DISCRIMINATION ORIENTED PROBABILISTIC TAGGING

Yi-Chung Lin, Tung-Hui Chiang, and Keh-Yih Su

Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.

Abstract

Conventional tagging models, with parameters estimated by the widely used maximum likelihood estimator, usually fail to achieve satisfactory performance in real applications. Since they achieve lexical disambiguation indirectly and implicitly via estimation, these models are usually unable to cover the statistical variation in the real text. In this paper, a discrimination oriented learning algorithm is proposed to directly pursue the goal of lexical disambiguation, so that the modeling error and the estimation error due to insufficient training data can be compensated. A 42% reduction in error rate, has been observed in the task of tagging Brown Corpus by using this proposed method.

1. Introduction

Tagging part of speech (or lexical disambiguation) in a sentence is an important problem in natural language processing. Traditionally, this task is achieved by ruling out the lexical ambiguities with a parser. However, as pointed out by Church[1], a parser is usually not capable to rule out all of those undesired ambiguities. Thus, passing all the combinations of different grammatical categories to the parser still let the problem to be unsolved. However, if there is a mechanism to select only a few combinations to the parser, with high possibility to be correct, it not only reduces the total processing cost, as parsing is a very expensive process, but also enhances the power of disambiguation, as fewer parse tree will be generated.

Several algorithms have been proposed in the literature to select the corrective category from all the possible tags for a given word. Greene and Rubin[2] developed TAGGIT with 3300 *context frame rules*. Each rule deletes one or more candidates from a list of possible tags for each word when its context is satisfied. TAGGIT achieves accuracy rate about 77% in the task of tagging Brown Corpus. Leech, Garside and Atwell[3] tag *LOB Corpus* with CLAWS[4], which is a bigram model with an IDIOMTAG procedure applied after initial tag assignment and before disambiguation, and 96.7% corrective tagging has been reported. Church[1] used a trigram model to tag Brown Corpus and achieved 95%-99% (depend on the definition of *correct*) accuracy. DeRose[5] developed VOLSUNG, which is similar to CLAWS, and reached accuracy rates of 96% without idiom tagging

and 99% with idiom tagging for the LOB Corpus. It also achieves 96% accuracy rate for tagging Brown Corpus. Recently, several probabilistic models based on trigram are investigated on different corpus[6][7], and have made some improvement.

All above models (except TAGGIT) use parameters estimated by maximum likelihood estimator. Correct disambiguation, however, depends only upon correct rank ordering of different category sequences. Therefore, maximizing likelihood does not imply minimizing the error rate of disambiguation[8][9]. Thus, a discriminative learning procedure is required to tune the model parameters to achieve high performance. Furthermore, due to insufficient amount of training data and incompleteness of model knowledge, the statistical variation between the testing set and the training set is usually not well characterized in those conventional approaches, therefore, minimizing the error rate in the training set does not necessarily imply maximizing the disambiguation accuracy in real applications. To achieve satisfactory result in real applications, this discriminative learning procedure must also be robust.

In this paper, a discrimination oriented learning procedure is proposed to fine tune the model parameters. Parameters are adjusted to shift the correct category sequence to the top rank among different combinations of categories during learning process. Great improvement, 42% reduction in error rate, have been observed in the task of tagging Brown Corpus.

2. Simulation Setup

2.1. Corpus Preparation

Brown Corpus is selected in this paper to compare different approaches, because it is the most well-known and widely-used corpus. Using *sentence closer* tag [10] as the delimiter between sentences, we extract 1,147,474 words (including sentence markers), of 54,597 sentences from Brown Corpus. No morphological analysis is done in preparing the training set. So words with different characters (such as *advantage* and *advantages*) are considered as different words. In the same way, the tags *PPS*, *MD* and *PPS+MD*, for the words *he*, *will* and *he'll*, are also treated as three different tags. Based on this, we construct a dictionary with 49,705 different words and a tag set with 187 different tags (not 87 tags stated in[10]).

Because it is the performance in the real applications (i.e., the testing set in our case) that we really care, the whole corpus are separated into two sets:

1. *Training set* — contains 919,247 words in 43,677 sentences, which is used to train the model parameters.
2. *Testing set* — contains 228,227 words in 10,920 sentences, which is used to estimate the accuracy rate of different tagging procedures.

2.2. Probabilistic Model

The purpose of lexical disambiguation is to find a correct part of speech sequence "$c_1, c_2, \cdots, c_N$" for a given sentence, "$w_1, w_2, \cdots, w_N$", where w_j is the j-th word of the given sentence and c_j is the part of speech assigned to the j-th word. This problem can be formulated as to find argmax$P(c_1^N|w_1^N)$, where c_1^N and w_1^N are the short-hand notations for "$c_1, c_2, \cdots, c_N$" and "$w_1, w_2, \cdots, w_N$" respectively. $P(c_1^N|w_1^N)$ can be further derived, using the multiplication rule in probability theory, as the following equation

$$P\left(c_1^N|w_1^N\right) = \prod_{j=1}^{N} P\left(c_j|c_1^{j-1}, w_1^N\right). \tag{1}$$

However, it is infeasible to directly estimate the parameter $P(\ c_j|c_1^{j-1}, w_1^N)$, for it demands a huge amount of data to train those a lot of parameters. To make it practical, assumptions must be made to simplify the evaluation process of $P(\ c_j|c_1^{j-1}, w_1^N)$. It is obvious that the correct category of a word in a sentence strongly depends on the word itself and the categories from the adjacent words. So, it is reasonable to make either of the following assumptions :

1. Assume $P(\ c_j|c_1^{j-1}, w_1^N) \approx P(c_j|w_j)P(c_j|c_{j-1})$. This is the bigram[1] model used in CLAWS[3].
2. Assume $P(\ c_j|c_1^{j-1}, w_1^N) \approx P(c_j|w_j)P(c_j|c_{j-2}, c_{j-1})$. This is the trigram model proposed by Church[1].

The probability $P(\ c_j|w_j)$ is called lexical probability, and $P(c_j|c_{j-1})$ or $P(c_j|c_{j-2}, c_{j-1})$ is called context (or transition) probability.

Using the above assumptions, the problem of lexical disambiguation can be formulated as to find argmax$(\prod_{j=1}^{N} P(c_j|w_j)P(c_j|c_{j-n}^{j-1}))$, where n=1 or 2. The *beginning of sentence* marker is assigned to c_0 and c_{-1} in the above formulation.

2.3. Baseline Performance

The context probabilities $P(c_j|c_{j-n}^{j-1})$ are first obtained from the training corpus by maximum likelihood estimator. For example, given a sentence "*I saw a beautiful girl*", one possible category sequence is "*pron v art adj n*", then the value of probability of *P(n|art,adj)* is estimated by *C(art adj n)/C(art adj)*, where *C(art adj n)* is the number of occurrences of the tri-POS[2] "*art adj n*" in the training corpus, and *C(art adj)*$= \Sigma_X$*C(art adj X)* where X is any possible tag. The lexical probability is estimated in a similar way.

Table 1 and 2 lists the performance of those bigram and trigram models. These results will be used as the baseline performance in the following tests. There are 1,147,474 words (including sentence markers) in the Brown Corpus, but only 40% of these words are *ambiguous* (i.e., words

[1] Based on the assumption that the next word which will be uttered depends only on the previous one or two words, bigram and trigram language models are widely used in speech recognition. Church[1] used the terms of bigram and trigram to indicate that the next category is strongly depends on the previous one or two categories, respectively. We will follow his notations in this paper.

[2] In this paper, a tri-POS is defined as a sequence three categories (i.e. sequence of "c_{j-2} c_{j-1} c_j". In the same way, bi-POS is defined as a sequence of two categories like "c_{j-1} c_j".

with two or more categories, and are called *ambiguous words*). Therefore, using word accuracy to measure performance is not a good way, because most words in the corpus can have only one category. In this paper, the word accuracy rate is reported on the *ambiguous word accuracy*, which is defined as N_A/N_W, where N_A is the number of ambiguous words which are correct tagged and N_W is the total number of ambiguous words in the corpus. The error rate of ambiguous words is defined as $1-N_A/N_W$. In the same way, the sentence accuracy rate is defined as N_C/N_S, where N_C is the number of sentences in which every word is correct tagged, and N_S is the number of sentences in corpus.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
bigram	55.65	91.66	8.34
trigram	64.96	93.95	6.05

Table 1 Baseline performance in the training set.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
bigram	53.34	91.04	8.96
trigram	55.34	91.44	8.56

Table 2 Baseline performance in the testing set.

Table 1 and 2 shows that the accuracy rate of trigram model in the training set is much better than that of bigram model, but, in the testing set, the performance of trigram model are just slightly better than that of bigram model. The high accuracy rate of trigram model in training set is due to the phenomena of *over-tuning[9]*. The large difference between the accuracy rate of the training set and that of the testing set for trigram model is mainly due to the insufficience of training data.

3. Discrimination Oriented Learning

In section 2.2, the disambiguation process is formulated as to find argmax$P(c_1^N|w_1^N)$, and the simplified form of $\prod_{j=1}^{N} P(c_j|w_j)P(c_j|c_{j-n}^{j-1})$ is used to calculate $P(c_1^N|w_1^N)$. For the convenience of real

applications, a score function is defined in here as

$$\begin{aligned}\text{Score} &= \log\left\{\prod_{j=1}^{N} P(c_j|w_j)\, P\left(c_j|c_{j-n}^{j-1}\right)\right\} \\ &= \sum_{j=1}^{N}\left\{\log\left(P(c_j|w_j)\right) + \log\left(P\left(c_j|c_{j-n}^{j-1}\right)\right)\right\} \\ &= \sum_{j=1}^{N}\left\{S(c_j|w_j) + S\left(c_j|c_{j-n}^{j-1}\right)\right\},\end{aligned} \tag{2}$$

where $S(c_j|w_j)$=log $(P(c_j|w_j))$, is called lexical score and $S(c_j|c_{j-n}^{j-1})$=log $\left(P\left(c_j|c_{j-n}^{j-1}\right)\right)$, is called context score. Then, the lexical disambiguation process is to calculate the score function for all the possible category sequences of a input sentence, and to choose the category sequence which has the highest score. In baseline models, the parameters used to calculate the score function are estimated without considering the competing category sequences. So, they can not minimize the error rate in the training corpus.

In order to minimize the error rate of the training corpus, a discrimination oriented learning procedure[11][9] is adopted to tune the parameters (i.e., the lexical and context scores) in this paper. Without loss of generality, we use the bigram model and a sentence with three different possible category sequences to show how to tune the parameters. Assume that the sentence "*Press the left button*" has only one ambiguous word "*left*" with possible tags *v*, *n* and *adj*. The correct category sequence should be "*v art adj n*" in this case. The disambiguation process, before learning, is listed in Table 3. As the candidate 1, a wrong category sequence, has the highest score, an error is made.

		Press	the	left	button	sub total	total
candidate 1	@	v	art	v	n		
lexical score		0	0	-0.3*	0	-0.3	-2.38
contex score		-0.7	-0.52	-0.7*	-0.16*	-2.08	
candidate 2	@	v	art	n	n		
lexical score		0	0	-0.7	0	-0.7	-2.92
contex score		-0.7	-0.52	-0.3	-0.7	-2.22	
candidate 3	@	v	art	adj	n		
lexical score		0	0	-0.52*	0	-0.52	-2.42
contex score		-0.7	-0.52	-0.52*	-0.16*	-1.90	

Table 3 Disambiguation process before learning. The symbol @ is *beginning of sentence* marker. The marker * denotes those parameters which will be adjusted.

Comparing candidate 1 and candidate 3 (the correct sequence), we find that the parameters *S*(*v*|*left*), *S*(*v*|*art*), *S*(*n*|*v*), *S*(*adj*|*left*), *S*(*adj*|*art*) and *S*(*n*|*adj*) are involved in the incorrect decision.

If we can increase the parameters *S*(*adj*|*left*), *S*(*adj*|*art*) and *S*(*n*|*adj*), and decrease the parameters *S*(*v*|*left*), *S*(*v*|*art*) and *S*(*n*|*v*), we can make the correct category sequence to have the highest score. To adjust these parameters, a score vector is first defined as

$$\begin{aligned}\vec{S} &= (s_1, s_2, s_3, s_4, s_5, s_6) \\ &= (S(adj|left), S(adj|art), S(n|adj), \\ &\quad S(v|left), S(v|art), S(n|v)),\end{aligned}$$

then the following equations are used to tune the score vector.

$$\vec{S}_{t+1} = \vec{S}_t + \triangle\vec{S}_t, \tag{4}$$

where

$$\begin{aligned}&\triangle\vec{S} = \varepsilon C H\left(\vec{X}, \vec{S}\right), \\ &H\left(\vec{X}, \vec{S}\right) = l'(d)\frac{1}{\left\|\vec{X}\right\|\sqrt{1-d^2}}T\left(\vec{S}\right)\vec{X}, \\ &d = \frac{\vec{S}^t\vec{X}}{\left\|\vec{S}\right\|\left\|\vec{X}\right\|}, \\ &l'(d) = \frac{d_0}{d_0^2 + d^2}, \\ &T\left(\vec{S}\right) = \frac{\left\|\vec{S}\right\|\vec{E} - \vec{S}\vec{S}^t}{\left\|\vec{S}\right\|^3}.\end{aligned} \tag{5}$$

In equation (5), ϵ is a small constant to control the convergence speed of learning process, C is positive-definite matrix and d_0 is the window size[11]. The vector $\vec{X}$ in this example is $(1,1,1,-1,-1,-1)$, such that $\vec{S}^t\vec{X}$ is the difference between the score of candidate 3 and that of candidate 1. As the details of the learning process have already been investigated in the literature[11], we will not give the detail derivations here. Using the above equations, the disambiguation process after learning is listed in Table 4.

		Press	the	left	button	sub total	total
candidate 1	@	v	art	v	n		
lexical score		0	0	-0.35*	0	-0.35	-2.51
contex score		-0.7	-0.52	-0.74*	-0.20*	-2.16	
candidate 2	@	v	art	n	n		
lexical score		0	0	-0.7	0	-0.7	-2.92
contex score		-0.7	-0.52	-0.3	-0.7	-2.22	
candidate 3	@	v	art	adj	n		
lexical score		0	0	-0.48*	0	-0.48	-2.29
contex score		-0.7	-0.52	-0.48*	-0.11*	-1.81	

Table 4 Disambiguation process after learning. The marker * denotes those parameters which should be adjusted during learning.

After discrimination oriented learning, the accuracy of lexical disambiguation is improved greatly. Comparing Table 1, 2 , 5 and 6, the error rate of ambiguous words of bigram model is reduced from 8.96% to 5.66% in the testing set, i.e., about 37% error rate reduction. For trigram model, the error rate of ambiguous words in testing set is reduced from 8.56% to 6.4%, about 25% error rate reduction. In the training set, the decrease of error rate of bigram and trigram models are 48% and 58% respectively, but these improvements are not important because the error rate of real applications is approximated by the performance in the testing set not the training set.

One phenomena should be noticed in Table 5 and 6. Although the accuracy rate of trigram is much better than that of bigram in the testing set, the accuracy rate of trigram is worse than that of bigram in the testing set. This problem is due to the limited size of training corpus and will be discussed in next section.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
bigram	73.94	95.66	4.34
trigram	83.89	97.44	2.56

Table 5 Performance in the training set after learning.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
bigram	65.91	94.32	5.68
trigram	63.65	93.60	6.40

Table 6 Performance in the testing set after learning.

4. Merging Unreliable Parameters

Due to the limited size of training corpus, trigram model suffers the problem of *over-tuning*, which usually occurs when the number of available training data is not large enough compared to the number of parameters. In this situation, the learning process will be lead to a pseudo optimal point in the training corpus, which sometimes even degrades the performance in the testing set. This phenomena is shown in Table 5 and 6 that the performance of trigram in testing set is poorer than that of bigram, although the performance of trigram is much better than that of bigram in the training corpus. One way to overcome this problem is to replace the unreliable parameters of trigram, i.e., whose number of occurrences in the training corpus are below a threshold, with the more reliable parameters of bigram. For example, if the tri-POS (*art v n*) and (*prep v n*) occurred less than R times in the training corpus, then the parameter $S(n|v)$, instead of $S(n|art,v)$ and $S(n|prep,v)$, will be used in the learning process.

The merging procedure described above is similar to the *backing-off* procedure[12][7]. However, the proposed approach differs from the backing-off approach in that the parameters corresponding to bi-POS will be adjusted during learning process, instead of using them directly as backing-off procedure does. The threshold R is found to be insensitive in a wide range from 1 to 50 and is set to 20 in our simulation. Figure 1 displays the behavior of learning process for the case of merging the unreliable parameters and Table 7 shows the final performance.

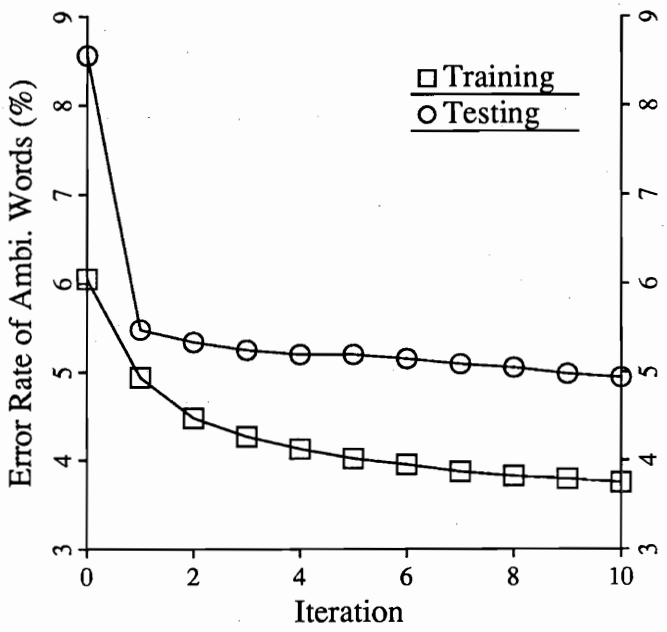

Figure 1 The error rates of merged trigram model during parameters-merged learning.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
training	76.51	96.23	3.77
testing	69.76	95.05	4.95

Table 7 Final performance of merged trigram model in both training set and testing set.

The reason for the improvement of performance is : although trigram carries more discriminative informations, they are poorly estimated (or trained) for not having enough data, and thus is quite unreliable to be used in the testing set. To replace those unreliable parameters with more reliable parameters from bigram, although they carry less discriminative informations, we sacrifice a small amount of modeling error for reducing a large amount of estimation error in the testing set, thus to improve the performance in the testing set. Figure 2 shows the improvements made by discrimination oriented learning and parameters-merged learning.

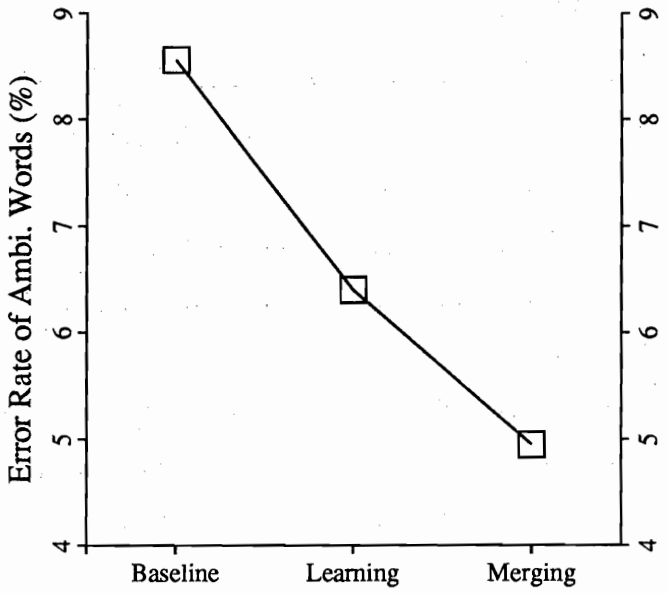

Figure 2 The performance improvement of trigram model. *Baseline* means parameters are estimated by maximum likelihood estimator. *Learning* means the parameters are tuned by discrimination oriented learning. *Merging* means the parameters are merged and then tuned.

5. Conclusion

Recently, probabilistic models are widely used for lexical disambiguation. In conventional probabilistic approaches, model parameters are estimated by maximum likehood estimator without considering the competing candidates, therefore, the can not minimize the error rate of lexical disambiguation. In this paper, a discrimination oriented learning method is proposed to tune the parameters. The method results in 37% and 25% error rate reductions of ambiguous words for bigram and trigram models in the testing set. To further improve the performance, a merging procedure is used to conquer the problem of over-tuning and make the model more robust. Using those merged parameters for learning, great improvement, 42% reduction in error rate, have been observed in the task of tagging Brown Corpus.

Reference

[1] K. W. Church, "A stochastic patrts program and noun phrase parser for unrestricted text," in *ACL Proceedings of the 2nd Conference on Applied Natural Language Processing*, Austin, TX, USA, pp. 299–307, Feb 9-12 1988.

[2] B. B. Greene and G. M. Rubin, *Automatical grammatical tagging of English.* Rhode Island: Department of Linguistics, Brown University, 1971.

[3] G. Leech, R. Garside, and E. Atwell, "The automatic grammatical tagging of the LOB corpus," *ICAME News 7*, pp. 13–33, 1983.

[4] B. M. Booth, "Revising CLAWS," *ICAME News 9*, pp. 29–35, 1985.

[5] S. J. DeRose, "Grammatical category disambiguation by statistical optimization," *Computational Linguistics*, vol. 14, pp. 31–39, Winter 1985.

[6] B. Merialdo, "Tagging text with a probabilistic model," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ontario, Canada, pp. 809–812, May 14-17 1991.

[7] B. Maltese and F. Mancini, "A technique to automatically assign parts-of-speech to words raking into account word-ending information through a probabilistic model," in *Proceedings of the 2nd European Conference on Speech Communication and Technology*, Genova, Italy, pp. 753–756, Sep 24-26 1991.

[8] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "A new algorithm for the estimation fo hidden markov model parameters," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, USA, pp. 493–496, Apr 1988.

[9] K.-Y. Su and C.-H. Lee, "Robustness and discrimination oriented speech recognition using weighted HMM and subspace projection approaches," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ontario, Canada, pp. 541–544, May 14-17 1991.

[10]W. N. Francis and H. Kucera, *Frequency analysis of English usage*. Houghton Mifflin Company, 1982.

[11]S. Amari, "A theory of adaptive pattern classfiers," *IEEE Trans. on Electronic Computers*, vol. EC-16, pp. 299–307, June 1967.

[12]S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 400–401, Mar. 1987.

Acquisition of Unbounded Dependency Using Explanation-Based Learning

Rey-Long Liu and Von-Wun Soo
Department of Computer Science
National Tsing-Hua University
HsinChu, Taiwan, R.O.C.

Abstract

A natural language acquisition model using Explanation-Based Learning (EBL) had been proposed to acquire parsing-related knowledge which includes Context-Free grammar rules and syntactic and thematic features of lexicons. The domain theory that is assumed to be innate to the model includes the theta-theory and the universal feature instantiation principles in Generalized Phrase Structure Grammar (GPSG). In this paper, we show in particular how unbounded dependency may be acquired in the natural language acquisition model. The acquisition problem of unbounded dependency may be further divided into two sub-problems: detecting whether there are moved constituents and finding the places to which the constituents are moved. For these problems, the universal innate domain theory facilitates and constrains the acquisition process which is otherwise intractable.

Keywords: Natural Language Acquisition, Explanation-Based Learning, Theta Theory, Universal Feature Instantiation Principles, Knowledge Assimilation.

1. Introduction

Parsing involves searching for a set of applicable knowledge pieces to transform a sentence into its corresponding syntactic and/or semantic structure (e.g. the parse tree). This problem solving process needs a knowledge base which is often enhanced, maintained, and tested periodically, especially when the system is applied to different domains. Since natural language is ever evolutionary in nature, extensibility of a natural language processing (NLP) system becomes one of the most critical concerns in real applications.

The Universal Grammar (UG, Chomsky[19]), which is claimed to be innate and universal among various natural languages, is believed to reflect children natural language acquisition phenomena. From this point of view, natural language acquisition may be approached by setting the parameters embedded in UG and learning the particular linguistic requirements (called Periphery Grammar) of the target language. Thus, the introduction of UG not only reduces the hypotheses space and hence makes learning more tractable, but also promotes the portability of of the system, since it not only facilitates adaptive acquisition in various application domains with the same target language (Lehman[10]), but also makes acquisition across different natural languages more possible.

Therefore, a natural language acquisition model (Liu[13]) had been proposed to automatically assimilate and maintain parsing-related knowledge, including Context-Free grammar rules and syntactic and thematic requirements of lexicons. In the model, the knowledge bases of the model consist of two parts: the static part and the dynamic part. The static part contains the universal linguistic principles, including the theta-theory and the universal feature instantiation principles in the Generalized Phrase Structure Grammar formalism (GPSG, Gazdar[4]). They are innate and invariant in learning. The dynamic part contains current parsing-related knowledge of the system (periphery grammar). Through learning, the periphery grammar in the dynamic part is enhanced by following the principles in the static part.

In this paper, we focus on the acquisition of unbounded dependency in the developed explanation-based natural acquisition model. Typically, an unbounded dependency occurs in a

construction in which there is an unexpected constituent outside a clause, while within that clause a constituent is correspondingly missing (Chomsky[19]). Wh-questions, relative clauses, and topicalizations, which all involve movement, are the representative examples of unbounded dependencies we consider in this paper.

In fact, the task of unbounded dependency acquisition involves two steps: detecting whether there are moved constituents, and then finding the place to which the constituents are moved. For example, in the sentence "The boy I see is a student", it is necessary for the learning system to determine whether the VP (Verb Phrase) "see" has a missing theme or not. If a theme is missing, the system learns that an NP (Noun Phrase) may be constructed by an NP followed by an S (Sentence) with a theme missing. On the other hand, if no themes are missing, the S cannot have a missing theme. For these problems, the universal innate linguistic principles facilitate and constrain the acquisition process which is otherwise intractable.

In the next section, we describe the framework of the explanation-based natural acquisition model. More detailed elaboration may be found in Liu[13]. In section 3, we show why and how the universal linguistic principles are employed to acquire unbounded dependency. In section 4, experimental results are shown to investigate the performance of the model. The model is also related to previous works and evaluated from various perspectives. In section 5, we conclude the article.

2. Explanation-based natural language acquisition

Explanation-Based Learning (EBL, Mitchell[17], Keller[8]) had been widely applied to learning domains in which intensive domain theory may be constructed before learning. Major components of EBL may include Goal Concept, Operationality, Training Example, Domain Theory, and Problem Solver. In learning, the problem solver uses the predefined domain theory to prove (or explain) the given positive training examples to be an instance of the goal concept. The sufficient conditions of the explanation are thus extracted and expressed in terms of the operationality criteria. In later problem solving, when the extracted conditions may be directly applied to the current problem, no further explanation processes are needed. Therefore, through

learning, the domain theory is "compiled" into a more efficient version.

A new explanation-based natural language acquisition model had been proposed to learn parsing-related knowledge for the parser (Liu[13]). The relationship between the traditional EBL and the language acquisition model can be summarized as follows:

- Goal concept: Grammatical sentence.
- Operationality: Recognizability of linguistic features of constituents.
- Training examples: Input sentences and their parse trees.
- Domain theory: Universal linguistic principles (static) + Current parsing knowledge (dynamic).
- Problem solver: The parser.
- Explanation tree: Parse tree annotated with sufficient constraints (features).

In the model, the problem solver is the parser which uses its parsing knowledge to parse an input sentence. If the highest level goal S-maj (a major sentence) can be achieved, the sentence is proven to be grammatical (the sentence can be successfully parsed). The condition parts of the rules in the knowledge base are expressed in terms of linguistic features such as VERB, NOUN, AGENT, OBJECT, PERSON, ... etc. These features are operational or "efficiently recognizable" (Keller[8]) in the system.

In real world problem domains (e.g. natural language processing), although a preliminary domain theory can be constructed (such as simple grammar rules), it is quite difficult to have a complete and correct domain theory (Hall[6]). The domain theory can be incomplete. It is separated into two major parts: a static part and a dynamic part. The static part includes universal linguistic principles which are invariant and innate to the system, while the dynamic part is augmented through learning.

When an input sentence cannot be proven to be grammatical (i.e. it cannot be successfully parsed), learning is triggered to enhance the dynamic part of the domain theory. The parsing knowledge in the dynamic part includes the argument structures of verbs (e.g. the verb "see" needs an EXPERIENCER argument and a THEME argument), thematic features of nouns (e.g. AGENT, OBJECT, ... etc.), general grammar rules (e.g. S --> NP VP), and some special phrase patterns (e.g. "Although S, S"). Initially, syntactic and thematic features of some verbs and nouns are provided to the dynamic part as the bootstrapping parsing knowledge.

2.1 The learning algorithm

As the dynamic part is inadequate to provide actions, learning is triggered. The system can first deduce a correct solution path from the given parse tree (Liu[11], Liu[13]). After executing each action in the solution path, an annotated parse tree can still be constructed as a sufficient condition to explain the input sentence as a grammatical sentence. The new parsing knowledge can be extracted from the annotated parse tree and then assimilated into the dynamic part of the domain theory. The algorithm of the learning module can be thus formalized as follows:

(1) Get the parse tree of the new sentence from the trainer;
(2) Iteratively invoke the parser to annotate all constituents in the parse tree (i.e. apply the current parsing knowledge and the universal linguistic principles to the parse tree);
(3) Extract new rules from the annotated parse tree.
(4) If the first subgoal of the extracted rule is a phrase, assimilate the new rule into the grammar rule base;
Else begin
(5) Try to generalize the rules in the lexicon entry (empirical generalization);
(6) Assimilate the rule into the lexicon entry;
end

In the following sections, we further elaborate the extra parse tree input (step 1), the use of universal linguistic principles (step 2), and the way of knowledge assimilation (step 4 and step 6). Finally, an example is shown to illustrate the learning algorithm.

2.2 The parse tree as external guidance

When there is missing knowledge in the domain theory, new knowledge might become too ambiguous to acquire, even though the learning system has exploited all its current knowledge to the largest extent. For example, consider the sentence "Taking exercises is good for your health". The target knowledge is the rule "NP[NUM=-plu,PER=3] --> VP[VF=prp]" which means that a singular (NUM=-plu) third-person (PER=3) Noun Phrase (NP) can be constructed by a Verb Phrase (VP) with present participle verb form (VF=prp). However, if no other information is provided, the learning module cannot segment the sentence into phrases. In that case, there are too many possible kinds of new knowledge. For example, the system can hypothesize

that "taking" can be an NP, an S-maj can be implemented by the pattern "taking NP VP", an S-maj can be expanded as "taking exercises VP", ... etc.

However, the given parse tree cannot be a correct "explanation tree" in which the system can find sufficient conditions for the sentence to be grammatical. For example, in a parse tree, the system can deduce a rule "S --> VP" (since S is the mother of VP in the parse tree) which is too general in the sense that the sentence "Eats the hotdog" will also be accepted. To find a sufficient condition, the parse tree should be annotated with critical features by the help of the static part of the domain theory.

2.3 The static part -- universal linguistic principles

In the model, the static (and predefined) part of the domain theory contains the "abstract" and universal linguistic principles which guide the acquisition of "operational" knowledge (parsing knowledge) in the dynamic part. It contains the minimal linguistic knowledge which is assumed to be innate to the system and is invariant during learning. It includes the theta-theory and the universal feature instantiation principles. These principles promote the portability of the system and make learning more tractable by reducing the hypothesis space in learning. The universal innate principles in the model are thus defined as follows:

- The theta-theory (Chomsky[19]) proposes a theta criterion which requires that, in the argument structure of a lexical head, each argument must bear one and only one theta-role. For example, in the sentence "John kissed Mary", the head "kissed" assigns the NP "John" the "AGENT" theta-role, and the NP "Mary" the "THEME" theta-role. No arguments can be assigned more than one theta-roles.

- The Head Feature Convention (HFC, Gazdar[4]) says that a mother's HEAD features should be identical to the HEAD features of its head daughter. For example, the verb "eating" is the HEAD of the verb phrase "eating the apple" (the verb phrase is the mother of the verb in a parse tree). Since verb form (VF) is a HEAD feature defined in GPSG, the verb phrase should share the feature "VF=prp" with the verb (via unification).

• The Foot Feature Principle (FFP, Gazdar[4]) allows FOOT features to propagate from any daughter to its mother in the parse tree. For example, the SLASH feature is a FOOT feature in GPSG. If a constituent has a SLASH feature with value NP, there is an NP missing in it. Consider the NP "the boy I like". There is an object NP missing in the verb phrase "like". By following FFP, this SLASH feature will be propagated to the clause "I like".

• The Control Agreement Principle (CAP, Gazdar[4]) says that controllees (such as VPs) agree with their controllers (such as NPs) by showing the features that are essentially properties of the controllers. The AGR feature in GPSG formalism needs to follow this principle. For example, for the verb "likes", an AGR feature with value "NP[NUM=-plu,PER=3]" (Subject-Verb agreement) is encoded. According to the feature, CAP will inform the parser to climb the parse tree upward to check whether there is a singular third-person NP. CAP can deal with semantic processing when the value of an AGR feature includes thematic properties AGENT, THEME, EXPERIENCER, ... etc.) of controllers.

For more detailed description, the reader should refer to Chomsky[19] and Gazdar[4]. The critical roles of these principles on the acquisition of parsing knowledge can be further illustrated by the following examples:

• Suppose the system attempts to learn from an English command sentence "Eat the hotdog", and it has the rules for parsing the NP "the hotdog" and the subcategorization information of "Eat" (e.g. "Eat" needs an NP as object) as the currently available parsing knowledge. If HFC is not employed, even though a parse tree is given, the system might induce the rule "S-maj --> VP" (it comes from the parse tree). The rule is too general in the sense that the sentence "Eats the hotdog" will also be accepted. On the other hand, by following HFC, the VP can be appropriately annotated by critical features which are the basis of the generality of the new rule. In this case, a better rule "S-maj --> VP[VF=bse]" (a VP with base verb form can be a major sentence) can be constructed to enrich the current parsing knowledge bases (see Fig. 1).

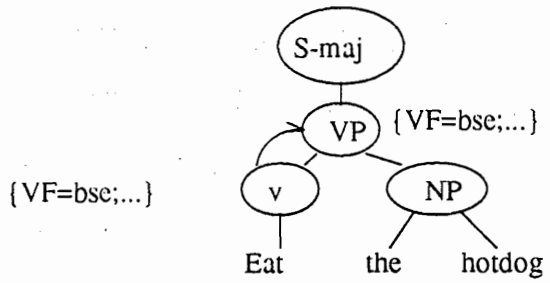

Fig. 1. Head Feature Convention

• Consider the sentence "Taking exercises is good for your health". Suppose the parsing module does not have a rule to construct an NP from a VP with present participle verb form. From the given parse tree and HFC, a VP[VF=prp] can be constructed by the parser. Therefore, the rule "NP --> VP[VF=prp]" can be induced. However, this rule is too general in the sense that the sentence "Taking exercises are good for your health" will also be accepted. On the other hand, if the VP "is good for your health" is parsed, by following CAP, it will restrict the number and person features of the NP to be singular and third-person. Therefore, the target rule "NP[NUM=-plu,PER=3] --> VP[VF=prp]" can be acquired (see Fig. 2).

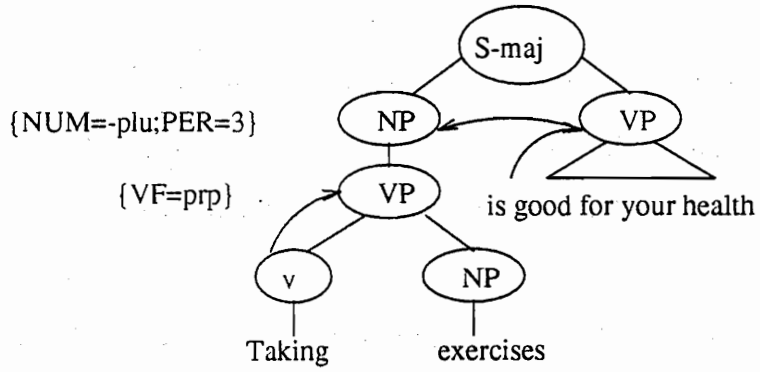

Fig. 2. Head Feature Convention & Foot Feature Principle.

2.4 Blame assignment and knowledge assimilation

In this paper, we focus on the problem of incomplete domain theory. Enhancing the

dynamic domain theory is simply adding and then properly generalizing new knowledge pieces. From this point of view, the problem of blame assignment is reduced to the problem of finding which knowledge pieces are the missing knowledge. When invoking the parser to parse the sentence (step 2 in the learning algorithm), the learning system keeps track of the activation of rules. When no rules can issue the current action in the solution path, there is a missing rule at this point. After the whole parse tree is annotated, the missing rule may be extracted and assimilated into the dynamic domain theory.

The acquired rules may be assimilated into either the lexicon entries (step 6 in the learning algorithm) or the general grammar rule base (step 4 in the learning algorithm). The way of assimilating knowledge is closely related to the way of retrieving knowledge to use. In the model, indexing is employed for fast assimilation and utilization of knowledge. If the first subgoal of the acquired rule is a phrase, the rule is placed into the grammar rule base. If the first subgoal is a word, the rule is assimilated into the lexicon entry of the word in the dictionary.

2.5 An example

When the parser fails to parse the input sentence, learning is triggered, and the user is asked to input a parse tree (Step 1). For example, for the above sentence "Taking exercises is good for your health", the parse tree might be:

```
(S (NP (VP (v taking)
           (NP (n exercises))))
   (VP (v is)
       (adj good)
       (PP (prep for)
                 (NP (pos your)
                     (n health))))).
```

As described in section 2.2, the system should have the ability to derive the critical features of constituents rather than directly extracts the rules from the parse tree. Therefore, after given a parse tree, the parser is invoked to separately parse the constituents in the sentence (Step 2). After that, the critical features (including syntactic and thematic features) of each parsed constituent are derived. In this case, we assume the subcat pattern "take NP[THM=OBJECT]" has

already been in the lexicon entry of "take". Therefore, the first VP "taking exercises" can be successfully parsed. Its feature "VF=prp" is also derived (since "VF" is a HEAD feature). At this time, the parsing module finds that it has a missing rule which allows it to construct an NP from the VP[VF=prp]. Therefore, now the possible new rule is "NP --> VP[VF=prp]".

After parsing the main VP "is good for your health", its feature "AGR=NP[NUM=-plu,PER=3]" is computed, where AGR is also a HEAD feature whose propagation in the parse tree must obey the Head Feature Convention. By following the Control Agreement Principle, the VP needs an NP which must be singular and the third person. This feature specification indicates that the NP constructed from the VP[VF=prp] should have the features "NUM=-plu" and "PER=3". Therefore, the final rule "NP[NUM=-plu,PER=3] --> VP[VF=prp]" can be successfully extracted from the annotated parse tree (Step 3). Since the first subgoal of the rule is a phrase, this rule is considered to be a general phrase structure rule which should be assimilated into the grammar rule base (Step 4).

It should be noted that, this way of computing critical features of constituents is a conservative way of acquiring new knowledge. That is, the computed features might be too specific. For example, consider the sentence "We live in an abundant life". Since the NP "We" has the thematic feature "THM=PERSON", after the sentence is processed, the system will restrict the AGENT of "live" to be an NP with the feature "THM=PERSON". Thus, when other input sentences involving "live" are entered (e.g. the sentence "The dog lives with us"), the learning module will try to generalize the rules which had already been acquired and stored in the lexicon entry of "live" (Step 5). The generalized rule is then assimilated into the lexicon entry of "live" (Step 6). Also note that, in some cases the system needs to generalize knowledge pieces among different lexicon entries.

3 Acquisition of unbounded dependency

Since constructions of unbounded dependency frequently occur in natural languages, its acquisition becomes an important task for natural language acquisition. As described above, portability is one of the major concerns of the learning model. Therefore, we need to introduce a

minimal and universal innate domain theory to constrain the hypothesis space, and simultaneously, maintain the portability of the system in the sense that it may be applied to various learning situations (e.g. Chinese).

Typically, unbounded dependency occurs in a construction in which there is an unexpected constituent outside a clause, while within that clause its corresponding constituent is missing. We consider in this paper such typical unbounded dependencies as in relative clauses, wh-movements, and topicalizations. To acquire them, the learning system needs to determine whether there are missing constituents and, if so, to which places the missing constituents are moved.

Berwick[2] employs the Subjacency Principle to locate the moved constituents in the sentence. The location process is simply triggered when the syntactic requirements (e.g. the subcategorization frames of verbs) are not satisfied (e.g. an NP is expected but does not appear at its corresponding place). However, when the syntactic requirements have not been completely acquired, the location process might be miss-triggered. For example, many verbs may be both transitive and intransitive. As a verb's transitive subcategorization frame has already been acquired, but its intransitive version has not, a new sentence with no NPs occurring at the object position of the verb causes two possibilities: either the verb may have a intransitive version or there is a missing NP that can be found in other places in the sentence (unbounded dependency). If the ambiguity cannot be resolved, erroneous knowledge, which is not only useless but also harmful to the learning system, may be acquired.

In this paper, FFP and the theta-theory work together to facilitate the acquisition of unbounded dependency. They are consulted as the learning system attempts to acquire the unbounded dependency.

3.1 Acquisition of unbounded dependencies in relative clauses

Movement in relative clauses may be characterized as A-Bar-movement in GB theory (Chomsky[19]). A constituent is moved from a position that is assigned both a theta-role and

Case to an A-Bar position. Consider the sentence "The boy I see is a student". The parsing module needs to acquire the target rule "NP --> SN S[/=NP]", where "SN" is a nonterminal for simple NPs (without embedding clauses), and "/" is the SLASH feature in GPSG terminology ("/=NP" means "missing an NP"). Similarly, without using FFP to propagate the SLASH feature of the VP "see" to the S "I see", the rule "NP --> SN S", which is too general, will be acquired. However, if the verb "see" is intransitive, the slash feature may not exist. How can the system determine whether the VP "see" has the "/=NP" feature? By following the theta-theory, the SN "the boy" must bear a theta-role. In the sentence, only the verb "see" may assign the "THEME" theta-role to it (the verb "is" only assigns a theta-role to the whole NP "The boy I see"). Therefore, the VP "see" is missing an NP.

The second step in the acquisition of unbounded dependency involves the locating of the moved constituent. In GPSG formalism, FFP propagates the slash feature upward *until* there is a rule which admits the subtree and mentions the corresponding slash feature in its LHS. However, this rule is just the target rule the system needs to acquire (e.g. "NP --> SN S[/=NP]"). In the model, the theta-theory and FFP need to work together to locate the moved constituents. The locating process propagates the slash feature upward, and as the *first* constituent with no theta-roles assigned is encountered in a subtree, the constituent is treated as the moved constituent, and the locating process then terminates.

Similarly, in the case of reduced relative clauses, such as "The boy running in the park" and "The boy seen in the room", FFP and the theta-theory may facilitate the acquisition of unbounded dependencies. In the former example, "running" is allowed to assign an "AGENT" theta-role to "the boy". Therefore, there are no missing NPs in the VP "running". In the latter sentence, since the verb "seen" with the passive participle form cannot assign theta-role to "the boy", an NP must be missing in the VP "seen".

3.2 Acquisition of unbounded dependencies in topicalizations

The way of acquiring topicalization constructions is quite similar to the way of acquiring relative clauses. Since there is an "extra" constituent (e.g. NP, AP, or PP) not been assigned any

theta roles, there must be some corresponding constituent missing in the structures after the extra constituent. Therefore, the acquisition of unbounded dependency may always be triggered, and locating process may also be succeeded in finding the extra constituent.

However, there might be multiple places from which the extra constituent is moved (Gazdar[4]). For example, consider the sentence "Sandy we want to succeed". It may be interpreted as "We want Sandy to succeed" or "We want to succeed Sandy". In the model, the acquisition of unbounded dependency is triggered whenever necessary. Therefore, the learning system will adopt the first interpretation. Fortunately, no matter which interpretation is adopted, from the acquisition point of view, the target rule "S --> NP S[/=NP]" may be learned.

3.3 Acquisition of unbounded dependencies in wh-movement

Wh-movement is also characterized as A-Bar-movement. Therefore, unbounded dependency in wh-movement may be learned in a similar way to acquiring relative clauses. The major difference is that, additional transformation is needed (e.g. in English, the auxiliary-verb inversion). Auxiliary-verb inversion can be treated as special phrase patterns which may be learned in the way discussed in section 2.3. For example, consider the wh-questions "What do you want?". The target rule is "S-maj --> what do S[/=NP]". It requires that, after matching "what" and "do", an S with a missing NP is expected for constructing an S-maj.

Generality of the acquired rules deserves further elaboration here. The reader may question why a better rule "S-maj --> wh aux S[/=NP]", where "wh" denotes a category covering wh-words and "aux" denotes a category covering auxiliary verbs, is not acquired. Unfortunately, universal linguistic principles give no help in this case, since it is the "Peripheral Grammar" that needs to take the responsibility of this kind of generalization. However, peripheral grammar is what the system tries to learn. Without any prior knowledge about the peripheral grammar of a particular natural language, over-generalization might be committed due to either the categories that are not well pre-classified or some special phrase patterns (e.g. not only S but also S) that cannot be generalized in this way. Therefore, the more specific version is preferred by the model. The specific rule "S-maj --> what do S[/=NP]" may be further generalized as more

empirical evidences are available (step 5 in the learning algorithm).

4. Experiment and evaluation

For efficiency, the system is implemented in C language on PC-386 computers. There are about five thousand lines of code in the program. The system can acquire thematic features of unknown nouns, argument structures of verbs, general phrase structure rules, and special patterns (such as "Not only S, but also S") which are all essential for a practical parser. About thirty general grammar rules and thousands of lexicon entries are currently in the dynamic part of the system. They are either initially given (for bootstrapping) or acquired by the system. The initially given knowledge includes the syntactic and thematic features of some nouns and verbs and a general set of phrase structure rules such as "S --> NP VP" that can be easily constructed (recall that the agreement in number between the NP and the VP is licensed by the Control Agreement Principle). The features of the words in sentences that trigger the acquisition of new rules should be available. Otherwise, no features can be propagated by the direction of the universal linguistic principles, and in turn, the acquired rules will be erroneous (recall sec. 2.2). On the other hand, when the system tries to acquire features of unknown words, all rules (e.g. argument structures of verbs) for parsing the sentence should be available. Rule acquisition and lexicon feature acquisition depend on each other in learning.

4.1 Efficiency of parsing

To show the parsing efficiency after learning, we show some interesting data concerning the effects of the introduced problem solving strategies. We had employed the strategies of common work sharing, dynamic conflict resolution, and knowledge indexing in the parsing module (Liu[13]). Common work sharing keeps a record of both succeeded and failed goals to eliminate redundant exploration. Dynamic conflict resolution resolves ambiguities in parsing by dynamically scanning the history of parsing and current input. Therefore, a set of parsing (diagnostic) rules in traditional Marcus' parsing (Liu[12]), which is quite difficult to maintain and acquire, may be avoided. Indexing adopts the concept of lexicon-driven NLP to assimilate and

retrieve relevant knowledge pieces.

In the experiment, we use 77 sentences to test the performance of the problem solver (parser) after learning. Most of the sentences come from a testing corpus originally collected from Chinese students' articles for grammar and style checking. The result is shown in Table 1. Since knowledge indexing maintains knowledge retrieval efficiency after learning, we focus on, under indexing, the effects of common work sharing and dynamic conflict resolution. As the result shows, common work sharing has significant contribution to efficiency. When it is incorporated, dynamic conflict resolution further improves the efficiency. Otherwise, the performance cannot be acceptable. It is interesting to note that, when common works are not shared among alternatives, the overhead caused by redundant invocation of conflict resolution even slows down the global efficiency.

Table 1. Accumulated run time (in second).

Strategies	Run Time
Indexing+Sharing+Resolution	62.44
Indexing+Sharing+Non-resolution	97.19
Indexing+Non-sharing+Resolution	3659.76
Indexing+Non-sharing+Non-resolution	3006.39

The result also shows that, if different learning approaches (Holder[7]), operationality criteria (Keller[8]), or intelligent knowledge selection methods (Minton[16]) are introduced without improving problem solving strategies, many "useful" or "good" knowledge pieces will be discarded because of the poor problem solving performance. As a result, the effective power of EBL may be limited, and even worse, the incomplete domain theory cannot be enhanced.

4.2 Minimal domain theory

As described above, the static part consists of the universal linguistic principles which are assumed to be invariant and innate to the system. To design an effective explanation-based natural language acquisition model, the application of universal linguistic knowledge is valuable. As the model is applied to other languages (e.g. Chinese), whether the static part is adequate or not becomes an interesting problem (Huang[30]). We believe that a more concrete and "univer-

sal" model can be expected only after analyzing various learning and processing requirements of different languages. This analysis can help us to define the minimal static domain knowledge which is the core of EBL.

In fact, more predefined domain knowledge also introduces more domain constraints which might turn to be obstacles in different learning situations (e.g. different target languages). According to the GPSG formalism, there are still five components that are responsible for licensing natural language sentences but not included as innate domain theory in our model. They are Feature Co-occurrence Restriction (FCR), Feature Specification Default (FSD), Lexical Immediate Dominance Rules (LIDs), Non-Lexical Immediate Dominance Rules (NLIDs), Metarules, and Linear Precedence Statements (LPS). These principles are either the target knowledge to be acquired (e.g. LIDs, NLIDs, LPS) or the principles that need fine-tuning (e.g. FCR, FSD, Metarules) among different natural languages. Although the introduction of FCR, FSD and Metarules makes knowledge representation more compact by reducing redundancies in knowledge bases, to acquire them needs a huge amount of empirical generalization which may be intractable, especially when empirical generalization is expensive in learning. Fortunately, they have no effects on the learnability of various parsing knowledge. In fact, by fast knowledge indexing, enumerating knowledge pieces (possibly redundant from the point of view of FCR, FSD and Metarules) in the general grammar rule base and the lexicon does not deteriorate parsing efficiency.

4.3 The validity and availability of the given parse trees

The kinds of input given to a learning system is essential and can vary from different learning methodologies and systems. The learning system utilizes the input to derive (or infer) new knowledge (such as a consistently generalized version of knowledge). In natural language acquisition, additional input is indispensable (the semantic bootstrapping hypothesis, Pinker[20]). In practice, the form and availability of the extra input have a strong effect on the plausibility (including portability and convergence quality) of the model.

In our model, giving a parse tree of an unrecognized sentence to the system seems to be a

strong assumption. From the parse tree, we can have not only categories of words but also phrase structures of the input sentence. However, there are still many things remaining to be learned. No parsers can completely parse sentences using general phrase structure rules only. The information in the parse tree is properly generalized according to the linguistic principles and current parsing knowledge. The system can thus derive practically essential knowledge (syntactic and thematic knowledge) based on the informative initial input knowledge.

In fact, the extra input can range from syntactic association to semantic association (or both) to the current sentence. The critical point is what kind of information the input provides. Giving syntactic information (Zernik[28], Lytinen[15], Liu[11], Liu[13]) to the system allows the acquisition of more syntactic (and perhaps semantic) information, while entering semantic information (Berwick[2], Siskind[24], Pinker[20], Zernik[27]) facilitates the acquisition of more semantic information.

Another aspect of providing extra input is the availability of the input. In practice, providing complicated semantic association is a very heavy burden for a naive user. In language acquisition, we can also rely on a large "pre-processed" corpus. However, to what extent the corpus should be pre-processed? As pointed out in section 2.2 (and in Zernik[28] also), a minimally pre-processed corpus allowing only co-occurrence acquisition contributes little in phrase structure and lexicon acquisition. Two constituents that are conceptually related (e.g. a verb and its argument) may not be co-located because they are distant from each other, while two constituents that are conceptually unrelated may still be co-located due to inadequate information in the minimally pre-processed corpus (Basili[1], Smadja[25]). Furthermore, co-location acquisition needs a large corpus and a large memory. To reduce these difficulties, a partial parser (Basili[1], Sekine[22], Smadja[25]), a tagger (Zernik[29]), and/or a set of predefined syntactic and semantic categories (Basili[1], Smadja[25]) need to be constructed before learning. However, the limitations (e.g. the incorrect analysis on the text and incomplete set of categories) coming from these preprocessing may also be introduced.

Machine-readable dictionaries were also the available sources of the training input in recent

years (Sanfilippo[21]). To acquire knowledge from them, a pre-processor (e.g. a parser) is needed for processing the description text part and the example part in lexical entries. When the system tries to learn from multiple dictionaries or multiple lexical entries, filtering and combining information from different sources are needed. These processing modules are the basic requirements, and hence the limitations, of the learning model.

Interactive acquisition (Lang[9], Liu[11], Lu[14], Simmons[23], Velard[26]) shows another alternative for giving additional information to the system. The confirmation information is available only if there is a well-trained trainer monitoring the learning behavior of the system. In addition, the number of questions needed for justifying the generated hypotheses may become a critical bottleneck (Liu[11]).

The parse trees assumed in the model can come from the trainer, the existing incomplete parsers, and the parse tree bank constructed for research evaluation (Grishman[5]). Currently, we are trying to transform an on-line parse tree corpus (PENN tree bank in the CD-ROM from Association of Computational Linguistics Data Collection Initiative) into the form suitable in the model. By exploiting the large available parse tree bank, the system can converge to a more complete parser without relying on the parse trees given by users.

4.4 Future work in the acquisition of unbounded dependency

The acquisition of unbounded dependency in "missing-object" constructions has not yet been well-developed in the model. For example, in the sentence "Kim is easy to please", there is an NP missing in the VP "please". However, for the sentence "Kim is eager to please", the VP "please" does not have any NP missing (Gazdar[4]). GPSG deals with the problem by using lexical immediate dominance (lexical ID) rules of "easy" and "eager". However, from the acquisition point of view, the incorporated universal linguistic principles have no help to the discrimination of the two sentence structures.

5. Conclusion

In this paper, we consider the effects of incorporating universal linguistic principles from

the viewpoint of computational natural language acquisition. Portability and learnability are the major concerns of the explanation-based natural language acquisition model. Currently, we find the theta-theory and the universal feature instantiation principles may play the critical role as the domain theory in EBL. The acquired knowledge can be properly generalized (without causing over-generalization) by following the guidance of these principles. In the acquisition of unbounded dependency, these principles facilitate not only the triggering of the chaining process, but also the locating of the moved constituents. The acquired operational knowledge, including Context-Free grammar rules and syntactic and thematic requirements of lexicons, becomes new domain theory for later parsing and learning.

Acknowledgement

This research is supported in part by NSC under the grant NSC81-0408-E-007-02.

References

[1] Basili R., Pazienza M. T., and Velardi P., *Computational Lexicons: the Neat Examples and the Odd Exemplars,* Proc. of the 3rd Conference on Applied NLP, pp. 96-103, 1992.

[2] Berwick R. C., *The Acquisition of Syntactic Knowledge,* The MIT Press, Cambridge, Massachusetts, London, England, 1985.

[3] Flann N. S. and Dietterich T. G., *A Study of Explanation-Based Methods for Inductive Learning,* Machine Learning, 4, 187-226, 1989.

[4] Gazdar G., Klein E., Pullum G. K., and Sag I. A., *Generalized Phrase Structure Grammar,* Harvard University Press, Cambridge, Massachusetts, 1985.

[5] Grishman R., Macleod C., and Sterling J., *Evaluating Parsing Strategies Using Standardized Parse Files,* Proc. of the Third Applied NLP, pp. 156-161, 1992.

[6] Hall R. J., *Learning by Failing to Explain: Using Partial Explanation to Learn in Incomplete or Intractable Domains,* Machine Learning, 3: 45-77, 1988.

[7] Holder L. B., *The General Utility Problem in Machine Learning,* Proc. of the Seventh

International Machine Learning Conference, pp. 402-410, 1990.

[8] Keller R. M., *Defining Operationality for Explanation-Based Learning,* Artificial Intelligence, 35: 227-241, 1988.

[9] Lang F.-M. and Hirschman L., *Improved Portability and Parsing through Interactive Acquisition of Semantic Information,* Proc. of the 2nd conference on Applied Natural Language Processing, pp. 49-57, 1988.

[10] Lehman J. F., *Adaptive Parsing: A General Method for Learning Idiosyncratic Grammars,* Proc. of the seventh international machine learning conference, pp. 368-376, 1990.

[11] Liu R.-L. and Soo V.-W., *Parsing Driven Generalization for Natural Language Acquisition,* Proc. of ROCLING, R.O.C., pp. 351-376, 1990.

[12] Liu R.-L. and Soo V.-W., *Dealing with Ambiguities in English Conjunctions and Comparatives by A Deterministic Parser,* International Journal of Pattern Recognition and Artificial Intelligence, December, Vol. 4, No. 4, pp. 629-649, 1990.

[13] Liu R.-L. and Soo V.-W., *Augmenting and Efficiently Utilizing Domain Theory in Explanation-Based Natural Language Acquisition,* Proc. of the 9th International Machine Learning Conference, ML92, 1992.

[14] Lu R., Liu Y., and Li X., *Computer-Aided Grammar Acquisition in the Chinese Understanding System CUSAGA,* Proc. of IJCAI, pp. 1550-1555, 1989.

[15] Lytinen S. L. and Moon C. E., *A Comparison of Learning Techniques in Second Language Learning,* Proc. of the 7th Machine Learning conference, pp. 377-383, 1990.

[16] Minton S., *Quantitative Results Concerning the Utility of Explanation-Based Learning,* Proc. of AAAI, pp. 564-569, 1988.

[17] Mitchell T. M., Keller R. M., and Kedar-Cabelli S. T., *Explanation-Based Generalization: A Unifying View,* Machine Learning, 1: 47-80, 1986.

[18] Pazzani M. J., *Detecting and Correcting Errors of Omission After Explanation-based Learning,* Proc. of IJCAI, pp. 713-718, 1989.

[19] Chomsky N., *Lectures on Government and Binding* Foris Publications - Dordrecht, 1981.

[20] Pinker S., *Language Learnability and Language Development,* The Harvard University Press, Cambridge, Massachusetts, London, England, 1984.

[21] Sanfilippo A. and Poznanski V., *The Acquisition of Lexical Knowledge from Combined Machine-Readable Dictionary Sources,* Proc. of the Third Conference on Applied NLP, pp. 80-87, 1992.

[22] Sekine S., Carroll J. J., Ananiadou S., and Tsujii J., *Automatic Learning for Semantic Collocation,* Proc. of the Third Conference on Applied NLP, pp. 104-110, 1992.

[23] Simmons R. F. and Yu Y.-H., *The Acquisition and Application of Context Sensitive Grammar for English,* Proc. of ACL, pp. 122-129, 1991.

[24] Siskind J. M., *Acquiring Core Meanings of Words, Represented as Jackendoff-style Conceptual structures, from Correlated Streams of Linguistic and Non-linguistic Input,* Proc. of the 28th annual meeting of ACL, pp. 143-156, 1990.

[25] Smadja F. A., *From N-Grams to Collocations: An Evaluation of EXTRACT,* Proc. of ACL, pp. 279-284, 1991.

[26] Velard P., Pazienza M. T., and Fasolo M., *How to Encode Semantic Knowledge: A Method for Meaning Representation and Computer-Aided Acquisition,* Computational Linguistic, Vol. 17, No. 2, pp. 153-170, 1991.

[27] Zernik U., *Learning Idioms -- With and Without Explanation,* Proc. of IJCAI, pp. 133-136, 1987.

[28] Zernik U., *Lexicon Acquisition: Learning from Corpus by Capitalizing on Lexical Categories,* Proc. of IJCAI, pp. 1556-1562, 1989.

[29] Zernik U. and Jacobs P., *Tagging for Learning: Collecting Thematic Relation from Corpus,* Proc. of COLING, pp. 34-39, 1990.

[30] Huang C.-R., *Certainty in Functional Uncertainty,* Journal of Chinese Linguistics, Vol. 20, No. 2, pp. 247-287, 1992.

STATISTICAL MODELS FOR WORD SEGMENTATION AND UNKNOWN WORD RESOLUTION

Tung-Hui Chiang, Jing-Shin Chang, Ming-Yu Lin and Keh-Yih Su

Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
{andy,shin,felipe,kysu}@ee.nthu.edu.tw

ABSTRACT

In a Chinese sentence, there are no word delimiters, like blanks, between the "words". Therefore, it is important to identify the word boundaries before processing Chinese text. Traditional approaches tend to use dictionary lookup, morphological rules and heuristics to identify the word boundaries. Such approaches may not be applied to a large system due to the complicated linguistic phenomena involved in Chinese morphology and syntax. In this paper, the various available features in a sentence are used to construct a generalized word segmentation model; the various probabilistic models for word segmentation are then derived based on the generalized model.

In general, the likelihood measure adopted in a probabilistic model does not provide a scoring mechanism that directly indicates the real ranks of the various candidate segmentation patterns. To enhance the baseline models, a robust adaptive learning algorithm is proposed to adjust the parameters of the baseline models so as to increase the discrimination power and robustness of the models.

The simulation shows that cost-effective word segmentation could be achieved under various contexts with the proposed models. It is possible to achieve accuracy in word recognition rate of 99.39% and sentence recognition rate of 97.65% in the testing corpus by incorporating word length information to a context-independent word model and applying a robust adaptive learning algorithm in the segmentation process.

Since not all lexical items could be found in the system dictionary in real applications, the performance of most word segmentation methods in the literature may degraded significantly when unknown words are encountered. Such an "*unknown word problem*" is also examined in this paper. An error recovery mechanism based on the segmentation model is proposed.

Preliminary experiments show that the error rates introduced by unknown words could be reduced significantly.

1. Introduction

Most natural language processing tasks, such as machine translation or spoken language processing, take *words* as the smallest meaningful units. However, no obvious delimiter markers can be observed between Chinese words except for some punctuation marks. Therefore, word segmentation is essential in almost all Chinese language processing tasks. (The same is true for other languages like Japanese.)

Matching input characters against the lexical entries in a large dictionary is helpful in identifying the embedded words. Unfortunately, an input sentence can usually be segmented into more than one segmentation patterns. For example, a Chinese sentence like:

對方姑娘而言，立志當政治家的沒有一個功成名就的。

may include the following ambiguous segmentation patterns based on simple dictionary lookup:

1.+ 對 方姑娘 而言 ， 立志 當 政治家 的 沒有 一 個 功成名就 的。
TO MS. FANG, those who decide to BE A STATESMAN never succeed and become famous.

2.* 對 方姑娘 而言 ， 立志 當政 治家 的 沒有 一 個 功成名就 的。
TO MS. FANG, those who decide to HOLD POWER and MANAGE A HOUSEHOLD never ...

3.* 對方 姑娘 而言 ， 立志 當政 治家 的 沒有 一 個 功成名就 的。
To the LADY of the COUNTER PARTY, those who decide to HOLD POWER and MANAGE A HOUSEHOLD never ...

4.* 對方 姑娘 而言 ， 立志 當 政治家 的 沒有 一 個 功成名就 的。
To the LADY of the COUNTER PARTY, those who decide to BE A STATESMAN never ...

⋮

where the first segmentation pattern is the preferred one. To find the correct segmentation pattern, it is necessary to use other information sources in addition to dictionary lookup. The main issue for dealing with the word segmentation problem is how to find out the *correct* segmentation from all possible ones.

There are several technical reasons that make the word segmentation problem nontrivial. First, the Chinese characters can be combined rather freely to form legal words. As such, ambiguous segmentation patterns may not be resolved by using simple dictionary lookup.

Second, a Chinese text contains not only words but also inflectional or derivational *morphemes*, *tense markers*, *aspect markers*, and so on. Because such morphemes and markers may often be combined with adjacent characters to form legal words as well as standing alone as a word, it is hard to deal with such ambiguities with simple morphological analysis.

Third, *unknown words* may appear in the input text. This fact may make many word segmentation models work badly in real applications, because most segmentation algorithms today assume that all words in the input text could be found in the system dictionary. In fact, unknown word resolution has become the major bottleneck with the current segmentation techniques.

To resolve these problems, various knowledge sources might have to be consulted. However, extensive use of high level knowledge and analysis may requires extremely high computation cost. Hence, segmentation algorithms that make use of discriminative and easily acquired features are desirable.

In the past, two different methodologies were used for word segmentation; some approaches are *rule-based* (Chen [3, 4], Ho [7], Yeh [10]) while others are *statistical* ones (Chang [2], Fan [6], Sproat [8]). Since it is costly to construct lexical or morphological rules by hand, no objective preference could be given for ambiguous segmentation patterns, and it is difficult to maintain rule consistency as the size of the rule base increases, it is less favorable to use a rule-based approach in large scale applications. On the contrary, as data are jointly considered in a statistical framework, statistical approaches usually do not suffer from the consistency problem. Also, global optimization can usually be modeled in statistical frameworks, rather than local constraints by rules. Therefore, statistical approaches are usually more practical in a large application like machine translation. However, the current statistical approaches usually use a maximum likelihood measure to evaluate preference without regarding to the discrimination power of such models. As a result, when the baseline models introduce errors, heuristic approaches, such as adding special information to the dictionary or resorting to later syntactic or semantic analyses are suggested (Chang [2]) to remedy the modeling and estimation errors. Such approaches not only destroy the uniformity of the statistical methods but also make maintenance difficult.

To resolve the above problems, several probabilistic models are proposed in this paper based on a generalized word segmentation model. The focus is to derive different formulations under different constraints of the available resources. In particular, features that could be acquired inexpensively will be used for cost-effective word segmentation so that deep analyses are needed only to the least extent.

In order to adapt the probabilistic models to reflect the real *ranks* of the candidate segmentation patterns and to suppress *statistical variations* among different application domains, a discrimination and robustness oriented adaptive learning algorithm (Su [9], Chiang[5]) is applied to enhance the performance. Moreover, the *unknown word problem* will be addressed and be examined against the proposed models; some experiment results are given and general guidelines to this problem will be suggested.

2. Word Segmentation Models

2.1 A Generalized Word Segmentation Model

For an input sentence with n Chinese characters $c_1, c_2, \cdots, c_n$ (represented as c_1^n hereafter), it might have several different ways of segmentation according to the system dictionary. The goal of word segmentation is to find the *most probable* segmentation pattern for the given character string. Since a segmentation pattern can be identified uniquely with the sequence of words of the segmented sentence. The goal is equivalent to finding a word sequence

$$\hat{W} \equiv \underset{W_i}{\operatorname{argmax}} P\left(W_i \mid c_1^n\right) \tag{2.1.1}$$

with the largest *segmentation score* $P\left(W_i \mid c_1^n\right)$. In this formula, $\underset{W_i}{\operatorname{argmax}} P(\cdot)$ refers to the argument, among all possible W_i's, that maximizes the probabilistic function $P(\cdot)$, and $W_i \equiv w_{i,1}^{i,m_i} = w_{i,1}, w_{i,2}, \cdots, w_{i,m_i}$ denotes the i-th possible word sequence with m_i words, whose j-th element is $w_{i,j}$.

In general, we could formulate the segmentation score by involving whatever features that are considered discriminative or available, subject only to the constraints of the complexity of the model and the number of parameters that need to be trained. In particular, we would like to use the segmented words (W_i), the word length information (L_i), the number of characters (n) in the input sentence and the number of words (m_i) for the i-th segmentation pattern as the features for word segmentation. ($L_i \equiv l_{i,1}^{i,m_i} = l_{i,1}, l_{i,2}, \cdots, l_{i,m_i}$ refers to the i-th sequence of word lengths, where $l_{i,j}$ denotes the length of the j-th word in the i-th possible

word sequence.) These features could be acquired inexpensively in general. Thus, they are adopted in the current task. With these features, we can identify a "segmentation pattern" uniquely with a (W_i, L_i, m_i) triple, and the goal of word segmentation would become to find the word segmentation pattern corresponding to

$$\underset{i}{\operatorname{argmax}}\, P\left(W_i, L_i, m_i \mid c_1^n, n\right) \tag{2.1.2}$$

Hence, we could define a *generalized segmentation score* as:

$$P\left(W_i, L_i, m_i \mid c_1^n, n\right) \tag{2.1.3}$$

Note that the variables, such as W_i and L_i, are not independent. Technically, however, these features are integrated in a single formula so that all models that are computationally feasible could be derived from this general formula; unavailable features will simply be ignored when deriving a particular model.

The generalized segmentation score can be estimated in several different ways depending on the available information resources. In the following sections, we will give a more detailed derivation of a particular model, which takes advantage of the segmented words and the word length information for segmentation. Other models can be derived in much the same way. So they are simply listed without proof.

2.2 Computational Models for Word Segmentation

Assume that a segmented text corpus is available, then we can use the frequency information of the words and their lengths (in characters) for segmentation. The corresponding segmentation score for the i-th segmentation pattern will be:

$$\begin{aligned}
&P\left(L_i, W_i, m_i \mid c_1^n, n\right) \\
&= P\left(l_{i,1}^{i,m_i}, w_{i,1}^{i,m_i}, m_i \mid c_1^n, n\right) \\
&\equiv P_i\left(l_1^m, w_1^m, m \mid c_1^n, n\right) \\
&= P_i\left(l_1^m, w_1^m \mid m, c_1^n, n\right) \times P_i\left(m \mid c_1^n, n\right) \\
&= \prod_{k=1}^{m_i} P_i\left(l_k, w_k \mid l_1^{k-1}, w_1^{k-1}, m, c_1^n, n\right) \times P_i\left(m \mid c_1^n, n\right) \\
&= \prod_{k} P_i\left(l_k \mid w_k, l_1^{k-1}, w_1^{k-1}, \cdots, n\right) \cdot P_i\left(w_k \mid l_1^{k-1}, w_1^{k-1}, \cdots, n\right) \times P_i\left(m \mid c_1^n, n\right)
\end{aligned} \tag{2.2.4}$$

For notational simplicity, $P_i(\cdot)$ is used specifically to denote the probability for the i-th segmentation pattern, and all the respective i indices are dropped from the equation. The multiplication theory for probability: $P(a, b \mid c) = P(a \mid b, c) \times P(b \mid c)$, is applied repeatedly in the derivation, which results in the product terms, indexed by k, in the last two formulae.

Since l_k is unique once w_k is given, we have $P(l_k \mid w_k, \cdots) = 1$ for the first term in the equation. If we assume that the k-th word depends only on the length l_{k-1} of the previous word, the second term in the last formula can be approximated as $P\left(w_k \mid l_1^{k-1}, w_1^{k-1}, \cdots, n\right) \approx P(w_k \mid l_{k-1})$. Furthermore, if we assume that the number of words m_i depends only on the length of the sentence n, then we have $P_i\left(m \mid c_1^n, n\right) \approx P_i(m \mid n)$. With these assumptions, the segmentation problem is equivalent to finding:

$$\begin{aligned} &\underset{i}{\operatorname{argmax}} P(W_i, L_i, m_i \mid c_1^n, n) \\ &\approx \underset{i}{\operatorname{argmax}} \prod_k P_i(w_k \mid l_{k-1}) \times P_i(m \mid n) \\ &= \underset{i}{\operatorname{argmax}} \sum_k \log P_i(w_k \mid l_{k-1}) + \log P_i(m \mid n) \end{aligned} \tag{2.2.5}$$

where $\log(\cdot)$ refers to a logarithmic function. (The log-scaled probabilities are used simply to reduce the computation time and avoid mathematical underflow.) There are several variants of the above equation, depending on different assumptions made in deriving the segmentation score. First, it is possible to drop the term $P_i(m \mid n)$ or $\sum \log P_i(w_k \mid l_{k-1})$, depending on what information is available, in the previous derivation steps. Alternatively, we can also assume that the word w_k does not depend on the length of the preceding word length l_{k-1}, and thus use $P_i(w_k)$ instead of $P_i(w_k \mid l_{k-1})$ in the formula. By changing the roles of w_k and l_k in the last step of derivation, we can use the transition probability $P_i(l_k \mid l_{k-1})$ instead of $P_i(w_k \mid l_{k-1})$ in the segmentation score. Therefore, the above formula along with its variants constitute a family of segmentation scores as shown below:

$$\begin{aligned} &\underset{i}{\operatorname{argmax}} P(W_i, L_i, m_i \mid c_1^n, n) \\ &\approx \underset{i}{\operatorname{argmax}} \begin{cases} \sum_{k=1}^{m_i} \log P_i(w_k) & (M1) \\ \sum_{k=1}^{m_i} \log P_i(l_k \mid l_{k-1}) & (M2) \\ \log P_i(m \mid n) & (M3) \\ \sum_{k=1}^{m_i} \log P_i(w_k \mid l_{k-1}) & (M4) \end{cases} \end{aligned} \tag{2.2.6}$$

Model M1 is a context-independent word model. It assumes that all words are independent of the other contextual information. Such a model is used in Chang [2] for the segmentation task.

Model M2 uses only the word length transition probabilities in determining the word segmentation patterns. Model M3, on the other hand, uses the number of characters and the number of words in a sentence as the features for segmentation. It seems that such features have nothing to do with the characteristics of Chinese words. However, as shown in Chang [2] and other literatures, most Chinese words are double-character words, single-character words and tri-character words; more than 99% of Chinese words fall within 4 characters. Hence, it is possible to make guesses based on word length information.

Moreover, the length information could be acquired without much extra cost when preparing a segmented corpus. Therefore, such features could provide an inexpensive way for word segmentation in applications where a large dictionary is not available or expensive to acquire. In fact, as will be seen in the performance evaluation section, the performance of such formulations is comparable with others. So it could be used, for instance, to bootstrap the automatic construction process of an electronic dictionary, where there is not a large dictionary initially.

Model M4 uses both word sequence and word length information for segmentation. If the word length information is ignored, this model reduces to M1. By using the extra word length information, which could be acquired from the same corpus for training model M1, this model could make use of more information and the performance is expected to be better if the training corpus is large enough to provide reliable estimation of the model parameters.

If a sentence is annotated with *lexical tags* (i.e., parts of speech) $T_{i,j} \equiv t_{i,j,1}, \cdots, t_{i,j,m_i}$, then it is possible to use such information to define a modified segmentation score. (Tag $t_{i,j,k}$ stands for the k-th part of speech in the j-th possible tag sequence of the i-th segmentation pattern.) One can achieve the same optimization criteria as that of the generalized segmentation score by noting that:

$$\begin{aligned} & \underset{i}{\operatorname{argmax}} P\left(W_i, L_i, m_i \mid c_1^n, n\right) \\ & = \underset{i}{\operatorname{argmax}} \sum_{all\ T_{i,j}} P\left(W_i, L_i, T_{i,j}, m_i \mid c_1^n, n\right) \\ & \approx \underset{i}{\operatorname{argmax}} \left[\max_{all\ T_{i,j}} P\left(W_i, L_i, T_{i,j}, m_i \mid c_1^n, n\right) \right]. \end{aligned} \tag{2.2.7}$$

The last formula means to find the tag sequence $T_{i,j}$ with the largest score as defined by

$$P\left(W_i, L_i, T_{i,j}, m_i \mid c_1^n, n\right) \tag{2.2.8}$$

for each possible segmentation pattern. Then select the segmentation pattern with the highest maximum score as the preferred segmentation pattern.

By following the same procedures as in Eq. (2.2.4) and making some assumptions, it is not difficult to find that the following word segmentation models could be used when the lexical tag information is available:

$$\begin{aligned} &\underset{i}{\operatorname{argmax}}\, P\left(W_i, L_i, m_i \mid c_1^n, n\right) \\ &\approx \underset{i}{\operatorname{argmax}} \begin{cases} \max\limits_{T_{i,j}} \sum\limits_{k=1}^{m_i} \log P_{ij}\left(t_k \mid t_{k-1}\right) & (M5) \\ \max\limits_{T_{i,j}} \sum\limits_{k} \log P_{ij}\left(w_k \mid l_{k-1}\right) + \sum\limits_{k} \log P_{ij}\left(t_k \mid t_{k-1}\right) & (M6) \\ \max\limits_{T_{i,j}} \sum\limits_{k} \log P_{ij}\left(w_k \mid t_{k-1}\right) + \sum\limits_{k} \log P_{ij}\left(t_k \mid t_{k-1}\right) & (M7) \end{cases} \end{aligned} \tag{2.2.9}$$

Here, we use $P_{ij}(\cdot)$ to specify the probability associated with the i-th segmentation pattern and the j-th tag sequence, with the corresponding indices within the parentheses omitted.

Model M5 is used to find the best parts of speech sequence associated with the ambiguous segmentation patterns. So the segmentation pattern that produces the most possible lexical tag sequence is regarded as the desired one. In Model M6, the parts of speech sequence is taken into account to facilitate word segmentation model M4. In model M7, the segmentation is considered best if the segmentation pattern maximizes the sequence of corresponding parts of speech and the sequence of words. Because both word sequence and lexical tag sequence are the target of optimization in this process, such a formula can be used, with some *reestimation* techniques, to segment the words and assign parts of speech to each word at the same time automatically.

3. Discrimination and Robustness Oriented Adaptive Learning

There are several technical problems with a general probabilistic model. First, the *model* might not be good enough to formulate the characteristics of the task under consideration. This problem can usually be relieved by using appropriate features and by considering more contextual information when constructing the model. Second, the parameters of the model might not be estimated correctly due to the lack of a large corpus. This problem can usually

be made less severe by using a larger database or better estimation techniques. Nevertheless, even if such *modeling* problem and the *estimation* problem could be resolved, it does not mean that the *ranks* of the estimated probabilistic measure are the same as the ranks of preference of the candidate segmentation patterns. Correct recognition, however, depends on the relative order of the ranks of the candidates.

The criteria of rank ordering and maximum likelihood are usually not equivalent, although they are highly correlated. Therefore, maximum likelihood estimation does not necessarily result in minimum error rate for data in the *training* set. For these reasons, the estimated parameters for the baseline models need to be adjusted to reflect the ranks of the candidate segmentation patterns. Hence, another (probably more) important issue is how to adjust the estimated likelihood measures so as to reflect the real ranks. We do this by adjusting the values of these probability terms based on the misjudged instances. By doing so, the set of parameters could be adjusted toward the goal of minimizing the error rate of the *training* corpus directly.

Furthermore, since statistical variations between a testing set and a training set are not taken into consideration in the baseline models, minimizing the error rate in the *training set* does not imply maximizing the recognition rate in an independent *testing set*, either. To enhance robustness, an extra step can be adopted to enlarge the difference in scores between the best scored candidate and the other candidates. This step will enhance the robustness of the model so that the performance will not be affected significantly by different text styles.

3.1 Adaptive Learning

The goal of adaptive learning is to provide a new parameter set, Λ', such that the new parameters in Λ' can provide more discrimination capability than the baseline parameter set Λ by adjusting the current parameters based on the misjudged training tokens. The basic idea is to adjust the parameters associated with the segmentation score of the correct candidate when the correct candidate is superseded by other candidates of larger scores; the adjustment will be continued until the modified score of the correct candidate is the largest among all candidates. Let y_k be the candidate whose segmentation score is the largest among all the candidates for the k-th training sentence, and let z_k be the correct candidate, then a distance measure $d_\Lambda(y_k, z_k)$ could be defined as a measure of separability between y_k and z_k. In particular, since we are concerned with the ranking order of the scores of the candidates, the *differences* of the segmentation scores could be used as the distance measure.

A larger difference between the segmentation scores for the correct candidate and the highest-scored candidate implies larger penalty of misjudgement. Thus, we can define a loss function as an increasing function of the distance, such as $\tan^{-1}(d_\Lambda/d_0)$ (Amari [1]), to indicate the penalty suffered from misjudgement.

To acquire a better parameter set, each parameter corresponding to the misjudged sentence is changed by a small amount in each iteration of learning so as to reduce the penalty of misjudgement; the amount of adjustment, say δA, will depend on the loss or penalty of misjudgement. Take the following segmentation patterns as an example:

1. 對 方姑娘 而言
   W1 W2 W3

2. 對方 姑娘 而言
   W1′ W2′ W3′

If model M1 is used, then the segmentation scores for these two patterns are determined by 5 parameters, namely, P1 = logP(W1), P2 = logP(W2), P3 = logP(W3) and P1′ = logP(W1′), P2′ = logP(W2′), P3′ = logP(W3′) (= P3, in this case), respectively. Assume that the initial values of these parameters are P1 = -1.8, P2 = -2.6, P3 = -1.7, P1′ = -1.6, P2′ = -2.3, and P3′ = -1.7, then the segmentation score of the first candidate (which is also the correct pattern) is -6.1 (= -1.8 -2.6 -1.7) and the segmentation score of the second candidate (which has the highest score) is -5.6 (= -1.6 -2.3 -1.7). Since this training sentence is misjudged, we may suffer from a loss whose penalty depends on the distance, namely the difference between the scores, (-5.6) - (-6.1) =0.5.

If the value of the loss function for this distance is 0.46, and the amount of adjustment, δA, for that amount of loss is 0.2, then we have a revised parameter set: P1 = -1.8+0.2 = -1.6, P2 = -2.6+0.2 = -2.4, P1′ = -1.6-0.2 = -1.8, P2′ = -2.3-0.2 = -2.5 and , P3 = P3′ = -1.7. Note that since P3 (P3′) happens to be adjusted in both patterns by the same amount, this parameter will not be changed after adjustment.

It is obvious that the correct candidate now has a higher score after parameter adjustment. Moreover, the parameters for the highest-scored candidate, which might be responsible for the misjudgement, are reduced after adjustment. So other misjudged sentences might also be affected by the adjustment of these parameters. If the correct candidate is still not the one with the highest score after the adjustment, the same procedure can be repeated; the

parameters of the correct candidate and the (possibly new) highest-scored candidate will be adjusted further until the correct candidate has the highest score.

Although the amount of adjustment for the various P(W)'s is shown to be the same in the current example, it may have to be weighted differently when we consider different information sources jointly. For instance, in model M6, we may use a smoothing technique to get a better estimated score by assigning different weights to the $P(w_k \mid l_{k-1})$ terms and the $P(t_k \mid t_{k-1})$ terms. Under such circumstance, the amount of adjustment for these two kinds of parameter sets will also be weighted by the same amount to account for their respective contributions.

Under appropriate conditions, it can be proved that the average amount of change in average loss will be *decreased* due to the adaptation (Amari [1]). Therefore, it is guaranteed that, by adjusting the parameters Λ of the baseline models in this manner, the discrimination power, in terms of the distances between the correct candidate and the other segmentation patterns, will be increased. Furthermore, since the amount of change in the parameters is directly proportional to the gradient of the loss function (Amari [1], Chiang [5], Su[9]), this also implies changing the parameters Λ in the direction in which the change in mean loss is the most drastic. Therefore, the speed of convergence is fast with this learning algorithm.

3.2 Robustness Enhancement

In addition to enhancing the *discrimination power* of the segmentation models, the *robustness* of the segmentation models is also an important concern. The robustness could be enhanced by increasing the "margin" of distances between the correct pattern and the other competing candidates (Su [9]). This can be done by adjusting the scores of the correct segmentation pattern and the one with the secondary highest score even after the correct segmentation pattern has been assigned the highest score. The adjustment of the parameters will stop only after the distance margin between the correct one and the candidate with the secondary highest score exceeds a given threshold. This will ensure that the correct candidate is separated from other competing candidates by at least the prescribed amount of margin. In this stage, the loss will be measured in terms of the distance between the top 2 candidates.

By enforcing a "margin" between the correct segmentation pattern and the most competitive candidate, the segmentation score will be more robust in the sense that any *statistical variations* between the *training corpus* and the *real instances* in the various applications could be properly suppressed. It is very important to enhance the robustness of the models in this

way, because the instances in real applications could not be predicted in advance. For more technical information on the robust adaptive learning algorithm, please refer to (Amari [1], Chiang [5], Su [9]).

4. Resolution of the Unknown Word Problem

Most word segmentation models in the literature are based on a simple assumption that all words in the text could be found in the system dictionary; there are no "unknown words" to the dictionary. However, as will be seen in a later section, such an assumption is usually unrealistic; the error introduced by unknown words, such as unknown proper nouns, constitutes a large fraction of the error rate in word segmentation. Therefore, it is important to take the unknown word problem seriously in dealing with real applications.

A word may become unknown to the system simply because it was not stored in the dictionary or because it belongs to some particular types of words, such as proper nouns, that can not be enumerated exhaustively. Sometimes, a substring of an unknown word is a legal word in the dictionary. In this case, the unknown word will be divided into pieces in the dictionary lookup process. It is also possible that an unknown word is a substring of some legal words in the dictionary. In this case, the unknown word will be hidden behind the legal word. All these error transformations: missing entry, separation of the unknown word into pieces, and hidden by a legal word, make it impossible to find all segmentation patterns by a simple dictionary lookup process.

The general solution is to take possible inverse error transformations in the vicinity of an unknown word; then evaluate the segmentation score or a revised version of it to select the most possible segmentation pattern, with unknown words recognized as a particular class of character stream of unknown length. This means to extend the segmentation patterns acquired from simple dictionary lookup by combining or dividing characters in a prescribed window where an unknown word is suspected to occur, and choose the most likely segmentation pattern from the set of extended segmentation patterns, including those candidates that are introduced by the unknown word problem. The general solution could be very complicated and will be addressed in other papers. Here, we just show a simplified version, and reveal some technical issues in unknown word resolution.

In particular, we could regard an unknown word, say w_u, as a unit of unknown length l_u that could possibly appear anywhere in the region where an unknown word is suspected to occur. We then use the dependency of the class of unknown words with their context to

determine the preference of the various segmentation patterns. The main task is to determine the positions and lengths of the unknown words in the suspected "unknown word regions" as shown below.

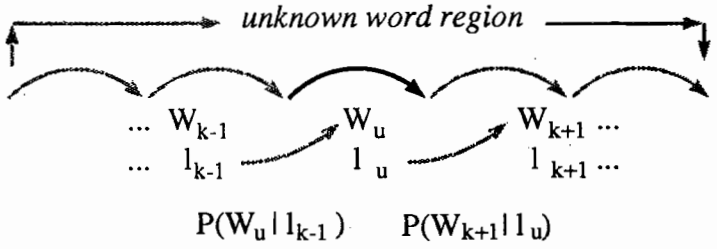

Figure 1 Evaluating segmentation score when unknown words are encountered.

For simplicity, assume that an unknown word region has been identified and exactly *one* unknown word is within the region, then we can formulate the segmentation score as in any of the previously mentioned models by replacing $w_{i,k}$ in one of the probability terms with w_u, and evaluate the segmentation scores for the various possible locations and lengths in the same way as if it was a known word. For example, if model M4 is applied to the suspected unknown word position and word length in Figure 1, we will have probability terms like:

$$score \approx \cdots \times P(w_u \mid l_{k-1}) \times P(w_{k+1} \mid l_u) \times \cdots \tag{4.1}$$

where $P(w_u \mid l_{k-1})$ is the probability that an unknown word will follow a word of length l_{k-1}, and $P(w_{k+1} \mid l_u)$ is the probability that the next word w_{k+1} will appear after an unknown word of length l_u.

The transition probabilities concerning the unknown words could be estimated from the training corpus by counting the relative frequencies of the lexical entries that could not be found in the system dictionary and the word lengths of their surrounding words.

Also, to rate the possibility that the suspected unknown word region does contain an unknown word, the above formulation must contain a factor of the form:

$$P\left(c_i^j \text{ contains an unknown word of length } l_u \text{ at position } k \mid c_1^n\right) \tag{4.2}$$

which serves to detect the unknown word regions. The detection of the unknown word regions is a nontrivial task. For the present, we just use the available word length information and the following simplified formula to account for the above factor:

$$P(L_{uwr}) \times P\left(w_u \in c_i^{i+L_{uwr}-1} \mid L_{uwr}\right) \times P\left(l_u \mid w_u \in c_i^{i+L_{uwr}-1}\right) \quad (4.3)$$

where $P(L_{uwr})$ is the prior probability that the unknown word region ("uwr") consists of isolated single characters of length L_{uwr}; $w_u \in c_i^{i+L_{unr}-1}$ stands for the event that an unknown word does exist in the unknown word region, and $P\left(l_u \mid w_u \in c_i^{i+L_{unr}-1}\right)$ is the probability that the unknown word length in such an unknown word region is of length l_u. The results will be investigated in the analysis section.

5. Test and Analysis

5.1 Simulation

To compare the performance of the various models, a Chinese text corpus with articles from different domains is constructed for evaluation. The contents of the corpus are mostly related to politics, economics and cinema review.

The sentences are segmented by hand so that they could be used for training or testing, as well as for comparison with machine processed results. The characters between punctuation marks are segmented into smaller tokens. Because there is no common standard about the definition of Chinese words, some rules of thumb are used for manual segmentation. In particular, the following principles of segmentation are taken to keep it as consistent as possible.

1. Frequently used compound nouns and idiomatic expressions are segmented as single words without further segmentation.
2. A segment that has a direct mapping with an English word is considered a Chinese word. This technical principle is adopted specifically for the machine translation system we are working with.
3. Small segments that could be derived with general morphological rules are merged and be regarded as one word. In general, such words can be formed in the lexical analysis phase with a simple finite state machine. Therefore, the merged segments are considered a word that should be output by the segmentation algorithm as one unit.
4. When a segment is segmented into smaller tokens and the semantics of this segment can not be recovered by the compositional semantics of the smaller tokens, then the original segment will be regarded as a single word.

5. A large segment that contains a predicate part, its arguments or complements, negation markers or aspect markers is divided into smaller segments corresponding to the respective parts. This makes it easy to map each part to its syntactic or semantic construct when used for natural language applications. In fact, the purpose of word segmentation is to find the terminal words to be used by a syntactic or semantic analyzer. Therefore, those segments that could be mapped directly to the syntactic or semantic constructs are identified as such terminal words.
6. When conflicts are encountered in applying these principles, judgement is given by the human according to the frequency of use.

The testing sentences are scanned and all ambiguous segmentation patterns allowed by dictionary lookup are constructed. The various segmentation patterns are then scored with the various segmentation models. Adaptive learning as well as robustness enhancement are performed to improve the segmentation models in some testing cases. The top-1 candidate is then compared with the hand parsed results to evaluate the performance of the model under consideration.

Instead of judging the correctness by human inspection *after* the machine processed results are produced, a file is prepared to hold hand-parsed segmentations for comparison *before* the evaluation is started; the file is kept untouched throughout the evaluation process for all models. Such arrangement ensures that the evaluation is not affected by personal judgement, which may vary from one time to another, and keeps a consistent criterion of correctness.

The dictionary contains 99,441 entries, and about 9,755 words are actually encountered in the corpus. The tag set for models M5 – M7 contains a total of 22 parts of speech for Chinese and 3 special tags. (The testing environment is shown in Table 7.) To see the effects of unknown words on the performance of word segmentation, some tests are conducted in two modes, one with unknown words in the testing sentences and the other with all unknown words inserted to the dictionary.

5.2 Performance Evaluation

Since most models exhibit high recognition accuracy, the error rate, defined as "100%-Accuracy" is emphasized in performance evaluation. (The word accuracy or sentence accuracy are shown in the parentheses for comparison with other reports though.) The word accuracy is defined as the number of correctly segmented words divided by the total number of words in manually segmented sentences. The sentence accuracy, on the other

hand, is defined as the number of correctly segmented sentences divided by the total number of sentences involved in testing. Here, a sentence actually refers to a segment between the punctuation marks. A sentence is said to be "correctly segmented" if none of the words in the sentence is incorrectly identified.

Baseline Performance

Table 1 and Table 2 show the baseline performance with models M1, M2, M3 and M4 as shown in Eq. (2.2.6). In Table 1, the training and testing sentences contain unknown words, which can not be found in the dictionary. In Table 2, all unknown words are entered to the dictionary as legal entries.

Model	Training Set Error (*Accuracy)		Testing Set Error (*Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
Max Match-1	4.01 (95.99)	20.74 (79.26)	4.23 (95.77)	20.68 (79.32)
Max Match-2	4.01 (95.99)	20.77 (79.23)	4.15 (95.85)	20.54 (79.46)
P(Lk|Lk-1)	8.70 (91.30)	45.54 (54.46)	9.41 (90.59)	47.86 (52.14)
P(m|n)	7.19 (92.81)	38.61 (61.39)	7.82 (92.18)	39.30 (60.70)
P(Wk)	3.62 (96.38)	19.81 (80.19)	3.94 (96.06)	19.97 (80.03)
P(Wk|Lk-1)	3.68 (96.32)	20.08 (79.92)	4.07 (95.93)	21.04 (78.96)
(*) The numbers in the parentheses show the accuracy rates				

Table 1 Baseline Performance WITH Unknown Words

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
Max Match-1	1.14 (98.86)	4.05 (95.95)	1.22 (98.78)	4.07 (95.93)
Max Match-2	1.14 (98.86)	4.07 (95.93)	1.12 (98.88)	3.78 (96.22)
P(Lk|Lk-1)	6.16 (93.84)	37.57 (62.43)	6.82 (93.18)	40.09 (59.91)
P(m|n)	5.24 (94.76)	28.53 (71.47)	5.71 (94.29)	29.60 (70.40)
P(Wk)	0.54 (99.46)	2.07 (97.93)	0.76 (99.24)	2.50 (97.50)
P(Wk|Lk-1)	0.47 (99.53)	1.77 (98.23)	0.73 (99.27)	2.50 (97.50)

Table 2 Baseline Performance WITHOUT Unknown Words

A commonly used heuristic approach, designated as "Max(imum) Match-1", is also shown for comparison. It scans the input from left to right and from right to left, respectively, to match against the dictionary entries; the one with a smaller number of words is considered the preferred segmentation pattern. During the scanning process, if two matches against the dictionary entries are possible from the current word boundary, then the one with a larger number of characters is selected as the correct match. If the total number of words in both scanning directions are the same, then the first distinct word, either from left or from right, is compared. The segmentation pattern corresponding to the word with a larger number of characters is selected as the preferred pattern. A variant of the maximum match approach, designated as Max Match-2, as proposed in Chen [4] (Heuristic rule #1), is also implemented for comparison. It scans the text left-to-right and uses a 3–word sequence, instead of a single word, to judge the preference of the first word in this sequence.

There are several interesting and important points to point out concerning the above performance. First, it is surprising that a "trivial" model like model M2 ($P_i(l_k \mid l_{k-1})$) or model M3 ($P(m_i \mid n)$), which uses only the word length, word count and character count information, achieve comparable performance in word accuracy as the other models that make use of word information.

As noted previously, Chinese words are mostly double-character words, single-character words and tri-character words. This implies that there might be useful information in the dependencies between word lengths and even character counts or word counts. Therefore, it is significant to use such features for segmentation. As can be seen from the tables, such a trivial model is not significantly worse than other more "reasonable" models. This means that word segmentation could be easily resolved statistically even with a simple model like model M2 or M3. Because the number of parameters for these two models are very small and the parameters do not refer to any lexical entries, they could be used in some applications where a large dictionary is unavailable.

Second, the unknown words introduce significant error rates. The word accuracy is degraded by about 2–3% in both training set or testing set, and the sentence accuracy is degraded by about 8%-19%. This means that the unknown word problem is a major source of errors for the word segmentation problem. The degradation is also observed between Table 3 and Table 4 even after adaptive learning is applied; in this case, the degradation in word accuracy is about 3% and the degradation in sentence accuracy is about 17–19%.

In Table 1, M1 model is slightly better than M4 model; in Table 2, M4 is slightly better

than M1. However, the difference in word accuracy is not more than 0.1% and the sentence accuracy differs by less than 1.1%. So it is hardly distinguishable. The same is true when we compare the corresponding rows in Table 3 and Table 4 where adaptive learning is applied. A larger difference is observed only when the tag transition probabilities ($P(t_k \mid t_{k-1})$) is jointly considered for segmentation as shown in Table 5. In general, the M4 model is slightly better than M1. Yet, both models are better with respect to the maximum match heuristics.

Adaptive Learning

Table 3 and Table 4 show the performance after the robust adaptive learning algorithm is applied to the baseline models. Since the maximum match algorithms use a deterministic process, they do not have the capability of learning. Hence, there is no corresponding entry in the tables.

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
P(Lk|Lk-1)	4.17 (95.83)	21.33 (78.67)	4.37 (95.63)	21.33 (78.67)
P(m|n)	4.33 (95.67)	22.18 (77.82)	4.43 (95.57)	21.47 (78.53)
P(Wk)	3.28 (96.72)	18.79 (81.21)	3.84 (96.16)	20.26 (79.74)
P(Wk|Lk-1)	3.23 (96.77)	18.28 (81.72)	4.00 (96.00)	21.04 (78.96)

Table 3 Performance WITH Unknown Words after LEARNING

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
P(Lk|Lk-1)	1.20 (98.80)	4.65 (95.35)	1.19 (98.81)	4.14 (95.86)
P(m|n)	1.26 (98.74)	4.99 (95.01)	1.23 (98.77)	4.21 (95.79)
P(Wk)	0.38 (99.62)	1.60 (98.40)	0.68 (99.32)	2.50 (97.50)
P(Wk|Lk-1)	0.11 (99.89)	0.48 (99.52)	0.61 (99.39)	2.35 (97.65)

Table 4 Performance WITHOUT Unknown Words after LEARNING

When comparing Table 3 and Table 4 with Table 1 and Table 2 respectively, some facts are observed. First the simple models M2 and M3 are greatly improved both in word accuracy and sentence accuracy by adaptive learning. The improved performance is comparable with the other models which use word information. The improvement for M1 and M4 models are

less obvious because the baseline performance is already very high before learning. In fact, one instance in Table 3 shows a little degradation in sentence accuracy due to over-tuning of the parameters. However, substantial error rate reduction can be observed in the other cases.

The above results confirm the underlying principle of adaptive learning that finding the correct ranks among the estimated scores, rather than finding a better estimate of the scores, plays an important role in statistical word segmentation (and virtually in all such statistical frameworks.) This may also imply that the initial baseline model might not be as important as the learning process, although it is important to have a good initial guess. Indeed, the criterion of the initial baseline models is to minimize the risk of misjudgement by maximizing the estimated probability measure. On the other hand, the robust adaptive learning algorithm try to find a direct mapping between the scores and the ranks of the candidates and try to overcome statistical variations between the training and testing sentences by minimizing the system error rate directly. Therefore, as observed in the tables, it is more robust for unseen text after learning.

Segmentation with Lexical Tags

Table 5 shows the performance when lexical tags (i.e., parts of speech) are used in word segmentation. These rows correspond to the models M5, M6, M7 in Eqn. (2.2.9). In comparison with Table 2, the baseline performance of model M5 ($P(t_k \mid t_{k-1})$), which uses lexical tags for segmentation, does not show more promising performance than M1 or M4, although its word accuracy can achieve as high as 97%. The model M1 ($P(w_k)$), when jointly considered with the lexical tag transition probability ($P(w_k) \times P(t_k \mid t_{k-1})$), is in fact degraded slightly. The baseline performance of M6 ($P(w_k \mid l_{k-1}) \times P(t_k \mid t_{k-1})$) is only slightly better than that of M4, where the tag transition probability is not used. The surprising results might be due to the very free linear order of the Chinese language.

Nevertheless, the overall performance of model M6 is the best among all when robust adaptive learning is applied. Word accuracy in this operation mode can achieve as high as 99.91% for the training set and 99.39% for the testing set. The sentence accuracy is 99.55% and 97.65% for the training set and the testing set, respectively. Since this model is to optimize the segmentation pattern and the tag sequence, it is useful for automatic tagging of plain Chinese text.

If adaptive learning is not applied to M6, its performance becomes slightly less satisfactory. Under this condition, the M4 model with adaptive learning has the best performance

among all interesting models. Since the same corpora for the M1 model could be used to acquire the required parameters $P(w_k \mid l_{k-1})$, the performance is achieved without extra cost beyond what is required for the context-independent word model (M1). Therefore, a good model along with robust adaptive learning could result in a cost-effective segmentation model without using extra resources.

Model	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
P(Tk|Tk-1)	2.52 (97.48)	14.39 (85.61)	2.65 (97.35)	14.19 (85.81)
after learning =>	0.82 (99.18)	3.14 (96.86)	0.92 (99.08)	3.21 (96.79)
P(W)*P(Tk|Tk-1)	0.66 (99.34)	2.89 (97.11)	0.89 (99.11)	3.57 (96.43)
P(W|L)*P(Tk|Tk-1)	0.47 (99.53)	1.77 (98.23)	0.71 (99.29)	2.43 (97.57)
after learning =>	0.09 (99.91)	0.45 (99.55)	0.61 (99.39)	2.35 (97.65)
P(W|T)*P(Tk|Tk-1)	1.47 (98.53)	6.79 (93.21)	1.50 (98.50)	6.04 (93.94)

Table 5 Baseline Performance WITHOUT Unknown Words but WITH Lexical Tag Information

Lexical Tags *vs.* Learning

In contrast to adaptive learning, using lexical tags does not seem to help much in word segmentation. This can be verified by comparing the baseline performance of the $P(w_k) \times P(t_k \mid t_{k-1})$ and $P(w_k \mid l_{k-1}) \times P(t_k \mid t_{k-1})$ models in Table 5 with the performance of $P(w_k)$ and $P(w_k \mid l_{k-1})$ models in Table 4; the small amount of degradation might imply that adaptive learning is more effective in improving the baseline models than using the lexical tag information (unless adaptive learning is also applied.)

Unknown Word Problem

As described previously, the error rate introduced by unknown words is significant. Many models in the literature are based on the assumption that all words in the text could be found in the system dictionary. It is evident, however, that such an assumption is unrealistic from the experiment results. This may imply that more research energy should be directed toward *unknown word resolution* rather than the development of alternative baseline models. Table 6 shows the performance for unknown word resolution with the model proposed in the previous section; the underlying model is a revised version of the M4 model.

	Training Set Error (Accuracy)		Testing Set Error (Accuracy)	
	word (%)	sentence (%)	word (%)	sentence (%)
before learning	38.06 (61.94)	85.04 (14.96)	39.64 (60.36)	86.38 (13.62)
after learning	1.78 (98.22)	8.35 (91.65)	3.59 (96.41)	15.26 (84.74)

Table 6 Performance for Unknown Word Resolution (Baseline and Learning for 10 iterations)

It is interesting to note that the performance of the *baseline* model is very low. This is probably a generic phenomena for all kinds of error correction problems; because the segmentation patterns are extended according to the error types, the candidate patterns are no more confined to the patterns that could be generated with dictionary lookup. Hence, the number of possible segmentation patterns increases drastically, and the performance of the baseline model tends to degrade. Another factor that accounts for the degradation in the baseline performance is the estimation error of the model parameters. Because all unknown words are regarded as a special class of words with the same statistical behavior, the estimated probabilities, such as the $P(w_u \mid l_{k-1})$ term, may not indicate the specific distribution of a specific unknown word under consideration. To resolve this problem, adaptive learning is essential. The learning results in the table show how unknown word errors can be recovered after adaptive learning is applied.

In comparison with the best baseline performance in Table 1 and the best learning results in Table 3, where unknown words are not handled, the error rates are reduced by 45–51% for words and 54–58% for sentences in the training set; in the testing set, the reduction in error rates amounts to 7–9% for words and 24–28% for sentences.

Of course, we also noted that some isolated single-character words are merged by mistake with this simplified error correction model. This may imply that the current features for detecting the unknown word region and the existence of the unknown words are not effective enough for detecting some instances of unknown word errors. If better features other than the sentence length, word count, and character count could be used, the improvement might be even more encouraging.

Cost Concern

The costs of the various models are directly related to the corpus size and the number of parameters to be estimated. Table 7 shows the testing environment, including the numbers of parameters for all models. Among the various models, model M2 and M3 have the smallest number of parameters. As shown in the above experiments, many models proposed here do not have significantly different performance in terms of accuracy on segmentation. The costs of the models are thus important in some applications. This seems to suggest that we could start with a simple baseline model and use an adaptive learning algorithm to acquire low cost yet high performance in word segmentation. It also suggests that we could use the less expensive models, for example, to bootstrap an automatic dictionary construction process from very limited available corpus resources.

Model	Number of Parameters	Model	Number of Parameters
P(Lk|Lk-1)	40	P(Tk|Tk-1)	625
P(m|n)	229	P(W)*P(Tk|Tk-1)	9,755+625
P(Wk)	9,755	P(W|L)*P(Tk|Tk-1)	14,473+625
P(Wk|Lk-1)	14,473	P(W|T)*P(Tk|Tk-1)	10,231+625
Training Set	41599 words / 5608 sentences		
Testing Set	10134 words / 1402 sentences		
Dictionary	99441 entries		
Lexical Tags	22 parts of speech & 3 special tags		
Ambiguity	8.6 candidates/sentences (both training set & testing set)		

Table 7 Testing Environment

6. Conclusion

In this paper, we have proposed a generalized word segmentation model for the Chinese word segmentation problem. We have shown how to use the various available information to resolve the segmentation problem based on the generalized model. It is shown that word segmentation can be resolved easily and inexpensively with the proposed statistical models. Word accuracy as high as 96% and sentence accuracy up to 80% can be achieved in the baseline model when there are unknown words. When there are no unknown words, the performance is about 99% for words and 97% for sentence.

In addition to the baseline models, a robust adaptive learning algorithm is proposed to enhance the performance of the baseline models so that these models could perform well even in handling unseen text. It is noticed that a good adaptive learning algorithm is critical to facilitate word segmentation. The reason is that a good robust adaptive learning algorithm could provide a scoring mechanism that directly minimizes the error rates both in the training corpus and the testing set. Therefore, it provides better discrimination power in ranking the large number of possible segmentation patterns.

We also find that the unknown words contribute a significant portion of the error rate. To be practical in real applications, the unknown word problem should therefore be taken seriously. In this paper, we have proposed an error correction mechanism for resolving the special unknown word problem. With such a mechanism, the error rates are reduced by 45–51% for words and 54–58% for sentences in the training set; in the testing set, the reduction in error rates amounts to 7–9% for words and 24–28% for sentences.

Throughout the framework, we had tried to use extra information from the least expensive features already available in a segmented corpus. By using the extra features of character count, word count and word length information, it is shown to improve the system performance with respect to the other models that do not use them. The use of such inexpensive features also make possible some applications where the available resource is limited.

Acknowledgement

We would like to express our gratitude to Shu-Jun Ke (Behavior Design Corporation) for her efforts in preparing the segmented corpus. Her work provides useful training and testing materials for verifying the various proposed models. We also would like to express our thanks to the Free China Times for making the text corpus available to us. Special thanks are given to the Behavior Design Corporation for providing the Chinese-English Electronic Dictionary to this research project.

References

[1] Amari, S., "A Theory of Adaptive Pattern Classifiers," *IEEE Trans. on Electronic Computers*, vol. EC-16, no. 3, pp. 299–307, June 1967.

[2] Chang, Jyun-Sheng, C.-D. Chen and S.-D. Chen, "Chinese Word Segmentation through Constraint Satisfaction and Statistical Optimization," (in Chinese) *Proceedings*

of ROCLING-IV, ROC Computational Linguistics Conferences, pp. 147–165, Kenting, Taiwan, ROC, 1991.

[3] Chen, K.-J., C.-J. Chen and L.-J. Lee, "Analysis and Research in Chinese Sentence Segmentation and Construction," *Technical Report, TR-86–004,* Taipei: Academia Sinica, 1986.

[4] Chen, K.-J., Shing-Huan Liu, "Word Identification For Mandarin Chinese Sentences," *Proceedings of COLING-92*, 14th Int. Conference on Computational Linguistics, pp. 101–107, Nantes, France, July 23–28, 1992.

[5] Chiang, T.-H., Y.-C. Lin and K.-Y. Su, "Syntactic Ambiguity Resolution Using A Discrimination and Robustness Oriented Adaptive Learning Algorithm," *Proceedings of COLING-92*, 14th Int. Conference on Computational Linguistics, pp. 352–358, Nantes, France, July 23–28, 1992.

[6] Fan, C.-K. and W.-H. Tsai, "Automatic Word Identification in Chinese Sentences by the Relaxation Technique," *Computer Processing of Chinese and Oriental Languages*, vol. 4, no. 1, pp. 33–56, 1988.

[7] Ho, W.-H., "Automatic Recognition of Chinese Words," master thesis, National Taiwan Institute of Technology, Taipei, Taiwan, 1983.

[8] Sproat, R. and C. Shin, "A Statistical Method for Finding Word Boundaries in Chinese Text," *Computer Processing of Chinese and Oriental Languages*, vol. 4, no. 4, pp. 336–351, 1991.

[9] Su, K.-Y. and C.-H. Lee, "Robustness and Discrimination Oriented Speech Recognition Using Weighted HMM and Subspace Projection Approach," *Proceedings of IEEE ICASSP-91*, vol. 1, pp. 541-544, Toronto, Ontario, Canada. May 14-17, 1991.

[10] Yeh, C.-L. and H.-J. Lee, "Rule-Based Word Identification for Mandarin Chinese Sentences — A Unification Approach," *Computer Processing of Chinese and Oriental Languages*, vol. 5, no. 2, pp. 97–118, March 1991.

A Modular and Statistical Approach to Machine Translation

Dah-Yih Wang & Jyun-Sheng Chang

Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan

Abstract

In this paper, we report our experiment on a modular statistical approach to machine translation system. The experimental MT system consists of modules implemented by statistical methods to handle different level of linguistic analysis. The overall architecture of the system resembles that of a transfer-based MT system, but with less explicit expert knowledge involved. Five hundred simple bilingual sentences with main verbs restricted to 30 commonly used verbs are used as training data. These sentences are syntactically and semantically tagged to provide statistical data for case role analysis and transfer. A bilingual dictionary and collocation data from a corpus of Chinese news are used in target generation. The system is tested against the original 500 sentences and additional 100 sentences with promising results.

1. Introduction

Changes in the philosophy of language and mind heavily influence the MT researchers in using different approaches. In the 1970s and 1980s, rule-based systems are philosophically based on Norm Chomsky's *deterministic rationalism*, which means, the meaning of a sentence is inferred by a successively modification of internal model. As a result, the translation process amounts to the mechanical determination by fixed rules. However, Chomskyan paradigm is by now widely rejected [Sampson 83].

Another view being widespreadly accepted is *fallible rationalism*, which means, the mind responds to experiential inputs not by a deterministic algorithm (rule), but by creatively formulating fallible hypothesis. On this view, it suggests MT researchers ought to exploit any techniques that offer the possibility of better approximation to acceptable translation.

This changing trend was reflected by the growing popularity of statistical-oriented approaches in computational linguistics community. For MT, rule-based approaches need complete understanding of the characteristics of the source and target language; on the contrary, statistical-oriented approaches uses little linguistics analysis and treats translation problem purely as a process of optimization of possibility. Both approaches have its own benefits and drawbacks. Generally speaking, they can compensate for each other. Hence, to seek a balance point between these two different approaches seems a feasible way to go.

1.1 Machine Translation Model

The models of MT range from rule-based to corpus-based. Others that lie between are example-based and hybrid systems. For simplicity, we only discuss the rule-based and corpus-based models here.

1.1.1 Rule-based Machine Translation

Rule-based machine translation model may be roughly classified as *transfer* and *interlingua* approach. "The interlingua approach is now largely disfavored in most practical systems. The distinction among *direct translation*, *transfer-based* and *interlingua* system is fairly captured by the well-known pyramid diagram in Fig-1 that is probably first found in [Vauquois 73]. This diagram shows the deeper the analysis of the source language (SL), the less complex is the mapping from source language to target language (TL) [Somers 87]". But how deep should the analysis be remains an open issue. Undoubtedly, proper analysis greatly reduced the complexity of the problem.

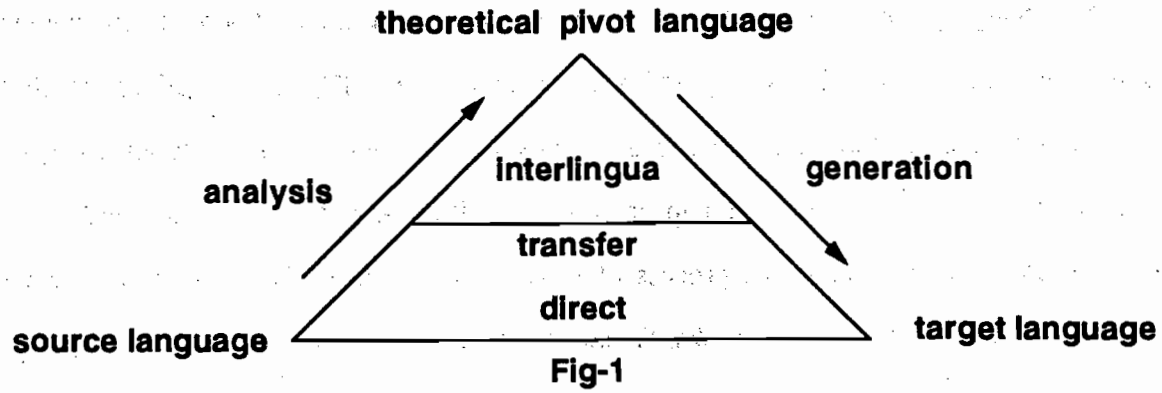

Fig-1

In most transfer-based MT systems, SL text is syntactically analyzed, then transformed into some intermediate representation (e.g., case role in case grammar), and finally TL text is generated. In summary, the whole process can be realized in three phases: analysis, transfer, and synthesis.

1.1.2 Corpus-based Machine Translation

[Brown 90] first proposed a new MT model, consisting of *translation model* (TM) and *language model* (LM). The former describes the local correspondences between the two words in two different language while the latter shows the linear relations among the words within the same language. More precisely, given a sentence in SL, the translation problem reduces to: (1) find the word-by-word correspondences of the input in the TL and (2) among the corresponding words in (1), find the most likely translation of the input w.r.t the TM and the most plausible target sentence w.r.t the LM.

1.2 Recent Statistical Computaional Linguistics Researches

The researchers on machine translation have paid much attentions to corpus-based approach for the past few years. This trend is due to the fact that machine translation involves in both complex and tremendous knowledge acquisitions. The rule-based

approach suffers from the disadvantages of time-consuming knowledge engineering and difficulty in maintaining data consistency.

Lately, much research effort in statistical approach has been devoted to fundamental works in computational linguistics. The following successful results encourages MT researchers to reconsider the MT problem from quite a different point of view.

- **Tagging part of speech**

Several studies attack the problem by optimizing the product of the probabilities of relative tag probability (RTP) and tag bi-gram, achieved a correctness of 95% [Derose 88, Church 89]. Also, a corpus-based segmentation of Chinese text reported a 90-95% accuracy [Chang 91].

- **Grouping non-recursive noun phrase**

Using the bi-gram probabilities of starting a noun phrase and ending a noun phrase, non-recursive noun phrases for unrestricted text can be grouped with a 95-99% accuracy [Church 88].

- **Finding clauses**

Similar technique also applies to finding clauses in unrestricted text with a mere 6.5% error rate [Ejerhed 88].

In addtion, some researchers also use statistical models to disambiguate word sense [Brown 91] and [Dagan 91], and to tag sentences for thematic relation learning. Nevertheless, not all the statistics-oriented natural language processings are satisfactory. With the progress in these fundamental problems, the framework of a modular and statistical MT system apparently based on sound ground.

1.3 Our Model

Traditional rule-based systems deal with different linguistics problems in several modules because MT problem involves many huge and minute knowledge sources on different linguistic levels (morphology, lexicon, syntax, semantic, etc.). In a statistical MT system, in order to isolate the effects of irrelevant parameters, the work of analysis, transfer and synthesis should be accomplished within different modules.

Our major concern for this study is how to take advantage of the statistical power in dealing with uncertain or inconsistent data in corpus-based system, and the generalization power as well as economic property of linguistics knowledge. Hence, we propose a statistics-oriented method that incorporates the linguistics knowledge as the backbone of information retrievals.

Our assumption is that if statistical approaches to group all kinds of phrase and embedded sentences (instead of parsing) can be fully developed in the near future, it would be worth paying more attentions to do analysis, transfer and synthesis not in so rigid ways as before. We thus, by the use of case grammar, attempt to construct statistical models, with less effort involved, to deal with *case role analysis*, *case role translation* (some kind of transfer) and *lexical choice*. These three modules together can form the kernel of a MT system. We hope that some inspiration from our experiment might help to sketch out the skeleton of a modular and statistical machine translation system in Fig-2.

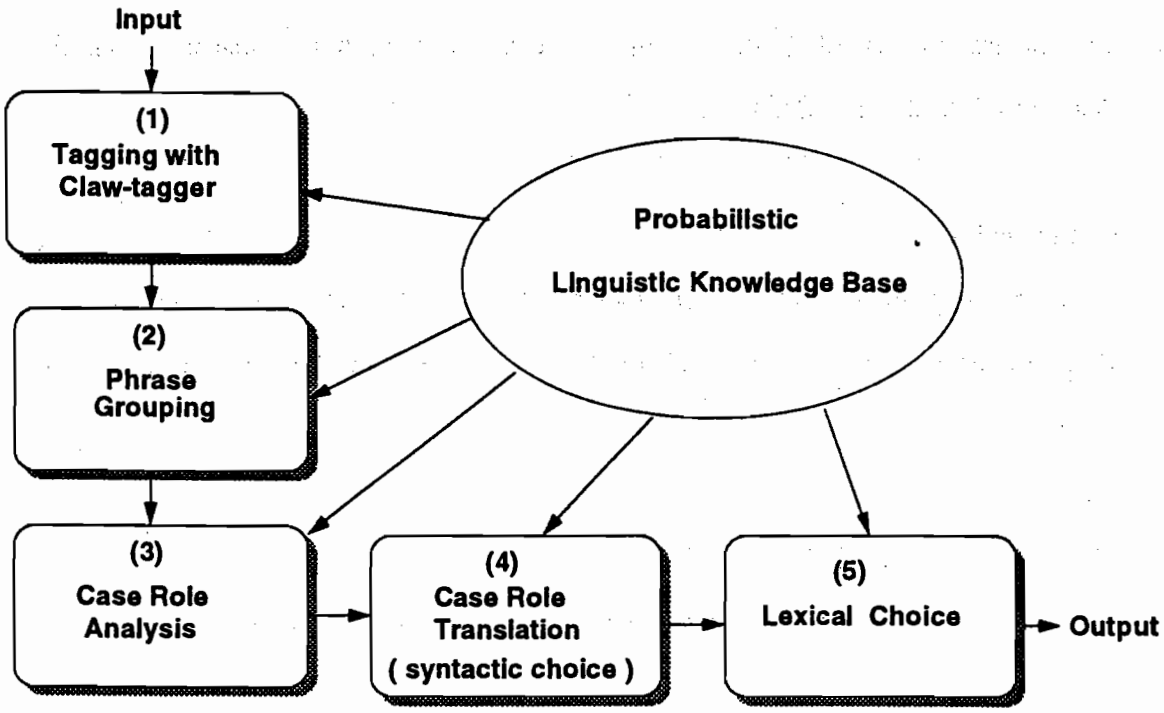

Fig-2

Our discussion includes (3), (4), and (5). (1) comes from the Claw tagging system. In (2), the statistical models for grouping non-recursive noun phrase comes from [K. D. Church 89]; the grouping of other kinds of phrase were implemented by some heuristics.

2. Case Role Analysis

Case grammar is widely adopted in MT researches because of its good property of capturing the deep structure of a proposition, and thus is suitable for analyzing source and generating the TL. For simplicity, in our experiment we only consider the easiest case, that is, simple sentence without any tense, aspect or mood.

2.1 Statistical Model for Case Role Analysis

A predicate may have many case frames; to tell one from the others may need a delicate mechanism to analyze the functional relationships among the constituents of a

structure. In order to avoiding such complex work, we attempt to construct a simpler statistical model to do the same things.

I. For inner roles:

We use the tri-gram information of inner roles and prepositions (case makers) for a specific predicate to substitute the need for the case frame. Take the case of *provide* for an example:

2-1. [Ag I] [V provide] [Th a book] to [Be him].
The tri-grams are: ("","",Ag), ("",Ag,V), (Ag,V,Th), (V,Th,to), and (Th,to,Be).

2-2. [Ag I] [V provide] [Be him] with [Th a book].
The tri-grams are: ("","",Ag), ("",Ag,V), (Ag,V,Be), (V,Be,with), and (Be,with,Th).

In addition to the tri-gram *contextual probabilities* (CP), we also need *relative case probabilities* (RCP). We define RCP to be the "relative probabilities of the tags the of a phrase head to assume a certain case role", i.e., $Pr(role|tag_{head})$. For example, a singular common noun NN[1] may act as a Theme with the probability of 0.6, as an Agent with probability 0.1, as an Experiencer with 0.03 probability, and as a Beneficiary with 0.02 probability. Then, the RCP of NN would be: Pr(Th|NN)=0.6, Pr(Ag|NN)=0.1, Pr(Ex|NN)=0.03, Pr(Be|NN)=0.04. Table-1 shows part of the RCP.

[1] All the tags used in the paper come from LOB tagset.

RCP	NN	NP[2]	PP1A[3]	PP3OS[4]	JJ[5]
Th	0.6	0.3	0.05	0.8	0.1
Ag	0.1	0.2	0.8	0	0.1
Ex	0.03	0.1	0.07	0.06	0.05
Be	0.04	0.1	0	0.08	0.05
Cp	0.02	0.01	0	0.01	0.45

Table-1 Relative Context Probability

In table-1, Pr(Ag|PP1A)=0.8 means that *I* tends to function as an Agent. Pr(Ag|PP30S)=0 means *them* never function as an Agent. Pr(Th|NN)>Pr(Ag|NN) means a common noun has a greater tendency to function as a Theme than as an Agent. We choose the tag of a phrase head because of two reasons: (1) Head is the most informative word in a phrase and (2) The n-grams can capture more information with unimportant words skipped.

The analysis process is to maximize the product of case role tri-grams for the predicate and RCP.

II. For outer roles:

Most outer roles can act as only one case role; this greatly reduces the ambiguity in analysis. Unfortunately, dealing with outer roles may be problematic in case role analysis because: (1) Outer roles occur with comparatively low frequency, simply training outer roles from corpus without special processing may suffer from the problem of undersampling. (2) The syntactic structures (surface structure) of outer roles are diverse,

[2] proper noun: *John, London*
[3] 1st singular nominatiove pronoun in subject position: *I*
[4] 3rd plural nominative pronoun in object position: *them*
[5] general adjective: *tall, good*

ranging from all kinds of phrase to subordinate clauses. Among them, some are analyzable; others are idiomatic.

Since there is no suitable statistical model at hand, we use mainly heuristics to deal with outer role analysis.

3. Case Role Translation

Transfer operations improve the quality of translation. Instead of examining the syntactic structures and idiosyncrasies of specific lexical items, we choose to do *case role translation* to facilitate the transfer process.

3.1 Why Transfer?

Even though the deep (semantic) structures are identical, there are surface (syntactic) structure differences between source and target language. See the following examples:

3-1. [Ag I] [V washed] [Th the car] [Pl in the garage] [Ti yesterday].
The translation "[Ag 我] [Ti 昨天] [Pl 在車庫] [V 洗] [Th 車子]" shows the syntactic differences (case role order) between Chinese and English.

3.2 Statistical Model for Case Role Translation

As before, we rely on both translation and language model to cope with case role translation. The major tasks of case role translation are as follows:

(1) Reorder the case roles.
(2) Translate the preposition of outer role into proper target words.
(3) Pick out some function words and put them in appropriate place.

For instance, the sentence "*I place the vase on the desk carefully*" has the case analysis:

3-1. [Ag I] [V place] [Th the vase] [Lo on the desk] [Ma carefully].

After the case translation, the result is "Ag Ma 地 把 Th V 在 Lo 上". These three tasks are realized separately as follows:

(1) (Ag V Th Lo Ma) is reordered to (Ag Ma Th V Lo).
(2) *on* is translated into 在...上.
(3) 地 and 把 are inserted in the proper positions.

I. Translation model:

The translation model provides the probabilities of correspondences between source and targets case roles with/without a case markers. See table-2.

with a stick	用 棍子	**wtih** Im	用 Im
run	地 跑	V	地 V
fast	得 快	Ma	得 Ma
during last year	在 去年 期間	**during** Du	在Du 期間
to the school	到 學校	to Lgo	到 Lgo
with courage	勇敢 地	with Ma	Ma 地
company	把 公司	Th	把 Th

Table-2

II. Language model:

It's not trivial to determine whether and where to insert the source-independent function words such as 得, 地, and 把 in the target sentences, because the inclusion of these words depends on the ordering of target case role. Consider the following examples:

3-2. [Ag John][V runs] [Ma fast].

[Ag 約翰][V 跑]得[Ma 快].

*[Ag 約翰][V 跑][Ma 快].

3-3. [Ag John][V runs][Lgo to the school][Ma quickly].

[Ag 約翰][Ma 很快]地[V 跑][Lgo 到學校].

*[Ag 約翰][V 跑][Lgo 到學校]得[Ma很快].

3-4. [Ag John][V runs][Th the company][Ma very successfully].

[Ag 約翰]把[Th 公司][V 經營]得[Ma 非常成功].

*[Ag 約翰][Ma 非常成功]地[V 經營][Th 公司].

From the observations above, the language model should insure proper target role ordering and the insertion of function words consistent with the ordering of the target roles.

Our language model encodes the possibilities of the mutual ordering among case roles, which are possibly merged with function words, in the form of tri-gram. The tri-grams of the language model in example 3-2 above would be

("","",Ag), ("",Ag,V), and (Ag,V,得Ma).

Similarly, example 3-3 has tri-grams as

("","",Ag), ("",Ag,Ma), (Ag,Ma,地V), and (Ma,地V,到Lgo).

The process of case role translation is simply to optimize the product of these two models.

4. Lexical Choice

4.1 Statistical Model for Lexical Choice

Different senses of a word in a context result in different target words are significant. To choose proper lexical items, we employ *global scope* and *local scope* to differentiate word sense implicitly. "Global scope" means the sense of a word is determined by other words in different structures. On the contrary, "local scope" means the sense of a word is determined by its neighbors within the same structure (the words to the left and/or right). In the following, we will describe the proper translations of a verb and another informative word (*informant*) from global scope. Other words are translated with the local scope.

I. Global scope:

We assumed that, in a sentence, the meaning of a verb is related to one of its argument. More precisely, we presume the most probable informative argument to be the head word of an inner role. For examples, in

(run, machine), (river, run), (take, bus), (take, job), (break, bank), and (window, break),

the translation of run, take, and break is determined by its Theme. How to select the informant is not trivial, we thus make the decision by a heuristic. The inner role is selected by the precedence "Cp > Th > Lo > Ag".

II. Local scope:

With the belief that words within a grammatical unit are strongly correlated, we deal with other words on the base of phrase, i.e., from a local scope. From observations, we know that heads and their modifiers have greater tendency to co-occur. Consequently, sampling the collocation information from corpora would be feasible.

To demonstrate how GSP and LSP work in lexical choice, consider following examples:

4-1. [Ag They] [V develop] [Th all the natural resources].
The proper translation can be "他們 開發 所有的天然資源"
The GSP is Pr(Verb|develop)*Pr(Informant|resource)*Pr(Verb,Informant).
The LSP is $Pr(T_{31}|all)*Pr(T_{32}|the)*Pr(T_{33}|natural)*Pr(T_{34}|resource)*X$ where
Collocation probability $X = Pr(T_{31},T_{32})*Pr(T_{32},T_{33})*Pr(T_{33},T_{34})$.

To get a feel of the difficulty involved in word selection, take a look at the possible translations of words listed in a dictionary:

develop: 引起 宏揚 沖洗 長 建設 振興 培養 產生 發育 發揮 **開發** 開闢 增進
natural: **天然** 平常 天生
resource: **資源** 安慰 消遣 機智

If we can extract sufficient collocation information from corpora, it is likely to encounter the co-occurrences of (開發,資源) in "開發台灣西部外海石油資源" and (天然,資源) in "天然資源並非取之不竭". Especially, to suit the need for a limited domain amounts to train the parameters from that domain rather than to build semantic hierarchy (network) by some domain-dependent features.

Technology of acquiring collocation information is beginning to mature and the burden of human knowledge acquisition will be alleviated at least partly [Smadja 90]. For this experiment, we use collocation probability to handle GSP. As for LSP, we use only the stand-alone probability of each word. The best translation of words is determined by the product of GSP and LSP.

5. Experimental Results

5.1 Training Data

To avoid additional work irrelevant to our discussion, our training data include only simple sentences with present aspect, active form, and non-recursive phrases. Five hundred bilingual (English-Chinese) sentences, with 30 commonly used verbs as the main verb, were adopted from two dictionaries[6]. The English sentences were syntactically tagged by Claw-tagger, and both English and Chinese sentences were semantically tagged (case role) by hand. After tagging, we grouped the phrases of the sentences then fed them to the system. These 30 verbs are averagely selected from 15 verb classes which are classified by Cook's *matrix model* in *Case Grammar Theory* [Cook 70], thus have a representative coverage in case role analysis. The tag set is from the LOB tag set, and case role set mainly borrows from [Tang 1975].

---

[6] These two dictionaries are *Longman Englis-Chinese Dictionary of Contemporary English*, Longman Group (Far East) Ltd. 1988 and 英語常用動詞用法詞典 (A Dictionary of Commonly Used English Verb), 商務印書館, 上海譯文出版社, 1986.

In addition to these 500 sentences, about 8,000 subject-verb (SV) or verb-object (VO) type of Chinese phrase head bi-grams are extracted from two sources[7] to facilitate the lexical choice of verb and its informant.

The translation of single words comes from *BDC Chinese-English Dictionary* version 1.0 (致遠科技公司).

5.2 Evaluation Criteria

Due to the lack of programs for extracting collocation information and the shortage of bilingual corpus, our models severely suffer from the problem of undersampling. Therefore, to evaluate the performances of the models needs special consideration.

Ⅰ. For case role analysis:

Our criterion for judging case role analysis is rather simple. Namely, if the any case is assigned to a phrase incorrectly, we regard the whole sentence as a wrong analysis.

Ⅱ. For case role translation:

If the source case roles assigned to a sentence is reordered to target case role incorrectly, or any case markers is improperly inserted, omitted, or placed, we regard the case role translation as a failure.

Ⅲ. For lexical choice:

Since our simplified model for lexical choice in local scope model (LSM) hasn't incorporated the collocation probability yet, our evaluation criterion for lexical choice is restricted to the suitability of a verb-informant pair.

[7] (a) 30,000 Chinese words from general domains. (b)1000,000 Chinese words of reportage from Union Press (聯合報).

5.3 Two Tests

We did two tests to evaluate the system performance according to the criteria defined above. In the first test, we test the system with the same training sentences to see its capability of learning. Secondly, we randomly selected 100 sentences from Brown Corpus of category A,B,C[8] under two constraints: (1) the usage of a verb cannot be a phrasal verb and (2) the inner and outer roles are within our recognition. The overall result shows a satisfactory capability of learning on the whole, as some of the testing sentences reveal[9]:

5-1. [The /ATI delegation /NN] (arrives /VBZ) in /IN [Beijing /NP] on /IN [Wednesday /NR]
AG V IN,LGO TI
代表團 訪問代表團 :2
出生 來臨 到 到了 到達 抵達 進站 開到 駕臨 :9
代表團 星期三 到達 北京

5-2. [John /NP] (breaks /VBZ) [the /ATI windows /NNS] with /IN [a /AT stone /NN]
AG V TH WITH,IM
中止 打破 折斷 沖破 刷新 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯 違背 摧毀 暴跌 潰決 壓破 斷 斷裂 鎮壓 離開 鑿開 :24
窗 窗子 窗戶 牖 :4
約翰 用石 把窗子 打破

5-3. (Break /VB) [the /ATI news /NN] to /IN [him /PP3O] { gently /RB }
V TH BE MA
消息 新聞 :2
中止 打破 折斷 沖破 刷新 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯 違背 摧毀 暴跌 潰決 壓破 斷 斷裂 鎮壓 離開 鑿開 :24
把消息 婉轉 地透露 給他

5-4. [They /PP3AS] (count /VB) [him /PP3O] among /IN [their /PP$ supporters /NNS]
AG V TH CP
支數 伯爵 告發條項 依賴 計算 計數 當作 算 認為 數 數目 點 :12
支持者 :1
他們 當作 他 是他們的支持者

5-5. [The /ATI train /NN] (moves /VBZ) { slowly /RB } along /IN [the /ATI river /NN side /NN]
TH V MA ALON,PA
火車 系列 訓練 鍛練 :4

---

[8] Category A: 定期刊物 報導文學 (reportage), Category B: 定期刊物 社論 (editorial), Category C: 書評 (reviews).

[9] NP is grouped by "[]", VP by "()", ADVP by "{}", and ADJP by "<>". Line 1: input. Line 2: After analysis. Line 3,4: senses of informant and verb. Line 5: output.

心動 打動 有所感觸 改變 步 步驟 走 招數 建議 看法改變 挪動 動 動彈 移 移動 移線
進行 進展 感動 搬 搬走 搬移 搬遷 調動 撼動 轉 轉移 :27
火車 延著河水邊 慢慢 地移動

5-6. [Last /AP year /NN] [we /PP1AS] (open /VB) [training /VBG classes /NNS] for /IN [the /ATI school /NN teachers /NNS]
TI AG V TH FOR,OBE
公開 打開 全天服務 伸開 拆封 空曠 展開 張開 爽朗 開 開始 開放 開啟 開著 開幕 開學
開闊 睜開 營業中 翻開 露封 :21
屆 班 級 級別 期 等 階級 種類 :8
去年 我們 為學校老師 開 訓練班

The result of second test is slightly less satisfactory than that of first test since our examples suffer from undersampling in case role analysis and case role translation. Although many case frames are within our recognition, yet the case role orders of testing sentences are different from that of training sentence. As for the overall performances of these two tests, see table-3.

Error Rate	Analysis	Case Translation[10]	Choice
Test1	5/500=1%	15/495=3%	1/495=0.2%
Test2	17/100=17%	6/83=7.2%	20/83=24.0%

Table-3

For more detailed examples, refer to appendix A.

6. Conclusions

6.1 Summary

We propose the MT model with statistical analysis, and modularity because of the following reasons: (1) encouraging results from recent statistical computational linguistics researches show the potentials in statistical MT, (2) the progress in automatic

[10] If the case analysis fails, then we did not do case translation.

6.2 Future Work

6.2.1 Extend to More Complex Syntactic Structures

Case roles can be assigned to not only phrases, but also to other structures (e.g., subordinate sentence and infinitive). Moreover, case relation can function at levels other than verb-phrase, such as *Characteristic/Composition* in "a book *of* poems" and *Partitive* in "the chairman *of* the board". That is, prepositions can also assign case roles to phrases.

To extend to complex syntactic structures, we might have to subdivide to case role set according to their different syntactic structures. For instance, although both an NP and an infinitive can function as a Th, we may assign them Th1 and Th2, respectively. However, this inevitably enlarges the size of n-gram matrix and consequently increases the cost of knowledge acquisition.

6.2.2 Substitute Semantic Tag for Case Role

During the development of the system, the case role assignment and the coverage of case role set is unclear. This may be a bottleneck in the long run. A more specific pivot language (such as semantic tag) may be an alternative to tag a structure semantically and automatically. Yet, the study of semantic tag still has a long way to go.

References

[Hunag 88] 黃金仁, 王良志, 格框機器翻譯的應用, 第四屆中華民國計算語言學討論會論文集, 99-125 頁

[Chang 91] 張俊盛, 陳志達, 陳舜德, 限制式滿足與機率最佳化的中文斷詞方法, 第四屆中華民國計算語言學討論會論文集, 墾丁, 147-166 頁, 1991.

[Derose 88] Steven J. Derose. Grammatical Category Disambiguation by Statistical Optimization, *Computational Linguistics*, 14(1):31-39, Winter 1988.

[Sampson 83] Geoffrey Sampson. Fallible Rationalism and Machine Translation. In Proceedings of First Conference of the European Chapter of the Association for Computational Linguistics, pages 86-89, Italy, 1983.

[Church 88] Kenneth Ward Church. A statistical Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of Second Conference on Applied Natural Language Processing*, pages 136-143, Austin, 1988.

[Brown 91] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics* pages 79-85 Volume 16, Number 2, June 1990.

[Brown 91] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. Word-sense Disambiguation Using Statistical Methods, pages 246-270, In *Proceedings of the annual Meeting of the Association for Computational Linguistics*, 1991.

[Brown,91] Peter F. Brown, Jennifer J. Lai, and Robert L. Mercer. Aligning Sentence in Parallel Corpora, In *Procedings of the Annual Meeting of the Association for Computational Linguistics*, pages 169-176, 1991.

[Dagan 91] Ido Dagan, Alon Itai, and Ulrike Schwall. Two Languages Are More Informative Than One. *29th ACL.*

[Somers 87], H.L. Somers 1987, *Valency adn Case in Computational Linguistics*, Edinburgh University Press. pages 262-278.

[Tang 75] Ting-Chi Charles Tang, *A Case Grammar Classfication of Chinese Verb*, Hai-Guo Book Company, Tapei, Taiwan. pages 26-43.

[Smadja 90] Frank A. Smadja and Kathleen R. MaKeown. Automatically Extraction and Representing Collocations for Language Generation, In *Proceedings of the Annual Meeting of The Association for Computational Linguistics*, pages 252-259, 1990.

[Su 91] Keh-Yih Su. An introduction to Corpus Based Statistical Oriented Techniques of Natural Languages Processing - A Tutorial, presented in *the 4-th ROC Computational Linguistics Conferenece* (ROCLING IV), Kenting, 1991.

[Pieraccini 90] Roberto Pieraccini, Esther Levin, and Chin-Hui Lee. Stochastic Representation of Conceptual Structure in the ATIS work, In *Proceedings of the 3rd DARPA Speech and Natural Language Workshop*, 1990.

[Tsutsumi 91] Taijiro Tsutsumi, Word-Sense Disambiguation By Examples. ICCCL 1991, pages 440-446.

[Slocum 85] Jonathan Slocum, A Survey of Machine Translation: Its History, Current Status, and Future Prospects. *Computational Linguistics* pages 1-17, Volume 11, Number 1, January-March 1985.

[Isabelle 85] Pierre Isabelle and Laurent Bourbeau , TAUM-AVIATION: Its Technical Features and Some Experimental Results, *Computational Linguistics* pages 19-27, Volume 11, Number 1, January-March 1985.

[Garside 87] Roger Garside, Geoffrey Leech, and Geoffrey Sampson, *The Computaional Analysis of English*, Longman Group UK Limited 1987.

[Ejerhed 88] Eva I. Ejerhed, Finding Clauses in Unrestricted Text by Finitary and Stochastic Method, In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 219-227, Austin, 1988.

[Cook 89] Walter A. Cook, *Case Grammar Theory* , Georgetwon University Press,1989.

Appendix A

(He /PP3A)(accepts /VBZ)(this /DT little /JJ gift /NN) under /IN(such /JJ condition /NN)
AG V TH UNDB,CON
同意 收 受 受理 服 容納 接受 接納 :8
禮物 禮品 :2
接受 禮物
在這樣的條件之下 他 接受 這少許禮物

(Finally /RB)(they /PP3AS)(accept /VB)(our /PPS terms /NNS)
TI AG V TH
同意 收 受 受理 服 容納 接受 接納 :8
專用術語 專門術語 條件 :3
接受 條件
最後 他們 接受 我們的條件

(The /ATI river /NN)(breaks /VBZ)(its /PPS banks /NNS)
AG V TH
中止 打破 折斷 沖破 割斷 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯
違背 摧毀 暴跌 潰決 擊破 斷 斷裂 鎮壓 離開 劈開 :24
坍塌 堤岸 銀行 :3
沖破 堤岸
河水 沖破 它的堤岸

(They /PP3AS)(break /VB)(the /ATI hard /JJ frozen /JJ earth /NN) with /IN(picks /NNS)
AG V TH WITH,IN
中止 打破 折斷 沖破 割斷 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯
違背 摧毀 暴跌 潰決 擊破 斷 斷裂 鎮壓 離開 劈開 :24
土 地球 :2
劈開 土
他們 用去除 把這個堅硬的冷凍土 劈開

(We /PP3AS)(break /VB)(the /ATI enemy's /NNS blockade /NN)
AG V TH
中止 打破 折斷 沖破 割斷 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯
違背 摧毀 暴跌 潰決 擊破 斷 斷裂 鎮壓 離開 劈開 :24
封鎖 :1
粉碎 封鎖
我們 粉碎 敵人的封鎖

(He /PP3A)(breaks /VBZ)(two /CD national /JJ records /NNS)(that /DT evening /NN)
AG V TH TI
中止 打破 折斷 沖破 割斷 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯
違背 摧毀 暴跌 潰決 擊破 斷 斷裂 鎮壓 離開 劈開 :24
紀錄 唱片 唱碟 載 錄下來 :5
打破 紀錄
他 那個晚上 打破 兩全國紀錄

(The /ATI warlord /NN government /NN)(breaks /VBZ)(the /ATI demonstration /NN)
AG V TH
中止 打破 折斷 沖破 割斷 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯
違背 摧毀 暴跌 潰決 擊破 斷 斷裂 鎮壓 離開 劈開 :24
示威 示威遊行 示威進行 示範 :4
鎮壓 示威遊行
軍閥政府 鎮壓 示威遊行

(The /ATI mirror /NN)(breaks /VBZ) into /IN(pieces /NNS)
TH V INTO,OO
反映 反射 面 鏡子 :4
中止 打破 折斷 沖破 割斷 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯
違背 摧毀 暴跌 潰決 擊破 斷 斷裂 鎮壓 離開 劈開 :24
鏡子 破
鏡子 破 成碎片

(Brittle /JJ things /NNS)(break /VB)(easily /RB)
TH V MA
事 事物 事情 底事 東西 物件 :6
中止 打破 折斷 沖破 割斷 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯
違背 摧毀 暴跌 潰決 擊破 斷 斷裂 鎮壓 離開 劈開 :24
東西 破
脆東西 破 得容易

(I /PP1A)(never /RB)(break /VB)(that /DT vow /NN)
AG PR V TH
中止 打破 折斷 沖破 割斷 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯
違背 摧毀 暴跌 潰決 擊破 斷 斷裂 鎮壓 離開 劈開 :24
誓言 :1
違背 誓言
我 從不 違背 那個誓言

(He /PP3A)(breaks /VBZ) under /IN (continuous /JJ questioning /VBG)
AG V UNDB,COM
他 我 :2
中止 打破 折斷 沖破 割斷 消失 破 破裂 粉碎 停止 崩潰 透露 違反 違犯
違背 摧毀 暴跌 潰決 擊破 斷 斷裂 鎮壓 離開 劈開 :24
他 崩潰
在不斷的盤問之下 他 崩潰

(We /PP1A)(change /VB)(the /ATI date /NN) to /IN(Feb /NP 28 /CD)
AG V TH TO,OO
兌換 找錢 找頭 改 改易 改變 更動 修改 掉 推移 換 替換 調換 變 變幻
變更 變易 變換 變遷 :19
日 日期 時日 棗子 號 :5
改 日期
我們 把日期 改 成二月二十八日

(I /PP1A)(change /VB)(my /PPS address /NN)(next /JJ week /NN)
AG V TH TI
兌換 找錢 找頭 改 改易 改變 更動 修改 掉 推移 換 替換 調換 變 變幻
變更 變易 變換 變遷 :19
地址 住址 致辭 發表聲明 :4
改 地址
我 下星期 改 我的地址

(I /PP1A)(change /VB)(places /NNS) with /IN(you /PP2)
AG V TH WITH,COAG
兌換 找錢 找頭 改 改易 改變 更動 修改 掉 推移 換 替換 調換 變 變幻
變更 變易 變換 變遷 :19
下 地 地方 安 安放 安置 位置 所在 放 放置 座位 處 :12
換 座位
我 和你 換 座位

(The /ATI box /NN)(contains /VBZ)(some /DTI drugs /NNS)
LO V TH
內含 有 含 含有 抑制 容 蘊含 :7
毒 毒品 [illegible] 藥品 :4
有 藥品
盒裡 有 一些藥品

(His /PPS country /NN)(develops /VBZ)(its /PPS traditional /JJ friendship /NN) with /IN(China /NP)
AG V TH WITH,COAG
引起 宏揚 沖洗 長 長成 建設 振興 培養 產生 發育 發展 發展起來 發揮
發揚 開發 開闢 增進 養成 曬 顯像 顯影 :21
友情 友誼 交情 情義 情誼 :5
發展 友誼
他的國家 和中國 發展 它的傳統友誼

(He /PP3A)(develops /VBZ)(a /AT good /JJ habit /NN)
AG V TH
引起 宏揚 沖洗 長 長成 建設 振興 培養 產生 發育 發展 發展起來 發揮
發揚 開發 開闢 增進 養成 曬 顯像 顯影 :21
習 習慣 習慣性 癮 :4
養成 習慣
他 養成 一個好習慣

(You /PP2)(develop /VB)(this /DT film /NN) for /IN(us /PP1OS)
AG V TH FOR,OBB
引起 宏揚 沖洗 長 長成 建設 振興 培養 產生 發育 發展 發展起來 發揮
發揚 開發 開闢 增進 養成 曬 顯像 顯影 :21
片子 底片 軟片 膠 膠片 膠卷 :6
沖洗 膠卷
你 為我們 沖洗 這膠卷

(A /AT large /JJ labouring /JJ class /NN)(develops /VBZ)(rapidly /RB) during /IN(past /JJ two /CD years /NNS)

TH V MA DUBI,DU
是 城 城 城别 期 等 階級 種類 :8
引起 宏揚 神洗 長 長成 建設 振興 培養 產生 發育 發展 發展起來 發揮
發揚 開發 開闢 增進 養成 游 願像 願影 :21
階級 發展
一個大工人階級 在過去兩年期間 發展 得迅速

(The /AT1 city /NN)(develops /VBZ) from /IN(a /AT fishing /JJ
village /NN)
TH V PROM,SO
市 城 城市 城區 都市 都會 :6
引起 宏揚 神洗 長 長成 建設 振興 培養 產生 發育 發展 發展起來 發揮
發揚 開發 開闢 增進 養成 游 願像 願影 :21
城市 發展起來
城市 是由一個的遊村 發展起來的

(I /PP1A)(drive /VB)(you /PP2)(home /NN)
AO V TH LOO
行駛 起步 打 喚醒開動 推動 推進 開 開車 開車送 幹勤 經 鍋氣 駕 駕車
駕駛 開動 :16
兄台 兄弟 你 你們 足下 您 :6
開車送 你
我 開車送 你 回家

At /IN(the /AT1 time /NN)(our /PP$ troops /NNS /)(drive /VB)
toward /IN(the /AT1 enemy /NN stronghold /NN)
TI AO V TOWA,LOO
軍隊 :1
行駛 起步 打 喚醒開動 推動 推進 開 開車 開車送 幹勤 經 鍋氣 駕 駕車
駕駛 開動 :16
軍隊 推進
進時 我們的軍隊 向敵人據點 推進

(The /AT1 reactionaries /NNS)(drive /VB)(all /ABN the /AT1
inhabitants /NNS) from /IN(the /AT1 island /NN)
AO V TH PROM,LOO
行駛 起步 打 喚醒開動 推動 推進 開 開車 開車送 幹勤 經 鍋氣 駕 駕車
駕駛 開動 :16
住戶 居民 :2
經 居民
反動頭 把所有的居民 從島上 趕走

(Diesel-engines /NNS)(drive /VB)(the /AT1 pumps /NNS)
AO V TH
行駛 起步 打 喚醒開動 推動 推進 開 開車 開車送 幹勤 經 鍋氣 駕 駕車
駕駛 開動 :16
水泵 抽動 唧筒 幫浦 :4
開動 水泵
開動 幫浦
柴油機 開動 幫浦

In /IN(1927 /CD)(the /AT1 Chinese /JNP revolution /NN)(enters
/VBZ)(a /AT new /JJ period /NN) in /IN(its /PP$ history /NN)
TI AO V TH IN,PL
入 上 吃頭 紀 參加 進 進入 進來 :8
段 時代 時期 發 期 期間 過期 :7
進入 時期
一九二七年 這個人民革命 在它的歷史 進入 一個新時期

(She /PP3A)(enters /VBZ)(all /ABN the /AT1 events /NNS) in
/IN(her /PP$ diary /NN)
AO V TH IN,LOO
入 上 吃頭 紀 參加 進 進入 進來 :8
事 事件 事端 :3
紀 事
她 把所有的事 紀 到她的日紀上

(We /PP1AS)(arrive /VB)(home /NR)(late /RB)
AO V LOO TI
他 百人 喜俏 我 我們 咱們 :6
出生 來臨 到 到7 到達 抵達 進站 開到 駕臨 :9
我們 到達
我們 很晚 到達 家裡

(Finally /RB)(our /PP$ holidays /NNS)(arrive /VB)
TI TH V
休假 例假 假日 節 :4
出生 來臨 到 到7 到達 抵達 進站 開到 駕臨 :9
假日 來臨

最後 我們的假日 來臨

(Her /PP$ baby /NN)(arrives /VBZ) during /IN(the /AT1 night /NN
)
TH V DURI,DU
嬰兒 :1
出生 來臨 到 到7 到達 抵達 進站 開到 駕臨 :9
嬰兒 出生
她的嬰兒 在夜晚期間 出生

(John /NP)(breaks /VBZ)(the /AT1 windows /NNS) with /IN(a /AT
stone /NN)
AO V TH WITH,IN
中止 打破 折斷 神破 刷新 消失 破 破裂 粉碎 停止 崩潰 遺露 違反 違犯
違背 摧毀 暴跌 潰決 堅破 新 新裂 損壞 能開 整開 :24
窗 窗子 窗戶 扇 :4
打破 窗子
打破 窗戶
約翰 用石 把窗子 打破

(The /AT1 window /NN)(breaks /VBZ) into /IN(pieces /NNS)
TH V INTO,OO
窗 窗子 窗戶 扇 :4
中止 打破 折斷 神破 刷新 消失 破 破裂 粉碎 停止 崩潰 遺露 違反 違犯
違背 摧毀 暴跌 潰決 堅破 新 新裂 損壞 能開 整開 :24
窗戶 破
窗戶 破 成碎片

(Brent /VB)(the /AT1 news /NN) to /IN(him /PP3O)(gentlely /RB
)
V TH BB MA
消息 新聞 :2
中止 打破 折斷 神破 刷新 消失 破 破裂 粉碎 停止 崩潰 遺露 違反 違犯
違背 摧毀 暴跌 潰決 堅破 新 新裂 損壞 能開 整開 :24
消息 遺露
把消息 婉轉 地遺露 給他

In /IN(Autum /NR)(the /AT1 leaves /NNS)(change /VB) from /IN(
green /NN) to /IN(brown /NN)
TI TH V PROM,SO TO,OO
分別 出弄 丟人 別 走 留 留下 退席 客 剩 剩下 跑開 樹葉 能開 扇 :15
兌換 伏過 伏誅 改 改易 改變 更動 修改 神 推移 換 替換 通換 變 變幻
變更 變易 變換 變通 :19
樹葉 變
秋天裡 樹葉 由綠色 變 成棕色

(We /PP1AS)(count /VB)(them /PP3OS) among /IN(our /PP$
friends /NNS)
AO V TH CP
支數 伯爵 告發條項 依照 計算 計數 當作 算 認為 數 數目 點 :12
友 友人 朋友 :3
當作 朋友
我們 當作 他們 是我們的朋友

(We /PP1AS)(develop /VB)(all /ABN the /AT1 natural /JJ
resources /NNS) in /IN(our /PP$ country /NN)
AO V TH IN,PL
引起 宏揚 神洗 長 長成 建設 振興 培養 產生 發育 發展 發展起來 發揮
發揚 開發 開闢 增進 養成 游 願像 願影 :21
資源 :1
開發 資源
我們 在我們的國家 開發 所有的地盤天然資源

(The /AT1 boy /NN)(drives /VBZ)(the /AT1 cattle /NNS) along
/IN(the /AT1 road /NN) in /IN(the /AT1 evening /NN)
AO V TH ALON,PA TI
行駛 起步 打 喚醒開動 推動 推進 開 開車 開車送 幹勤 經 鍋氣 駕 駕車
駕駛 開動 :16
牛 :1
經 牛
男孩 在n傍晚時 沿著路 趕 牛

(They /PP3AS)(drive /VB) to /IN(the /AT1 station /NN)
AO V TO,LOO
他們 她們 我們 票證 :4
行駛 起步 打 喚醒開動 推動 推進 開 開車 開車送 幹勤 經 鍋氣 駕 駕車
駕駛 開動 :16
他們 開
他們 開車

他們 開車 到車站

(He /PP3A)(drives /VBZ)(me /PP1O) to /IN(the /ATI station /NN)(this /DT morning /NN)
AO V TH TO,LOO TI
行駛 起步 釘 喚醒驅動 推動 推進 開 開車 開車送 幹勁 逗 鋪瓦 置 置車
置駛 驅動 :16
m 叫醒 我 :3
開車送 我
他 今天早上 開車送 我 到車站

(Drive /VB)(the /ATI nail /NN) through /IN(the /ATI wood /NN)
V TH THRO,LOO
浮釘 釘子 :2
行駛 起步 釘 喚醒驅動 推動 推進 開 開車 開車送 幹勁 逗 鋪瓦 置 置車
駕駛 驅動 :16
釘子 釘
把釘子 釘 入木頭

(The /ATI workers /NNS)(drive /VB)(the /ATI tunnel /NN) through /IN(the /ATI mountain /NN)
AO V TH THRO,LOO
行駛 起步 釘 喚醒驅動 推動 推進 開 開車 開車送 幹勁 逗 鋪瓦 置 置車
駕駛 驅動 :16
地下道 地道 函洞 隧道 :4
開 隧道
工人 把隧道 開 入山

(He /PP3A)(enters /VBZ)(the /ATI hall /NN) with /IN(his /PP$ sister /NN)
AO V LO WITH,COAO
入 上 吃領 記 參加 進 進入 進來 :8
大廳 門廳 堂 殿堂 廳 廳堂 :6
進入 大廳
他 和他的妹妹 進入 大廳

(We /PP1AS)(fix /VB)(the /ATI pole /NN) in /IN(the /ATI ground /NN)(properly /RB)
AO V TH IN,LO MA
改正 固定 定下 修理 梳 準備 填 :7
杆 杆子 被騙人 桿子 饵 :5
固定 杆子
我們 適當 固定 杆子 在地上

(Aunt /NN)(fixes /VBZ)(the /ATI breakfast /NN) for /IN(me /PP1O)
AO V TH FOR,OBB
改正 固定 定下 修理 梳 準備 填 :7
早飯 早膳 早餐 早點 晨間自助餐 :5
準備 早餐
伯母 為我 準備 早餐

(I /PP1A)(fix /VB)(everything /PN) in /IN(advance /NN)
AO V TH TI
改正 固定 定下 修理 梳 準備 填 :7
每一件事 事事 物物 樣樣 :4
準備 每一件事
我 事先 準備 每一件事

(I /PP1A)(fix /VB)(the /ATI radio /NN) for /IN(John /NP)
AO V TH FOR,OBB
改正 固定 定下 修理 梳 準備 填 :7
收音機 :1
修理 收音機
我 為約翰 修理 收音機

(I /PP1A)(always /RB)(keep /VB)(my /PP$ appointment /NN) as /IN(time /NN)
AO PR V TH MA
放 保存 保持 保留 保置 待 持續 赴 留下 開 照顧 經營 養 遵守 照 :15
任命 決定 約定 約會 :4
赴 約會
我 總是 準時 赴 我的約會

(They /PP3AS)(keep /VB)(a /AT small /JJ shop /NN) in /IN(the /ATI city /NN)
AO V TH IN,PL
放 保存 保持 保留 保置 待 持續 赴 留下 開 照顧 經營 養 遵守 照 :15
店 店鋪 商店 鋪 鋪子 購物 鋪 鋪子 :8
開 店
開 店鋪
他們 在城市 開 一個小店

(His /PP$ illness /NN)(keeps /VBZ)(him /PP3O) in /IN(the /ATI hospital /NN)(6 /CD weeks /NNS)
AO V TH IN,LO DU
放 保存 保持 保留 保置 待 持續 赴 留下 開 照顧 經營 養 遵守 照 :15
m 他 :2
待 他
他的毛病 把他 待 在醫院 6星期

(We /PP1AS)(keep /VB)(a /AT seat /NN) for /IN(him /PP3O)
AO V TH FOR,OBB
放 保存 保持 保留 保置 待 持續 赴 留下 開 照顧 經營 養 遵守 照 :15
位子 坐位 容納 席位 座位 就座 :6
保留 座位
我們 為他 保留 座位

(I /PP1A)(always /RB)(keep /VB)< silent /JJ >
AO PR V CP
放 保存 保持 保留 保置 待 持續 赴 留下 開 照顧 經營 養 遵守 照 :15
木納 沈默 沉默 寂靜 默省無聲 靜默 :6
保持 沈默
我 總是 保持 沈默

(He /PP3A)(lead /VB)(me /PP1O) (his /PP$ bike /NN)
AO V BB TH
予以 借 借給 貸出 增添 :5
自行車 單車 腳踏車 :3
借 腳踏車
他 借 我 他的腳踏車

(Lend /VB)(me /PP1O)(your /PP$ flashlight /NN)
V BB TH
予以 借 借給 貸出 增添 :5
手電筒 電筒 :2
借 手電筒
借 我 你的手電筒

(They /PP3AS)(lend /VB)(us /PP1OS)(their /PP$ area /NNS)(generously /RB)
AO V BB TH MA
予以 借 借給 貸出 增添 :5
牛 :1
借 牛
他們 慨然 借 他們的牛 給我們

(They /PP3AS)(lend /VB)(all-out /JJ supports /NNS) to /IN(our /PP$ school /NN)
AO V TH BB
予以 借 借給 貸出 增添 :5
支持 支援 支撐 助陣 扶 扶持 愛戴 擁護 擁蓋 :9
予以 支持
他們 予以 大力的支持 給我們的學校

(The /ATI prisoner /NN)(moves /VBZ)(his /PP$ feet /NN)(slowly /RB)
AO V TH MA
心動 打動 有所感應 改變 步 步驟 走 招致 造成 看法改變 移動 動 動彈
移 移動 移植 進行 造成 感動 搬 順走 搬移 搬運 調動 動動 轉 轉移 :27
尺 呎 足 英尺 英呎 腳 :6
移動 腳
囚徒 把他的腳 移動 得慢慢

line 177
(Move /VB)(your /PP$ car /NN)
V TH
心動 打動 有所感應 改變 步 步驟 走 招致 造成 看法改變 移動 動 動彈
移 移動 移植 進行 造成 感動 搬 順走 搬移 搬運 調動 動動 轉 轉移 :27
克拉 汽車 車 車子 普通綜合險 :5
移動 車子
移動 你的車子

line 178
(The /ATI talk /NN)(moves /VBZ) from /IN(London /NP) to /IN(Paris /NP)
TH V PROM,LSO TO,LOO
交談 言 話 說 談話 演 演一演 演員 演論 講 :10

心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
話 感動
展列 轉移
話 從信教 感動 到巴黎

line 179
(The /AT1 government's /NN$ opinions /NN$)(move /VB)
TH V
意見 意思 說 道 :4
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
意見 改變
政府的意見 改變

line 180
(They /PP3A$)(move /VB) from /IN(the /AT1 present /JJ house /NN
)(yesterday /NR)
AQ V FROM,LSO TI
他們 她們 我們 領導 :4
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
他們 移動
他們 跟走
我們 跟
他們 昨天 從這個在場房子 移動

(The /AT1 work /NN)(moves /VBZ)(quickly /RB) during /IN(these
/DT$ two /CD weeks /NN$)
TH V MA DURI,DU
工 工作 工程 事 做事 從事 處理 部 協會 :9
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
工作 進展
部 跟
工作 在這些兩星期期間 進展 得很快

(The /AT1 train /NN)(moves /VBZ)(slowly /RB) along /IN(the
/AT1 river /NN side /NN)
TH V MA ALON,PA
火車 系列 訓練 鍛鍊 :4
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
火車 移動
火車 延著河水邊 慢慢 地移動

(The /AT1 assembly /NN line /NN)(moves /VBZ)(smoothly /RB)
TH V MA
白領 句 行 排 電話線路 線 橫列 連線 :8
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
線 進行
配件線 進行 得順利

(I /PP1A)(move /VB)(quickly /RB) to /IN(the /AT1 table /NN)
AQ V MA TO,LOO
我 咱家 其道 其他 那人 :5
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
我 移動
我 很快 地移動 到桌子邊

(People /NN$)(move /VB) through /IN(the /AT1 halls /NN$)
TH V THRO,LOO
人 人民 人們 :3
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
人們 移動
人們 移動 過大廳

(The /AT1 earth /NN)(moves /VBZ) round /IN(the /AT1 sun /NN)
TH V ROUN,PA
土 地球 :2
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
地球 繞
地球 繞著太陽 轉

(Their /PP$ revolutionary /JJ drive /NN)(deeply /RB)(moves /VBZ
)(us /PP1OS)
TH MA V BX
行駛 起步 釘 喚醒鼓動 推動 推進 開 開車 開車送 幹勁 趕 鎮氣 管 駕車
駕駛 鼓動 :16
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
幹勁 感動
他們的革命幹勁 深深 地感動 我們

(The /AT1 story /NN)(moves /VBZ)(them /PP3OS)(very /QL much
/RB)
TH V BX DQ
故事 :1
心動 打動 有所感應 改變 步 步驟 走 招數 連續 看法改變 鄰動 動 動彈
移 移動 移植 進行 進展 感動 跟 跟走 跟移 跟通 調動 傳動 轉 轉移 :27
故事 感動
故事 使他們 非常很多 感動

(They /PP3AS)(open /VB)(the /AT1 dialogue /NN) with /IN(us
/PP1OS)
AQ V TH WITH,COAO
公開 打開 全天服務 伸開 折封 空曠 展開 領開 典明 開 開始 開放 開啟
開著 開幕 開學 開關 膠開 盛宴中 越開 露封 :21
問答 對白 對答 對話 對談 :5
展開 對話
展開 對談
他們 和我們 展開 對話

(They /PP3AS)(open /VB)(a /AT conversation /NN)(instantly /RB
)
AQ V TH MA
公開 打開 全天服務 伸開 折封 空曠 展開 領開 典明 開 開始 開放 開啟
開著 開幕 開學 開關 膠開 盛宴中 越開 露封 :21
會話 對談 :2
展開 對談
他們 把對談 展開 得立即

With /IN(their /PP$ help /NN)(we /PP1AS)(open /VB)(a /AT
small /JJ bookshop /NN)
WITH,IN AQ V TH
公開 打開 全天服務 伸開 折封 空曠 展開 領開 典明 開 開始 開放 開啟
開著 開幕 開學 開關 膠開 盛宴中 越開 露封 :21
書店 :1
開 書店
靠著他們的幫助 我們 開了 一個小書店

(Last /AP year /NN)(we /PP1AS)(open /VB)(training /VBG
classes /NN$) for /IN(the /AT1 school /NN teachers /NN$)
TI AQ V TH FOR,OBB
公開 打開 全天服務 伸開 折封 空曠 展開 領開 典明 開 開始 開放 開啟
開著 開幕 開學 開關 膠開 盛宴中 越開 露封 :21
屆 班 級 級別 期 等 階級 種類 :8
開 班
去年 我們 為學校老師 開 訓練班

(They /PP3AS)(place /VB)(the /AT1 picture /NN)(too /QL high
/RB) on /IN(the /AT1 wall /NN)
AQ V TH MA ON,LO
下 地 地方 安 安放 安置 位置 所在 放 放置 座位 處 :12
情況 描 描繪 畫 照片 像 圖片 圖形 :8
放 置
他們 把畫 在牆上 放 得太高

(I /PP1A)(place /VB)(the /AT1 book /NN) under /IN(the /AT1
desk /NN)
AQ V TH UNDB,LO
下 地 地方 安 安放 安置 位置 所在 放 放置 座位 處 :12
本 定 定下 訂 書 書本 書卷 紀人 帳面 簿 :10
放 書
我 把書 放 在書桌底下

(He /PP3A)(places /VBZ)(his /PP$ cap /NN) on /IN(a /AT chair
/NN)
AQ V TH ON,LO
下 地 地方 安 安放 安置 位置 所在 放 放置 座位 處 :12
汽車蓋 金屬帽 便帽 蓋帽 帽子 無邊便帽 軍帽 鴨嘴帽 鋼盔蓋 :9
放 帽子
他 放 他的帽子 在椅子上

(They /PP3AS)(provide /VB)(valuable /JJ data /NNS) for /IN(soil /NN improvement /NN)
AG V TH BB
出 供給 規定 提供 :4
事實資料 資料 :2
提供 資料
他們 提供 有價值資料 給土壤改良

(The /ATI state /NN)(grudgingly /RB)(provides /VBZ)(little /AP money /NN) for /IN(water /NN conservation /NN projects /NNS)
AG MA V TH BB
出 供給 規定 提供 :4
金錢 幣 銀錢 錢 錢財 :5
提供 錢
州 勉勉強強 地提供 少許錢 給水保護計劃

(I /PP1A) (always /RB) (regard /VB) (him /PP3O) (highly /RB)
BX PR V TH MA
n 他 :2
有關 注視 看 重視 視為 當 認為 關心 :8
他 看
我 總是 把他 看 得很高

(I /PP1A) (regard /VB) (him /PP3O) as /IN (my /PPS brother /NN)
BX V TH CP
有關 注視 看 重視 視為 當 認為 關心 :8
兄弟 弟弟 :2
當 兄弟
我 把他 當 成我的兄弟

For /IN (a /AT moment /NN) (she /PP3A) (regards /VBZ) (me /PP1O) with /IN (wide /JJ eyes /NNS)
FOR.DU BX V TH MA
有關 注視 看 重視 視為 當 認為 關心 :8
m 叫醒 我 :3
注視 我
有一會兒 她 眼睛睜得大大 地注視 我

(They /PP3AS) (regard /VB) (him /PP3O) with /IN (interest /NN)
BX V TH MA
有關 注視 看 重視 視為 當 認為 關心 :8
m 他 :2
注視 他
看 他
他們 感興趣 地看 他

(That /DT) (regards /VBZ) (me /PP1O)
TH V COTH
m 叫醒 我 :3
有關 注視 看 重視 視為 當 認為 關心 :8
我 有關
那個 跟我 有關

(Smoke /NN) (rises /VBZ) from /IN (the /ATI factory /NN chimneys /NNS)
TH V FROM.LSO
吃煙 吸食 抽煙 吹煙 煙 煙燻 :6
上升 上昇 上漲 升 升起 泛出 浮 浮現 起 起立 起床 高漲 提高 散會 發生 增長 興起 矗立 :18
煙 升起
煙 從工廠煙囪 升起

(The /ATI dog /NN) (rolls /VBZ) on /IN (the /ATI floor /NN)
TH V ON.PL
狗 :1
小圓麵包 匹 打滾 使自己 卷 滾 捲 搖擺 捲繞 開 搖晃 滾 滾動 碾壓 橫滾 翻滾 :16
狗 打滾
狗 在樓上 打滾

(The /ATI train /NN) (rolls /VBZ) (slowly /RB) into /IN (the /ATI station /NN)
TH V MA INTO.LGO
火車 系列 訓練 繼續 :4
小圓麵包 匹 打滾 使自己 卷 滾 捲 搖擺 捲繞 開 搖晃 滾 滾動 碾壓 橫滾 翻滾 :16
火車 開
火車 慢慢 開 進車站

(He /PP3A) (rouses /VBZ up /RP) from /IN (sleep /NN)
TH V FROM.LSO
他 我 :2
引起 叫醒 弄醒 激 醒 驚醒 :6
他 醒
我 叫醒
他 從睡夢 醒起

(The /ATI news /NN) (rouses /VBZ) (great /JJ indignation /NN)
AG V TH
引起 叫醒 弄醒 激 醒 驚醒 :6
不平 憤慨 :2
引起 憤慨
消息 引起 大憤慨

(The /ATI bus /NN) (runs /VBZ) from /IN (Yenan /NP) to /IN (Xian /NP)
TH V FROM.LSO TO.LGO
公共汽車 公車 巴士 :3
串 走私 延伸 治理 持續 流 執行 帶動 撐 跑 跑步 跑來 進行 開 經營 運轉 管理 賽跑試跑 淌 撈兒 競選 變 變得 :23
公共汽車 開
公共汽車 從延安 開 到西安

(The /ATI trolley-bus /NN) (runs /VBZ) (every /JJ three /CD minutes /NNS)
TH V FR
電車 :1
串 走私 延伸 治理 持續 流 執行 帶動 撐 跑 跑步 跑來 進行 開 經營 運轉 管理 賽跑試跑 淌 撈兒 競選 變 變得 :23
電車 開
電車 每三分鐘 開

(The /ATI road /NN) (runs /VBZ) for /IN (many /AP miles /NNS) by /IN (the /ATI sea /NN)
TH V FOR.LB BY.PA
公路 馬路 路 路程 道 道路 :6
串 走私 延伸 治理 持續 流 執行 帶動 撐 跑 跑步 跑來 進行 開 經營 運轉 管理 賽跑試跑 淌 撈兒 競選 變 變得 :23
公路 延伸
路 延伸
路 沿著海 延伸 許多英里

(The /ATI forest /NN) (runs /VBZ) (intermittently /RB) for /IN (200 /CD kms /NNS)
TH V MA FOR.LB
森林 樹林 :2
串 走私 延伸 治理 持續 流 執行 帶動 撐 跑 跑步 跑來 進行 開 經營 運轉 管理 賽跑試跑 淌 撈兒 競選 變 變得 :23
森林 延伸
森林 交錯著 延伸 200公里

(Their /PPS food /NN supply /NN) (runs /VBZ) (low /RB)
TH V CP
串 走私 延伸 治理 持續 流 執行 帶動 撐 跑 跑步 跑來 進行 開 經營 運轉 管理 賽跑試跑 淌 撈兒 競選 變 變得 :23
不夠 低 低下 低微 低沈音錄音帶 卑下 :6
變得 不夠
他們的糧食供應 變得 不夠了

(The /ATI train /NN) (runs /VBZ) on /RP (Sunday /NR)
TH V TI
火車 系列 訓練 繼續 :4
串 走私 延伸 治理 持續 流 執行 帶動 撐 跑 跑步 跑來 進行 開 經營 運轉 管理 賽跑試跑 淌 撈兒 競選 變 變得 :23
火車 開
火車 星期日 開

(The /ATI tears /NNS) (run /VBZ) down /RP (his /PPS face /NN)
TH V DOWN.LO
淚 眼淚 淚下 淚珠 :4
串 走私 延伸 治理 持續 流 執行 帶動 撐 跑 跑步 跑來 進行 開 經營 運轉 管理 賽跑試跑 淌 撈兒 競選 變 變得 :23
眼淚 流
眼淚 流下 他的臉

漢語的動詞名物化初探--
漢語中帶論元的名物化派生名詞

葉美利* 湯志眞** 黃居仁** 陳克健*
*中央研究院資訊科學所
**中央研究院歷史語言研究所

摘要

動詞在句子中既可以當謂語，又可以當主、賓語，加上中文又缺乏英語的‘that’，‘to’以及‘-ing’等標示；因此造成角色上的歧義，引起剖析上的困擾。我們觀察其語法表現，發現出現在主、賓位的動詞中，有些具備動詞的語法特性，有些則呈現名詞的語法特性。我們認爲呈現名詞語法性質的那些詞實爲動詞經名物化所形成的派生名詞，因爲缺乏構形標記，所以形式與動詞完全一樣。本文研究名物化的現象並採用陳與黃 [1]訊息爲本的格位語法爲漢語的派生名詞提供一個語法表達模式。

1. 緒論

動詞[註1]在句中除了擔任謂語之外， 也可以出現在主、賓語的位置。如何判斷動詞在句子裡所扮演的角色，因而成爲剖析上的一大課題。我們以動詞與名詞的語法特性來測試，發現主、賓位的動詞有的具備動詞的語法特性，有的則呈現名詞的語法特質。兩種情形最大的不同點在於，前者出現的句子，主要動詞可次類劃分(subcategorize)動詞組[註2]爲主語或賓語；而後者出現的句子，其主要動詞只次類劃分名詞組爲主、賓語。

本文採用的訊息爲本的格位語法 (Information-based Case Grammar，ICG)[1] ，在詞項中除了標示論元的語意角色之外，也標明其語法形式。所以，根據主要動詞的語法訊息，應該就可以預測第一種情形中，動詞所扮演的角色。第二種情形中，主要動詞劃分的主、賓語是名詞，但卻出現一個一般認爲是動詞的詞語，並且還具備名詞的語法特性。如何處理這種現象，使該詞獲得適當的語法訊息，是本文的研究重點。根據我們的觀察，出現在這類情形中的詞，雖然形式與動詞一樣，但是以動詞的語法特性來測試，卻不具備這些特質，反之，卻可以通過名詞的語法特質測試。這些詞雖然不具備動詞的外在句法結構形式，卻具有動詞內在的語意框架(semantic frame)，承襲了動詞的論元角色，只是論元的體現形式 (realization)不同於動詞。因此，我們不主張將之多重分類 [註3]。我們認爲具備名詞特性的詞是動詞

經名物化 (nominalization) 所形成的派生名詞(derived nouns)[註4]。本文第三節依是否保留原動詞的語意論元將派生名詞分為可帶論元與不可帶論元兩大類，並依論元之體現將可帶論元的派生名詞分成十小類，同時討論如何利用訊息為本的格位語法中的語法表達模式來完成這類派生名詞組的語法與語意分析。

2. 動詞名物化

2.1. 動詞或名詞?

幾乎所有談論漢語名物化的文章都把名物化定義為：位於主、賓語位置的動詞、形容詞。名物化之所以會引起廣泛的討論 [2&3]，主要是因為，以注語言學家都以功能來劃分詞類，並且認為動詞的主要功能是充當句子的謂語，而名詞則擔任主語、賓語等論元。所以，動詞、形容詞跑去當論元就是動詞名用，或是名物化 [2]。根據我們的觀察，同樣是主、賓位上的動詞，外在的句法表現卻大不相同：有些主、賓位的動詞具有動詞的語法特性，有些卻不然，一律以名物化一詞來概之，似乎並不恰當。比較下列兩組句子：

(1) a. 研究是一個問題。
    b. 他們打算研究。
(2) a. 研究進行得很順利。
    b. 他們破壞了研究。

上面兩組句子中，雖然「研究」都出現在主位或賓位，然而只有 (1)句中的「研究」具有動詞的語法特性[4]，可以(A)以「不」來否定(3a，4a)、(B)形成正反問句(3b，4b)、(C)直接前加助動詞 (3c，4c)、(D)以修飾動詞的副詞來修飾(3d，4d)、(E)帶上由「得」字所引導的結果或情狀補語(3e, 4e)[註5、6]，而(2)句的「研究」卻不具備這些特點(5，6)。

(3) a. 不研究是一個問題。
    b. 研不研究是一個問題。
    c. 要研究是一個問題。
    d. 立刻研究是一個問題。
    e. 研究得很透澈是一個問題。
(4) a. 他們打算不研究。
    b. 他們打算研不研究?
    c. 他們打算要研究。
    d. 他們打算立刻研究。
    e. 他們打算研究得透澈一點。

(5) a. *不研究進行得很順利。
b. *研不研究進行得很順利。
c. *要研究進行得很順利。
d. *立刻研究進行得很順利。
e. *研究得很透澈進行得很順利。
(6) a. *他們破壞了不研究。
b. *他們破壞了研不研究。
c. *他們破壞了要研究。
d. *他們破壞了立刻研究。
e. *他們破壞了研究得很澈底。

另外，(1)句中的「研究」，賓語可以直接出現在動詞後面，而(2)句的「研究」卻不行。

(7) a. 研究生態很困難。
b. 他們打算研究生態。
(8) a. ?研究生態進行得很順利。
b. *他專門破壞研究生態。

反之，如果賓語以修飾語的形式出現於「研究」前面(8c，8d)，句子的合法度就不會產生問題。

(8) c. 生態的研究進行得很順利。
d. 他專門破壞生態的研究。

「研究」在(1)、(2)兩句的主、賓語位置呈現截然不同的表現，這與句子的主要動詞有關。(1)句中的主要動詞「是」與「打算」都是原來在格框中就分別認可動詞組擔任主語與賓語的動詞。而位在其主、賓位上的動詞「研究」，確實也帶有動詞的語法特性。(2)句的主要動詞「進行」、「破壞」一般不認為其格框帶有形式為動詞組的論元，而位於其主、賓位上的「研究」非但不具備動詞的性質，反而可前加常與名詞搭配出現的定量複合詞組。

(9) a. 這三項研究進行得很順利。
b. 他破壞了那兩項研究。

根據這些表現，我們推測(2)句中的「研究」可能是一個名詞。(2a) 中的「研究」不但可以用領屬名詞組修飾(10a)，還可以出現在「把」字句的

賓語位置(10b)，「被」字句的主位(10c)，同時又可以與一般名詞並列，這些都是一般認為只有名詞會出現的位置，更加証明 (2)句中的「研究」是一個名詞。

(10) a. 他們破壞了我的研究。
b. 他們把研究破壞了。
c. 研究被他們破壞了。
d. 他們破壞了研究和設備。

反之，「打算」的賓語是動詞組， 因此無法以領屬名詞組修飾，不能出現在「把」字句的賓位[註7]、「被」字句的主位，也不能與名詞並列。

(11) a. *他們打算我的研究。
b. *他們把研究打算。
c. *研究被他們打算。
d. *他們打算研究和設備。

綜合以上討論， 我們認為(1)、(2)兩句中的「研究」不同，一是動詞，一是名詞。

2.2. 名物化派生名詞

前面提到(2)句中的「研究」為名詞。它與動詞「研究」有什麼關係呢？我們認為(2)中的「研究」(下文中以「研究N」來代表) 是動詞「研究」(「研究V」)經過名物化所形成的派生名詞，就如英語的派生名詞 ‘ creation，agreement，criticizism ...’ 是由動詞 ‘create，agree，criticize ... ’ 名物化而來的。只是因為漢語缺乏構形變化，所以動詞「研究」與所派生出來的名詞，形式一模一樣。事實上，英語亦有零構形的派生名詞，如‘research，program，plan ...’ ，而漢語的「信賴度、影響力、依賴性」等，則可視為是一種有構形變化的派生名詞 [註8]。這些派生詞是由動詞「信賴、影響、依賴」加上後綴(suffixes)「度、力、性」名物化而來的。「信賴、影響、依賴」亦可形成零構形的派生名詞「信賴N、影響N、依賴N」[註9]。請比較下面的例子[註10]：

(12) a. 車主對保養廠的信賴
b. 車主對保養廠的信賴度
(13) a. 黨部在嘉義縣對民眾的影響
b. 黨部在嘉義縣對民眾的影響力

(14) a. 小孩對父母親的依賴
b. 小孩對父母親的依賴性

「影響力」、「信賴度」與「依賴性」等有構形的名物化是動詞經一構形變化後衍生出來的名詞，可以由構詞律來處理。關於構詞律的處理，請參考洪等[8]。本文主要討論零構形派生名詞的處理方式。

3. 派生名詞的分類

3.1. 論元結構與派生名詞的分類

前面提到派生名詞「信賴N、影響N、依賴N」與「信賴度、影響力、依賴性」都是由動詞「信賴V、影響V、依賴V」名物化而來的。動詞名物化後派生出來的名詞，雖然詞類已經改變，卻還具有原動詞的論元角色，只是體現方式不同。動詞的論元體現為主語或賓語；名詞的論元則體現為修飾語。例句(18)中，「國人」為「認同」之客體(theme)，而「傳統家電」為其目標(goal)[註11]。

(18) a. 國人對傳統家電的認同
b. 國人對傳統家電的認同感
c. 國人對傳統家電很認同。
d. 國人很認同傳統家電。
(19) a. 法律對個人的約束
b. 法律對個人的約束力
c. 法律約束個人。

但是，並非所有的派生名詞都承襲原動詞的論元結構，如(20b)的「參考」就不可以帶論元角色。

(20) a. 你可以參考這篇文章。
b. *你對這篇文章的參考
c. 你可以把這篇文章當作一種參考。

除了上面的派生名詞之外，非派生的名詞如「興趣」、「誠意」、「信心」也可以帶論元。

(21) a. 他對文學的興趣是從大學時代培養起來的。
b. 經過這次意外後，客戶對這些產品的信心大減。

因此，我們有必要將名詞分為兩大類，一類帶論元(argument-taking nouns，

以 AN來代替)，一類不帶論元(non-argument-taking nouns，在詞庫小組的分類中屬NA類[10])[註12]。

3.2. 論元的體現與帶論元名詞的分類

上一節提到名詞可依是否帶論元分為兩大類。對於不帶論元的名詞，詞庫小組曾依可不可數以及抽象或實體等條件來細分[10]。帶論元的名詞的分類標準，主要是所帶的論元以及論元的體現方式[註13]。

第一類 (AN1)帶主事與目標兩個論元，而且目標由「對」引介。屬於這一類的有「規劃」、「訓練」與「影響」、「影響力」等。

(22) a. <東歐民主運動><對藝術>的影響將源源不斷地湧現。
agen goal
b. <東歐民主運動>的影響將源源不斷地湧現。
agent

如果主事與目標都出現的話，主事體現為名詞組(NP)，目標為由‘對’引介的名詞組(PP[對])，且主事在目標之前，目標與派生名詞之間有一個標示中心語的標記(head marker)‘的’(22a)[14]；若主事單獨出現，則其體現為‘NP’，主事與派生名詞中間有‘的’(22b)。以下為這類名詞語法表達形式的片段[註14]，我們把論元的體現以限制的方式標於各論元之後，以線性次序津‘BP’(Basic Pattern)來表達論元之間的次序。

```
semantic:  ┌  .
           │  .
           └arguments: agent，goal

syntactic: ┌ class:  AN1
           │ features:
           └ constraints:  ┌form:  .
                           │       .
                           │       .
                           │   agent[NP]
                           │   goal[PP[對]]
                           └ BP: ┌ B1 agent < goal < 的 < *
                                 └ B2 agent < 的 < *
```

第二類(AN2)帶經驗者(experiencer, exp)與目標兩個論元，目標和第一

類一樣，由介詞來引介。除了經驗者與主事上的不同之外，這類的情形大致與第一類相同。

(23) a. 隨著時光流逝，<他><對我>的怨恨也日益加深。
exp goal
b. 沒想到我這麼做反而招致<他>的怨恨。
exp

第三類帶論元的名詞 (AN3)和前兩類一樣，也帶有一個目標，且目標可以‘對’來引介，但是另一個論元爲客體。

(24) a. 有關單位特地派人前來表達<政府><對此事>的重視。
theme goal
b. 此案已獲得<有關單位>的重視。
theme

第四、五、六類帶論元的名詞(AN4、AN5、AN6)都有一個形式可以爲動詞組(VP)或句子(S)的目標；第四類和第一類一樣，還有一個論元是主事；第五類和第二類一樣，有另一個論元爲經驗者；而第六類則和第三類一樣，另一個論元爲客體。

第四類：

(25) a. 他提出<校長有貪污情形應予以處罰>的主張。
goal
b. 政府並未同意<另外開放民間投資興建>的建議。
goal
c. 他們呼籲大家支持<該會><全民反核>的主張。[註15]
agent goal

第五類：

(26) a. 有<他是否又在騙人>的懷疑
goal
b. 表達<他們><被上級忽視>的不滿
exp goal
d. 表達<他>的不滿
exp

第六類：

(27) a. 成為<懂品味，有身份>的象徵
goal
b. 揭示了<搬運機><高速化>的傾向
theme goal
c. 因此，<府會關係破裂>的傳聞不逕而走
goal

第七類帶論元的派生名詞 (AN7)帶主事、目標與一個形式為動詞組或句子的客體。三個論元很少一起出現，可能是因為兩到三個名詞組一起出現的話，不僅容易引起混淆，唸起來也不順。 (28a)為主事與客體一起出現的情形，(28d)為目標與客體共同出現的例子。

(28) a. 執政黨如此做，不外是希望換取
<黃信介><民進黨不杯葛今天開議議事>的承諾。
agent theme
b. 有關單位希望攤販能遵守<每天中午十二時一律休市>的承諾。
theme
c. <趙署長>的承諾，使他信心大增。
agent
d. 有關單位昨天下達<警方><依法取締外籍勞工>的命令。
goal theme

第八類 (AN8)如「買賣」、「拍賣」、「發展」、「建立」、「建設」、「設計」等，帶的論元為主事與客體，客體與派生名詞之間可以有‘的’(29a) ，也可以不要有‘的’(29b)。主事多不出現，如果出現，多半會有受事[註16]。

(29) a. <板橋車站>的設計
theme
b. <商品>買賣
theme
c. <漢城與莫斯科><外交關係>的建立
agent theme

第九類(AN9)只帶一個客體，這類名詞包括「失敗」、「軟弱」、「進步」、「轉變」等。

(30) a. 他們事前沒有計畫，導致<這次行動>的失敗。
theme
b. 大家同心協力促進<社會國家>的進步。
theme

最後一類帶論元的名詞(AN10)只帶主事一個論元，其體現爲‘NP’。

(31) a. <他們>的探險還未結束。
agent
b. <兩國>的戰爭一直擴大。
agent

各類的詳細語法表達請參見附錄。

3.3. 派生名詞的附加成份及其句型

上一節討論的是帶論元派生名詞的論元及基本句型，這裡討論這類派生名詞的附加成份及這些附加成份的句型。根據我們的觀察，這類名詞常帶的附加成份有時間(time)、地點 (location)、數量詞 (quantifier)及修飾語(property)。以下簡單說明這些附加成份常出現的形式及其線性次序‘AP’(Adjunct Pattern)。

時間附加成份常以中心語爲時間名詞的名詞組(NP[Nd])、由表時間觀念的準量詞和定詞組合而成的定量複合詞組(DM[Nfg，+definite，+temporal_relation])、方位詞組(GP)或介詞組(PP)的形式出現。

(32) a. 他因此削減了<以往>對該組織的善意支持。(NP)
b. 這正應了我<上星期>在自由副刊上對他的批評。(DM)
c. 這正應了我<這些日子來>在自由副刊上對他的批評。(GP)
d. 這正應了我<臨出國前>在自由副刊上對他的批評。(PP)

地點這個附加成份通常以介詞組的形式出現，請參考(32b、32c、32d)。數量詞以定量複合詞組的形式出現(33)；修飾語的形式通常是名詞 (34a)、動詞(34b)或非謂形容詞(34c)。

(33) 這正應了我臨出國前在自由副刊上對他的<那三項>批評。

(34) a. 他因此削減了以注對該組織的<善意>支持。
b. 解決國內對直升制度的<猛烈>批評。
c. 有關單位已取消對後期發展地區的<基本>限制。

這些附加成份與論元的次序大致是：(1) 時間在地點之前(35a)；(2) 時間與地點和外在論元[註17]之間沒有線性次序上的限制(35b，c))；(3) 數量詞和內在論元之間沒有明顯的線性次序限制(35d)，然數量詞通常出現在修飾語前面(35e)；(4)中心語則出現在最後。

(35) a. 這正應了我<以前><在報刊上>對他的<一些><猛烈>批評。
b. 這正應了<以前>我<在報刊上>對他的<一些><猛烈>批評。
c. 這正應了<以前><在報刊上>我對他的<一些><猛烈>批評。
d. 這正應了我<以前><在報刊上><一些>對他的<猛烈>批評。
e. *這正應了我<以前><在報刊上>對他的<猛烈><一些>批評。

以下是附加成份的形式以及句型：

```
constraints:form: time[{Np[Nd]，DM[Nfg，+definite，
                      +temporal_relation]，GP，PP}]
                  location[PP]
                  quantifier[DM{Nfa，Nfc，Nfd}]
                  property[{N，V，A}]
            AP:   A1:time < location < goal <  *
                  A2:quantifier < property < *
```

3.4 名物化派生名詞的預測規律

帶論元的名物化派生名詞由動詞名物化而來，自然承襲原動詞的論元結構。因此，如果有一個動詞帶主事與目標兩個論元，目標的形式爲名詞組，則該動詞名物化後產生的帶論元派生名詞屬第一類 (AN1)。以‘支持’一詞爲例，擔任動詞(36a)時的語法訊息可由該類動詞(VC2)的語法表達(37)獲得[註18]，擔任名詞時(36b)，語法訊息則由第四類帶論元派生名詞(AN1)的語法表達模式(38)取得。兩者論元相同，只是形式與句型不同。

(36) a. 約有百分之八十的美國人支持布希的做法。
b. 謝米爾讚揚美國對以色列的大力支持。

(37)

```
semantic:  meaning:
           features:
           arguments: agent，goal
           adjuncts:
syntactic: class:  VC2
           features:
           constraints: form:  .
                               .
                               .
                               agent[{NP，PP[被，爲，由，遭，
                                     歸，+agent]}]
                               goal[{NP，PP[{把、將}，+goal]}]
                        BP: B1 agent[{NP，PP[由]}] < * < goal
```

(38)

```
semantic:  meaning:
           feature:
           arguments: agent，goal
           adjuncts: time，location，quantifier，property
syntactic: class:  AN1
           features:
           constraints: form: time[{Np[Nd]，DM[Nfg，+definite，
                              temporal_relation]，GP，PP}]
                              location[PP]
                              quantifier[DM{Nfa，Nfc，Nfd}]
                              property[{N，V，A}]
                              agent[NP]
                              goal[PP[對]]
                        BP: B1 agent < goal < 的 < *
                            B2 agent < 的 < *
                        AP: A1 time < location < goal < *
                            A2 quantifier < property < *
```

所以，基本上我們可由動詞的語法類預測名物化以後產生出來的帶論元的派生名詞屬於那一個小類。但是，同一類動詞名物化之後，衍生出來的可能是帶論元的派生名詞，也可能是不帶論元的派生名詞；如「參考」與「影響」同屬單賓動作動詞 (VC2)，但是名物化後，一個產生的名詞不帶論元，

一個帶論元。因此我們無法完全由動詞的類別來預測所產生的結果，所以我必須在動詞的詞項中另外標註[註19]。本文以 '[+argument]' 來標示名物化後會產生帶論元的派生名詞的動詞。

綜合以上所述，我們可以由動詞帶的 '[argument]' 特徵來預測名物化後所衍生出來的名詞帶不帶論元。而由動詞所屬的語法類則可預測這個派生名詞屬於那個小類。以下是我們依據這兩個標準所歸納出來的規律，VA等各類動詞所代表的詞類意義請參見詞庫小組<<國語的詞類分析修訂版>>[10]。

1. [{VC2，VB1，VD}，+argument] --> AN1
2. [{VI1，VJ2}，+argument] --> AN2
3. [{VI2，VJ1}，+argument] ---> AN3
4. [{VE2，VF1}， +argument] --> AN4
5. [VK1，+argument] --> AN5
6. [VK2，+argument] --> AN6
7. [{VE1，VF2}，+argument] -->AN7
8. [VC3，+argument] --> AN8
9. [VH，+argument] --> AN9
10. [VA，+argument] --> AN10

以上我們將派生名詞分類並提供了語法表達。剖析器利用動詞及名物化派生名詞的語法訊息，因為句型上的不同，所以可以分辨及分析這兩種不同的結構。

4. 結論

本文將引起剖析困難的動詞當主、賓語的現象依句法上的表現分為兩種：第一種出現的句子，主要動詞可劃分動詞組或句子為論元，因此哪個是主要動詞，哪個為主、賓語，可由動詞的格框訊息得知。第二種情形，動詞雖然保持原來的形式，實質上卻已名物化為一個派生名詞。因此在外在的句法結構上呈現名詞的句法特性。名物化派生名詞雖然不具備動詞的語法特性，卻有動詞的語意框架，承襲了動詞的論元結構。本文將名物化派生名詞依帶論元與否分為兩大類，並按所帶的論元及其體現將帶論元的派生名詞分成十個小類，提供語法表達。因為名物化派生名詞會承襲原動詞的論元，所以我們可以由動詞所帶的 '[argument]' 特徵及所屬的語法類來預測動詞名物化後所產生帶論元的名詞屬於哪個小類。因此，無需多重分類，即可獲得某一特定派生名詞的語法訊息。

注釋：

1. 我們把形容詞視爲一種靜態動詞。
2. 或者是帶動詞組的句子。
3. 我們認爲需要多重分類的情形，是像「好」的狀態不及物動詞(1a)與程度副詞的用法(1b)。名物化派生名詞與動詞間存有語意結構上的關係，如論元的承襲，因此名物化派生詞則可由規律來預測；而「好」的動詞與程度副詞用法之間，卻找不到這種關係，所以必須列兩個詞類。

(1) a. 張三眞好。
    b. 我好喜歡張三。

4. 此外，兩個詞除了詞類不同之外，語音形式相同，語意又相關，如果列爲兩個詞，可能就無法掌握這種關係。另外，究竟是動詞名物化還是名詞詞動化(verbalized)，目前我們以語意爲判斷標準。一般而言名詞動詞化爲動詞多半會含有使動(causative)或者是起始(inchoative)的意味。

5. 名詞和動詞都可以當謂語，但是謂語名詞卻不具有 (A)-(E)所列的特質，試比較 (2)與(3)。可見，眞正可以區分動詞與名詞的，是(A)-(E)四項語法特性。而不是擔認主、賓語或謂語。只是，一般而言，謂語多由動詞來擔任。

(2) a. 他來。
    b. 他不來。
    c. 他來不來。
    d. 他會來。
    e. 他立刻就來。
    f. 他來得很早。
(3) a. 今天星期五。
    b. *今天不星期五。
    c. *今天星不星期五。
    d. *今天會星期五。
    e. *今天立刻星期五。
    f. *今天星期五得很早。

6. 事實上，‘研究有困難’與‘他們打算研究’兩個例子並不是平行的，如下例所示，‘研究是一個問題’中的「研究」也有可能是名詞，而‘他們打算研究’中的「研究」卻只能是動詞。

(4) a. 這項研究是一個問題。
    b. 生態的研究是一個問題。
(5) a. *他們打算這項研究。
    b. *他們打算生態的研究。

7. 劉承慧指出，能不能出現在「把」字結構，除了賓語必需是名詞以外，也與主要動詞有關 (6)。但是，「打算」劃分的是一個動詞組，因此根本就不符合第一個‘賓語是名詞組’的要求。

(6) *我把小王看見了。

8. 呂[5]就認為「員、家、人、度、法、學、力、氣、性」等可以算是後綴。
9. 趙[6])亦把 ϕ 列入名詞的詞尾，認為「司馬」(管馬的人)、「編輯」、「幹事」、「跑街」等動詞同時也是帶零詞尾的執事名詞。
10. 例句(12b)與(13b)係採自林[7]，為了方便呈現我們的重點，我們將(12b)稍加修改。
11. 論元語意角色的指派方面，本文主要遵循林[9]。
12. 帶論元的名詞在句中又可以有過程 (process，如(7a))與結果 (result，如(7b)) 兩種詮釋[11&12&13]。這與語境有關，為句子層次上的問題。本文處理的重點是詞彙訊息的表達。關於這個問題，我們將另外為文探討。

(7) a. 他對我的影響一直持續到大學時代。
b. 他對我的影響造成今天這樣的局面。

13. 這裡的分類所用的論元組合主要是參照詞庫小組的動詞分類，一來是因為這類名詞與動詞的分類標準相同，再者本文主要在處理動詞名物化的帶論元派生名詞，而帶論元的名物化派生名詞會承襲動詞的論元。
14. 這裡先舉一個例子做為示範，下一節還要介紹附加成份及其句型，並將細部的語法表達附於文後。
15. 這類與第六、七類派生名詞可以有一個以上‘的’(8a)，或者‘的’也可以出現在主事後面(8b)。

(8) a. 他們呼籲大家支持該會的全民反核的主張。
b. 他們呼籲大家支持該會的全民反核主張。

16. 若受事不出現，則主事名詞組多會有領屬者的詮釋(10)。

(10) 他的設計

17. 一般而言，外在論元如動態動詞的主事或狀態動詞的容體與經驗者，多體現為基本句式的主語。而內在論元如目標或容體則多體現為賓語。
18. 動詞分類請參考中文詞知識庫小組<<國語的詞類分析修訂版>>。
19. 同樣的情形也見於有語音形式的後綴如‘性’，以「代表」與「依附」兩個動詞為例，兩者為同一類動詞，但是加上這個後綴之後，前者衍生帶論元的派生名詞，後者衍生的卻是不帶論元的派生名詞。

參考書目：

[1] 陳克健、黃居仁，訊息爲本的格位語法——個適用於表答中文的語法模式，中華民國第二屆計算語言學研討會論文集(ROCLING II)，1989，95-119，南港:中央研究院。

[2] 朱德熙、盧甲文、馬貞，關於漢語動詞，形容詞"名物化"的問題，<<北京大學學報・人文學科>> 1961-4。

[3] 史振曄，試論漢語動詞形容詞的名詞化，<<中國語文>> 1960，12。

[4] 湯廷池，動詞的語法屬性，<<國語語法研究>>，學生書局，1979。

[5] 呂叔湘，<<漢語語法論文集增訂本>>，商務印書館，1984。

[6] 趙元任，<<中國話的文法>>，丁邦新譯，香港中文大學出版社，1968。

[7] 林甫雯，討論內部結構爲 V-N 的複合名詞，詞庫小組工作報告，1991。

[8] 洪偉美、黃居仁、湯志眞、陳克建，中文派生詞的構詞規律初探，發表於第三屆世界華文教學研討會，1991。

[9] 林甫雯，ICG中的論旨角色，詞庫小組工作報告，1992。

[10] 中文詞知識庫小組，<<國語的詞類分析修訂版>>，中央研究院計算機中心，1989。

[11] Elina Rigler，The Semantic Classification of Deverbal Nouns，Working Papers in Language Processing No2，1988.

[12] Jane Grimshaw，Argument Structure，the MIT Press，1990.

[13] 湯志眞，漢語的‘的’與英語的‘'s’，發表於第三屆世界華文教學研討會，1991，台北。 即將刊登於中央研究院歷史語言研究所集刊， 第63本第3份。

[14] C.R. Huang，Mandarin Chinses NP de - A Comparative Study of Current Grammatical Theories，Institute of History and Philology Academia Sinica Special Publication No. 93，1989.

☆☆本文之研究得國科會名物化動詞片語的表達與剖析方法研究計劃(NSC81-0408-E001-02)之經費贊助，特此申謝。感謝詞庫小組全體同仁在論文寫作期間提供協助，尤其是林甫雯、劉承慧、魏文眞與張莉萍，他們細心閱讀本文初稿並提供寶貴的意見。最後，還要感謝黃瑞珠與馬北江的熱心協助。

A LEXICON-DRIVEN ANALYSIS OF CHINESE SERIAL VERB CONSTRUCTIONS

Ching-Long Yeh
Department of Information Engineering

Tatung Institute of Technology
Taipei, Taiwan 10451
Phone: (02) 5925252 EXT. 3484
Email:CLYEH@TWNTTIT.BITNET

Hsi-Jian Lee[1]
Department of Computer Science
and Information Engineering
National Chiao Tung University
Hsinchu, Taiwan 30050
Phone:(035) 712121 EXT. 3735 or 3703
Email:hjlee@HJLEE.CSIE.NCTU.EDU.TW

ABSTRACT

Serial verb constructions (SVCs), a series of VPs juxtaposed without any marker between them, is a specific structure in Chinese, which can not be treated as ordinary VPs. Structural ambiguity is the most serious problem for analyzing SVCs. In this paper, we investigate resolution of structural ambiguities of SVCs as well as the related problem, determinism during the course of parsing. We show that some types of SVCs, such as pivotal constructions, sentential subjects and sentential objects, can be dealt with as ordinary VPs through their lexical representations. In addition, we use a reconstructive phrase structure rule for describing the remaining SVCs, two more separate events and descriptive clauses. This reconstructive rule plays the role of eliminating nondeterminism during the course of parsing. At first, these types of SVCs are temporarily analyzed as an S followed by a VP; then after completion of the right-hand side of this rule, reconstruction rules are consulted to build the actual structure of the SVC sentence. The parsing results of SVC sentences are naturally expressed in conventional head-complement structure in HPSG, without inventing any new structure.

[1] To whom correspondence should be addressed.

I. INTRODUCTION

A Chinese declarative sentence, like an English sentence, is basically composed of an NP and a VP. Generally, a Chinese NP is composed of a head noun and some preceding modifiers;a Chinese VP consists of a head verb, one or two complement NPs and some adjuncts [1-3]. Some kinds of Chinese VPs have different structures. For example, the VP, 有 一本書 很 有趣, in sentence (1) consists of a head verb, 有, a complement NP, 一本書, and another VP, 很 有趣. In the corresponding English sentence, "He has a book which is very interesting", a marker, *which*, is used to denote the beginning of a relative clause. In such manner, it is easy to divide these two VPs because of the marker. However, there is no such marker in Chinese, which will make parsing difficult. In addition to sentence (1), there are other types of sentences having similar structure, as in sentences (2) to (6). In general, these types of sentences contain two or more verb phrases juxtaposed without any marker between them, termed serial verb constructions (SVCs) [2,3].

(1) 他 有 一本書 很 有趣.

He has a book which is very interesting.

(2) 他 去 學校 打 籃球.

He went to school to play basketball.

(3) 我 求 他 代表 我.

I begged him/her to represent me.

(4) 他 有 一本書 我 很 喜歡.

He has a book which I like very much.

(5) 他 說 他 要 去 台北.

He said he want to go to Taipei.

(6) 五個人 坐 一輛摩托車 很 危險.

It is very dangerous that five people ride on a motorcycle.

All of the above sentences have the same form,

(NP) V1 (NP) (NP) V2 (NP),

where the NPs in parenthesis are optional and V1 and V2 represent the first and the second verbs, respectively [2]. These sentences have different syntactic structures because of different types of verbs and relationship between them. Generally there are the following types of SVCs shown in Table I [2]. The syntactic structure in head-complement tree form of each type of SVCs are shown Fig.1. The labels C and H adjacent to arcs in the trees denote complement and head, respectively.

There are two main approaches for analysis of SVCs: one is based on phrase structure rules (PSRs) [4-7] and the other is based on Case Grammar [8-9]. It is difficult to obtain a

uniquely correct result by merely using simple syntactic types because an SVC sentence can be any structure in Table I. Therefore additional information is required to rule out structures not preferred. Subcategorization structure of verbs is a useful clue to guide the construction of an appropriate syntactic structure [4,5]. For example, a pivotal verb, such as 勸 (suggest), subcategorizes an NP and a VP as its direct and indirect objects, respectively; the verb 說(say) needs a saturated sentence as its object.

Table I. Types of Serial Verb Constructions.

Types	Descriptions	Examples
(i)	Two more separate events	他 上樓 睡覺 (He went upstairs to sleep). 他 拿 一雙筷子 吃飯 (He uses a pair of chopsticks to eat rice).
(ii)	Pivotal constructions	我 要 她 代表 我 (I asked her to represent me). 我 勸 他 學 醫 (I suggested him to study medicine).
(iii)	Descriptive clauses	我 有 一個妹妹 喜歡 游泳 (I have a sister who likes to swim). 他 有 一本書 很 有趣 (He has a book which is very interesting).
(iv)	Sentential subjects	五個人 坐 一輛摩托車 很 危險 (It is very dangerous that five people ride a motorcycle). 機器 翻譯 一個 句子 要 五分鐘 (It needs five minutes for machine to translate a sentence).
(v)	Sentential objects	他 說 妳 很 漂亮 (He said you are very beautiful). 他 否認 他 做錯 了 (He denied that he was wrong).

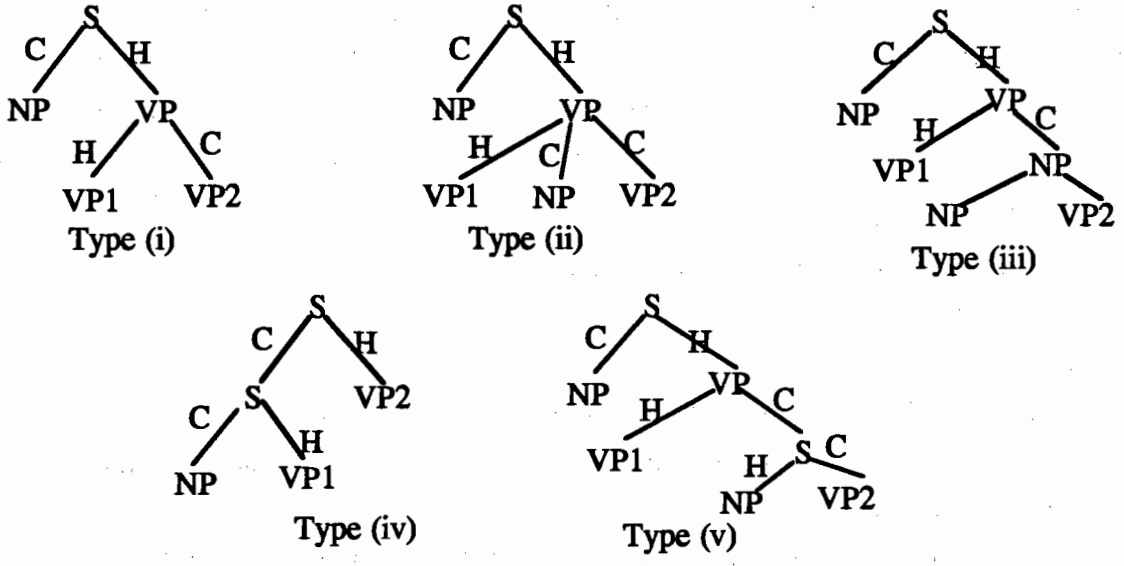

Fig. 1 Types of syntactic structures of SVCs in head-complement tree form.

Classification of verbs based on their meaning is another effective method to help determine the types of SVCs. Yang [8] divided verbs into seven groups according to their semantic categories: causative, emotional, possessive, narrative, special-1, special-2, and normal. Chang [4,5] divided stative verbs into three classes: (1) NP-statives which describe the properties of individuals; (2) VP-statives which modify the verb phrases; and (3) S-statives whose subjects are propositions. Pun [9], based on the theory of Case Grammar, nominated fourteen verb classes, where each one has different basic slots.

Preference rules make up the deficiency of the above methods to choose a preferred structure. Chang [4,5] used a preference rule that argument readings are preferred adjunct readings. That is, pivotal constructions and sentential objects are preferred over descriptive clauses, two more separate events, and sentential subjects. If there are alternative reading survived, Chang applies the last rule to choose a preferred structure in the order of descriptive clauses, two separate events, and sentential subjects.

From the above discussions, subcategorization structures and classification of verbs are essential means for structural disambiguation in SVCs even if they are treated in different manners. In this paper, we investigate the analysis of SVCs in our HPSG-based parser by utilizing subcategorization information and classification of verbs. The former works on SVC analysis focused on the resolution of structural ambiguities[4-7]. We will focus on determinism of parsing in addition to structural disambiguation in SVCs.

Our parser is basically a unification-based, lexicon-driven left-corner parser [10]. Certain types of SVCs can be analyzed by relying on the specific subcategorization structures of the verbs [3]. The lexical entries of verbs in remaining types of SVCs have the same subcategorization frame as ordinary verbs [11]. Classification of verbs is thus used to disambiguate these types of SVCs. There are exceptions in descriptive clauses which can not be disambiguated according to verb classification, as in sentences (7) and (8) [2].

(7) 他 做 了 一道菜 我 很 喜歡.

He cooked a dish which I like very much.

(8) 我們 種 那種菜 吃.

We raised that kind of vegetable to eat.

In these types of SVCs, the second VPs are either an object-missing sentence, S/NP[Obj] or an object-missing verb phrase, VP/NP[Obj]. Both of these structural features can be used to guide the constructions of SVCs.

Our unification-based left-corner parser scans the input sentences from left to right. In such manner, each the prefixes of SVC sentences will form a subconstituent which is itself a saturated sentence. The remaining of the sentence will form a phrasal verb, which plays the

decisive role to build up the syntactic structure of an SVC sentence. We thus do not give a PSR for each kind of SVCs, which will result in nondeterminism. Instead, we use a general PSR with loose conditions on the subconstituents in the rule. The PSR only confines that the subconstituents is a saturated sentence followed by a phrasal verb and, temporarily, it does not state how the target constituent is configured. Then, after the second verb phrase is actually constructed, a set of reconstruction rules which defines the relationships between the subconstituents in SVCs and how the complete structure is constructed is consulted to build up the actual structure of the SVC sentence.

In the following section, structural ambiguities and their resolution in SVCs are demonstrated by examples. In Section III, we show a method to eliminate nondeterminism in the course of parsing SVC sentences. In Section IV, we show the implementation of SVC analysis in our HSPG parser. Finally, concluding remarks are made in Section V.

II. STRUCTURAL AMBIGUITIES AND THEIR RESOLUTION

In this section, we show the situations of structural ambiguities in SVCs and their resolution. According to the structures of SVCs shown in Fig. 1, SVCs can be described by the partial set of PSRs shown in Fig. 2.

PSRs	Descriptions
VP –> VP VP	Two more separate events.
VP –> VP NP VP	Pivotal constructions.
VP –> VP NP1 NP1 –> NP VP	Descriptive clauses.
VP –> VP S	Sentential objects.
S –> S VP	Sentential subjects.
VP –> V VP –> V NP	VP-forming rules

Fig. 2 Partial set of PSRs for SVCs.

Based on the above PSRs, an SVC sentence such as 我 勸 他 學 醫 (I suggested him to study medicine) can be analyzed as any one of the structures in Fig. 1, which results in ambiguity. However, there is only one is correct among these structures. We first employ the subcategorization structure of the main verb 勸. Since the pivotal verb, 勸, subcategorizes an NP as its first object and a VP as the second object. Thus, by this information, a structure of Type (ii) is selected. Similarly, the main verbs in SVCs of sentential subjects and sentential objects have their specific subcategorization; thus these types of SVCs can also be identified by subcategorization information. Table II is a summary of the specific subcategorizations.

Other types of SVCs, descriptive clauses and two more separate events, do not have specific subcategorization for disambiguation. A method based on classification of verbs can be useful. If the class of verbs in the second VPs of these two types of SVCs can be distinguished, then the ambiguity of these two types of SVCs can be removed. Chang's classification is effective for this purpose [4,5]. In Chang's classification, (1) NP-statives describe the properties of individuals, such as 聰明 (clever); (2) VP-statives modify verb phrases, such as 專心 (engrossed); and (3) S-statives whose subjects are propositions, such as 危險 (dangerous). Preference rules of SVCs indicate that the second VPs in SVCs of descriptive clauses, two separate events and sentential subjects are NP-statives, VP-statives and S-statives, respectively. In the sentence, 他 上樓 睡覺, the second VP is a VP-stative in Chang's classification and the first verb,上樓, does not subcategorize a VP object; therefore, it is identified as an SVC of two separate events. The second VP in sentence (1) is used to modified an NP which is NP-stative; thus a structure of descriptive clauses is established. The classification of verbs is finer in the Case Grammar approach [8,9], which will not be discussed here.

Table II. Subcategorization structures of some verbs of SVCs.

Types	Subjects	Object1	Object2
Pivotal	NP	NP	VP
Sentential subjects	S	NP	null
Sentential objects	NP	S	null

There are still cases in descriptive clauses which can not be identified by the above methods, such as sentences in (7) and (8). These sentences have structural evidence in the second VPs. The second VP is an object-missing sentence or S/NP[Obj] in sentence (7), and an object-missing VP, or VP/NP[Obj] in sentence (8). Thus the parser can analyze these sentences by taking advantage of these structural evidence.

III. ELIMINATION OF NONDETERMINISM IN PARSING SVC SENTENCES

Efforts of parsing SVCs mostly focus on resolving structural ambiguities [4-7]. The related problem concerning nondeterminism during the course of parsing are not addressed at all. In the following, we investigate the elimination of nondeterminism, which will further promote the efficiency of parsing SVCs.

In the bottom-up parsing, based on the partial set of PSRs listed in Fig. 2, nondeterminism occurs when a VP subconstituent is constructed because all VP rules of SVCs in Fig. 2 are candidates in the next rule activation. From the observation of syntactic structures shown in Fig. 1, the leaves in all SVC sentences are of the same linear form: an S followed by a VP, each of which is shown in shaded areas in Fig. 3.

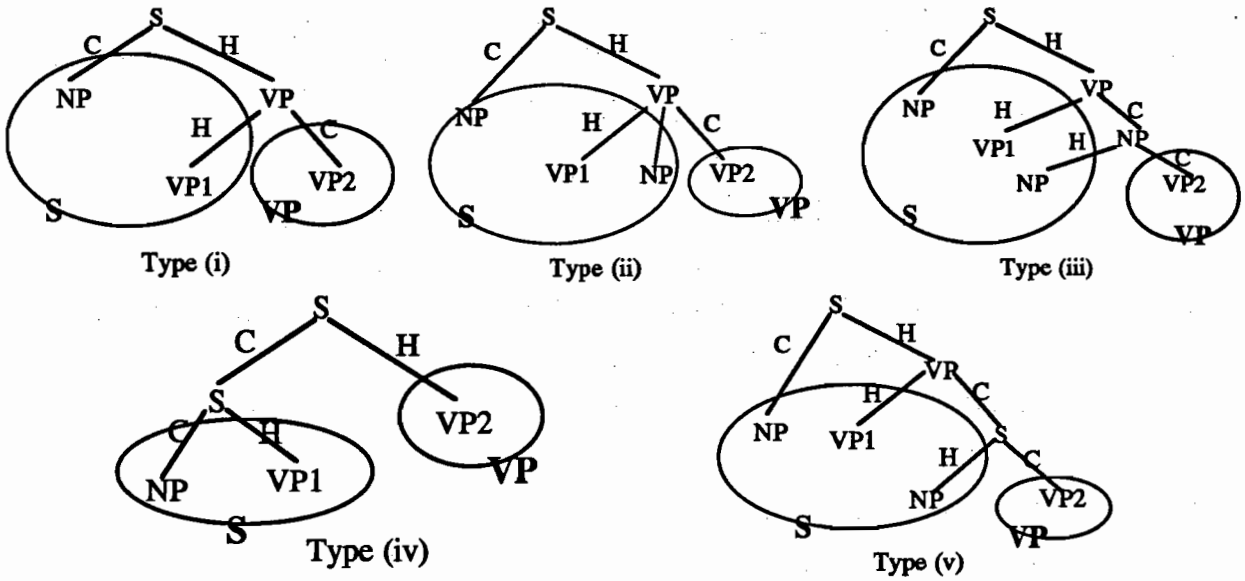

Fig. 3 All types of syntactic structures of SVCs redrawn in S-VP form.

Thus if we use a PS rule, S_SVC --> S VP, to describe SVC sentences, then nondeterminism just described can be eliminated. All SVC sentences are temporarily treated as a sentence composed of an S and a VP. Then after the right-hand side of this S_SVC rule is completed, a set of reconstruction rules are needed to reorganize the subconstituents in S and VP to establish the actual structure of the SVC sentence. In fact, the reconstruction rules play the role of structural disambiguation of SVCs.

According to the discussion in the previous section, we adopt subcategorization information of verbs to identify pivotal construction, sentential subjects and sentential object. We use the verb classification of Chang [4,5] to distinguish SVCs of two more separate events and descriptive clauses. In the following, we show in order the reconstruction rules for these two classes of SVCs in Table III. Note that S and VP in the table are the right-hand side of the S rule, S --> S VP, and S/NP[Obj] and VP/NP[Obj] denote an object-missing sentence and an object-missing verb phrase, respectively.

Table III. Reconstruction rules for SVCs.

Rule #	Descriptions	Conditions	Actions
1	For pivotal constructions	head_verb(S)=pivotal	Construct a Type (ii) structure.
2	For sentential subject	subj(head_verb(S))= sentence obj(head_verb(S))=NP	Construct a Type (iv) structure.
3	For Sentential object	subj(head_verb(S))=NP obj(head_verb(S))=sentence	Construct a Type (v) structure.
4	For two separate events	head_verb(VP)=VP-stative	Construct a Type (i) structure.

5	For descriptive clauses	head_verb(VP)=NP-stative	Construct a Type (iii) structure.
6	For descriptive clauses	VP=S/NP[Obj]	Construct a Type (iii) structure.
7	For descriptive clauses	VP=VP/NP[Obj]	Construct a Type (iii) structure.

IV. IMPLEMENTATION AND RESULTS

In this section, we show the implementation of the S-rule and reconstruction rules for analyzing SVCs in our HPSG parser without altering the existing structure of the parser. In addition, we will show a prominent merit of our parser in dealing with SVCs; that is, pivotal constructions and sentential objects can be treated as ordinary sentences by our lexicon-driven method.

A. Overview of an HPSG parser

HPSG (Head-driven Phrase Structure Grammar) is a lexicon-driven grammar formalism [12]. It reduces PSRs of its predecessor, GPSG [13], by enriching the content of lexical entries. An HPSG consists of a list of universal principles, lexical entries, and language-specific grammar rules. The most often used universal principles are the head feature principle (HFP), the subcategorization principle (SP), and the adjuncts principle (AP). The HFP declares that a phrase shares the same head features with its head daughter; the SP states that in any phrase, each complement daughter must be unifiable with a member of the head daughter's subcat-list, a list of subcategorization specification that remain to be satisfied; and the AP states that any adjunct daughter must be unifiable with some member of the head daughter's adjuncts specification.

The structure of our HPSG can be depicted schematically as shown in Fig. 3, where each component is described in the following.

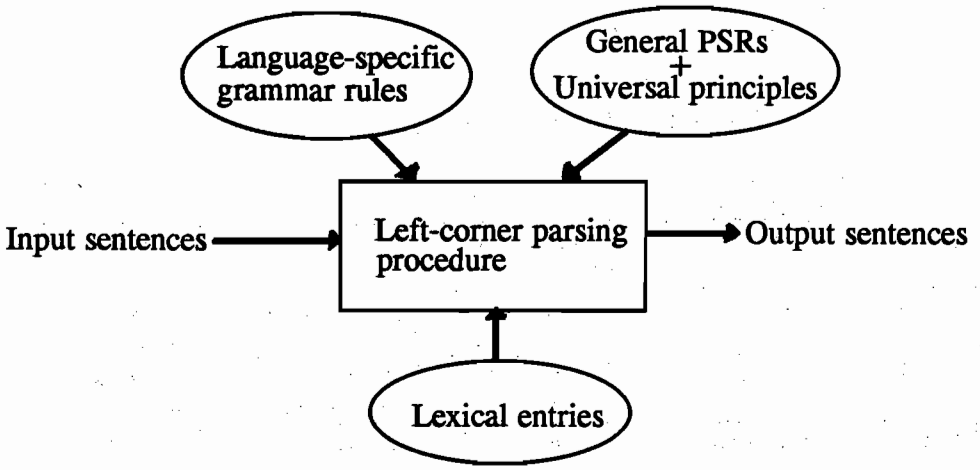

Fig. 4 Schematic diagram of the HPSG parser.

The HPSG rules translated in PATR-II formalism [14], implemented in Prolog, are shown in Fig. 4, where each one actually is a rule-of-rule. That is, a translated HPSG rule is the sole argument of the functor *rule_sel* (rule selection), which is selected according to the conditions listed in the body of the corresponding *rule_sel*. The body of a *rule_sel* plays the role of deterministic selection of an HPSG rule.

```
% Head pre-complement structures.
rule_sel( (R ule X ---> [X1, X2]:-
        X1:syn:loc  ===  X2:syn:loc:subcat:first:syn:loc,  %
    SP
        X2:syn:loc:subcat:rest === end, % SP
        X:syn:loc:subcat === X2:syn:loc:subcat:rest, % Sp
        X:syn:loc:head === X2:syn:loc:head, % HFP.
        X:syn:loc:lex === '-',
        X:head_dtr === X2,
        X:cmp_dtr === X1)):-
    current_dag(C),
    rem_sen([Next_word|_]),N ord Next_word,
    (subcat_feature(C);
     C isa_vp, N isa_nominalization_particle).
```

```
% Head post-complement structures.
rule_sel((R ule X ---> [X1,X2]:-
        X1:syn:loc:subcat:first:syn === X2:syn, % SP
        X:syn:loc:subcat === X1:syn:loc:subcat:rest, % SP
        X:syn:loc:head === X1:syn:loc:head, % HFP,
        X:syn:loc:lex === '-',
        X:head_dtr === X1,
        X:cmp_dtr === X2)):-
    current_dag(C),
    (C isa_preposition;
     C isa_vp2). % C is a VP with subcat length >1.
```

```
% A lexical X is changed to a phrasal X.
rul_sel( (R ule X ---> [X1]:-
        X:syn:loc:subcat === X1:syn:loc:subcat, % SP
        X:syn:loc:head === X1:syn:loc:head, % HFP.
        X:syn:loc:lex === '-',
        X:head_dtr === X1)):-
    current_dag(C),
    (C isa_lexical_v;
     C isa_lexical_n;
     C isa_genitive_marker;
     C isa_nominalization_particle).
```

```
% Head pre-adjunct structures.
rule_sel((R ule X ---> [X1,X2]:-
        X:syn:loc:subcat === X2:syn:loc:subcat, % SP
        X:syn:loc:head === X2:syn:loc:head, % HFP
        X2:syn:loc:head:adjuncts === X1:syn:loc:head,
        X:syn:loc:lex === '-',
        X:head_dtr === X1,
        X:syn:loc:adj_dtr_type === pre,
        X:adj_dtr === X1)):-
    current_dag(C),
    C isa_pre_adjunct_category.
```

```
% Head post-adjunct structures.
rule_sel((R ule X ---> [X1,X2]:-
        X:syn:loc:subcat === X1:syn:loc:subcat, % SP
        X:syn:loc:head === X1:syn:loc:head, % HFP
        X1:syn:loc:head:adjuncts === X2:syn:loc:head,
        X:syn:loc:lex === '-',
        X:head_dtr === X1,
        X:syn:loc:adj_dtr_type === post,
        X:adj_dtr === X2)):-
    current_dag(C),
    C isa_phrase,
    (rem_sen([Next_word|_]);rem_sen([Next_word])),
    N ord Next_word,
    N isa_post_adjunct_category.
```

Note:

(i) The symbols X, X1 and X2 are variables denoting the feature structures of constituents.

(ii) The symbols concatenated by colons are path names of the corresponding feature structure.

(iii) The identity symbol, ===, represents the destructional unification.

Fig. 4: HPSG rules translated into PATR-II form.

The HPSG lexical entry, expressed in feature structure form, is shown below.

(12)
```
phon:
syn:    loc:    head:
                subcat:
                lex:
        bind
```

Lexical entries for Chinese verb 賣 (sell) and noun 西瓜 (water melon) represented in PATR-II form are shown below.

(13)W ord '賣':-

W:phon === '賣',

W:syn:loc:head:maj === v,

W:syn:loc:subcat:first:syn:loc:head:maj === n,

W:syn:loc:subcat:rest:first:syn:loc:head:maj === n,

W:syn:loc:subcat:rest:rest:first:syn:loc:head:maj===n,

W:syn:loc:subcat:rest:rest:rest === end.

(14)W ord '西瓜':-

W:phon === '西瓜',

W:syn:loc:head:maj === n,

W:syn:loc:head:type === common,

W:syn:loc:head:hier === fruit.

Note that the *adjuncts* features of lexical entries are inserted by using a meta-lexicon procedure, which adds the *adjuncts* feature to the corresponding lexical entries automatically.

The parsing procedure is a left-corner one [14], as shown below.

(15)
```
recognize(Dag1,B,C) :- leaf(Dag0,B,E), left_corner(Dag0,Dag1,E,C).
left_corner(Dag1,Dag2,C,C):-unify(Dag1,Dag2).
left_corner(Dag1,Dag2, C,D) :-
        rule_sel((_ ule Dag0 -- [Dag1|Dags]:- XXX)),call(XXX),
        recognize_rest(Dags,C,H),left_corner(Dag0,Dag2,H,D)
leaf(Dag, [Word|C],C):- Dag ord Word.
recognize_rest([],A,A).
recognize_rest([Dag|Dags],C,D):- recognize(Dag,C,E), recognize_rest(Dags,E,D).
```

The first rule, *recognize*, states that a sentence is recognized as category Dag1 if it proves that a leaf of category Dag0 constitutes a left-corner of Dag1. Detailed descriptions of the rest clauses can be obtained from [10,14].

B. Lexicon-driven parsing of SVCs

SVCs of pivotal constructions and sentential objects can be analyzed as ordinary sentences by using our lexicon-driven parser. The lexicon-driven parser relies heavily on the subcategorization frame of verbs to construct the head-complement structures. The combination of the head and the adjuncts depends on the *adjuncts* features of the head. To deal with the verb of pivotal constructions, 勸 (suggest), for instance, a lexical entry is given below.

(16) W ord '勸' :-

W:phon === '勸',

W:syn:loc:head:maj === v,

W:syn:loc:lex === '+',

W:syn:loc:subcat:first:syn:loc:head === n, % The first obj. is an NP.

W:syn:loc:subcat:first:syn:loc:lex === '-',

W:syn:loc:subcat:rest:first:syn:loc:head:maj === v, % The second obj. is a VP (a

W:syn:loc:subcat:rest:first:syn:loc:lex === '-', % subject-missing sentence).

W:syn:loc:subcat:rest:first:syn:loc:subcat:rest === end,

W:syn:loc:subcat:rest:rest:first:syn:loc:head:maj === n, % The subj. is an NP.

W:syn:loc:subcat:rest:rest:first:syn:loc:lex === '-',

W:syn:loc:subcat:rest:rest:rest === end. % End marker of subcat frame.

For the case of sentential objects, 說 (say), for example, a lexical entry is shown in (17).

(17) W ord '說':-

W:phon === '說',

W:syn:loc:head:maj === v,

W:syn:loc:lex === '+',

W:syn:loc:subcat:first:syn:loc:head:maj === v, % The obj. is a saturated sentence.

W:syn:loc:subcat:first:syn:loc:subcat === end,

W:syn:loc:subcat:rest:first:syn:loc:head:maj === n, % The subj. is an NP.

W:syn:loc:subcat:rest:first:syn:loc:lex === '-',

W:syn:loc:subcat:rest:rest === end. % End of subcat frame.

In the following steps, we show how the pivotal construction sentence, 我 勸 他 學 醫 (I suggested him to study medicine), is constructed according to the lexical entry of the verb 勸 and the HPSG rules in Fig. 4. In the following demonstration, for convenience, we only list the main part procedure of rule invocation.

Step 1: Current Word: 我.

The NP 我 first activates the *head pre-complement rule,* and require a VP (S/NP[subj]) to form a complete sentence.

Step 2: Current Word: 勸,

The verb 勸 activates the *head post-complement rule*, and an NP (the first object of 勸) is required by the rule to form a larger VP which in turn leaves over a VP to form a saturated VP.

Step 3: Current Word: 他.

The NP 他 unifies successfully the remaining part in Step 2. Again this resulting VP activates the *head post-complement rule*, and a VP (the second object of 勸) is required to form a complete VP.

Step 4: Current Word: 學.

The verb 學, a transitive verb, activates the *head post-complement rule*, and an NP (the object of) is required.

Step 5: Current Word: 醫.

The NP, 醫, satisfies the remaining part in Step 4; thus a complete VP, 學 醫, is constructed, which in turn unifies successfully the remaining part of Step 3. A complete pivotal construction is thus constructed.

C. The S-rule for SVCs and reconstruction of SVC structures

In Section III, we use an S –> S VP rule to describe some SVC sentences in order to remove nondeterminism on PSRs selection. Then a set of reconstruction rules are applied on the right-hand side of this S-rule to reorganize the real structure of the SVC sentence. In the HPSG parser, this S-rule is thus expressed as the following form.

(18)
```
rule_sel((R ule X ---> [X1,X2]:-
                    X1:syn:loc:head:maj ===v, % A sentence.
                    X1:syn:loc:subcat === end,
                    X1:syn:loc:lex === '-',
                    X2:syn:loc:head:maj === v, % A phrasal verb.
                    X2:syn:loc:lex === '-',
                    assert(svc(X,X1,X2)) )):-
          current_dag(C), C isa_saturated_sentence.
```

Note that in this rule we only confine loose restrictions for X1 and X2, i.e. the S and the VP in the right-hand side, respectively. At the end of the list of identities, an *svc(X1,X2,X3)* is asserted into the database, which is used for latter reconstruction. The reconstruction rules are represented as follows.

(19)
```
reconstruction(Dag0,Dag1,Dag2):-
        retract(svc(Dag0,Dag1,Dag2)),
        (
% Case 1. Two more separate events. E.g., [他]i 唱歌 ei 跳舞.
        Dag2:syn:loc:head:type === vp_stative,
```

```
        Dag2:syn:loc:subcat:rest === end, % S/NP[Subj]
        Dag2:syn:loc:subcat:first:syn:loc:head === % Dag2's null NP ===
          Dag1:cmp_dtr:syn:loc:head,          % Dag1's subject.
        Dag0:syn:loc:head === Dag1:syn:loc:head, % HFP
        Dag0:syn:loc:subcat === Dag1:syn:loc:subcat, % SP
        Dag0:cmp_dtr === Dag1:cmp_dtr,
        Dag0:head_dtr:head_dtr === Dag1:head_dtr,
        Dag0:head_dtr:head_dtr === Dag1:head_dtr,
        Dag0:head_dtr:cmp_dtr === Dag2;
% Case 2. Descriptive clauses. E.g., 他 有 [一本書]i ei 很 漂亮.
        Dag2:syn:loc:head:type === np_stative,
        Dag2:syn:loc:subcat:rest === end, % S/NP[Subj]
        Dag2:syn:loc:subcat:first:syn:loc:head === % Dag2's null NP ===
          Dag1:head_dtr:cmp_dtr:syn:loc:head,   % Dag1's object.
        Dag0:syn:loc:head === Dag1:syn:loc:head, % HFP
        Dag0:syn:loc:subcat === Dag1:syn:loc:subcat, % SP
        Dag0:cmp_dtr === Dag1:cmp_dtr,
        Dag0:head_dtr:head_dtr === Dag1:head_dtr,
        Dag0:head_dtr:cmp_dtr === Dag2;
% Case 3. Descriptive clauses. E.g., 他 有 [一本書]i 我 很 喜歡 ei.
        Dag2:syn:loc:head:maj === v,
        Dag2:syn:loc:subcat === end,
        Dag2:head_dtr:cmp_dtr:syn:loc:head:maj === n ,   % S[Obj:null]
        Dag2:head_dtr:cmp_dtr:syn:loc:null === '+', % S[Obj:null]
        Dag2:head_dtr:cmp_dtr:syn:loc:head ===      % X2's null NP ===
          Dag1:head_dtr:cmp_dtr:syn:loc:head, % X1's object.
        Dag0:syn:loc:head === Dag1:syn:loc:head, % HFP
        Dag0:syn:loc:subcat === Dag1:syn:loc:subcat, % SP
        Dag0:cmp_dtr === Dag1:cmp_dtr,
        Dag0:head_dtr:head_dtr === Dag1:head_dtr,
        Dag0:head_dtr:cmp_dtr === Dag2;
% Case 4. Descriptive clauses. E.g., [我們]i 種 [菜]j ei 吃 ej.
        Dag2:cmp_dtr:syn:loc:head:maj === n ,   % VP[Obj:null]
        Dag2:cmp_dtr:syn:loc:head:type === null , % VP[Obj:null]
        Dag2:cmp_dtr:syn:loc:head ===       % X2's null NP ===
          Dag1:head_dtr:cmp_dtr:syn:loc:head, % X1's object.
        Dag2:syn:loc:subcat:first:syn:loc:head === % X2's subject ===
          Dag1:head_dtr:cmp_dtr:syn:loc:head,   % X1's subject.
        Dag0:syn:loc:head === Dag1:syn:loc:head, % HFP
        Dag0:syn:loc:subcat === Dag1:syn:loc:subcat, % SP
        Dag0:cmp_dtr === Dag1:cmp_dtr,
        Dag0:head_dtr:head_dtr === Dag1:head_dtr,
        Dag0:head_dtr:cmp_dtr === Dag2;
% Case 5. Sentential subjects. E.g., 五 個 人 坐 一 輛 摩拖車 很 危險.
        Dag2:syn:loc:head:type === s_stative,
        Dag2:syn:loc:subcat:rest === end, % S/NP[Subj]
        Dag2:syn:loc:subcat:first:syn:loc:head:maj === v, % X2 subcats an S.
        Dag2:syn:loc:subcat:first:syn:loc:subcat === end, %
        Dag0:syn:loc:head === Dag2:syn:loc:head, % HFP
        Dag0:syn:loc:subcat === Dag2:syn:loc:subcat, % SP
        Dag0:cmp_dtr === Dag1,
        Dag0:head_dtr === Dag2).
```

Recall that *reconstruction* is activated after the right-hand side of S –> S VP rule is completed. Thus the activation of *reconstruction* is inserted after the *recognize_rest* procedure in the seconds *left_corner* rule, which becomes the following.

(20)
```
left_corner(Dag1,Dag2, C,D) :-
        rule_sel((_ ule Dag0 -- [Dag1|Dags]:- XXX)),call(XXX),
        recognize_rest(Dags,C,H),
        case([svc(_,_,_) -> reconstruction(Dag0,Dag1,Dag2)]),
        left_corner(Dag0,Dag2,H,D)
```

In the following, we show, in brief steps, the process of parsing a sample sentence, 他 有 一本書 很 有趣.

Step 1: Current word: 他.
The NP headed by this noun activates the head-pre-complement rule and a VP is required to form a complete sentence.

Step 2: Current word: 有.
This verb is first changed to phrasal verb which in turn activates the head-post-complement rule. An NP is required to form a VP headed by 有.

Step 3: Current words: 一 本 書.
An NP headed by 書 is formed, which satisfies the requirement of Step 2. A sentence is thus established; the S --> S VP rule is activated according to this sentence. To complete this S-rule, a VP is required.

Step 4: Current word: 很.
This adverb activates the head-pre-adjunct rule and a VP is required.

Step 5: Current word: 有趣.
This is an intransitive verb. It is first changed into a phrasal verb, which meets the requirement of Step 5. A VP, 很 有趣, is formed, and then it satisfies the remaining VP in Step 3. Before the completion of the S-rule a *svc(X0,X1,X2)* is asserted into the database.

Step 6: After the completion of the right-hand side of the S-rule, i.e., execution of *recognize_rest* in *left_corner*, a check on *svc(_,_,_)* is true, which activates *reconstruction* to establish the actual SVC structure. In this situation, Case (ii) in *reconstruction* is fired.

D. Sample results of parsing SVC sentence

In the following we show the parsing results of two sample sentences, in abbreviated feature structure form.

SVC Type: Two more separate events.

Sentence: 他 讀書 很 專心.

```
syn:  loc:  head: ...
            subcat: ...
cmp_dtr:  syn:  loc:  head: ...
                      lex:-
          head_dtr:  phon:他
                     syn:  loc:  head: ...
                                 lex:+
                           bind:-
head_dtr:  head_dtr:  syn:  loc:  subcat: ...
                            head: ...
                            lex:-
                      head_dtr:  phon:讀書
                                 syn:  loc:  head: ...
                                             subcat: ...
                                             lex:+
                      cmp_dtr:  syn:  loc:  head: ...
           cmp_dtr:  syn:  loc:  head: ...
                                 subcat: ...
                                 lex:-
                                 adj_dtr_type:pre
                           lex:-
                      head_dtr:  syn:  loc:  head: ...
                                             subcat: ...
                                             lex:-
                                 head_dtr:  phon:專心
                                            syn:  loc:  head: ...
                                                        subcat: ...
                                                        lex:+
                      adj_dtr:  phon:很
                                syn:  loc:  head: ...
                                            lex:+
                                      bind:-
```

SVC Type: Descriptive clauses.

Sentence: 他 有 一 本 書 我 很 喜歡.

```
syn:  loc:  head: ...
            subcat: ...
cmp_dtr:  syn:  loc:  head: ...
                      lex:-
          head_dtr:  phon:他
                     syn:  loc:  head: ...
                                 lex:+
                           bind:-
head_dtr:  head_dtr:  syn:  loc:  subcat: ...
                            head: ...
                            lex:-
                      head_dtr:  syn:  loc:  subcat: ...
                                             head: ...
                                             lex:-
                                 head_dtr:  phon:有
```

syn: loc: head: ...
subcat: ...
lex:+
cmp_dtr: syn: loc: head: ...
subcat: ...
lex:-
adj_dtr_type:pre
head_dtr: syn: loc: head: ...
subcat: ...
lex:-
adj_dtr_type:pre
head_dtr: syn: loc: head: ...
subcat: ...
lex:-
head_dtr: phon:書
syn: loc: head: ...
lex:+
bind:-
adj_dtr: phon:本
syn: loc: head: ...
subcat: ...
lex:+
bind:-
adj_dtr: phon:一
syn: loc: head: ...
lex:+
bind:-
cmp_dtr: syn: loc: head: ...
subcat: ...
lex:-
lex:-
head_dtr: syn: loc: subcat: ...
head: ...
lex:-
head_dtr: syn: loc: head: ...
subcat: ...
lex:-
adj_dtr_type:pre
head_dtr: syn: loc: head: ...
subcat: ...
lex:-
head_dtr: phon:喜歡
syn: loc: head: ...
subcat: ...
lex:+
adj_dtr: phon:很
syn: loc: head: ...
lex:+
bind:-
cmp_dtr: syn: loc: head: ...
lex:-
null:+
bind:-

```
cmp_dtr:   syn:   loc:   head: ...
                        lex:-
           head_dtr:   phon:我
                       syn:   loc:   head: ...
                                     lex:+
                                   bind:-
```

V. CONCLUDING REMARKS

We have shown a reconstructive method for parsing Chinese SVC sentences by using a lexicon-driven parser. In this work, only one phrase structure rule is added into the parser for SVCs. The lexicon-driven mechanism shows promising in processing SVCs of pivotal constructions, sentential subjects and sentential objects for the head verbs in these types of SVCs have their specific subcategorization. They are handled as ordinary VPs by using the existing parser without adding any phrase structure rule. The only phrase structure rule for SVCs is used to describe SVCs of descriptive clauses and two more separate events. Nondeterminism during the course of parsing SVCs does not occur because there is only one phrase structure rule for SVCs. The phrase structure rule is attached with a set of reconstruction rule which is used to build the actual structures of SVCs and resolve structural ambiguities. We have tested every type of SVCs in our parser, and performance is similar as processing ordinary declarative sentences. The resulting structures fit the conventional HPSG format, so that it can be treated as ordinary declarative sentences in latter phases, such as semantic interpretation, structural transfer in MT, etc. At present, the parser performs well for every SVCs consisting of two VPs. However, there are still further work for long SVCs. Consider the SVCs containing more than two VPs, as in sentences (21) and (22).

(21) 他 去 學校 找 同學 打 籃球.

He went to school to find classmates to play basketball.

(22) 我 有 一個 妹妹 喜歡 去 學校 找 同學 打 籃球.

I have a sister who likes to go to school to find classmates to play basketball.

At present, the reconstructive approach for SVCs can not parse these sentences. By observations, the troublesome VP series in these sentences are mostly in conjunctive structures. Consequently, the cases of long series of VPs will be handled in our further work on analysis of Chinese sentence linking.

REFERENCES

[1] Y. R. Chao, *A Grammar of Spoken Chinese,* University of California Press, Berkeley, 1968.

[2] C. N. Li and S. Thompson, *Mandarin Chinese: a Functional Reference Grammar*, University of California Press, Berkeley, 1981.

[3] S. Lu, *800 Modern Chinese Phrases*, The Commercial Press, Hong Kong, 1984 (in Chinese).

[4] C. H. Chang and G. K. Krulee, "Predication ambiguity in Chinese and its resolution," *Proc. of ICCPCOL'91*, Taiwan, 1991, pp. 109-114.

[5] C. H. Chang, *Resolving Ambiguities in Mandarin Chinese: Implication for Machine Translation*, Ph.D. Thesis, Northwestern University, Evanston, Illinois, 1991.

[6] M. S. Sun, "Resolving ambiguities in Chinese parsing," *Proc. of ICCPCOL'91*, Taiwan, 1991, pp. 121-126.

[7] L. S. Lee, L. F. Chien, L. L. Lin, J. Huang, and K. J. Chen, "An efficient natural language processing system specially designed for the Chinese language," *Computational Linguistics*, Vol. 17, No. 4, 1991, pp. 347-374.

[8] Y. Yang, *Studies on an Analysis System for Chinese Sentences*, Ph.D. Thesis, Kyoto University, Japan, 1985.

[9] K. H. Pun, "Analysis of serial verb constructions in Chinese," *Proc. of ICCPCOL'91*, Taiwan, 1991, pp. 170-175.

[10] C. L. Yeh and H. J. Lee, "An HPSG parser implemented on compilation of unification-based grammar formalism," *Proc. of Natural Language Processing Pacific Rim Symposium*, Singapore, 1991, pp. 1-8.

[11] L. Zheng, "Silent subjects in Chinese descriptive clauses," *Lingua*, pp. 341-349, 1991.

[12] C. Pollard and I. Sag,*Information-based Syntax and Semantics: Vol. I, Fundamentals*, CSLI Lecture Notes, No.13, 1987.

[13] P. Sells, *Lectures on Contemporary Syntactic Theories*, CSLI Lecture Notes, No. 3, Chicago University Press, Chicago, 1985

[14] G. Gazdar and C. Mellish, *Natural Language Processing in Prolog*, Addison-Wesley, 1989.

REDUPLICATION IN MANDARIN CHINESE:

THEIR FORMATION RULES, SYNTACTIC BEHAVIOR AND ICG REPRESENTATION

Feng-yi Chen[*], *Ruo-ping Jean Mo*[*], *Chu-Ren Huang*[**], *Keh-Jiann Chen*[*]

**The Institute of Information Science, Academia Sinica*
***The Institute of History and Philology, Academia Sinica*
Nankang, Taipei, Taiwan,
Republic of China

ABSTRACT

Morphologically derived words can neither be identified by dictionary look-up nor be accounted for with a syntactic parser in NLP. Mandarin Chinese involves several productive morphological rules. This paper proposes a set of rules to identify reduplicatives in Mandarin Chinese. This set of rules will be used to complement dictionary look-up and DM generation rules in the word segmentation module. The co-occurrence restriction of adjuncts in reduplication is also discussed and expressed in ICG mechanism to improve parsing results.

I. Introduction

Mandarin Chinese reduplicatives are constructed by repeating the whole or part of a lexical item. Verbal reduplicatives may denote delimitative as well as tentative aspects(嘗試貌) or intensifying meaning. For instance, both verbs *chang* 'to sing' and *pingan* 'to be safe and secure' can be reduplicated, as shown in (1).

(1) a. tamen pingshr shihuan *chang chang ge*
they usually like sing sing song
'They usually like to sing a little bit'

b. shiauhai dou *ping ping an an*
children all flat flat peaceful peaceful
'Children are all very safe and secure'

In addition to verbs, onomatopoeia, measure words, and morphological derived determinative-measure compounds can also undergo the process of reduplication. Because of its high productivity and its being fed by another morphological rules, exhaustively listing reduplicatives in the lexicon is not a viable alternative.

The current version of CKIP word segmentation system [3] is based on a lexicon of about ninety thousand words and a set of determinative-measure rules [7]. Without rules to account for reduplicatives, not all correct word breaks can be found. Example extracted from machine-readable Chinese corpus is given in (2)[1].

(2) a. juang man le *yi dai dai* de pingguo
fill full LE one bag bag DE apple
'(The container etc.) is filled with bags of apples.'

---

1. The following examples presented in this paper are mainly adopted from this machine-readable corpus whose texts are mostly from a Chinese newspaper, *Tz You Shr Bau* 'Liberty Times' from October 1990 to February 1991, which contains 10 million words or so.

b. yijr tzai gung shiu shrchang jung *jang jang die die*
always in supply demand market center rise rise fall fall
'(The price) has always been rising and falling alternatively in the market.'

In order to solve the above-mentioned problem, this paper proposes a set of reduplicative-formation rules to build reduplicatives within the word segmentation module. We will first discuss the scope and types of reduplicatives in Mandarin Chinese, and their syntactic behavior and semantic variations.

II. Reduplicatives - their scope and types

Morphologically, reduplicatives are formed by total or partial repetition of a lexical item. However, not every lexical item containing partial repetition is regarded as a reduplicative in this paper. This is because reduplicatives are handled in terms of morphological rules in this paper. And some of the reduplicated types do not follow these requirements: they have limited productivity and show idiosyncratic grammatical behavior, which mean their forms are not predictable by general rules (cf. 3a, 3b, 3c, and 3d).

(3) a. the reduplication of adverbs

chang chang
often often
'often'

jin jin
only only
'only'

b. the reduplication of nouns

en en yuan yuan
favor favor hatred hatred
'gratitude and grudge'

shr shr wu wu
thing thing object object
'things and objects'

c. the xlixy reduplicatives

luo li luosuo
chatter inside wordy
'verbose or wordy'

tu li tuchi
earth inside rustic
'rustic'

d. the xyy reduplicatives

liu you you
green oil oil
'bright green'

shiau ha ha
laugh Ha Ha
'laugh heartily'

leng bing bing
cold ice ice
'icy'

shie lin lin
blood drench drench
'bloody'

In (3c), for example, only those adjectives with pejorative meaning, such as *tuchi* 'rustic' and *luosuo* 'wordy', can undergo this type of reduplication. Since their set is quite small, we will simply list them all in the lexicon. As for the xyy type, the meaning as well as the reduplicated yy form are lexically determined by the head x, which may be a noun, verb or adjective [1]&[4]. Try to compare the examples below.

(4) a. *hei* *chi chi*
black paint paint
'very dark'

*a'. *bai* *chi chi*
white paint paint

b. *ching piau piau*
light float float
'lightly'

*b'. *jung* *piau piau*
heavy float float

From example (4), we may observe that the reduplicated yy types, *chi chi* and *piau piau* must co-occur with *hei* 'black' and *ching* 'light', respectively. Using regular expressions to construct these reduplicatives is not feasible because there is no context-free constraint to rule out the non-existing forms. Therefore, they will still be stored in the lexicon.

To sum up, only those reduplicatives of high productivity and predictability are formed by rules and are included in this paper. Moreover, the categories that can have reduplication are limited to verbs, determinative-measure compounds, measures and onomatopoeia.[2] In what follows, a detailed discussion of various reduplicated forms and a set of formation rules will be offered according to different meaning properties.

2.1. Reduplication to Express Tentative Aspect

Generally speaking, the process of reduplication may add a sense of tentativeness to any action verbs which contain no modifier-head internal structure and have no meaning contradicted to the semantic function of reduplication, such as controllable verbs. Lastly, a reduplicatable action verb

2. Among these four categories that can undergo reduplication, except for the verb's, the forms of the other three are rather simple. Hence, in the paper, we will only list their formation rules without further explanation. As for the semantic functions of these reduplicatives, a sense of vividness is imposed if the original form is an onomatopoeia; otherwise, 'each' or the way of measuring is added.

cannot be an achievement verb. Take *chujia* 'become a monk' for example. Although this verb is both active and controllable, it cannot be reduplicated because it is an achievement verb. This verb is an achievement verb because the goal of becoming a monk is attained at the endpoint of the denoted action. In terms of lexical semantics, a tentative aspect stipulates that an act be broken down and carried out in a piecemeal fashion. The instantaneity of an achievement verb contradicts this interpretation. This is how our rule excludes *chujia* 'become a monk'. Additional functional types as shown in the following sections.

2.1.1. XX Type[3]

If the input is a monosyllabic action verb, some other words like *yi* 'one', or *le* 'PERFECTIVE' may be inserted between the two Xs; but if the original form is a disyllabic one, then no word may occur in between. Examples are shown in (5).

(5) a. *shiou (yi) shiou* ye hua de shiangwei
smell (one) smell wild flower DE fragrance
'try and smell the fragrance of wild flowers a little'

b. *jengli jengli* shuguei
arrange arrange bookcase
'arrange the bookcase a little'

Though there is a little difference in forms and number of syllables, only one rule is proposed.

RD1 --> X ({*yi, le*}) X
conditions: (1) X = VA, VB, VC, VD, VE, VF
(Action Verb)

3. The capitalized X used here means that X may be a syllable or a word and x, y or z just represents a syllable.

(2) IF X is monosyllabic
THEN {*yi, le*} is allowed

However, from the data collected, we find that some disyllabic stative verbs may also take this XX form. Closer investigation shows that their reduplicated counterparts are more active-like. The interpretation of the reduplicated (6c) is derived from the active reading of *Kelian* "to pity" in (6b), not the stative reading in (6a). Thus the generalization that only action verbs have tentative reduplicative forms is correct.

(6) a. ta shiangdang *kelian*
he quite pitiful
'He is quite pitiful.'

b. jingfang shiangdang *kelian* dueifang
police quite pitiful other-side
'The police pity the other party very much.'

c. *kelian kelian* women
pitiful pitiful we
'Just pity us a little bit!'

2.1.2. xxy(z) Type

Only VO compounds are allowed to take this reduplicated type. After reduplication, the result also has an additional meaning of 'doing a little bit', just like the tentative aspect reduplication of XX.

(7) a. ching ta lai *ping ping li*
please he come judge judge reason
'Just ask him to come (and try) to make a judgement.'

b. shiuang minjung lai *kai kai yanjie*
hope populace come open open view
'(We) wish the people will come and have their perspectives (somewhat) widened.'

These VO compounds may be two or three syllables, so the reduplicated type may either xxy or xxyz. And the formation rule is expressed in the following.

RD5 --> x ({*le, yi*}) xy(z)
conditions: xy(z) = VA13, VA3, VA4, VB

2.2. Reduplication to Express Vividness

The reduplicated process will impose a sense of vividness to stative verbs, or to intensify the attributes described by them. The major types of reduplicated stative verbs are presented below.

2.2.1. xx Type

Monosyllabic stative verbs can have this reduplication form. However, unlike monosyllabic action verbs, only *de* 'DE' or *di* '-ly' are allowed to follow it, as in (8). To distinguish reduplicated stative from action verbs, RD2 is proposed.

(8) dutz *kung kung (de)*
belly empty empty
'The belly is empty.'

RD2 --> xx ({*de, di*})
conditions: x =VH (Stative Verb)

2.2.2. xxyy Type

In general, only disyllabic stative verbs may undergo this kind of reduplication, and except for few which take sentential complements, most of

them fall under the category of intransitives.

(9) a. dajia dou hen *kaishin*
everybody all very happy
'Everybody is very happy.'

a'. dajia dou *kai kai shin shin*
everybody all ha- ha- ppy- ppy
'Everybody is very happy.'

b. ta hen *gaushing* toutzren dou neng dacheng gungshr
he very glad investor all able attain concerns
'He is very glad that investors can reach concerns.'

b'. rang dajia dou neng *gau gau shing shing* (de)
let everybody all able ha- ha- ppy- ppy
'Let everybody be very happy.'

The rule to form this type of reduplication is expressed in RD7.

RD7 --> xxyy
conditions: xy = VH11, VH21, DH[+onomatopoeic]

But, after examining more linguistic data, we discovered that some of the action verbs, whether intransitive or not, can also have this xxyy reduplicated type. They spread sporadically over the whole active set.

(10) a. yin juntz *jin jin chu chu* diauyu chang
addiction gentleman enter enter out out fish field
'(Addicted) smokers frequent fishing arenas.'

b. *yau yau huang huang* bu chu fangjian
shake shake wobble wobble walk out room
'(S/he) walked out the room unsteadily.'

The semantic function of this reduplicated type also differs with respect to the input: if the original word is stative, then the morphological process will make

it sound more vivid. Besides, the attributes it describes may also be intensified. But to those active input, only a sense of "doing a little bit' is added.

2.3. Reduplication of Onomatopoeic and Measure Words

Onomatopoeic words and measure words can also have XX reduplicated form. They are accounted for with RD4 and RD10, respectively. The Kleene stars '*' in rules means that onomatopoeic or measure words can be repeated twice or more to form reduplication. RD4 can be applied to both monosyllabic and disyllabic onomatopoeia.

RD4 --> DH* ({*de, di*})
conditions: DH (manner adverb) which is specified with the feature [+onomatopoeic]

RD10 --> RNOP1 ((*you*)) RNOP1)*

RNOP1 --> (IN1)(DESC) M[4]

It deserves mentioning that disyllabic onomatopoeia can undergo reduplication of xxyy type. The input to xxyy-type onomatopoeic reduplication has already been accounted for in RD7.

III. The Syntactic Constraints of Reduplication

Unlike typical morphological processes, Mandarin Chinese reduplication does not change either the argument structure or the category. This rule changes the semantics and some minor syntactic behavior, such as the allowed adjuncts and syntactic patterns. This section concerns with these reduplicated constructions and their representation in the Information-based

4. According to Mo et al. [7] 'IN1' in this rule represents numeral compounds and 'DESC' are descriptive words, such as *da* 'big', or *shiau* 'small'.

Case Grammar (ICG), which is proposed by Chen and Huang [2].

3.1. Distribution of Reduplication

3.1.1. Reduplicated Stative Verbs

Though both reduplicated and unreduplicated forms can function as main predicates and manner adverbs, the usage of reduplicated stative verbs is more restricted than their counterparts. Words after being reduplicated can neither co-occur with degree adverbs nor appear in the construction of comparison. This suggests that reduplication assigns the semantic feature of [-SCALE] because non-scaler predicates can neither occur in a comparative construction nor be modified by a degree adjunct [5]. In other word, reduplication turns a scaler predicate into an absolute predicate.

(11) a. yanjing *shiau shiau* de
eye small small DE
'(His/her) eyes are very small.'

*a'. yanjing feichang *shiau shiau* de
eye very small small DE

b. tamen fuchi liang yishiang *ping ping shuen shuen* de
they couple both always flat flat smooth smooth DE
'They have always been going smoothly as a couple.'

*b'. tamen fuchi liang yishiang hen *ping ping shuen shuen* de
they couple both always very flat flat smooth smooth DE

c. wo *gau* ta san gungfen
I tall he three centimeter
'I am three centimeter taller than he'

*c'. wo *gau gau* ta san gungfen
I tall tall he three centimeter

3.1.2. Reduplicated Action Verbs

Though the categories and argument structures of action verbs after reduplication remain the same, there are more syntactic limitations on reduplicated forms. We observe that these reduplicatives are mutually exclusive with the adjuncts of frequency, quantifier, duration, and postverbal location. Moreover, they are incompatible with the aspects, *le* 'PERFECTIVE', *je* 'DURATIVE', and *guo* 'EXPERIENTIAL' as well as the *bei* construction. As described in section 2.1, reduplication of action verbs denotes the tentative and delimitative aspects which mean some actions are regarded as a piecemeal fashion internally, so any expressions to signal the instantaneous completion of an action, such as the occurrence of aspect, *le* 'PERFECTIVE' and *guo* 'EXPERIENTIAL' certainly violate the semantic function of reduplication. Accordingly, a postverbal locative phrase which refers to the place where an action is achieved is also forbidden. Again, the incompatibility between *bei* construction and reduplication is because *bei* constructions interpret the event in its totality, contrary to the internal event-structure of the tentative aspect[5]. Beside the contradiction of semantic effects, the exclusion of some expressions may be because of the redundancy in meaning. For example, the durative aspect, *je* 'DURATIVE', is suggested by Li and Thompson [6] to express "ongoing, or durative nature of an event' which has been conveyed by reduplicated forms. In addition, tentative reduplicated constructions may also pertain the quantitative meaning of an action which the adjuncts of frequency and quantifier tend to express. It is not necessary to contain two or more expressions which are the same. Therefore, reduplicated forms do not occur with the durative aspect, adjuncts of duration, frequency and quantifier. These restrictions of co-occurrence will be illustrated by the following examples.

---

5. Li and Thompson [6] stipulates that a *bei* sentence "describes an event in which an entity or person is dealt with, handled, or manipulated in some way." This definition entails that the event is considered as a whole.

(12) a. ching *tzuo* yihuei (duration)
please sit a while
'Please sit for a while'

*a'. ching *tzuo tzuo* yihuei
please sit sit a while

b. *tzuo* tzai shrtou shang (postverbal location)
sit in stone above
'Sit on the stone'

*b'. *tzuo tzuo* tzai shrtou shang
sit sit in stone above

c. *tzou* yi bian (frequency)
walk one time
'Walk once!'

*c'. *tzou tzou* yi bian
walk walk one time

d. *tauluen* guo je ge wenti
discuss EXP this CL question
'This question has been discussed'

*d'. *tauluen tauluen* guo je ge wenti
discuss discuss EXP this CL question

3.2. Representation of Reduplication in ICG

Following the above discussion of grammatical feature of reduplication, this section proposes ICG representations of reduplication for efficient parsing. The representation includes syntactic information, such as category, basic patterns (BP) and adjunct precedence (AP), and semantic information, such as semantic features. According to the observation described in section 3.1.1 and 3.1.2, the co-occurrence restriction of adjuncts will be denoted with AP, as in (13).

(13) a. adjunct restriction in Stative verb

syn : constraints: *AP : A1 : v[+rd] > < {degree, comparison}*

b. adjunct restriction in Action Verb

syn: constraints: *AP: A2: v[+rd] > < {complement[ASP], frequency, quantifier, duration, agent[{PP[bei],P[bei]}]}*

As for the other information, (a) once recognized as reduplicated constructions, reduplicatives will inherit the same syntactic categories as their original forms, even though they are not stored in the lexicon; (b) reduplicated stative verbs will acquire the feature [+vivid] and reduplicated action verbs, [+tentative]; (c) the feature [+rd] will be specified, while reduplication is identified. Once the [+rd] is specified, A1 in stative verb and A2 in action verb will be applied during the process of parsing language. For further explanation, the stative verb, *kuaile* 'happy' and the action verb, *da* 'hit' and their reduplicated forms are taken as examples.

(14) a. *kuaile*

sem : meaning : happy
feature : +manner
adjuncts : ...

syn : class : VH21
features :
constraints: form
BP B1 : experiencer < * ;

a'. *kuai kuai le le*

sem : meaning : be very happy
feature : +manner, +*vivid*
adjuncts : ...

syn : class : VH21
features : +*rd*
constraints: form
BP B1 : experiencer < * ;

AP A1 : v[+rd] >< {degree, comparison}

b. *da*

sem : meaning : hit
feature :
adjuncts : ...

syn : class : VC2
features :
constraints: form
BP B1: agent[{NP,PP[you]}] < * < goal[NP];
B2: agent[{NP,PP[you]}] < goal[PP] <*;
B3: goal[NP] < agent[{PP,P[bei]}] < *;

b'. *da (yi) da*

sem : meaning : hit a little
feature : +*tentative*
adjuncts : ...

syn : class : VC2
features : +*rd*
constraints : form
BP B1: agent[{NP,PP[you]}]<*< goal[NP];
B2: agent[{NP,PP[you]}]< goal[PP]<*;

AP A2: v[+rd] >< {complement[ASP],
frequency, quantifier, duration,
agent[{PP[bei],P[bei]}]}

From the above examples, both reduplicated expressions are specified by the feature [+rd] in the syntactic feature. And, try to compare (14b) and (14b'): according to the co-occurrence restriction of adjuncts, action verbs with the feature [+rd] are incompatible with *bei* construction, thus example (14b') does not contain the third basic pattern.

IV. Concluding Remarks

In this paper, we present not only the scope and types of reduplicatives but also a set of formation rules to enhance our word segmentation module. Based on the syntactic constraints of reduplicated constructions, we express the co-occurrence restriction of adjuncts in ICG to help parsing. In addition to context-free rules as in the formation of determinative-measure compounds, context-sensitive rules to construct reduplication are required. However, these context-sensitive rules for reduplicatives are now implemented by context-free rule augmented with conditional checks and do not pose any problem for parsing efficiency.

REFERENCES

[1] Chao, Yuen Ren. 1968. *A Grammar of Spoken Chinese*. Berkeley: University of California Press.

[2] Chen, Keh-Jiann and Chu-Ren Huang. 1990. "Information-based Case Grammar." *Proceedings of COLING' 90*. pp.54-59.

[3] Chen, Keh-Jiann and Liu Shing-Huan. 1992. "Word Identification for Mandarin Chinese Sentence." *Proceedings of COLING' 92*. Vol.1. pp.101-107. Nantes, France.

[4] Huang, Chu-Ren. 1992. "Adjectival Reduplication in Southern Min: A Study of Morpholexical Rules with Syntactic Effects." In *Chinese Languages and Linguistics. Vol.1. Chinese Dialects*. 407-422. Academia Sinica, Taipei.

[5] Huang, Chu-Ren and Wei-Mei Hong. 1992. "Coordination and Dependency: A Study of Mandarin Comparative *bi*." Paper presented in *the First International Conference on Chinese Linguistics (ICCL I)*, Singapore. June 24-26, 1992.

[6] Li, Charles N. and Sandra A. Thompson. 1981. *Mandarin Chinese: A Functional Reference Grammar*. Berkeley: University of California Press.

[7] Mo, Ruo-ping Jean, Yao-Jung Yang, Keh-Jiann Chen and Chu-Ren Huang. 1991. "Determinative-Measure Compounds in Mandarin Chinese." *Proceedings of ROCLING IV*. pp.111-134.

A PARALLEL AUGMENTED CONTEXT-FREE PARSING SYSTEM FOR NATURAL LANGUAGE ANALYSIS

Hsin-Hsi Chen Jiunn-Liang Leu Yue-Shi Lee

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.
hh_chen@nlg.csie.ntu.edu.tw

Abstract

Parsing efficiency is one of the important issues in building practical natural language processing systems. This paper proposes a design and an implementation of a parallel augmented context-free parsing system for natural language analysis. Natural language grammars are more than context-free, so that unification formalisms are adopted to enforce the linguistic constraints and to transfer the linguistic information. Lexical and structural ambiguities are the famous problems in parsing natural language sentences. Traditional LR approaches to deal with these problems are pseudo parallelism or blind parallelism. They fork many processes to take care of parsing. Apparently, it results in the scheduling problem in shared-memory model or the communication problem in distributed-memory model. This paper presents a merge mechanism to compose the same jobs into one. It can not only eliminate the duplications, but also reduce the number of forked processes to the great extent. The gapping problems are also treated in this parallel parsing system. Currently, it is implemented in Prolog and in Strand, and running on Sun-series workstations.

1. Introduction

There is a growing interest in applying parallel computation techniques to natural language processing (NLP) [1-2]. Two approaches may be adopted: massively parallel systems and non-massively parallel systems [3]. These papers [4-8] show the typical examples for the former systems. They try to map natural language grammars into connectionist networks. Because the functionality of the nodes in the network is very primitive, they are involved in the following problems: (1) Is the network independent of the length of input sentence? (2) Does the network accept recursive grammar rules? (3) What threshold values and weights are assigned to nodes and links in the network? On the other side, these papers [9-15] deal with the design of non-massively parallel parsing systems. Most of the papers touch on parallelizing CYK Parsing algorithm, LR Parsing algorithm, Chart Parsing algorithm, *etc.*, however, only few presented

methods to capture the specific linguistic phenomena. To evaluate these types of parallel parsing systems, besides the performance criteria, i.e., scheduling of the processes in the shared-memory model [16] or the communication cost among processes in the distributed-memory model [17], the expressive capability is also an important issue. This paper will propose a parallel augmented context-free parsing system for natural language analysis. The linguistic phenomena are considered in depth in our design, gapping problem in particular. From the comparisons among different parsing strategies [18], Tomita's extended LR parser [19] is a better selection in computational linguistics. This paper will also follow the concept to design the parallel parsing system.

2. LR Parsing

Shift and reduce are two basic operations in LR parsing. LR parser uses two tables (Action and Goto tables) and one stack to control the parsing procedure. The Action table shows when to shift, to reduce, to terminate successfully, or to signal a syntactic error. The Goto table defines the next state after a nonterminal is matched and shifted. The stack contains a sequence of parse states. The following is a sample grammar:

(1) S --> NP VP
(2) S --> S PP
(3) NP --> n
(4) NP --> det n
(5) NP --> NP PP
(6) VP --> t1 NP
(7) VP --> t2 S
(8) VP --> iv
(9) PP --> prep NP

Table 1 shows its corresponding Action and Goto tables.

Table 1. The Parsing Table for the Sample Grammar

	det	n	t1	t2	iv	prep	$	S	NP	VP	PP
0	s2	s1						3	4		
1			r3	r3	r3	r3	r3				
2		s5									
3						s6	acc				7
4			s10	s11	s12	s6				13	9
5			r4	r4	r4	r4	r4				
6	s2	s1							8		
7						r2	r2				
8			r9	r9	r9	s6/r9	r9				9
9			r5	r5	r5	r5	r5				
10	s2	s1							14		
11	s2	s1						15	4		
12						r8	r8				
13						r1	r1				
14						s6/r6	r6				9
15						s6/r7	r7				7

Table 2 demonstrates the parsing steps for the sentence "I saw her duck".

Table 2. The Detailed Steps for Parsing the Sentence "I saw her duck"

step	stack	comment	input string
1	[_,0]	initial state	{n}{t1,t2}{n,det}{n,iv}$
2	[_,0] [n,1]	action(0,n)=s1	{t1,t2}{n,det}{n,iv}$
3.1	[_,0] [NP,4] [t1,10]	action(1,t1)=r3, goto(0,NP)=4 action(4,t1)=s10	{n,det}{n,iv}$
3.2	[_,0] [NP,4] [t2,11]	action(1,t2)=r3, goto(0,NP)=4 action(4,t2)=s11	{n,det}{n,iv}$
4.1	[_,0] [NP,4] [t1,10] [n,1]	action(10,n)=s1	{n,iv}$
4.2	[_,0] [NP,4] [t1,10] [det,2]	action(10,det)=s2	{n,iv}$
4.3	[_,0] [NP,4] [t2,11] [n,1]	action(11,n)=s1	{n,iv}$
4.4	[_,0] [NP,4] [t2,11] [det,2]	action(11,det)=s2	{n,iv}$
5.1	[_,0] [NP,4] [t1,10] [n,1]	action(1,n)=fail	
5.2	[_,0] [NP,4] [t1,10] [NP,14]	action(1,iv)=r3, goto(10,NP)=14 action(14,iv)=fail	
5.3	[_,0] [NP,4] [t1,10] [det,2] [n,5]	action(2,n)=s5	$
5.4	[_,0] [NP,4] [t1,10] [det,2]	action(2,iv)=fail	
5.5	[_,0] [NP,4] [t2,11] [n,1]	action(1,n)=fail	
5.6	[_,0] [NP,4] [t2,11] [NP,4] [iv,12]	action(1,iv)=r3, goto(11,NP)=4 action(4,iv)=s12	$
5.7	[_,0] [NP,4] [t2,11] [det,2] [n,5]	action(2,n)=s5	$
5.8	[_,0] [NP,4] [t2,11] [det,2]	action(2,det)=fail	
6.1	[_,0] [NP,4] [t1,10] [NP,14]	action(5,$)=r4, goto(10,NP)=14	$
6.2	[_,0] [NP,4] [t2,11] [NP,4] [VP,13]	action(4,$)=r8, goto(4,VP)=13	$
6.3	[_,0] [NP,4] [t2,11] [NP,4]	action(5,$)=r4, goto(11,NP)=4	$
7.1	[_,0] [NP,4] [VP,13]	action(14,$)=r6, goto(4,VP)=13	$
7.2	[_,0] [NP,4] [t2,11] [S,15]	action(13,$)=r1, goto(11,s)=15	$
7.3	[_,0] [NP,4] [t2,11] [NP,4]	action(4,$)=fail	
8.1	[_,0] [S,3]	action(13,$)=r1, goto(0,S)=3	$
8.2	[_,0] [NP,4] [VP,13]	action(15,$)=r7, goto(4,VP)=13	$
9.1	[_,0] [S,3]	action(3,$)=acc	
9.2	[_,0] [S,3]	action(13,$)=r1, goto(0,S)=3	$
10	[_,0] [S,3]	action(3,$)=acc	

The words "I", "saw", "her" and "duck" have categories {n}, {t1,t2}, {det,n} and {n,iv} respectively. The last three have more than one category. The effect is multiplicative rather than additive. Four stacks are generated at steps (4.1) - (4.4). Besides multi-category problem, the conflict entry in the action table also introduces nondeterminism. For example, parse the sentence "I saw her duck with a telescope". When the preposition "with" is inspected, a conflict entry (shift 6/reduce 6) will be met. The knowledge to resolve PP-attachment problem is originated from diverse resources [20]. Even if the knowledge is encoded, which action is selected correctly must be deferred to the later stage(s). Conventional approach to deal with these problems is pseudo parallelism or blind parallelism. The former explores the alternatives in a special sequence, e.g. breadth-first in our example. The latter forks many processes to take care of the subsequent actions. These processes may spend much time doing the same jobs. That decreases the significance of the parallelism. Synchronizing by shift operation [12] or data availability [13] was proposed to avoid the duplications. They tried to merge the stacks generated by different processes into tree-structured stacks (TSSs). Subsequently, Tanaka and Suresh [21] took another view on the interpretation of the elements in the pushdown stacks. These elements are called *dot reverse items* (*drits*). A *drit* is a dotted rule [A -> X1 X2 ... Xk • X$k+1$ ... Xm, i], which is similar to the Earley's item. However, its meaning is reverse. In Earley parsing [22], we plan to construct a sequence of item lists, I1, I2, ..., In such that a dotted rule [A -> α • β, i] $\in$ Ij iff S $\Rightarrow^*$ γ A δ, γ $\Rightarrow^*$ ω1 ω2 ... ωi, and α $\Rightarrow^*$ $\omega i+1$ $\omega i+2$ ... ωj. The item *drit* means [A -> α • β, j] $\in$ Ii iff S $\Rightarrow^*$ γ A δ, β $\Rightarrow^*$ $\omega i+1$ $\omega i+2$ ... ωj, and δ $\Rightarrow^*$ $\omega j+1$ $\omega j+2$ ... ωn. A sentence is *recognizable* by a grammar iff [s -> • γ, n] $\in$ I0.

Such an interpretation matches the direction of reduction, so that the merge can be done to the most depth. Consider the following two stacks possessed by two processes respectively. Each element in the stacks has two arguments. The first denotes a set of position numbers and the second is a state.

(a) ••• [{a},S1] [{b},S2] [{c},S3] [{e},S4]

(b) ••• [{a},S1] [{d},S2] [{c},S3] [{e},S4]

If the next action is a "reduce x" where x is "A -> B C D", we will get six *dirt*s shown as follows:

(1) [A -> B C • D, e] $\in$ I_c

(2) [A -> B C • D, e] $\in$ I_c

(3) [A -> B • C D, e] $\in$ I_b

(4) [A -> B • C D, e] $\in$ I_d

(5) [A -> • B C D, e] $\in$ I_a

(6) [A -> • B C D, e] $\in$ I_a

We can observe that (1) and (2), (5) and (6) have the same *drits*, i.e, the two processes do the same jobs. If we merge these two stacks into a TSS with the principle "merging the position numbers of those items with the same state from top of stacks", we can get a TSS like: "••• [{a},S1] [{b,d},S2] [{c},S3] [{e},S4]". Reducing the TSS by the same rule generates the same *drits* as before, however, it avoids the redundancy and decreases the number of processes to the great extent . Thus, the scheme can not only achieve the effects of chart parsing [23], but also is suitable to develop a new parallel parsing model.

3. A Parallel Parsing System

3.1 Parallel Recognizer

The fundamental concept of the recognizer is like the conventional LR algorithm except that the position numbers are used in the stacks. The following describes the basic recognizer:

(1) Initialize the stack to [{0},0], where the first 0 represents the word position and the next 0 denotes the initial state.
(2) Look up the first word of the remaining sentence in the dictionary, and return the feature structure(s)[1] of the word.
(3) Look up LR table by word category and the current state, and return a list of actions.
(4) Perform each action in the list.
    (a) accept: Terminate with success.
    (b) error: Terminate with failure.
    (c) shift: The position number is increased by 1 and go to step (5).
    (d) reduce: Do the reduce operation without changing the position number, and and try step (3) again.
(5) Merge the stacks with the same shift operation.
(6) Consume this word.
(7) If there are words left then go to step (2) else halt.

There are several places to enforce the parallelism:

(1) The table look-up at step (2) can be done in parallel.
(2) The actions in the action list can be performed in parallel.
(3) The merge operation at step (5) can be performed in parallel with the reduce operation at step (4.d).
(4) The new state created by the shift operation can be forwarded beforehand.
(5) The parse tree generation is overlapped with other actions (see next section).

Figure 1 demonstrates the architecture of the new parallel recognizer.

---

1 Unification is adopted. The unification-based formalism refers to [20].

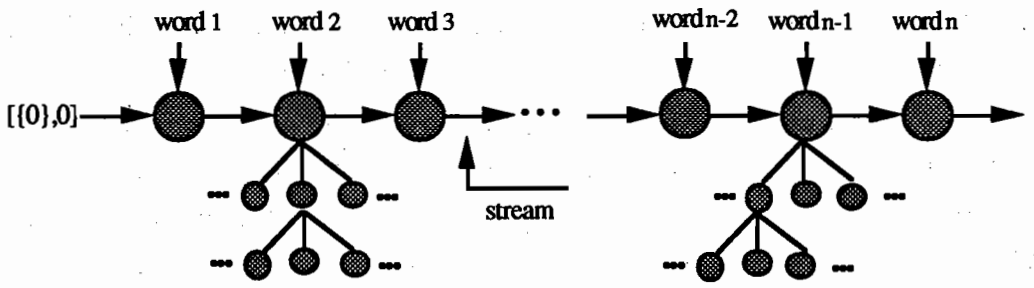

Figure 1. System Architecture of a Parallel Recognizer

A *word* process is initiated for each word to control all of its operations, i.e. shift and/or reduce operations. These processes are linked in a pipeline and communicate with each other by channels. The channel can transfer two kinds of information: merged stacks and a global table (see next section). When a word process receives the information (not necessary complete information) from its left neighbor word process, it begins its parsing steps immediately. Current system is developed by the language Strand88 [24], which is a parallel programming language. It can be run on parallel environments like transputers or be simulated on Unix systems. Strand provides a concise notation to describe process interactions. If a word process receives an incomplete information, it proceeds the parsing as possible as it can until it meets a variable. It is the characteristics of Strand language. The following shows the setup of the pipeline mechanism:

```
pglr(Sentence) :-
    pglr(0,Sentence,[[elt([0],0)]],_).
pglr(Pos,[],_,_).
pglr(Pos,[Word|Words],InStream,OutStream) :-
    dict:word(Word,WordStream),
    Pos1 is Pos + 1,
    goal(Pos1,Word,WordStream,InStream,MidStream),
    pglr(Pos1,Words,MidStream,OutStream).
```

A sequence of TSSs is transmitted from the left hand side to the right hand side via the special communication channel *stream*. The sequence is generated by the *word* process i ($1 \le i \le n$). We adopt the data structure for the stream: [TSS$i1$, TSS$i2$, ..., TSSim]. Each stack TSSij ($1 \le j \le m$) is represented as a list of elements of the form *elt(a list of position numbers,state number)*. Initially, the stream is: *[[elt([0],0)]]*. Because a word process may generate more than one TSS, a merger is used to merge m TSSs into a stream, and send them in sequence to its right neighbor. Figure 2 demonstrates a sample communication channel.

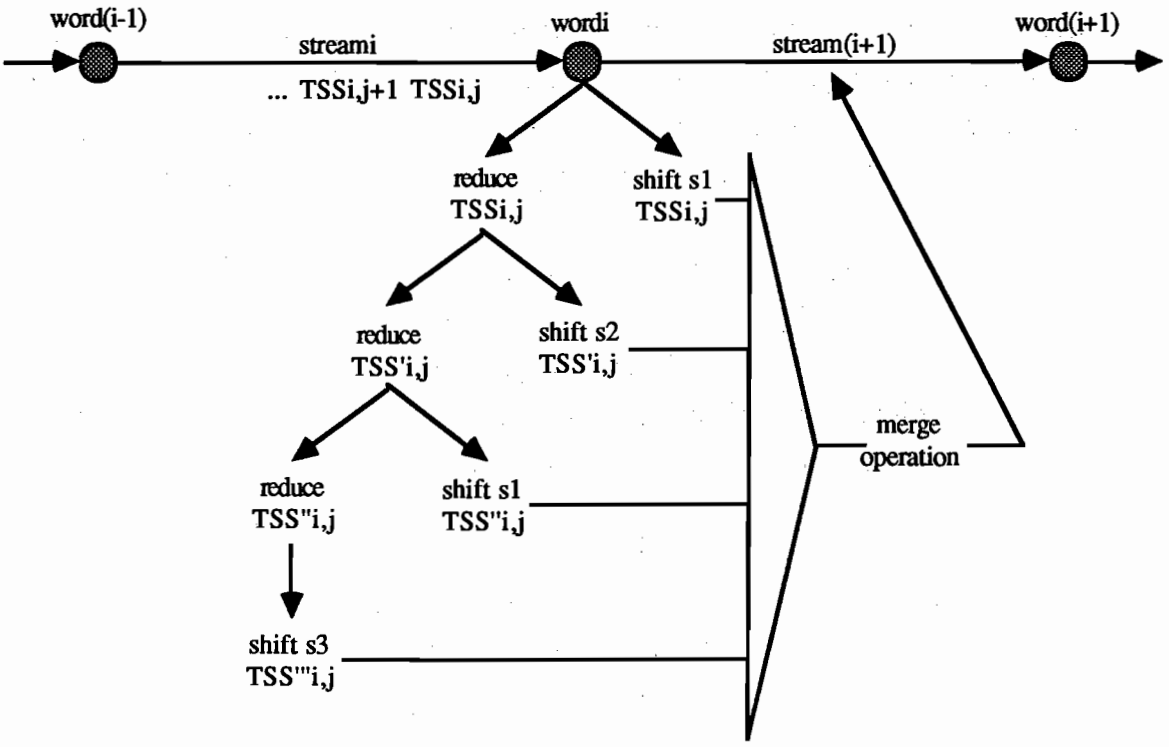

Figure 2. Communication Channel

The following shows the detailed step:

(a) s1: Keep {<s1, stackij, Tail(i+1)1>} and Stream(i+1)1, and
send out [[elt([i],s1)|Tail(i+1)1]|Stream(i+1)1].
That is, Stream(i+1) := [[elt([i],s1)|Tail(i+1)1]|Stream(i+1)1].

(b) s2: Keep {<s1, stackij, Tail(i+1)1>, <s2, stack'ij, Tail(i+1)2>} and Stream(i+1)2 and send out [[elt([i],s2)|Tail(i+1)2]|Stream(i+1)2].
That is, Stream(i+1)1 := [[elt([i],s2)|Tail(i+1)2]|Stream(i+1)2].

(c) s1: Merge stackij and stack"ij into nstack. Keep {<s1, nstack, Tail(i+1)1>, <s2, stack'ij, Tail(i+1)2>} and Stream(i+1)2.

(d) s3: Keep {<s1, nstack, Tail(i+1)1>, <s2, stack'ij, Tail(i+1)2>, <s3, stack'"ij, Tail(i+1)3>} and Stream(i+1)3, and
send out [[elt([i],s3)|Tail(i+1)3]|Stream(i+1)3].
That is, Stream(i+1)2 := [[elt([i],s3)|Tail(i+1)3]|Stream(i+1)3].

If the left stream is exhausted, let

Tail(i+1)1 := nstack, Tail(i+1)2 := stack'ij, Tail(i+1)3 := stack'"ij and
Stream(i+1)3 := [].

Otherwise, do the same job again. Strand language provides a predefined process *merger*. It allows many processes to write on a single stream. This approach has an advantage: the different shift message can be forwarded to next process before the reduce operation is terminated. It results in a better performance. The definition of

goal for a word process is given below. The process *pathval* retrieves the category information from a feature structure. The process *subgoal* transforms the TSSs in *InStream* into a merge list according to the LR parsing table. The list *MergeList* is a communication channel between processes *subgoal* and *dispatcher*. The process *dispatcher* sends the merged TSS into the right hand side neighbor one at a time via the channel *OutStream*.

```
goal(Pos,Word,[H],InStream,OutStream) :-
    pathval(H,[cat],Cat),
    subgoal(Pos,Cat,InStream,MergeList),
    dispatcher(OutStream,MergeList).
goal(Pos,Word,[H|T],InStream,OutStream) :-
    T =\= [] |
        pathval(H,[cat],Cat),
        subgoal(Pos,Cat,InStream,MergeList),
        dispatcher(OutStream1,MergeList),
        goal(Pos,Word,T,InStream,OutStream2),
        merger([merge(OutStream1),merge(OutStream2)],OutStream).
```

In natural languages, a word may have more than one category. This problem can be treated easily in the parallel parsing. Assume a word has N categories. The system can fork N processes and copy TSSs to each process to deal with those N categories. Theory 1 tell us: "Given any two stacks, no matter what states of their top of stacks are, if they receive different categories, they will not shift to the same state." That is, the new stacks cannot be merged. Based on the theory, the TSSs generated by any process can be sent to the next word process immediately without waiting for the generation of other TSSs. This can reduce the merge time. Table 3 lists the TSSs generated by word processes for the sentence "I saw her duck."

Table 3. TSSs Produced by Word Processes for the Sentence "I saw her duck"

node	tree-structured stacks (TSSs)	input string
1	[_,0] [n,1]	{n}
2	[_,0] [NP,4] [t1,10] [_,0] [NP,4] [t2,11]	{t1,t2}
3	[_,0] [NP,4] [t1,10]---[n,1] [_,0] [NP,4] [t2,11]┘ [_,0] [NP,4] [t1,10]---[det,2] [_,0] [NP,4] [t2,11]┘	{n,det}
4	[_,0] [NP,4] [t2,11] [NP,4] [iv,12] [_,0] [NP,4] [t1,10]---[det,2] [n,5] [_,0] [NP,4] [t2,11]┘	{n,iv}

Theorem 1. Given two stacks in a word process, no matter what states of their top of stacks are, they will not shift to the same state if they receive different categories.

Proof:

When the LR parser reads a word, it may execute reduce actions successively. At the end, it will meet a shift action, an accept action or an unacceptable signal. Assume the states of the top of two stacks are S1 and S2 respectively. Two cases are shown as follows.

(1) S1 = S2. It is trivial that the two stacks will not go into the same state.

(2) S1 ≠ S2. Assume the state S1 receives category c1 and the state S2 receives category c2. The configuration set of S1 can be divided into two groups: those configurations have the form "Pn -> Rn • c1 Un" and those do not. The configuration set of S2 can also be divided into "Pm -> Rm • c2 Um" and those do not. According to S1, the next state which receives c1 can be divided into two groups: "Pn -> Rn c1 • Un" and the prediction set of Un. According to S2, the next state which receives c2 can be divided into two groups: "Pm -> Rm c2 • Um" and prediction set of Um. Because {Pn -> Rn c1 • Un} ≠ {Pm -> Rm c2 • U2} and {Pn ->Rn c1 • Un} ≠ the prediction set of Um, the next states after receiving c1 and c2 cannot be the same. ■

3.2 Parsing Tree Generator

The conventional LR parsing algorithm keeps partial parse trees in the stacks. In the current implementation, only position numbers are recorded. This is because the interpretation of *drits* is from right to left, and it avoids the overheads during the merge and split operations. Under such a situation, if there does not exist an efficient parsing tree generation algorithm, the benefits from merge operation are lost. This section presents a parsing tree generator. It is active when any reduce action is performed. It will lookup tables, extract the Rhs of the production rule, apply the unification formulas and produce Lhs. There are two tables used: one is a global table received from the previous word process and the other is a delta table produced from its parent action process. The global table is the union of delta tables produced by the left-hand side word processes. Given a sentence "I saw her duck", Table 4 lists the delta tables generated by the word processes.

Table 4. The Delta Tables Produced by Word Processes for "I saw her duck"

node	word	partial trees produced by the node	global table
1	I	Δ1={1.<n,0,1>}	[]
2	saw	Δ2={2.<t1,1,2>,3.<t2,1,2>,4.<NP(1),0,1>}	Δ1
3	her	Δ3={5.<n,2,3>,6.<det,2,3>}	Δ1UΔ2
4	duck	Δ4={7.<n,3,4>,8.<iv,3,4>,9.<NP(5),2,3>}	Δ1UΔ2UΔ3
5	$	Δ5={10.<NP(6,7),2,4>,11.<VP(8),3,4>,12.<VP(2,10),1,4>, 13.<S(9,11),2,4>,14.<S(4,12),0,4>,15.<VP(3,13),1,4>, 16.<S(4,15),0,4>}	Δ1UΔ2UΔ3UΔ4

Each entry with a unique index has three arguments: the first is a partial tree, and the last two denote the left and the right positions respectively. For the performance issue, all the two tables are sorted and packed. The system uses the merge sort to arrange the partial trees. The right position is regarded as a primary key, and the left is a secondary key. The primary key is in descendant order and the secondary key is in ascendant order. This is the most efficient arrangement. Observe the delta tables in Table 4. The delta table created by each word process has a very interesting feature: "The right positions of all partial trees equal to the node number minus one except the leaf node." This is because the action process performs the reduce action until it meets a shift action, and each reduction will promote a partial tree up one level. Under the arrangement, we just append the global table to the delta table without applying the merge sort to these two tables. There is an important result in sorting the delta table: to keep the table entries unredundant. Because the system records the partial trees by a global table, it cannot distinguish which partial trees were produced by which processes when reduce action occurs. In fact, the distinction is not necessary. If the system cannot manage the tables efficiently, it will take a lot of time to search tables and will also have redundant solutions. Thus, we put an ambiguous forest [19] in the same table item and keep only one copy of subtrees that have the same structure and range. Figure 3 summarizes the flow of table constructions.

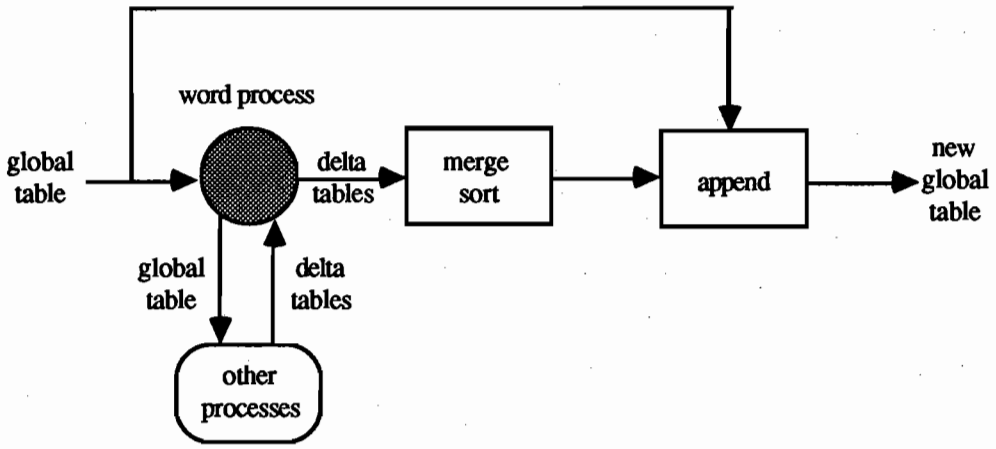

Figure 3. Table Management

4. Resolving Gapping Problem

Gapping is a common phenomenon in natural language sentences. Topicalization and relativization are two famous examples. In the sentence "The apples, I like", the constituent "The apples" is displaced from the object position to the topic position. These phenomena are regarded as movement transformations. To capture them, these papers [25-26] extended the conventional augmented context-free grammar formalism with two extra symbols ">>>" and "<<<" shown as follows:

(1) C --> C1, C2, ..., C*(i-1)*, C*i* <<< trace, C*(i+1)*, ..., C*n*.
This rule can be interpreted as "C is composed of C1, C2, ..., C*i*, ..., C*n*, where C*i* is moved from the position dominated by C*(i+1)*, ..., or C*n* and a trace is left at that position". The position of C*i* is called a *landing site.*

(2) C --> C1, C2, ..., C*(i-1)*, trace >>> C*i* , C*(i+1)*, ..., C*n*.
The interpretation of this rule is similar to the above except that the constituent C*i* is moved from its left hand side.

(3) C --> C1, C2, ..., C*(i-1)*, trace, C*(i+1)*, ..., C*n*.
This rule can be read as "C is composed of C1, C2, ..., C*(i-1)*, C*(i+1)*, ..., C*n*, and an empty constituent is left between C*(i-1)* and C*(i+1)*". The position of trace is called an *empty site.*

Under this grammar formalism, only the landing site and the empty site are specified. It is different from the slash technique in that no explicit slash feature is specified in the grammar. Consider a sample grammar shown below:

```
(1) syn_rule S1Bar --> TOPIC <<< TRACE, S:
        [TOPIC,head] === [TRACE,head].
(2) syn_rule S1Bar --> S.
(3) syn_rule S --> NP, VP:
        [NP,head] === [VP,subj].
(4) syn_rule NP --> *Det, *N:
        [NP,head] === [N,head].
(5) syn_rule NP --> *N:
        [NP,head] === [N,head].
(6) syn_rule VP --> *TV, NP:
        [VP,subj] === [TV,subj],
        [TV,obj] === [NP,head].
(7) syn_rule VP --> *TV, TRACE:
        [VP,subj] === [TV,subj],
        [TV,obj] === [TRACE,head].
```

Rule (1) deals with the topicalization and the others are the normal grammar rules. An empty constituent appears in rule (7).

Figure 4 shows one of the relationships between the displaced constituent and its corresponding empty constituent, which is a leftward movement. Rightward movement is symmetric, so their treatments are the same.

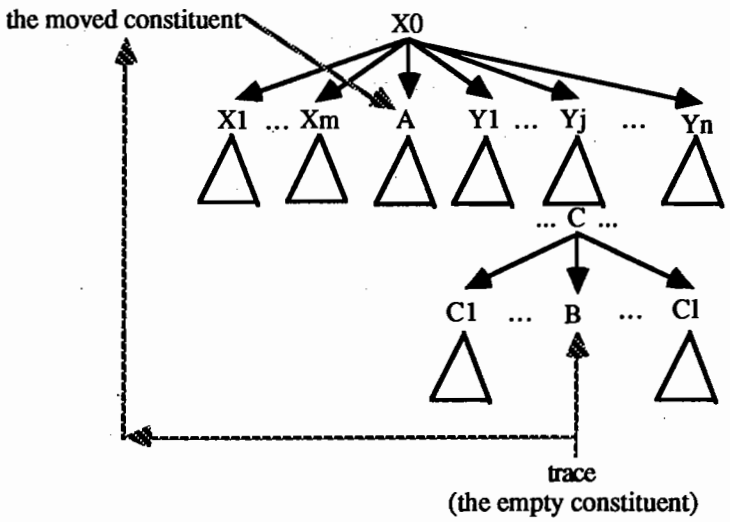

Figure 4. Leftward Movement

Before the grammar is translated, a preprocessing procedure computes the domination path, e.g. Y*j* dominates C and C dominates B. The empty constituent is raised up to the level dominated by X0 via C and Y*j*. For example, the above grammar is preprocessed like:

(1') syn_rule S1Bar --> TOPIC, S:
[TOPIC,head] === [S,trace,head].

(2') syn_rule S1Bar --> S:
[S1Bar,trace] === [S,trace].

(3') syn_rule S --> NP, VP:
[NP,head] === [VP,subj],
[S,trace] === [VP,trace].

(4') syn_rule NP --> *Det, *N:
[NP,head] === [N,head],
[NP,trace] === none.

(5') syn_rule NP --> *N:
[NP,head] === [N,head],
[NP,trace] === none.

(6') syn_rule VP --> *TV, NP:
[VP,subj] === [TV,subj],
[TV,obj] === [NP,head],
[VP,trace] === none.

(7') syn_rule VP --> *TV:
[VP,subj] === [TV,subj],
[TV,obj] === [VP,trace,head].

Rule (7') specifies that all the information about the empty constituent in the original rule is inherited by the mother category, i.e., VP. Because S dominates VP, rule (3') shows this information is also passed to S. Rule (1') depicts that the information is unified to the moved constituent. Rules (4'), (5') and (6') do not dominate any trace category, so the formulas "[FS,trace] === none" are added. The preprocessor automatically generates the *trace* feature for rules. It not only avoids the burden of grammar writing, but also detects the grammar errors beforehand. Finally, consider two general cases for preprocessing.

(a) For a rule C --> C1, C2, ..., C*(i-1)*, C*i* <<< trace, C*(i+1)*, ..., C*n*, if there exists more than one C*k* ((i+1) ≤ k ≤ n) that dominates *trace*, *trace* may be transferred up from different paths. During preprocessing, we split such a rule into several rules with the same Lhs and Rhs, and different sets of unification formulas. Because our parsing system can handle the conflict condition, these rules can be tried in parallel.

(b) For a rule C --> C1, C2, ..., C*n*, if there exists more than one C*k* (1 ≤ k ≤ n) that dominates *trace*, *trace* may be transferred up through the mother category. In this way, trace is considered as a disjunction feature to transfer all the possible information up.

5. Concluding Remarks

This paper proposes a design and an implementation of a parallel parsing system for natural language analysis based on LR parsing algorithm. It adopts dot reverse items instead of the conventional Earley items. This interpretation can not only achieve the same effect as Chart parsing, but also reduce the number of processes to the great extent. An efficient table management algorithm is also presented to construct the parsing trees. A global table for parse tree generation is set up incrementally. It is transferred from the leftmost word process to the rightmost process. Because the delta table generated by an intermediate word process is mutual exclusive of the global table sent from its left hand side word process, and the former is much smaller than the latter, it is easy to keep tables sorted and packed. For the well-treatment of the gapping phenomena, the formalism to specify the landing site and the empty site is introduced. A grammar translator adds disjunctive trace features to unification formulas automatically. Currently, it can capture the relationship of serial binding. The parallel parsing system is implemented with Prolog and with Strand, and running on Sun-series workstations.

Acknowledgements

Research on this paper was partially supported by National Science Council grant NSC-81-0408-E002-514, Taipei, Taiwan, R.O.C.

References

[1] A. Nijholt, "Parallel Parsing Strategies in Natural Language Processing," *Proceedings of International Parsing Workshop on Parsing Technologies*, 1989, pp. 240-253.

[2] R. Akker, H. Alblas, A. Nijholt and P.O. Luttighuis, "An Annotated Bibliography on Parallel Parsing," *Memoranda Informatica 89-67*, Department of Computer Science, University of Twente, the Netherlands, 1989.

[3] H. Schnelle, "Panel Discussion on Parallel Processing in Computational Linguistics," *Proceedings of 12th International Conference on Computational Linguistics*, 1988, pp. 595-598.

[4] D.L. Waltz and J.B. Pollack, "Massively Parallel Parsing: A Strongly Interactive Model of Natural Language Interpretation," *Cognitive Science*, Vol. 9, 1985, pp. 51-74.

[5] T. Li and H.-W. Chun, "A Massively Parallel Network-Based Natural Language Parsing System," *Proceedings of the Second International Conference on Computers and Applications*, 1987, pp. 401-408.

[6] B. Selman and G. Hirst, "Parsing as an Energy Minimization Problem," in *Genetic Algorithms and Simulated Annealing*, Lawrence Davis (Editor), Morgan Kaufmann Publishers, 1987, pp. 141-154.

[7] H. Nakagawa and M. Tatsunori, "A Parser based on Connectionist Model," *Proceedings of 12th International Conference on Computational Linguistics*, 1988, pp. 454-458.

[8] H. Schnelle and R. Wilkens, "The Translation of Constituent Structure Grammars into Connectionist Networks," *Proceedings of 13th International Conference on Computational Linguistics*, 1990, pp. 53-55.

[9] X. Huang and G. Louise, "Parsing in Parallel," *Proceedings of 11th International Conference on Computational Linguistics*, 1986, pp. 140-145.

[10] A. Haas, "Parallel Parsing for Unification Grammars," *Proceedings of International Joint Conference on Artificial Intelligence*, 1987, pp. 615-618.

[11] E.L. Lozinskii and S. Nirenburg, "Parsing in Parallel," *Computer Language*, Vol. 11, No. 1, 1986, pp. 39-51.

[12] H. Tanaka and H. Numazaki, "Parallel Generalized LR Parsing Based on Logic Programming," *Proceedings of International Workshop on Parsing Technologies*, 1989, pp. 329-338.

[13] H. Numazaki and H. Tanaka, "A New Parallel Algorithm for Generalized LR Parsing," *Proceedings of 13th International Conference on Computational Linguistics*, 1990, pp. 305-310.

[14] Y. Matsumoto, "Handling Coordination in a Logic-Based Concurrent Parser," *Natural Language Understanding and Logic Programming*, 1991, pp. 1-12.

[15] A. Kouji, "Parallel Parsing System Based on Dependency Grammar," *Natural Language Understanding and Logic Programming*, 1991, pp. 147-157.

[16] R. Grishman and M. Chitrao, "Evaluation of a Parallel Chart Parser," *Proceedings of the 2nd Conference on Applied Natural Language Processing*, 1988, pp. 71-76.

[17] H.S. Thompson, "Chart Parsing for Loosely Coupled Parallel Systems," *Proceedings of International Workshop on Parsing Technologies*, 1989, pp. 320-328.

[18] P. Shann, "The Selection of a Parsing Strategy for an On-Line Machine Translation System in a Sublanguage Domain," *Proceedings of International Workshop on Parsing Technologies*, 1989, pp. 264-276.

[19] M. Tomita, "An Efficient Augmented-Context-Free Parsing Algorithm," *Computational Linguistics*, Vol. 13, No. 1-2, 1987, pp. 31-46.

[20] K.-H. Chen and H.-H. Chen, "Attachment and Transfer of Prepositional Phrases with Constraint Propagation," Submitted to *Computer Processing of Chinese and Oriental Languages* (first revision).

[21] H. Tanaka and K.G. Suresh, "YAGLR: Yet Another Generalized LR Parser," *Proceedings of ROCLING IV*, Taiwan, 1991, pp. 21-31.

[22] A.V. Aho and J.D. Ullman, *The Theory of Parsing, Translation, and Compiling*, Vol. 1: Parsing, Prentice-Hall, 1973.

[23] M. Kay, "Algorithm Schemata and Data Structures in Syntactic Processing," in *Readings in Natural Language Processing*, B.J. Grosz (Editor), Morgan Kaufmann, 1986, pp. 35-70.

[24] I. Foster and S. Taylor, *Strand: New Concept in Parallel Programming*, Prentice Hall, 1989.

[25] H.-H. Chen, I.-P. Lin and C.-P. Wu, "A New Design of Prolog-Based Bottom-Up Parsing System with Government-Binding Theory," *Proceedings of the 12th International Conference on Computational Linguistics*, 1988, pp. 112-116.

[26] H.-H. Chen, "A Logic-Based Government-Binding Parser for Mandarin Chinese," *Proceedings of the 13th International Conference on Computational Linguistics*, 1990, pp. 48-53.