股經過四日下跌後上週五出現反彈，那斯達克指數累積下挫 500 點後回升 115 點，報 4,572 點。藍籌股則個別發展，首季最後一個交易日投資者入市興趣不大，杜瓊斯指數微跌 58 點，報收 10,921 點。

(200a) 美國科技股經過四日下跌後上週五出現反彈，

(200b) 那斯達克指數累積下挫 500 點後回升 115 點，

(200c) 報 4,572 點。

(200d) 藍籌股則個別發展，

(200e) 首季最後一個交易日投資者入市興趣不大，

(200f) 杜瓊斯指數微跌 58 點，

(200g) 報收 10,921 點。

Thereafter we can start restructuring the discourse. By syntactic clue and semantic agreement checking (200b-c) and (200f-g) form topic chains. The other comma-separated segments do not form any chain, so we can assign an event frame for every one of them. These event frames, together with the frame obtained from topic chains, will form the basis of further discourse inference. With the help of a domain-specific knowledge base, it is possible to deduce a correct discourse structure of the paragraph. Of course (200) is just a simple example. There are more non-trivial constructs such as "embedded topic chain", in which a topic chain exist in another topic chain. Further research towards this area is fruitful.

Finally we are planning to develop a computational-unification-based grammar model to replace the separate modules described in this thesis. We believe that such a model will allow easy multiple-sentences parsing and will be a major breakthrough in Chinese natural language processing research.

References

[1] Bussmann, H. 1996. *Routledge Dictionary of Language and Linguistics*. New York, NY: Routledge.

[2] Allen, J. 1995. *Natural Language Understanding*. Redwood City, CA: Benjamin/Cummings Publishing.

[3] Slocum, J. 1985. A Survey of Machine Translation. *Computational Linguistics* 11,

1, pages 1-17.

[4] Lufkin, J. M. 1989. Current Trends in Technical Translation. *IPCC*, pages 238-243.

[5] Chomsky, N. 1956. Three models for the description of language. *IRE Transactions PGIT*, 2, pages 113-124.

[6] Wahlster, W. (ed.). 2000. *Verbmobil : foundations of speech-to-speech translation.* Berlin: Springer.

[7] Woods, W.A. 1980. Cascaded ATN grammars. *AJCL* 6, 1, pages 1-12.

[8] Pereira, F. and S. Shieber. 1987. *Prolog and Natural Language Analysis*. Stanford, CA: Center for Study of Language and Information.

[9] Shieber, S. 1986. *An Introduction to Unification-Based Approaches to Grammar.* Stanford, CA: CSLI.

[10] Shieber, S. 1992. *Constraint-Based Grammar Formalisms*. Cambridge, MA: MIT Press.

[11] Covington, M. A. 1994 *Prolog for Natural Language Processing*. Englewood Cliffs, N.J.: Prentice Hall

[12] Gazdar, G., E. Klein, G. K. Pullum, and I. Sag. 1985. *Generalized Phrase*

Structure Grammar. Oxford: Basil Blackwell.

[13] Pollard, C. and I. Sag. 1987. *Information-Based Syntax and Semantics, Vol.1*. Stanford, CA: Center for the Study of Language and Information.

[14] Pollard, C., and I. A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago, IL: CSLI.

[15] Bennett, P. 1995. *A Course in Generalized Phrase Structure Grammar*. London: UCL Press

[16] Barton, G. 1985. On the Complexity of ID/LP Parsing. *Computational Lingustics* 11, pages 205-18.

[17] Earley, J. 1970. An Efficient Context-Free Parsing Algorithm. *ACM Communications* 13, 2, pages 94-102.

[18] Charniak, E. 1993. *Statistical Language Learning*. Cambridge, MA: MIT Press.

[19] Manning, D., H. Schutze. 1999. *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press.

[20] Yu, S. et al. 1998. *The Grammatical Knowledge Base of Contemporary Chinese-A Complete Specification*. Beijing, Tsinghua University Press.

[21] Hung, H. C. et al. 1997. *Problems in the Chinese Language: An Anthology*. Hong

Kong: The Commercial Press

[22] Yan, Hu. 1998. *Chinese-English Machine Translation*. M.Eng. Dissertation. Beijing, Tsinghua Unversity.

[23] Zhu, D. 1985. *Yufa Dawen*. Beijing, Commercial Press.

[24] Li, N. and S. Thompson. 1980. *Mandarin Chinese: A Functional Grammar*. Chicago: CUP.

[25] Chen, H.H. 1996. Logic-based Parsing of Chinese. In *Monograph of the Journal of Chinese Linguistics*, pages 47-76.

[26] Chomsky, N. 1981. *Lectures on Government and Binding*. Cinnaminson, NJ: Foris Publications.

[27] Chen, K. J. and C. R Huang. 1996. Information-based Case Grammar: A Unification-based Formalism for Parsing Chinese. In *Journal of Chinese Linguistics*, Monograph Series Number 9, pages 23-45.

[28] Fillmore, C. J. 1968. The case for case. In E. Bach and R. Harms (eds.), *Universals in Linguistic Theory*. New York: Holt, Rinehart, and Winston, pages 1-90.

[29] Lee, H. J., and P. R. Hsu. 1991. Parsing Chinese Sentences in a Unification-Based Grammar. *CPCOL* 5, 3&4:, pages 271-284.

[30] Lee, J., J. C. Dai, and Y. S. Chang. 1992. Parsing Chinese Nominalizations Based on HPSG. *CPCOL* 6, 2, pages 143-158.

[31] Dai, J. C., and H. J. Lee. 1994. A Generalized Unification-Based LR Parser for Chinese. *CPCOL* 8, 1, pages 1-18.

[32] Chiang, T.H. et al. 1996. Statistical Word Segmentation. In *Journal of Chinese Linguistics*, Monograph Series Number 9, pages 147-173.

[33] Sproat, R. et al. 1994. Stochastic Finite-State Word-Segmentation Algorithm for Chinese. *Computational Linguistics*, Summer.

[34] Yao, Y., and Lua, K. T. 1998. *A Probabilistic Context-Free Grammar Parser for Chinese. CPOL 11, .4*:393-405.

[35] Nagao, M. 1984. A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In A. Elithorn and R. Banerji (eds.), *Artificial and Human Intelligence: Edited Review Papers Presented at the International NATO Syposium on Artificial and Human Intelligence*. Amsterdam: North-Holland, pages 173-180.

[36] Liu, J. N. K., and L. Zhou. 1998. A Hybrid Model for Chinese-English Machine Translation. In *Proceedings of the IEEE International Conference of Systems, Man, and Cybernetics*, pages 1201-1206.

[37] Wu, D. 1995. Grammarless Extraction of Phrasal Translation Example from

Parallel Text. *TMI-95*, 354-372.

[38] Chu, C. 1998 *Discourse Grammar for Mandarin Chinese*. New York, NY: Lang.

[39] Wu, W.T. 1999. *Chinese Computational Semantics-analysis of relationships, semantic relationships and formality (漢語計算語義學-關係, 關係語義場和形式分析)*. Publishing House of Electronics Industry, Beijing, China.

[40] Lee, K. H. et al. 2000. Chinese Word Segmentation on Selected Character and Extraction of Cause-result Sentences. In *Proceeding of the International Conference of Computer Processing of Oriental Languages*, pages 97-100.

[41] Li, W. 1997. *CPSG: A Lexicalized Chinese Unification Grammar And Its Application*. Ph.D. thesis, Simon Fraser University, Linguistics Dept.

[42] Wu, A. and Jiang, Z. 2000. Statistically-Enhanced New Word Identification in a Rule-Based Chinese System. In *Proceedings of the Second Chinese Language Processing Workshop* in the *38th Annual Meeting of the Association for Computational Linguistics*.

[43] Packard, J. 2000. *The Morphology of Chinese: a linguistic and cognitive approach*. New York, NY: Oxford.

[44] Viterbi, A. J. 1967. Error bounds for convolution codes and an asymptotically optimal decoding algorithm. *IEEE Trans. on Information Theory* 13, pages 260-269.

[45] Klein, E. 2000. A Constraint-based Approach to English Prosodic Constituents. In *Proceeding of the 38th Annual Meeting of the Association for Computational Linguistics*.

[46] Zhu D. 1982. *Yufa Jianyi*. Beijing, Commercial Press.

[47] Lee, K.H., Roy Chan. 2000. Domain-Specific Chinese-English Machine Translation: Syntactic Analysis in GPSG. In *Proceeding of the International Conference on Chinese Language Computing*, pages 19-23.

[48] Mo, R. P. et al. 1996. Determinative-Measure Compounds in Mandarin Chinese Formation Rules and Parser Implementation. In *Journal of Chinese Linguistics*, Monograph Series Number 9, pages 123-145.

[49] Fu, A. 1999. *Lecture notes in machine translation*. Beijing, Chinese Academy of Social Science.

[50] Pun, K.H. and B. Lum. 1989. Resolving Ambiguities of Complex Noun Phrase in a Chinese Sentence by Case Grammar. In *Communication in Processing of Oriental Languages* Vol.4 Nos.2&3, pages 185-202.

[51] Mo, R. P. 1992. A Conceptual Structure for Chinese Analysis. *Chinese Knowledge Information Processing Group Technical Report CKIP-92-04*, Institute of Information Science Academia Sinica, Taiwan, Republic of China.

[52] Mei, J. et al. 1996. <<*同義詞詞林*>> *(Cilin)*. Shanghai, China: 上海辭書出版

社.

[53] Lua, K. T. 1993. A Study of Chinese Word Semantics. *CPCOL* .7,1, pages 37-60.

[54] Wong, K. S. (ed.). 1999. <<*商務新詞典*>> *(The Commercial Press New Dictionary of Words)*. Hong Kong, China: Commercial Press,.

[55] Chan, Roy and K. H. Lee. 2001. Towards a Robust Unification-based Parser for Chinese NLP. In *Proceeding of the International Conference of Computer Processing of Oriental Languages*, pages 126-131.

[56] McConnel, S. 2000. *"PC-PATR" – A Unification-Based Syntactic Parser, Version 1.2.2*, SIL International, TX.

Appendix I

The following is an example run of SERUP.

Phase One: Input text BEFORE lexical segmentation:

預期美國提早減息的美夢落空下， 美股前晚大跌拖累港股。
昨天全線急挫， 全日大跌 214 點， 收報 1130 點。 有消息透露，美資富達基金擔心美國經濟放緩會觸發大量基金投資者贖回基金單位，所以在本周二開始，匯控公布業績翌日，大手拋售亞太區股份，包括健力寶，匯豐控股等藍籌股套現以應付贖回的壓力。

Phase Two: Input text AFTER lexical segmentation:

預期,美國,提早,減息,的,美夢,落空,下,，,美股,前晚,大跌,拖累,港股,。,
昨,天,全,線,急挫,，,全日,大跌,214,點,，,收報,1130,點,。,有,消息,透露,，,美資,富達,基金,擔心,美國,經濟放緩,會,觸發,大,量,基金,投資,者,贖回,基金,單位,，所以,在,本,周二,開始,匯控,公布,業績,翌日,，,大手,拋售,亞太區,股份,，,包括,健,力,寶,，,匯豐控股,等,藍籌股,套現,以,應付,贖回,的,壓力,

Note that after lexical segmentation, the character sequences "昨,天" (yesterday), "全,線" (whole series), "健,力,寶" (Jianlibao, a company name) are left unidentified.

Phase Three: Input text after new word identification

預期 美國 提早 減息 的 美夢 落空 下 C 美股 前晚 大跌 拖累 港股 S
昨天 全線 急挫 C 全日 大跌 214 點 C 收報 1130 點 S 有 消息 透露 C 美資 富達 基金 擔心 美國 經濟放緩 會 觸發 大 量 基金 投資 者 贖回 基金 單位 C 所以 在 本 周二 開始 C 匯控 公布 業績 翌日 C 大手 拋售 亞太區 股份 C 包括 健力寶 C 匯豐 控股 等 藍籌股 套現 以 應付 贖回 的 壓力 S

After the completion of new word identification algorithm, the character sequences "昨天" (yesterday), "全線" (whole series), "健力寶" (Jianlibao, a company name) are identified finally. Moreover, all the punctuation marks are translated to the easier readable English acronym. For instance, C is comma, and S is full-stop.

Phase Four: POS-tagging

預期#v 美國#n 提早#d 減息#v 美夢#n 落空#v 下#f C#w
美股#n 前晚#t 大跌#v 拖累#v 港股#n S#w 昨天#t 全線#n 急挫#v C#w
全日#t 大跌#v 214#m 點#q C#w 收報#v 1130#m 點#q S#w
有#v 消息#n 透露#v C#w 美資#n 富達基金#n 擔心#v 美國#n 經濟#n 放緩#v
會#v 觸發#v 大#b 量#q 基金#n 投資者#n 贖回#v 基金#n 單位#n C#w
所以#c 在#p 本#r 周二#t 開始#v 匯控#n 公布#v 業績#n 翌日#t C#w
大手#d 拋售#v 亞太區#n 股份#n 包括#v 匯豐控股#n 等#u 藍籌股#n 套現#v

C#w 以#p 應付#v 贖回#v 的#u 壓力#n S#w

Phase Five: Clause segmentation

預期 美國 提早 減息 的 美夢 落空 下 C 美股 前晚 大跌 拖累 港股 S

昨天 全線 急挫 C

全日 大跌 214 點 C

收報 1130 點 S

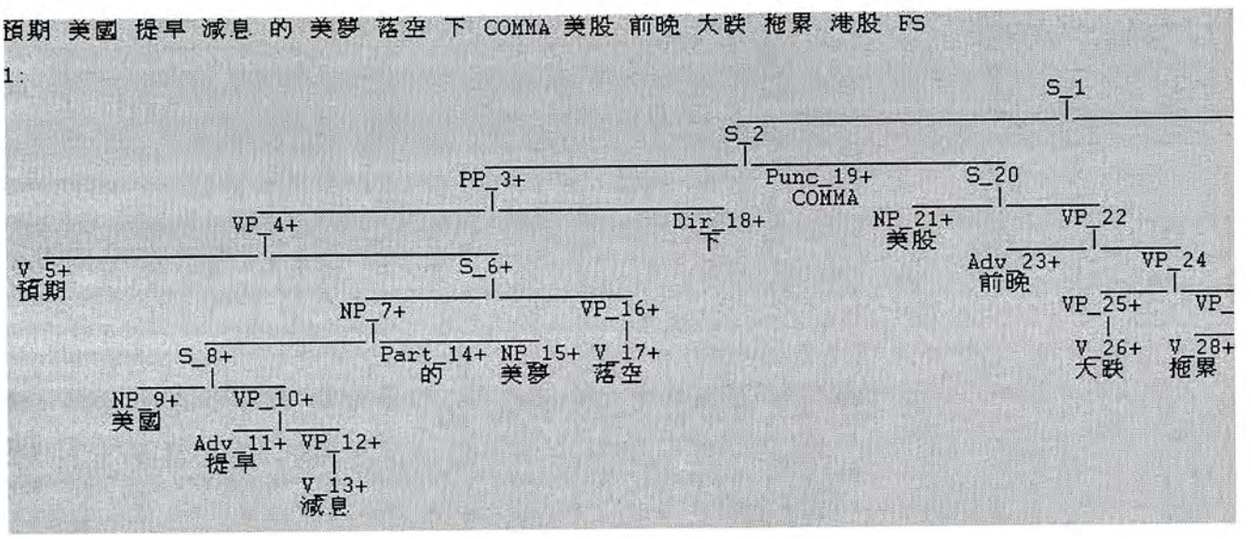
有 消息 透露 C 美資 富達 基金 擔心 美國 經濟放緩 會 觸發 大 量 基金 投資 者 贖回 基金 單位 C

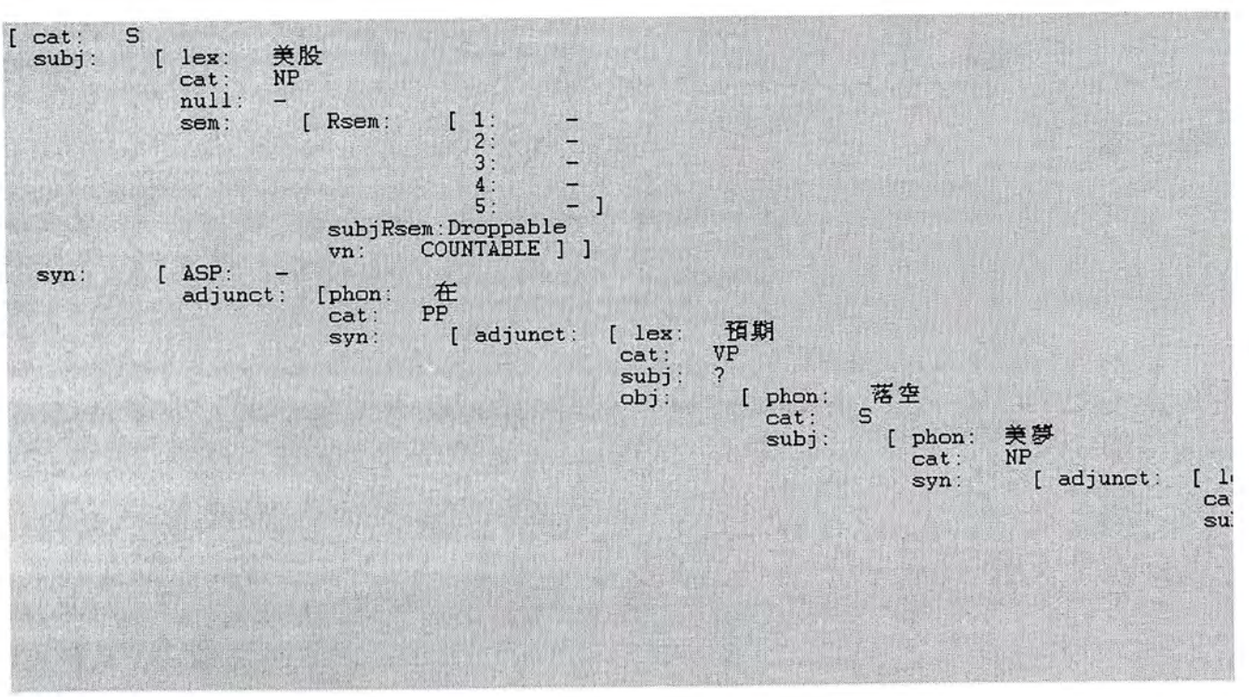
所以 在 本 周二 開始 C 匯控 公布 業績 翌日 C 大手 拋售 亞太區 股份 C 包括 健力寶 C 匯豐 控股 等 藍籌股 套現 以 應付 贖回 的 壓力 S

Phase Six: Syntactic parsing

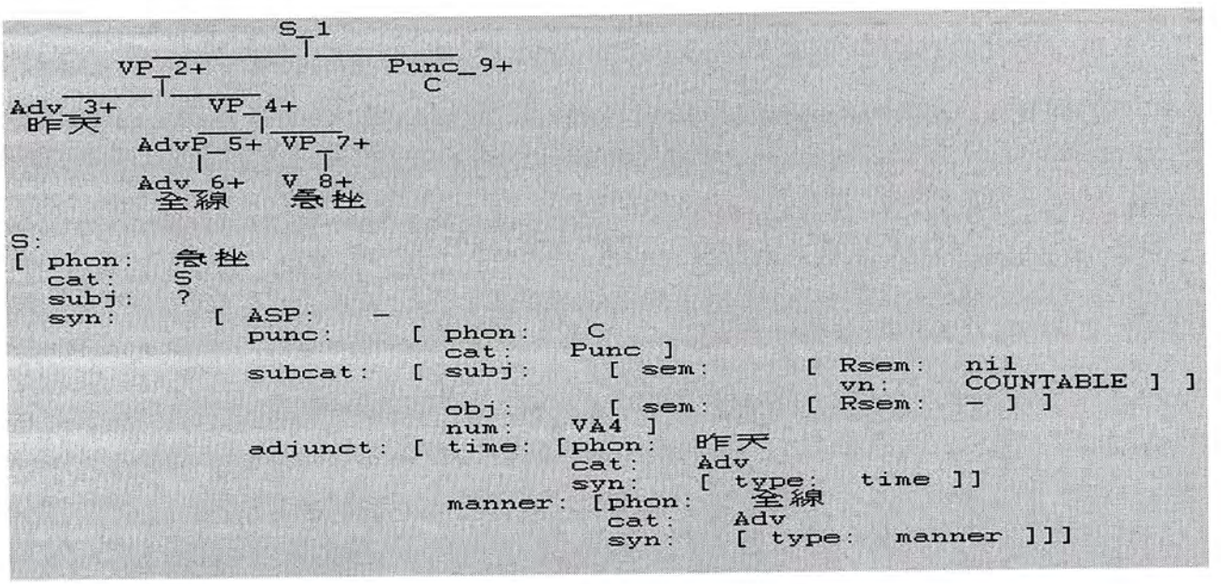
The full parsing result of this article is too gigantic that only the feature structures of the shorter sentences are shown here.

預期 美國 提早 減息 的 美夢 落空 下 C 美股 前晚 大跌 拖累 港股 S





昨天 全線 急挫 C


S_1
VP_2+
Punc_9+
C
Adv_3+
昨天
VP_4+
AdvP_5+
VP_7+
Adv_6+
全線
V_8+
急挫
S:
[phon: 急挫
cat: S
subj: ?
syn: [ASP: —
punc: [phon: C
cat: Punc]
subcat: [subj: [sem: [Rsem: nil
vn: COUNTABLE]]
obj: [sem: [Rsem: —]]
num: VA4]
adjunct: [time: [phon: 昨天
cat: Adv
syn: [type: time]]
manner: [phon: 全線
cat: Adv
syn: [type: manner]]]

Appendix II

The following are several sample PC-PATRII Chinese grammar rules in SERUP.

Sample Rule 1: Punctuation rule

```
Rule {S-> S Punc}
   S -> S_1 Punc
     <S slash> = <S_1 slash>
     <S phon> = <S_1 phon>
     <S syn> = <S_1 syn>
     <S subj> = <S_1 subj>
     <S obj> = <S_1 obj>
     <S_1 syn punc> = -
     <S syn punc> <= <Punc>
```

Sample Rule 2: Slash instantiation rule

```
Rule {S-> NP S/NP}
; as in 東西 我 吃了
      S -> NP S_1
     <S slash> = -
     <S_1 syn punc> = -
     <S_1 slash> = NP
     <S phon> = <S_1 phon>
     <S syn> = <S_1 syn>
     <NP null> = -
     <S obj> = <NP>
     <S subj> = <S_1 subj>
     {
     <S_1 syn subcat obj sem vn> = <NP sem vn>
          /
     <S_1 syn subcat obj sem Rsem> = <NP sem Rsem 1>
          /
     <S_1 syn subcat obj sem Rsem> = <NP sem Rsem 2>
          /
     <S_1 syn subcat obj sem Rsem> = <NP sem Rsem 3>
```

```
}
```

Sample Rule 3: "Ba2" construct in verb phrase with H[SUBCAT 1]

```
Rule {Verb Phrase Subcat1 - Ba2 construct}
  VP -> Prep NP V
      <V syn subcat> = 1
      <Prep> == [phon: 把]/[phon: 將]
      <VP phon> = <V phon>
      <VP syn> = <V syn>
      <V syn ASP> = 了
      <VP obj> = <NP>
      <VP slash> = -
      {
      <V syn subcat obj sem vn> = <NP sem vn>
            /
      <V syn subcat obj sem Rsem> = <NP sem Rsem 1>
            /
      <V syn subcat obj sem Rsem> = <NP sem Rsem 2>
            /
      <V syn subcat obj sem Rsem> = <NP sem Rsem 3>
      }
```

Appendix III

The following shows a portion of the lexicon in PC-PATRII format in SERUP.

```
\w 吃
\c V
\f <syn subcat> = 2
   <syn subcat obj sem vn> = FOOD
   <syn subcat obj sem Rsem> = Edible
   <syn subcat subj sem vn> = ANIMATE
   <syn subcat subj sem Rsem> = NIL

\w 東西
\c NP
\f <sem vn> = PHYSICALOBJ
   <sem subjRsem> = -
   <sem Rsem 1> = Buyable
   <sem Rsem 2> = Edible

\w 美國
\c NP
\f <sem vn> = ORGANIZATION
   <sem subjRsem> = -

\w 提早
\c Adv
\f <syn type> = time

\w 減息
\c V
\f <syn subcat> = 2
   <syn subcat subj sem vn> = ORGANIZATION
   <syn subcat subj sem Rsem> = NIL

\w 我
\c NP
```

```
\f <syn pro> = +
   <sem vn> = ANIMATE
  <sem subjRsem> = -

\w 大跌
\c V
\f <syn subcat> = 2
  <syn subcat subj sem vn> = COUNTABLE
  <syn subcat subj sem Rsem> = nil

\w 拖累
\c V
\f <syn subcat> = 2
  <syn subcat subj sem vn> = EVENT
  <syn subcat subj sem Rsem> = CanAffect
  <syn subcat obj sem vn> = COUNTABLE
  <syn subcat obj sem Rsem> = CanbeNegativelyAffected


\w 港股
\c NP
\f <sem vn> = COUNTABLE
   <sem subjRsem> = Droppable

\w 的
\c Part

\w 美夢
\c NP
\f <sem vn> = WISH
   <sem subjRsem> = Vanishable

\w 落空
\c V
\f <syn subcat> = 1
   <syn subcat subj sem vn> = PLAN
   <syn subcat subj sem Rsem> = Vanishable
```