

THE IDENTIFICATION OF THEMATIC ROLES
IN PARSING MANDARIN CHINESE

*Keh-jiann Chen**, *Chu-Ren Huang***, and *Li-ping Chang****

* Institute of Information Science, Academia Sinica
** Institute of History and Philology, Academia Sinica
***Computing Center, Academia Sinica

ABSTRACT

In parsing, the identification of thematic roles not only is fundamental to semantic interpretations but also can reduce syntactic branches and ambiguities. Since the syntactic structures for natural languages are usually complicated and ambiguous, there is no uniquely deterministic parsing strategy. The paper observes that, in addition to real world knowledge, there are four parameters to identify the roles of constituents in a sentence. They are syntactic categories and semantic features of constituents, case frames and case restrictions of verbs, syntactic configurations and word order, and oblique case assigners. Further, the paper shows how the parameters including syntactic and semantic information are represented in feature structures, and how they function in identifying thematic roles. The thematic roles of constituents can be determined by accumulating parametric information. Thus, it is believed that the unification parsing strategy can make parsing more deterministic and efficient.

1. INTRODUCTION

Identification of the thematic role of each constituent in a sentence is a crucial step in the process of understanding natural languages, for well-identified thematic roles reflect the semantic relations between the constituents. In addition to natural languages understanding, thematic roles also play crucial roles in parsing Mandarin Chinese. In Chinese, for lack of morphological markings of derivational variations, there exist many ambiguities in the tree structure of a sentence [陳 88]. Early identification of thematic roles can usually reduce syntactic

branches and ambiguities. For example, temporal expressions are instantiated by at least five different syntactic categories as shown in (1). They can be either noun phrases(NPs), compounds with Determinatives and Measures (D-M compounds), postposition phrases(PostPs), preposition phrases(PPs) or adverbs(ADVs)[張 88].

- (1)a. *tzaushang* 'morning' / *shiatian* 'summer' -- NPs
- b. *shangshingchi* 'last week' / *houtian* 'the day after tomorrow' -- D-M compounds
- c. *kaihuei hou* 'after the meeting' / *shengyi shrbai chian* 'before business fail' -- PostPs
- d. *chen shangke* 'when going to the class' / *tzai gungtzuo shr* 'in working' -- PPs
- e. *tsengjing* 'once' / *yungyuan* 'forever' -- ADVs

If a parser fails to identify any of the above as a temporal expression as early as possible, unnecessary ambiguities may arise. For instance, NPs and D-M compounds could be treated as arguments of a sentence, and the verb in a PostP is likely misread as the main verb.

With regard to the definition of thematic roles at sentential levels, we generally follow the system of case roles of [Fillmore 68] and thematic roles of [Jackendoff 83]. In addition to arguments, we also take the thematic roles of adjuncts into our consideration. Our system includes modals involving obligation, permission, possibility, etc., and adverbial modifiers of manner, reason, negation, etc. On the other hand, the thematic role at NP level is defined according to the semantic relation between the head noun and the modifier as described in the next paragraph. At other phrasal levels, the semantic relation between the head and a modifier or an argument

is straightforward. For instance, the role that the whole PP plays in a sentence is significant while the relation between the preposition and its object is purely formal.

The thematic roles of arguments and adjuncts represent the semantic relations between them and the head. Such relations can either be very precise or very rough depending on how the thematic roles were defined and assigned. For example, at NP levels, the semantic relations between the modifier and head are very complex, including the relations of predication, quantifier, possessor, apposition and properties. In particular, the relations of properties can be sub-classified into the relation of location, use, time, whole-part, source, shape, etc. However, it is unrealistic for a parser to distinguish them for the implausibility of stacking all the fine-grained real world knowledge to make such trivial distinctions. Hence, our system of thematic roles are stipulated under the premise that a computational identification is plausible. This paper attempts to show how accurate thematic roles can be achieved using syntactically and semantically represented information. In the following section, we will introduce the parameters for the identification of thematic roles. The representations of the grammatical information and their functions in identifying the thematic roles are discussed in the third section. Their actual identification process is proposed in the fourth section.

2. THE PARAMETERS FOR IDENTIFYING THEMATIC ROLES

It is clear that the thematic role of an argument is dependent on its grammatical function and governing verb [Bresnan

82]. However, it does not mean that the determination of the thematic role solely relies on the surface syntactic configurations. For instance, the PP - attachment problem in English as shown in (2), can not be resolved without further information.

(2) John opened the door with the key.

The PP 'with the key' can modify the NP 'the door', or the VP or the whole sentence. However, the most plausible interpretation is that the PP modifies the VP. What, then, are the prime factors to be considered in identifying thematic roles?

In Chinese, thematic roles and syntactic configurations (or word order) also show the intimate correspondences between the thematic role AGENT and the subject position of the active verb, and between the role GOAL and the object position. This is exemplified by *Changsan* as an AGENT in (3)a, and as a GOAL in (3)b.

(3) a. *Changsan da Lisz.*
Changsan beat Lisz
'Changsan beats Lisz.'

b. *Lisz da Changsan.*
Lisz beat Changsan
'Lisz beats Changsan.'

However, for the relatively free word order of Chinese, it is difficult to determine the thematic roles only by the syntactic configurations, as in (4).

(4) a. Menkou lai le yige ren.
doorway come Asp a man
'There comes a man at the door.'

a'. Changsan lai le Taipei.
Changsan come Asp Taipei
'Changsan came to Taipei.'

- b. Chiangshang gua le yifu hua.
 wall on hang Asp a picture
 'There hung a picture on the wall.'
- b'. Yifu hua gua chiangshang.
 a picture hang wall on
 'A picture was hung on the wall.'
- c. Jeben shu wo kan guo.
 this book I read Asp
 'The book, I have read.'
- c'. Wo jeben shu kan guo.
 I this book read Asp
 'I have read the book.'

Therefore, the determination of thematic roles not only depends on the case frame of the governing verb but also the case restrictions, i.e. a thematic argument must meet the semantic restrictions imposed on that thematic role.

Furthermore, the syntactic category and semantic class(or features) of the constituent also serve as central or partial determining factors in identifying the role of that constituent in the following environments where syntax alone is not sufficient.

a) With Constituents Matching Semantic Restrictions Imposed by the Case Frame of the Verb

- (5) *Jeben shu wo kan guo.*
 this book I read Asp
 'This book, I have read.'

The verb *kan* 'read' is subcategorized for the case frame<(AGENT)(GOAL)>, and the semantic restriction on AGENT is that it must be animate and the restriction on GOAL is that it must be concrete. Thus in (5) *wo* 'I' has to represent the AGENT and *jeben shu* 'this book' has to represent the GOAL.

b) Temporal and Locative Expressions

Since the syntactic categories of these expressions are quite complex, e.g. they can be NPs, PPs, PostPs, ADVs. etc., the thematic role of them can only be determined by their semantic classes.

- (6) *Ta tzuotian mei lai.*
he yesterday not come
'He did not come yesterday.'

As shown in (6), there are two NPs preceding the main verb, the semantic class of *tzuotian* 'yesterday' precludes the possibility of its being an argument of the main verb.

c) Identifying the Roles of Adverbs and Modals

The roles of adverbs are subclassified into manner, degree, quantity, negation, reason, etc. according to their semantic classes. Modals, including (auxiliary) verbs and adverbs, are either subclassified into possibility, obligation, permission, or assigned semantic features according to their meanings.

Both modals and adverbs are sentential adjuncts, so it is easy to identify their thematic roles only from the syntactic categories or their semantic classes, without resorting to syntactic configurations as shown in (7)-(8).

- (7) *Ta hen gaushing.* ---> DEGREE
he very happy
'He is very happy.'
- (8) *Ni yinggai chiu.* ---> MODALS, + obligation
you should go 'You should go.'

d) Identifying the Thematic Roles of PPs

Generally speaking, the determination of the thematic role of a PP depends on the oblique case assigner (i.e. the

preposition) as well as the semantic class of the prepositional object.

- (9) a. *Ta tzai jia kan shu.*
he at home read book
'He read at home.'
- b. *Ta tzai jejung chingkung shia kan shu.*
he in such condition down read book
'He read in such condition.'

The thematic role of the PP headed by *tzai* can be time, location, condition, etc. However, the thematic role of the PP *tzai jia* 'at home' in (9)a can only be LOCATION, as determined by the semantic class of the prepositional object *jia* 'home'. Similarly, the PP *tzai jejung chingkung shia* 'in such condition' in (9)b can only have the CONDITION role.

e) Identifying the Thematic Roles of PostPs

In Chinese, a PostP is often composed of a temporal or locative constituent followed by a localizer as shown in (10). Thus, the thematic role of a PostP can only be determined by the semantic type of its argument.

- (10)a. *liangnian yishang* 'over two years' ---> TIME
two years over
- b. *juomian yishang* 'above the desk' --->LOCATION
desk over

f) Identifying the Semantic Relations Between NP Modifiers and Its Head Noun

The identification of semantic relations between the modifier and the head noun is dependent on the syntactic category and/or the semantic class of the modifier as exemplified by (11).

- (11)a. *neige baubau* 'that baby'
- b. *Changsan de baubau* 'Changsan's baby'

c. Changsan neige ren 'Changsan, that man'

The semantic relations in (11) are quantifier, possessor, and apposition respectively.

To summarize the above discussion, we conclude that four parameters determine the thematic roles of constituents:

a. the syntactic category and semantic features of the constituent,

b. the case frame and semantic restrictions of the verb,

c. the syntactic configuration and word order, and

d. oblique case assigner, including prepositions and postpositions.

In addition to the parameters, identification of thematic roles sometimes depends also on real world knowledge, as exemplified by the PP in (12) [Winograd 83].

(12) I saw a man on the hill with a telescope.

In Chinese, as mentioned in section 1, the semantic relations between the pre-nominal adjunct of property and the head in an NP depend on real world knowledge, as in (13).

- (13) a. shueiguo mianbau ---> MATERIAL
'fruit bread'
- b. shueiguo shiaufan ---> GOODS
'the peddler who peddles the fruit'
- c. shueiguo pan ---> PURPOSE
'fruit plate'

Making use of real world knowledge can help to identify thematic roles precisely, but the use and the representation of real world knowledge relies on further research in the field of artificial intelligence. Therefore, real world knowledge applied in this paper is limited to the following domain:

- a. the case frame and case restrictions of a verb,
- b. the conceptual hierarchy [Chen 88], in which the semantic features for concepts and the semantic relations between concepts were encoded.

The parameters proposed in this section have been adopted to solve PP-attachments in English, but the actual processes in each adoption are divergent, such as marker - passing in [Hirst 84], two-staged ATN in [林 89], and rule-based approach in [Dahlgren 86]. There is yet to be a thorough discussion on the identification of the thematic roles in Chinese. In the next section, we will propose the idea for the process and give the implementation detail in the section 4.

3. REPRESENTATION AND UNIFICATION OF PARAMETRIC INFORMATION

Syntactic structures for natural languages are usually complicated and ambiguous. There is no uniquely deterministic parsing strategy for natural languages. Marcus[80] considered deterministic parsing for natural languages is possible, although only under the condition that the parser can look three constituents ahead and disregard ambiguities such as PP-attachment. In other words, in syntax-only parsing, backtracking is unavoidable without looking ahead [Marcus 80]. However, language comprehension of human beings does not seem to involve looking ahead nor backtracking. In other words, identification of thematic roles can be done left to right without the hypothesis-correcting process of looking ahead or backtracking. It is reasonable to assume that this is because that syntactic

information, semantic information as well as real world knowledge are involved in human comprehension. From a communicative point of view, human languages should avoid ambiguities and non-determinism in order to reduce misunderstanding. Therefore we believe that human language comprehension can be done by simply accumulating information monotonously. Violations of the above principle are rare but do appear in some languages. Garden-path sentences present typical examples as in (14) [Hirst 84].

(14)a. The horse raced past the barn fell.

b. The old dog the foot steps of the young.

However, we fail to discover similar examples in Chinese. Hence we believe that the identification of thematic roles in this language can be done by way of information accumulation with simple inferences and reasoning. The feature structures and unification formalisms advocated by many current theorists [Bresnan 82, Gazdar 85,87, Kay 85, Pollard 87, Shieber 86, Uszkoreit 86] offer a good approach to encode and accumulate the information needed in identifying thematic roles. Accordingly, we propose to unify syntactic analysis and semantic interpretation into a single process. Thus not only full advantages of unification parsing strategy are taken but both syntactic and semantic information also complement each other to make parsing more deterministic and efficient. In the next section, we will take Chinese examples to show how the parameters are represented and how they function in identifying the thematic roles.

4. THE IDENTIFICATION OF THEMATIC ROLES

As described in section 2, in addition to real world knowledge, there are four parameters to identify thematic roles of constituents in a sentence. The representation for the parametric information is specified in the lexical feature structure of the Information-based Case Grammar(ICG)[陳 89] . The information of the four parameters includes the syntactic information, like syntactic category, and word order, etc., and semantic information, like case restrictions, semantic features, etc. ICG offers lexicon-based syntactic representations. Each lexical entry contains the syntactic and semantic feature structures for the lexical head and projection phrases, as denoted in (15)*.

(15) word : $\left\{ \begin{array}{l} \text{sem} \\ \text{syn} \end{array} \right\} \left\{ \begin{array}{l} \text{meaning:} \\ \text{features:} \\ \text{arguments:} \\ \text{adjuncts:} \\ \text{class:} \\ \text{constraints:} \end{array} \right\} \left\{ \begin{array}{l} \text{Form:} \\ \text{BP (Basic Pattern):} \\ \text{AP (Adjunct Precedence):} \end{array} \right\}$

Let us recall that the first parameter in identifying thematic roles involves syntactic categories and semantic features. The information is specified in the syntactic class column and in the semantic features column respectively, as shown in (16)-(18).

(16) *jungchiou* 'Mid-Autumn':
 $\left\{ \begin{array}{l} \text{sem} \\ \text{syn} \end{array} \right\} \left\{ \begin{array}{l} \text{features: + time} \\ \dots \\ \text{class: Nd (temporal noun)} \\ \dots \end{array} \right\}$

* For convenience and readability, only partial information of the description of the information structure is given in this paper. Readers are referred to [陳 89] for detailed description and discussion of the grammar formalism.

(17) *keneng* 'possibly':

{	sem {	features: + modal, + possibility
	syn {	class: VL3 (modality)
		...

(18) *kungshou* 'empty-handed':

{	sem {	features: + manner
	syn {	class: Dh (ADV)
		...

The second parameter involves case frames and case restrictions of verbs. The information is specified as the values of the semantic arguments, as in (19)-(20).

(19) *da* 'beat':

{	sem {	arguments: {	AGENT: {	features: + animate
			GOAL: {	features: + physical

(20) *chiuan* 'persuade':

{	sem {	arguments: {	AGENT: {	features: + human
			GOAL □ *: {	features: + human
			THEME: {	feature: +active
				arguments: AGENT: □

The third parameter involves the syntactic configuration and word order. In ICG, the syntactic rules of the phrasal and the sentential structure can be represented in the information structure of the head by means of the immediate dominance rules (ID rules) and linear precedence rules (LP rules). The immediate dominance relations are specified in the semantic information column encoding the argument specification and the adjunct specification, and the syntactic form and the linear precedence of the constituent are specified in part of the syntactic constraints column [陳 89], as shown in (21)-(22).

* '□' indicates the co-referential relation, i.e. the GOAL and the AGENT of the embedded sentence are identical.

- (21) *chiuan* 'persuade':
- | | |
|------|--|
| sem: | arguments: { AGENT, GOAL, THEME |
| | adjuncts: time, loc, modality, reason,
instrument, manner, ... |
| syn: | constraints: { form: { time[{NP, DM, PP, ADV, +time}]
loc[PP, +location]
modality[VL3]
manner[ADV, +manner] |
| | BP: AGENT[NP] < * < GOAL[NP] < THEME[VP]
AP: { 1. {modality, time, loc} < *
2. AGENT < manner < * |

- (22) *da* 'beat':
- | | |
|------|--|
| sem: | arguments: { AGENT, GOAL |
| | adjuncts: time, loc, modality, manner, reason,
instrument, ... |
| syn: | form: (omitted) |
| | BP: { 1. AGENT[NP] < * < GOAL[NP]
2. GOAL[NP] < AGENT[PP[<i>bei</i>]] < * |
| | AP: (omitted) |

The last parameter is case markers. The oblique case assigners generally are represented as prepositions in Chinese, for example, *ba* is a GOAL marker, *bei* is an AGENT marker. However, a preposition may mark more than one case depending on the object of the preposition[黃 88]. For example, when the object of the preposition *bei* is an animate NP, *bei* in (23) marks an AGENT case; when the object is an abstract NP, *bei* in (24) marks a CAUSER case; when the object is a non-animate NP, *bei* in (25) marks an INSTRUMENT case.

- (23) *Changsan bei Lisz da.*
Changsan BEI Lisz beat
'Changsan was beaten by Lisz.'
- (24) *Changsan bei emeng jingshing.*
Changsan BEI nightmare awake
'Changsan was awakened by nightmare.'
- (25) *Changsan bei jen tszpuo le shoujr.*
Changsan BEI needle sting Asp finger
'Changsan was stung on the finger.'

In ICG, the case assigned by the preposition is represented

as semantic features in the information structure, exemplified by *bei* in (26).

(26) *bei*:

{	sem	{	features: 1.AGENT 2.CAUSER 3.INSTRUMENT
			arguments: DUMMY: { features: 1. +animate 2. -physical 3. -animate
{	syn	{	form: DUMMY[NP]
			BP: * << DUMMY

As shown in (26), the No.1 semantic feature has to correspond with the No.1 DUMMY feature, and so on so forth. In other words, when *bei* marks an AGENT, the object of *bei* must be animate. Other predictions follow similarly.

The thematic roles of the PPs in (23)-(25) is fully identified by the preposition *bei* and its object. But the identification of thematic roles for certain other PP's can not be completely resolved by the case marker and its object, since it still depends on the information of verbs. Further discussion of this issue will be given later involving (32)-(35).

With the representation of the four parameters in identifying thematic roles described, we can classify five different cases of how the parameters work in application.

4.1. IDENTIFIED ONLY BY THE FIRST PARAMETER -- THE SYNTACTIC CATEGORY AND THE SEMANTIC FEATURE OF THE CONSTITUENT

- a. Adverbs: the thematic roles of adverbs are exhaustively and unambiguously classified into quantity, evaluation, negation, degree, manner, reason, time, location, etc.[張 89], according to their semantic class.
- b. Modals: modals are classified into obligation, permission, possibility, etc.[張 89] according to their semantic feature.

- c. A small part of NPs: the identification of the thematic roles of most NPs is related with case restrictions, case frames, even with prepositions. However, there are some special types of NPs which are identified only by semantic features, for instance, time NPs.

For the above categories, their thematic roles can be immediately identified, so many unnecessary syntactic branches and ambiguities can be avoided.

4.2. IDENTIFIED BY TWO PARAMETERS -- THE CONSTITUENT AND THE OBLIQUE CASE ASSIGNER

- a. PPs: the uncertainty in identifying thematic roles of PPs can often be resolved by the semantic features of the prepositional object as exemplified by (27)-(29).

- (27) *bei Lisz* 'by Lisz':
 sem { features: AGENT, +human
 arguments: DUMMY: { meaning: Lisz
 features: +animate
- (28) *bei emeng* 'by nightmare':
 sem { features: CAUSER, +phenomenon
 arguments: DUMMY: { meaning: nightmare
 features: -physical
- (29) *bei jen* 'by the needle':
 sem { features: INSTRUMENT, +artificiality
 arguments: DUMMY: { meaning: needle
 features: -animate

Based on the conceptual hierarchy [Chen 88], human inherits the feature animate, so according to (26) the feature structure of *bei*, *bei Lisz* 'by Lisz' in (27) is identified with the No.1 role AGENT. The thematic roles of PPs in (28) and (29) are identified in the same way. However, for certain prepositional phrases, their resolution of thematic roles cannot be uniquely determined by two parameters, the uncertain thematic roles will be further resolved at

sentential level. We will discuss them in 4.3.

- b. PostPs: a postposition in Chinese often marks TIME or LOCATION, and the constituent preceding a postposition may neither be temporal nor locative, as shown in (30).

(30)a. *shueichr chian* 'before the water pond'

b. *chushr chian* 'before things-go-wrong'

Then, how can one identify the thematic role of the PostP based on the semantic features of the pre-argument of the postposition? Take the postposition *chian* 'before' as example, its feature structure is as follows.

(31) *chian* 'before':

sem	{	features:	1. TIME	2. LOCATION
		arguments:	DUMMY:	{ features: 1. (+ time, + event) 2. + physical
syn	{	form:	DUMMY[{NP, VP, S}]	
		BP:	DUMMY << *	

Since *shueichr* 'pool' in (30)a has the feature physical, so the thematic role of the PostP in (30)a can only be LOCATION, as stipulated by the feature specifications. By the same token, the syntactic category and the semantic feature of *chushr* 'things-go-wrong' in (30)b matches the No.1 semantic feature of the postposition *chian* 'before', so the thematic role of the PostP in (30)b is TIME. Thus, the thematic roles of all PostPs can be best identified at the phrasal level.

4.3. IDENTIFIED BY THREE PARAMETERS -- THE CONSTITUENT, CASE FRAME AND CASE RESTRICTION, AND OBLIQUE CASE ASSIGNER

In this section, we will focus on the unresolved thematic roles of some PPs after the phrasal level parsing. These will be determined by three parameters after parsing the sentence, typical examples are given in (32).

- (32)a. *Shiuesheng yi wenchuan shitau.*
 student YI hot spring bathe
 'The students bathe in the hot spring.'
- b. *Beitou yi wenchuan wenming.*
 Beitou YI hot spring famous
 'Beitou is famous for hot springs.'
- c. *Yi wenchuan eyan, Yangmingshan tzueihau.*
 YI hot spring the best
 'As for hot springs, the Yangming mountain offers the best.'

Though the three PPs are headed by the same preposition *yi*, their thematic roles are different owing to the different contextual environments, The PP is an INSTRUMENT in (32)a, a CAUSER in (32)b, and a TOPIC in (32)c.

The thematic roles of the PPs in (32) can not be adequately determined only by the feature structure information of *yi* in (33).

- (33) *yi*:
- | | | | | | | | |
|---|-----|---|------------|--|----------|----------|------------------------|
| { | sem | { | features: | 1.INSTRUMENT | 2.MANNER | 3.CAUSER | 4.TOPIC |
| | | | arguments: | DUMMY: { | | | |
| | | | | | | | features: |
| | | | | | | | 1. -animate |
| | | | | | | | 2. {-physical, +event} |
| | | | | | | | 3. {+entity, +event} |
| | | | | | | | 4. {+entity, +event} |
| { | syn | { | form: | 1. DUMMY[NP], 2.3.4. DUMMY[{NP,VP,S}] | | | |
| | | | BP: | B1: 1.2.3.* << DUMMY | | | |
| | | | | B2: 4.* << DUMMY << {laishuo, laikan, shuolai, tinglai, kanlai, eyan, eluen} | | | |

After parsing the PP *yi wenchuan* in (32), the unified information shows the following result:

- (34) *yi wenchuan* 'with the hot spring'
- | | | | | |
|---|-----|---|------------|----------------------------|
| { | sem | { | meaning: | hot spring |
| | | | features: | 1.INSTRUMENT 3.CAUSER |
| | | | arguments: | DUMMY: { features: terrain |
| { | syn | { | class: | PP |

MANNER and TOPIC are out of consideration respectively for conflicts in semantic features and the BP of the PP. With regard to the distinction between INSTRUMENT and CAUSER, it depends on further information to be supplied by the main verb. If the verb

is active, then the thematic role of the PP is an INSTRUMENT; if the verb is stative, the thematic role is a CAUSER. The parameters functioning in identifying the thematic role of the PP headed by *yi* are as the following:

(35)

case marker	syntactic & semantic restrictions of the constituent	class of verb	role
<i>yi</i>	- animate	active	INSTRUMENT
	- physical or + event	active	MANNER
	having the following constituent { <i>laishuo</i> , <i>eyan</i> , ...}	active, stative	TOPIC
	+event	stative	CAUSER

Although the above discussion seems a bit complicated, in fact, the process is very simple when the related deterministic information is represented at different levels. For example, the information structure of the *yi* PP unifies the information of the preposition *yi* in (33) and the semantic information of its possible argument to get a set of the unresolved thematic roles as shown in (34). The additional information needed is represented in the information structure of the verb, as the following rough representation:

(36) active verbs:

syntactic information: { form: [INSTRUMENT [PP [{ *yung*, *yi*, *kau*, *jie*, ... }]]]
MANNER [PP [{ *yi*, *yung*, *ping*, *kau*, ... }]]]
CAUSER [PP [{ *bei*, *jiaou*, *rang*, *wei*, ... }]]] }

(37) stative verbs:

syntactic information: { form: [MANNER [PP [{ *yi*, *yung*, *jie*, *jang*, ... }]]]
CAUSER [PP [{ *yi*, *bei*, *jiaou*, *wei*, ... }]]] }

In (37), the INSTRUMENT case role is not specified in the information structure of stative verbs; In (36), *yi* is not

involved in the CAUSER. Further, when the information of the preposition *yi* in (33) and the stative verb *wenming* 'famous' unify, the thematic role of the PP in (32)b can only be a CAUSER because it is the only possible role stipulated by the verb. On the other hand, when the information of the PP unifies with the information of the active verb *sitzau* 'bathe', the thematic role of the PP in (32)a has to be an INSTRUMENT because that is the only possible one specified by the verb.

4.4. IDENTIFIED BY THREE PARAMETERS -- THE CONSTITUENT, THE VERB AND WORD ORDER

The thematic roles of most arguments unmarked by case assigners are identified by three parameters.

- (38) *Jeben shu wo kan guo.*
 this book I read Asp
 'This book, I have read.'

In (38), *Jeben shu* 'the book' is the GOAL of the verb *kan* 'read', and *wo* 'I' is the AGENT. The related parametric information is encoded in the information structure of the verb *kan*, as shown in (39).

- (39) *kan* 'read':
- | | | | | |
|---|-----|---|----------------------------|---------------------------------|
| { | sem | { | arguments: { | |
| | | | AGENT: features: + animate | GOAL: features: + physical |
| { | syn | { | BP: { | B1: AGENT < * < GOAL |
| | | | | B2: AGENT < GOAL[+definite] < * |
| | | | | B3: GOAL[+definite] < AGENT < * |
| | | | | |

In parsing, functional identification of the thematic roles is unresolved after parsing the NPs *jeben shu* 'the book' and *wo* 'I'. Their thematic roles are not resolved until the unification of the information of the verb *kan* 'read'.

4.5. IDENTIFIED BY ALL FOUR PARAMETERS

- (40) *Changsan bei Lisz da.*
Changsan BEI Lisz beat
'Changsan was beaten by Lisz.'

In (40), the thematic role of the NP *Changsan* is identified with the information of the constituent, the case frames and restrictions of the verb, and word order. In addition, it is also related to the case role of the PP *bei Lisz*, for the preposition marks the AGENT case. Based on the information structure of the verb *da* 'beat', given in (22), it requires a GOAL and an AGENT, and one possible syntactic encoding has the GOAL occur preceding the *bei*-marked AGENT. Thus the four parameters converge to determine the thematic role of the NP *Changsan* as a GOAL.

The actual examples described in section 4 roughly show how the parametric information is represented and how it functions in identifying thematic roles at different phrasal level. Information accumulation is a general method to identify thematic roles. It is not only applicable to the processing of Chinese. For instance, we think the PP - attachment problem in English also can be resolved by the same way.

With regard to real world knowledge, it has not been implemented for representational difficulties. However, the concept of information accumulation is also applicable to the processing of real world knowledge when available.

5. CONCLUSION

In parsing, the identification of thematic roles is a very important procedure. It not only is fundamental to semantic interpretations but also reduces syntactically ambiguous branches

if performed timely.

The identification of thematic roles is just a start point to semantic interpretation and to the understanding of natural languages. Some of the thematic roles discussed are still very crudely defined owing to the limitation of applicable information. The available information only includes a) syntactic categories and semantic features of the constituent, b) the case frame and case restrictions of the verb, c) the syntactic configurations and word order, and d) the oblique case assigners. Correct identification of thematic roles depends on how much information is available. The more available information is, the better our identification should be. Even though the roles we have chosen to use are roughly those proposed by Jackendoff[83], we did make some refinement. Furthermore, if real world knowledge is better represented in the future, the thematic role can be more precisely and quickly identified.

ACKNOWLEDGEMENT

Research of this paper was partially supported by NSC grant #78-0408-E001-001 and the Electronic Research and Service Organization, Industrial Technology Research Institute, Taiwan, R.O.C. Some of the examples are taken from the research result of Chinese Knowledge Processing Group at Computing Center of Academia Sinica. We alone, of course, are responsible for any possible error contained in this paper.

REFERENCE

- 林雅萍 1989, "嵌入 X' 理論於 ATN 系統中以解決介詞片語所屬問題", 碩士論文。
台北: 台灣大學。
- 陳克健 1988, "中文剖析的問題與對策", 中華民國第一屆計算語言學研討會論文集。
pp. 21-28, 南港: 中央研究院計算中心。
- 陳克健, 黃居仁 1989, "訊息為本的格位語法 -- 一個適用於表達中文的語法模式",
第二屆計算語言學研討會論文集。
- 張莉萍 1988, "漢語的時間詞組和語言剖析", 中華民國第一屆計算語言學研討會
論文集。 pp. 75-86, 南港: 中央研究院計算中心。
- 張麗麗及中文詞知識庫小組 1989, 國語的詞類分析, 修訂版, 中央研究院計算中心
技術報告。
- 黃瑞珠 1988, "由剖析的觀點分析漢語介詞組", 中華民國第一屆計算語言學研討會
論文集。 pp. 127-144, 南港: 中央研究院計算中心。
- Bresnan, J. (ed.) 1982. *The Mental Representation of Grammatical Relations*. Cambridge, Mass. : MIT Press.
- Chen, K. J. and C. S. Cha 1988. "The Design of a Conceptual Structure and Its Relation to the Parsing of Chinese Sentences." ICCPCOL'88, Toronto.
- Dahlgren, K. and J. McDowell 1986. "Using Commonsense Knowledge to Disambiguate Prepositional Phrase Modifiers." AAAI-86, 589-593.
- Fillmore, C. 1968. "The Case for Case." In E. Bach and R. Harms (Eds.), *Universals in Linguistic Theory*. New York: Holt, Rinehart, and Winston.
- Gazdar, G., E. Klein, G. K. Pullum, and I. A. Sag 1985. *Generalized Phrase Structure Grammar*. Cambridge: Blackwell, and Cambridge, Mass.: Harvard University Press.
- Gazdar, G. et al. 1987. *Category Structures*. CSLI Report 102, Stanford: CSLI.

- Hirst, G. J. 1984. "Semantic Interpretation Against Ambiguity."
Ph.D. Diss. Brown University.
- Jackendoff, R. 1983. *Semantics and Cognition*. MIT Press.
- Kay, M. 1985. "Parsing in Functional Unification Grammar." In
D. Dowty, L. Karttunen, and A. Ziwicki (Eds.), *Natural
Language Parsing*. Cambridge: Cambridge University Press.
- Marcus, M. P. 1980. *A Theory of Syntactic Recognition for
Natural Languages*. MIT Press.
- Pollard, C. and Ivan A. Sag 1987. *Information-based Syntax
and Semantics, Vol.1 Fundamentals*, CSLI Lecture Notes
Series No. 13. Stanford: CSLI.
- Shieber, S. 1986. *An Introduction to Unification-Based
Approaches to Grammar*. CSLI Lecture Notes Series No.4.
Stanford: CSLI.
- Uszkoreit, H. 1986. "Categorial Unification Grammars." In
Coling 1986. Bonn: University of Bonn. Also appeared as
Report No. CSLI-86-66, Stanford: CSLI.
- Winograd, T. 1983. *Language as a Cognitive Processes*. Vol.1,
Syntax, Addison-Wesley.

A Unification-based Approach to Lexicography for Machine Translation System

Shu-Chuan Chen*, Mei-Hui Wang*, and Keh-Yih Su**

***BTC R&D Center
2F, 28 R&D Road II
Science-based Industrial Park
Hsinchu, Taiwan, R.O.C.**

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

ABSTRACT

In an operational machine translation system, a variety of texts will be encountered even if its domain of dexterity is restricted to a specific field. This diversity of texts poses problem on handling different usages or translations of identical lexical items. This paper presents a unification-based method for lexicography that can greatly lessen this problem. In the paper, we give a detailed discussion and example of the unification technique. We also show that by unifying lexical information in different dictionaries, the time spent in dictionary construction is saved; dictionary storage space is minimized; the integrity of distinct dictionaries is preserved; the option regarding which dictionaries to be unified is kept open; and all of the lexical information needed to construct a complete dictionary based on the vocabulary for a specific customer project is available. In view of the fact that categorial ambiguity might occur as a result of unification, score function is added as a solution. With these advantages, we regard the unification approach to lexicography as viable in enhancing the translation performance of a practical machine translation system.

1. Introduction

In an operational machine translation (MT) system, even if its domain of usefulness is restricted to a specific field¹, a rich variety of texts will still be encountered. For instance, if the domain is limited to articles on computer science, texts in the areas of user manuals, programming languages, hardware, etc. are all possible inputs. These texts may differ in the use of individual words, the size of glossaries, the patterns of syntactic constructions, and so on.

For an operational system like ArchTran, which is a commercialized English-Chinese machine translation system developed at BTC R&D Center, the main concern in the face of diversity of texts is the ability to deal with different usages or translations of identical terms².

The problem concerning different usages or translations of identical terms is two-fold. On the one hand, different usages or translations may result from **ambiguity in word sense**. On the other hand, the differences may be due to the **requirements of customers**.

The problem concerning word sense ambiguity is that a good number of words have more than one possible meaning, and different meanings may call for different translations. For example, the word *current* may be in the sense of "water flow" in one text, and "electricity flow" in another. The former use of the word will be translated into Chinese as "水流", and the latter as "電流" accordingly.

To disambiguate the semantics of a polysemous word found in a text in order to render the correct translation, the following knowledge sources should be incorporated into the MT system: morphological information (a word used as a countable or uncountable noun may mean differently); syntactic information (different internal arguments may give rise to different meanings of a verb); semantic information (selectional restrictions); and pragmatic or contextual information (using the technique of "script"). Nevertheless, for a second generation MT system like ArchTran, not all the information needed for disambiguation is available or complete [Boit87]. Therefore, other means of disambiguation have to be incorporated as well.

In the ATLAS-G system [Fuji89], finding the correct translation, i.e. meaning, is in part done interactively by selecting and remembering the most appropriate translation for a given word in a given text. The problem with this approach is that once chosen, a translation will be assumed for the rest of the text. If the selected translation is suitable for just a few occurrences of the word, the translation of the other occurrences will be in error.

As for the problem of satisfying customer's requirements, a customer may wish a specific translation for a term, and the MT system must be able to do that. For example, one customer may prefer the term *operating system* to be translated as "作業系統", while another as "操作系統".

An obvious solution to the problem of satisfying a customer's request of a specific translation is to change the translation listed in the system dictionary into the one preferred by

the customer each time a text is translated. This, however, is problematic, since the translation has to be changed from time to time to be in compliance with a particular text. Besides, if more than one text is being translated at the same time, the change may be suitable for the word in one text but not the others.

As an alternative solution, one may propose to construct a separate and self-contained dictionary for each text. However, chances are the glossaries of different texts may differ only in a relatively small number of words, thus building separate dictionaries is not feasible. Because by doing so, the time spent on lexicography and the storage space taken up by the dictionaries with a huge amount of shared words and lexical information are wasteful.

Another possible solution is to create a run-time dictionary that stores only those words whose meanings or translations are specified by the user interactively, and the life span of the dictionary lasts just for the text currently under translation. This method suffers the same drawback as the ATLAS-G system. Furthermore, because a run-time dictionary is not accessible to other texts being translated at the same time and also because it is not accessible to similar texts to be translated at a later time, the power of a time-sharing computer is not fully utilized.

Discussions on disambiguating word senses abound in the MT literature [Alle87, Hirs87, Hutc86, Nire87]. These discussions focus mainly on the use of the various knowledge sources mentioned before. A second generation MT system, as pointed out, is limited in its access to these knowledge sources. On the other hand, little discussion can be found on the issue of producing translations preferred by customers. The solutions examined above concerning "customer tailored" translations are unsatisfactory. In this paper, a unification approach to dictionary information combination is proposed as a new and viable way to deal with different usages and translations of identical words that occur as a result of diversity in texts. The unification technique has been implemented in the ArchTran system and proved to be of fruitful result.

2. Principles in Constructing ArchTran Dictionaries

The way in which the ArchTran dictionaries are constructed and the way in which the dictionary information is unified during parsing are the key to solving the problem of different usages and translations of identical words. Hence, before going into the details of the use of unification, a brief introduction of the principles behind lexicography in the ArchTran system is in order.

Below are some of the major principles governing dictionary construction in ArchTran:

Principle 1 : Use all possible information, whether morphological, syntactic, or semantic, to disambiguate word senses of a lexical item. The corresponding translations of these senses are recorded in the dictionary.

Principle 2 : Create separate dictionaries to store words used in different domains and for different customers. No duplicate information is allowed in these dictionaries.

Under this principle, ArchTran developed three types of dictionaries. One is constructed to store words that can be found in all sorts of texts, called **general dictionary**. The second dictionary is called **technical dictionary**, which encompasses the words used in a particular field, such as machinery, computer science, etc. The third is a **customer dictionary** for storing technical terminology that differs from or lacks in both the general and the technical dictionaries. That is, the terminology in the customer dictionary is specific to the texts of a particular customer. It should be noted that for a given technical domain, there may be more than one customer dictionary, because customer dictionary can be further branched into several sub-dictionaries to store terms for different customers or for different projects.

Consider the following example that illustrates the functions of the three types of dictionaries. The word *computer* will be listed in the general dictionary for its established usage in nearly every walk of life. The word *firmware*, used solely in the domain of computer science, will be listed in the computer technical dictionary. And the word *Macrokey*, a term denoting a software package developed by BTC that enables users to define their keyboard functions, is listed in the customer dictionary for a particular project.

It should be noted that the very criteria that determine in which dictionary a word should be stored also regulate other information of a word, such as categories, word senses, internal arguments of verbs, and so on. For instance, suppose that a lexical item has three distinct word senses: A, B, and C. If A can be found in the texts of various fields, it is stored in the general dictionary. If B is used in the field of computer science solely, it is stored in the computer technical dictionary. And if C is used exclusively in the texts of a specific company or project, it is stored in a customer dictionary accordingly.

These three types of dictionaries and their sub-dictionaries are organized in a hierarchy by generality. In the case that a term is listed in more than one dictionary, its use is supposed to be most specific in the texts of a specific customer project, less in a technical domain, and least in general use. The hierarchical structure of the general, technical, and customer dictionaries in the ArchTran system are illustrated in Figure 1.

Building three different types of dictionaries serves several important purposes. The first and the main purpose is to **render the most suitable translation** for a polysemous word or to meet customer's requirement. If the MT system can not successfully disambiguate the senses of a word using all the knowledge sources noted in Principle 1 above, restricting the domain of translation will be of help. The accuracy in disambiguating the semantics of a word can be enhanced, since in a specific domain, the number of possible meanings of a polysemous word is, in most cases, limited. And only this limited number of meanings needs to be differentiated and recorded in the dictionaries. Since the meaning or translation listed in the customer dictionary is the most likely one to be used in the texts of a specific domain than

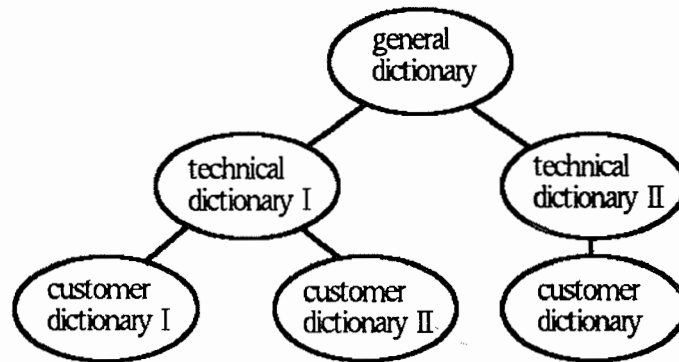


Figure 1: The hierarchical structure of the general, technical, and customer dictionaries in terms of generality

that in the technical dictionary, therefore, during translation it has the priority of being applied before that in the technical dictionary. The same holds for the entry in the technical dictionary and the general dictionary. Thus, the most suitable meaning, or translation, can be correctly produced by this priority ordering. This point will be further exemplified in Section 3.2

The second purpose is to **save dictionary construction time**. For an MT system to translate texts of different fields, it is important to build separate technical and customer dictionaries to store the terminology. As each dictionary is defined as to the kind of lexical items and lexical information it should store, no information will be duplicated in these dictionaries. Thus, eliminating duplicate storing of the same data will make dictionary construction time-efficient.

The third purpose of building three types of lexicons is to **save dictionary storage space**. The storage space taken up by dictionaries is significantly cut down, since each dictionary stores no more words or lexical information than it is purported to.

The last purpose is to **maintain integrity of dictionary**. The integrity of the general dictionary and technical dictionary can be maintained, since no changes will be made directly on the lexical items in these dictionaries every time a particular translation is preferred by a customer.

As there are three types of dictionaries in ArchTran, and each stores no more lexical items or lexical information than is specified, the need of unifying these dictionaries is obvious during translation, because only by unifying these dictionaries can the most suitable translation be obtained and a complete set of glossaries be available.

In the next section, the technique of unification will be discussed at length.

3. Unification Operation in ArchTran

3.1 Unification in Lexicography

Unification is an operation employed in quite a number of linguistic and computational theories. Basically, unification is similar to the notion of set union when the elements to be unified are atomic elements. Unification departs from set union when unifying complex-feature-based information elements. Unification is said to "fail" when the values of the same features to be unified clash; the operation succeeds when the values of the same features match. If unification succeeds, the "merge" operation may be subsequently performed [Shie86, Huan88].

The example below shows how unification is used in ArchTran for lexicography. Provided that there are two dictionaries in ArchTran that have an identical entry LEX. We will call the LEX in these two dictionaries LEX1 and LEX2, respectively. Let us assume that LEX1 and LEX2 differ in both category and meaning (they may differ in other aspects and the same principle applies), and the lexical information of LEX1 and LEX2 can be notated by feature structures as shown in Figure 2:

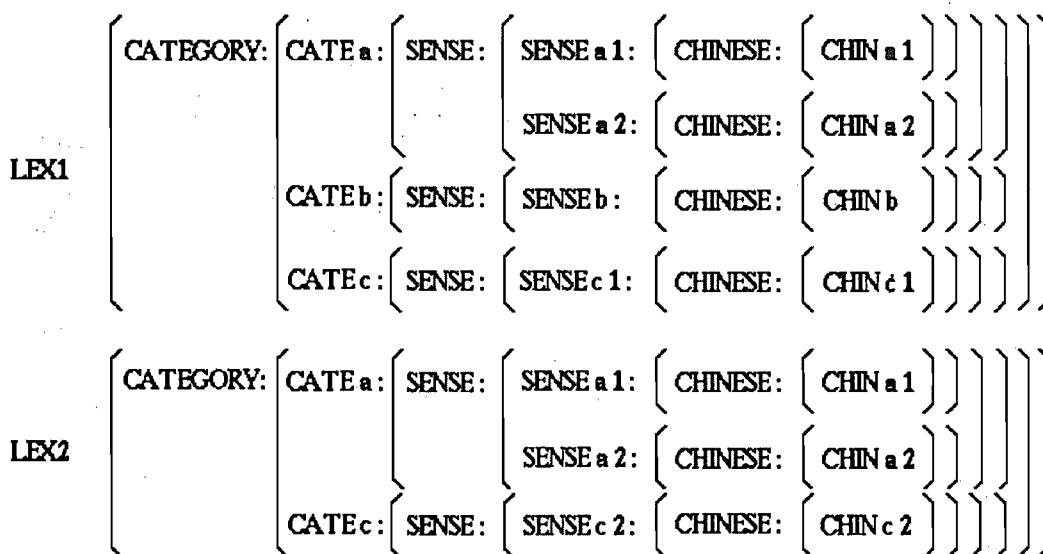


Figure 2: Differences between LEX1 and LEX2

In Figure 2, CATEGORY is a feature. CATEa, CATEb, and CATEc are values of CATEGORY and are features themselves. SENSE is the value of CATEa, CATEb, and CATEc and is a feature itself. SENSEa1, SENSEa2, SENSEb, SENSEc1, SENSEc2 are values of SENSE and are features as well. CHINESE is the value of SENSEa1, SENSEa2, SENSEb, SENSEc1, SENSEc2 and is a feature itself. CHINa1, CHINa2, CHINb, CHINc1, and CHINc2 are

The result of unifying LEX1 and LEX2 is shown in Figure 3:

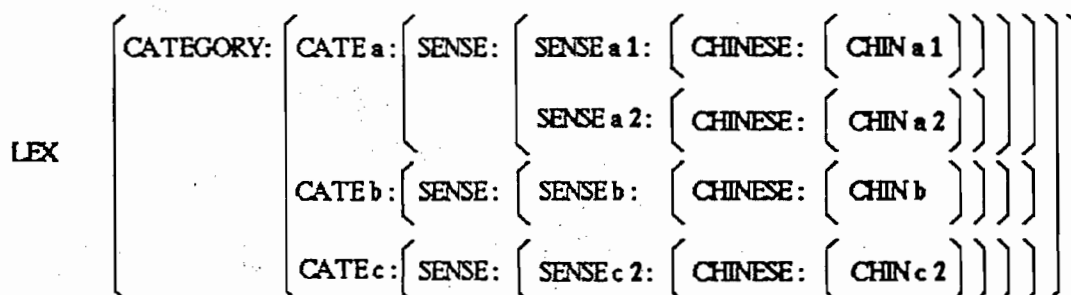


Figure 3: Result of unifying LEX1 and LEX2

From Figure 2 and Figure 3, we can see that for the feature CATEa, the values in LEX1 and LEX2 match with each other and can be subsequently merged. As for CATEb, since there is no counterpart in LEX2, it will be included. For CATEc, the values of SENSEc1 and SENSEc2 are in conflict and, as a result, unification fails. In ArchTran, an important operation when unification fails is **overwriting** [Shie86], by which we mean that the lexical information in one dictionary will replace that of the other. In this case, SENSEc2 in LEX2 overwrites SENSEc1 in LEX1.

A question that arises here is that which dictionary has the right to overwrite. As noted above, ArchTran has three types of dictionaries: customer dictionary, technical dictionary, and general dictionary, and they are organized in a hierarchy by generality. Therefore, for a given lexical item, as its use is most specific in the texts of a specific customer project, less in a technical domain, and least in general use, its data in the customer dictionary overwrite those in the technical dictionary, which in turn overwrite those in the general dictionary.

In the following section, we will give a concrete example to illustrate the use and effect of unification in the ArchTran system in combining the information of all the entries of a term found in different dictionaries.

3.2 An Example

Consider the word *stream*. It can be used as noun and verb, and both categories are stored in the general dictionary. One of the meanings of the noun is "brook", and its corresponding Chinese translation is given as "溪流". One of the meanings of the verb when used as a transitive verb is "cause to flow", and its corresponding Chinese translation is given as "使流出". Provided that other senses of *stream* are not distinguished in the system, these are the only two senses of the word listed in the general dictionary.

In the field of computer science, the same word when used as a noun means "stream of data", and it is translated into "資料流". When it is used as a transitive verb, it means "execute sequentially" and is translated into "依序執行". These two senses are recorded in the technical dictionary for computer domain.

Now the lexical information of the word *stream* recorded in the technical and the general dictionaries is as shown in Figure 4:

technical dictionary	{	CATEGORY:	{	NOUN:	{	SENSE:	{	STREAM OF DATA	:	{	CHINESE:	{	資料流	}}	}}	}}	}}	}}	}}
general dictionary	{	CATEGORY:	{	NOUN:	{	SENSE:	{	BROOK	:	{	CHINESE:	{	溪流	}}	}}	}}	}}	}}	}}

Figure 4 : Lexical information of the word "stream" in the technical and the general dictionaries

Suppose that no translations are specified regarding the translations of both the verb and the noun, after unifying the two dictionaries, the resultant lexical information of *stream* used in translating a text in the domain of computer is shown in Figure 5:

stream	{	CATEGORY:	{	NOUN:	{	SENSE:	{	STREAM OF DATA	:	{	CHINESE:	{	資料流	}}	}}	}}	}}	}}	}}

Figure 5 : Lexical information of the word "stream" after unifying the technical and the general dictionaries

As can be seen in Figure 5, the translation of *stream* when used as a verb will be "依序執行" and when used as a noun will be "資料流", rendering the most suitable translations for the word used in a text in computer science.

Suppose that in translating computer user manuals for a customer, the verb *stream*, used to mean the same as that in the technical dictionary, is preferred to be translated as "執行". This customized translation will be stored in the customer dictionary. Now the lexical information of the word *stream* listed in the customer dictionary is as shown in Figure 6:

customer dictionary [CATEGORY: [VERB: [SENSE: [EXECUTE SE-QUENTIALLY : [CHINESE: [執行]]]]]]]]

Figure 6 : Lexical information of the word "stream" in the customer dictionary

After unifying the three dictionaries, the resultant lexical information of *stream* used in translating the text is shown in Figure 7:

stream [CATEGORY: [NOUN: [SENSE: [STREAM OF DATA : [CHINESE: [資料流]]]]]] [VERB: [SENSE: [EXECUTE SE-QUENTIALLY : [CHINESE: [執行]]]]]]]]

Figure 7 : Lexical information of the word "stream" after unifying the customer, technical, and general dictionaries

As can be seen in Figure 7, the translation of *stream* when used as a verb will be "執行", meeting the requirement of the customer.

In the following section, we will discuss the merits of employing unification in dictionary information combination.

4. Merits of Using Unification in Lexicography

4.1 Merits of Unification

Besides satisfactorily handling the problem of different usages and translations of identical terms found in various texts, there are at least three other advantages of using unification in dictionary construction:

- [1] **Providing customized vocabulary without resorting to run-time dictionary** : The use of unification has the same effect as a run-time dictionary in providing customized vocabulary, but it does not have its drawback noted in Section 1; that is, a run-time dictionary is accessible only to one single text running at a given time.
- [2] **Providing options in unifying various dictionaries** : We can specify which technical dictionary and which customer dictionary to be unified with the general dictionary for each text to be translated. Thus, the system dictionaries can handle texts of different fields and from different customers.

[3] **Complete dictionary available based on the vocabulary for a customer project** : A seeming disadvantage of employing unification is that the customer dictionary is not self-contained, since it consists only of those words or data that are distinct from those in the other two types of dictionaries. This problem can be easily solved by unifying all the dictionaries into one. And a complete dictionary is available if needed.

These advantages support the use of unification in lexicography. Nevertheless, there is a problem with the effect of unification. This consequence and its remedy will be examined in the next section.

4.2 Resolution of Categorical Ambiguity Resulting from Unification

As discussed in Section 2, the question as to in which dictionary a specific word should be stored is determined by the domain where it appears. In other words, only in a particular field, a particular attribute of a word is likely to appear (of course, it is not absolutely certain as to where a particular use of a word will definitely appear or not appear). Thus unifying dictionaries sometimes brings about more categorical ambiguities than is desired. For example, if the use of the word *default* as a verb is dominant or solely in a particular text, the category is consequently stored in the corresponding customer dictionary. Suppose that the more commonly used category noun is stored in the general dictionary, by unifying the two dictionaries, categorical ambiguity will probably result when translating a text.

ArchTran has devised a method to handle this problem by examining a portion of the text before translation, and then assigning a score to the category of a lexical item in the customer dictionary (or the one in the technical dictionary, if the word has no entry in the customer dictionary) relative to the category in the technical or the general dictionary. The one with a higher functional frequency score will suppress those with a lower score. To continue with the example in Section 3.2, suppose that by examining part of the text to be translated we find that for the word *stream*, verb is the dominant category, then the verb in the customer dictionary will be given a higher score than the noun in the technical dictionary. Thus, during translation the verb will be chosen for *stream* if both categories are found in the output structures, or ambiguous parse trees, for a sentence. If the fact has been shown to be the contrary, then the verb in the customer dictionary will be given a lower score than the noun in the technical dictionary. Thus, the noun will be chosen for *stream*. Furthermore, if the functional frequency of the two categories are on a par, equal weighting will be given to them. Which category will be chosen is determined by the weighting of the phrase structure they are in.

The method of deciding the correct category for a word in the ArchTran system will be improved in the near future using probabilistic model.

In the next section, we will examine the unification operation in more detail from the perspective of sentence processing.

5. Unification Methods

We have already discussed the reason why we use dictionary unification and the dictionary unification principles in ArchTran. In this section we will present possible unification methods. In general, there are three ways to unify the lexical information of identical lexical items in different dictionaries.

- [1] **Unifying dictionaries before parsing.** This means that several different dictionaries are unified into one dictionary before any parsing begins. Therefore, only the unified dictionary will be used during dictionary look-up. The advantage of this method is that only one unification action is needed for each lexical item, and thus it saves parsing time. But the shortcoming is that a huge amount of storage space is required for duplicate lexical information in the system, since the merged dictionary and the source dictionaries coexist in the system.
- [2] **Unifying dictionaries during parsing.** This means only the lexical items that need to be unified are unified in the course of dictionary look-up and no external dictionary space is needed. The major advantage of this method is the saving of storage space. Nevertheless, this method also has a shortcoming. It requires a special purpose module to handle the unification in the run time and thus increases sentence processing time. Besides, unification has to repeat when a word needs unifying is encountered again.
- [3] **Unifying dictionaries with a cache during parsing.** This means cache storage is used to hold the information of the lexical items that are most recently unified. That is, when a word is looked up, the cache will be checked to see whether the word is already there or not. If the word is not in the cache and it is stored in more than one dictionary, all its entries in the various dictionaries will first be unified and then put into the cache. As the cache will be checked when a word is encountered, for a word that is already in the cache, no more unification operation is required next time it is input. This method is similar to that of unifying during parsing, except for the step of checking the cache. The advantage of using cache is that there is no external dictionary space needed and it increases the speed of information retrieval by retrieving information from an internal memory space. But the limitation of using cache is that run-time memory can hold only a limited number of unified words. Another shortcoming is the relative complexity in software, because an additional module has to be added to handle caching.

Comparing the above three methods, we chose to adopt the second method, that is, unifying dictionaries during parsing without a cache. There are three reasons for this decision. First, unlike the use of a merged dictionary, it needs no additional dictionary space. Second, as far as time is concerned, although it requires more time than simply looking up a merged dictionary, the time spent in performing run-time unification is rather small in relation to the

whole MT processing time. Third, it is simpler to implement than using cache and there is no run-time memory limitation problem.

6. Unification Implementation

How does the ArchTran system unify dictionaries? Before answering this question, we will give a brief introduction of the organization of ArchTran.

ArchTran can be decomposed into four general components. **Scanner** looks up dictionaries for the information of lexical items. **Parser** uses the lexical information and analysis grammar rules to analyze the input sentences. And then the **transfer** and **synthesis** modules transfer the English sentence structures into their corresponding Chinese sentence structures. Because the acquisition of lexical information is handled by the scanner, we added the unification module at the scanning stage.

In order to unify dictionaries, there is an interactive user interface **environment control** added to ArchTran, through which user can specify which dictionaries to unify and also specify their hierarchical relation to determine their order in unification. The scanner then looks up the dictionaries specified by the environment control and retrieves the information of lexical items. If there is an identical entry stored in different dictionaries, the scanner calls the unification module to unify the information of the word according to the unification principles.

7. Conclusions

In this paper, we discussed why and how a unification-based lexicography is adopted in the ArchTran English-Chinese machine translation system. Besides satisfactorily handling the problem of different usages or translations of identical terms found in various texts, we also showed that by unifying lexical information in different dictionaries, the time used in dictionary construction is saved; dictionary storage space is minimized; the integrity of distinct dictionaries is preserved; the option regarding which dictionaries to be unified is kept open; and all of the lexical information needed to construct a complete customer-oriented dictionary is available by unifying the relevant dictionaries. Although categorial ambiguity might occur as a result of unification, score function is added as a solution. Unification was proved to be a viable approach for lexicography in an operational MT system.

Using unification in lexicography is one of ArchTran's first attempts to extend the scope of application of the technique. Future research will aim at adopting unification into the ArchTran analysis grammar.

Notes

1. This is the concept of sublanguage-oriented MT system. But the scientific fields to which the ArchTran system is applicable is not so limited as, for example, that of the

METEO system, which aims at translating meteorological reports [Hut86]. The ArchTran system intends to translate texts of various scientific fields, such as computer, machinery, and so on.

2. From our experience in translating computer articles, it is observed that for different texts of the same domain, the size of vocabulary does not vary to the same extent as the different usages of identical terms. In addition, size of vocabulary is seldom a concern as cost for memory becomes cheaper and cheaper. As for the patterns of syntactic constructions, for sentences within a specific field, they usually do not exhibit an unwieldy variety of structures.

3. The values of the feature SENSE, i.e. SENSEa1, SENSEa2, etc., can be regarded either as semantic types [Alle87] or as any other representation of word sense. It is beyond the scope of this paper to engage in a discussion of word semantics. Besides, for the sake of simplicity, in this example each sense is given a distinct Chinese translation. This ignores the fact that words may be polysemous and therefore more than one sense may be expressed by a single Chinese word. It also ignores the fact that words may be synonymous and therefore more than one Chinese word may be used to express a sense.

References

[Alle87] Allen, J., *Natural Language Understanding*. the Benjamin/Cummings Publishing Company, Inc., U.S.A., 1987.

[Boit87] Boitet, C., *Software and Lingware Engineering in Modern M(A)T Systems*. GETA, University of Grenoble & CNRS, 1987. Prepared for *Handbook of Machine Translation* (Niemeyer 1987).

[Fuji89] Fujitsu. "ATLAS-G for High-level Industrial Document Translation" in *Electronics News from Fujitsu*. Volume 11, No. 3, March 1989, pp. 1-3.

[Hirs87] Hirst, G., *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge University Press, Cambridge, Great Britain, 1987.

[Huan88] Huang, C-R., "Unification" in *Proceedings of R.O.C. Computational Linguistics Workshops I*. pp. 29-54, Academia Sinica, Taipei, Taiwan, 1988.

[Hut86] Hutchins, W.J., *Machine Translation: Past, Present, Future*. Market Cross House, West Sussex, Great Britain, 1986.

[Nire87] Nirenburg, S., *Machine Translation : Theoretical and Methodological Issues*. Cambridge University Press, Cambridge, Great Britain, 1987.

[Shie86] Shieber, S.M., *Introduction to Unification-based Approaches to Grammar*. Stanford : Center for the Study of Language and Information, U.S.A., 1986.

NTUMT Strategy
for Prepositional-Phrase Attachment

Ka-Wai Chui, Yia-ping Lin,

徐嘉慧

林雅萍

Shuanfan Huang & I-Peng Lin

黃宣範

林一鵬

National Taiwan University

國立台灣大學

Proceedings of ROCLING II (1989)

中華民國第二屆計算語言學研討會論文集 163-186 頁

Abstract

This paper aims to propose a new parsing strategy to tackle the notorious prepositional-phrase attachment problem (PP attachment problem) in our NTUMT system.

First of all, correct PP attachment is determined in our PP-Attachment Table (PAT), which requires both syntactic and semantic analyses of verbs, nouns and prepositions in lexicon. PAT is indeed the component where all the idiosyncratic attachment conditions are specified for prepositions.

As to our parsing strategy, it can be considered as an interaction between two drivers -- Intelligent ATN (IATN) and Phrase Structure ATN (PATN). IATN scans the input sentence leftwards and activates PATN to construct the first bar-level structures. PATN is then responsible for building up structures whenever IATN gives the command. Correct PP attachment is governed by the seven states in the IATN Grammar, giving priority to verbs. Since our system just outputs one parsing tree, for a PP which is ambiguous in attaching both to the preceding verb and the preceding noun, the PP will be assigned to be verb modifier.

NTUMT Strategy for Prepositional Phrase Attachment

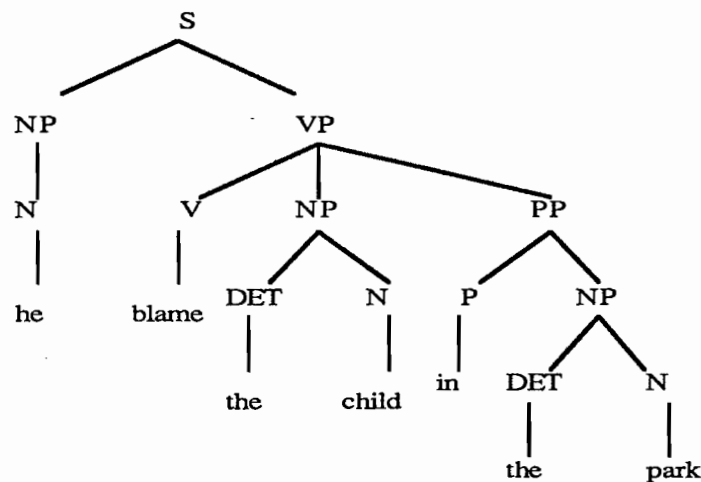
1. The Problem

English prepositional phrases (PP), the postverbal ones in particular, have always been a major problem in parsing. The problem has to do with correctly attaching them to other sentence constituents. This problem of prepositional-phrase attachment (PP attachment) can be exemplified in the following sentence:

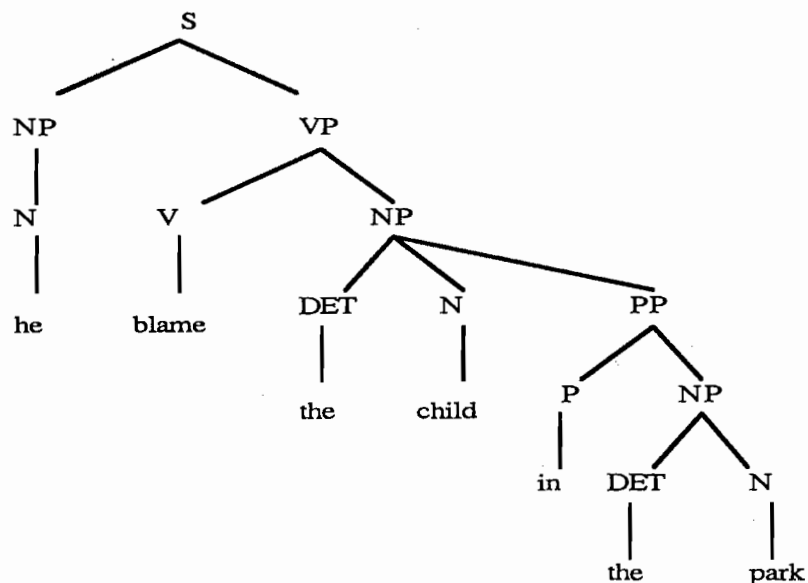
(1) **He blamed the child in the park**

Sentence (1) is ambiguous in that the PP *in the park* can either be a verb modifier, meaning that the whole event happens in the park; or a noun modifier, showing that the child he blamed is in the park, not elsewhere. Two different tree diagrams, (1a) and (1b), then result:

(1a)



(1b)

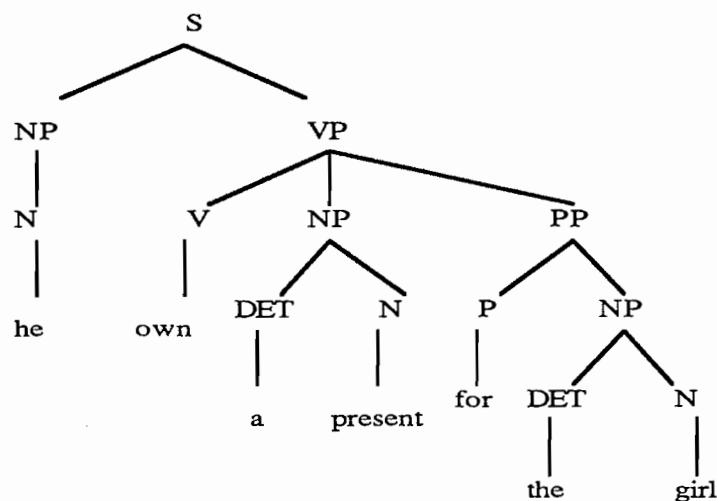


The syntactic ambiguity in sentence (1) does not constitute any semantic anomaly. Disambiguation of this type relies on contextual information. However, some type of attachment is semantically unacceptable. Consider the following sentence:

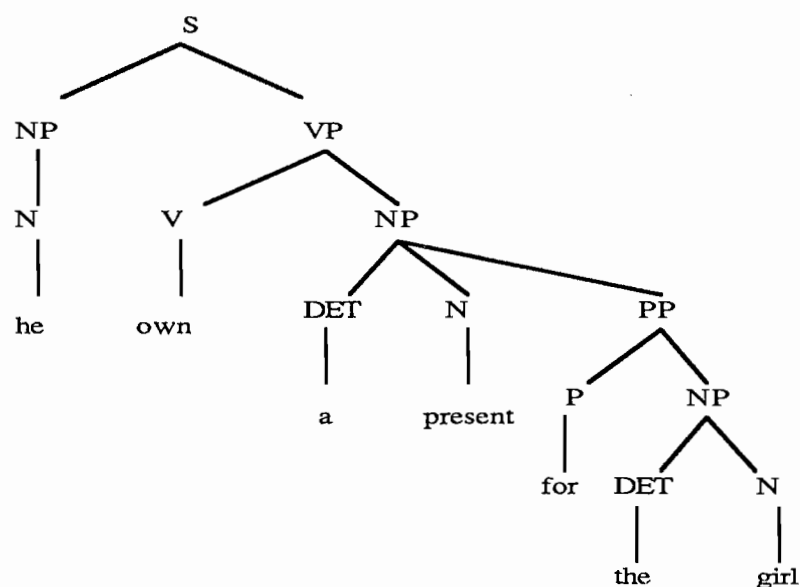
(2) He owned a present for the girl

Sentence (2) shares with sentence (1) in yielding two parsing trees according to our context-free phrase structure rules:

(2a)



(2b)



For the PP *for the girl* to be verb modifier as in (2a) is semantically anomalous. The correct attachment is attaching the PP to the noun *present*. Disambiguation of this type does require both syntactic and semantic analyses of verbs, nouns and prepositions.

In short, PP attachment is not a trivial problem. Of the 929 sentences found in volume 5 and 6 of the English textbooks used by the 3rd-year junior high school students, 36.4% of which include at least one PP. Detailed statistics are displayed in the following table:

	volume 5	volume 6	Total
sentences with at least one PP	152	186	338
sentences without PP	341	250	591
Total	493	436	929

Any English parser should be capable of handling the ambiguous PP attachment as in (1) and (2), but rejecting those semantically unacceptable ones like (2a).

2. Literary Review

In the past, different approaches have been proposed to tackle this PP attachment problem, including Frazier and Fodor's Right Association and Minimal Attachment (1979), Fodor's Lexical Preference (1981), Hirst's Principle of Parsimony (1984), Wilks' Preference Semantics, as well as Schubert's solution by taking syntax, semantics and pragmatics into account. Since these previous treatments are far from satisfactory, Xiuming Huang (1987) presented his most recent resolution for PP attachment in his XTRA system.

Although Huang relies on the integration of syntactic analysis with semantic and contextual interpretation by means of the case preferences of verbs, nouns and prepositions, the *pp-attachment* strategy adopted in his XTRA system suffers from a number of defects. The most serious of which is the inadequacy of the 'seven clauses', which have to be applied sequentially until one succeeds. His clause 1 states that '% check the noun phrase immediately preceding the pp for any case preferences. If its preferences are satisfied then attach the pp to the (Object) np, producing *Rebuilt-Object* (p.115).' Huang obviously takes noun case preference as priority, starting with the noun phrase immediately preceding the PP and working leftwards. This ensures correct PP attachment in those sentences like the following:

(3) He lost the ticket to Paris

According to Huang, the semantic formula for one sense of the noun *ticket* has a direction case (p.114):

sem(ticket1, ... , preps([prep(to), prep-obj(*pla), case(direction)]))

Since *ticket* includes a case preference of direction that matches the preplate for *to*, the PP *to Paris* performs as noun modifier exclusively. Nevertheless, there are counterexamples showing that such priority for nouns may trigger off wrong attachment. Consider the following sentence:

(4) He sent the ticket to Paris

Clause 1 is invalid in (4) because the PP *to Paris* being assigned to the preceding noun, just like (3), violates the semantic rule. Besides, it is the verb *sent* which subcategorizes obligatorily this direction case. PP attachment in (4) depends on the verb, rather than on the noun. Though the PP is by no means subcategorized by the verb *lost* in (3), it is believed that correct attachment can still rely on the properties of the verb and the preposition itself in that the direction indicated by *to Paris* has to co-occur with those locomotion verbs. Since *lost* indicates a state instead, it fails to take a direction case. The PP will thus be assigned to the preceding noun automatically. In short, not only does the discussion above indicate that the sequential application of Huang's seven clauses may constitute wrong attachment, but using noun case preference as priority further neglects the importance of verb subcategorization, as well as the semantic properties of verbs.

In the following, the parsing strategy adopted in our NTUMT system will be introduced. The strategy ensures correct PP attachment for the 338 sentences in the two English textbooks, which even include two and three postverbal prepositional phrases like the followings:

(5) He drove his car to the market in town

(6) He saw the money on the desk in the room next to mine

Since our system just outputs one parsing tree, for a PP which is ambiguous in attaching both to the preceding verb and the preceding noun, just like sentence (1), the PP will be assigned to become verb modifier as verbs always maintain priority in our system.

3. NTUMT Strategy for PP Attachment

In our NTUMT system, the syntactic and semantic analyses of verbs, nouns and prepositions in lexicon are an integral part of the system. The PP Attachment Table (PAT) is devised to base on these types of information in determining correct PP attachment. Besides, the essence of our parsing strategy gives priority to verb, and emphasizing the surface order of input sentences in that the passive counterpart of (4) allows the PP to be noun modifier:

(7) The ticket to Paris was sent by him

As a matter of fact, (7) does not cause any attachment problem.

In the following, the syntactic and semantic analyses of verbs, nouns and prepositions in lexicon, the PP Attachment Table, together with the parsing strategy, will be introduced respectively.

3.1 Lexicon

3.1.1 verbs

The syntactic and semantic analyses of verbs are crucial to PP attachment. Not only do they provide the subcategorization information which shows whether the PP in question is an argument or a modifier, such as *to Paris* of *sent* in (4); but they also specify the co-occurrence restrictions with PP by virtue of their own semantic feature, such as *lost* in relation to the PP in (3).

Take *push* for instance, it is such a ditransitive verb that it requires two arguments-- a noun phrase and a prepositional phrase. According to our classification, the syntax of *push* belongs to T9:

(push push (V (SUBCAT T9)))

The semantic interpretation of verb comprises case and feature assignment. Hence, *push*

assigns *patient* to the following concrete noun, the case of *goal* to the prepositional phrase. Together with the semantic property of the verb, being assigned the feature+*action*, all these information of *push* are represented in lexicon as follows:

(push push (V (SUBCAT T9) (F +action)) (T9 ((SUBJ agent +animate) (OBJ1 patient +concrete) (PP goal NIL))))

3.1.2 nouns

As argued in section 2, using case preference of noun to solve the PP attachment problem does not guarantee correct attachment in every situation, mainly because PP is usually nominal modifier, rather than argument. Take example (3) for instance, the noun *ticket* co-occurring with a direction case *to Paris* may also co-occur with a benefactive role as in (8):

(8) He lost the ticket for Mary

Therefore, it is unreasonable to specify just the direction case while the benefactive case is out of consideration. However, specifying exhaustively the possible optional PP a particular noun may take is also uneconomical, especially the information from verb and preposition are so explicit and bountiful. In lexicon, optional PP will not be specified for nouns.

Of course, we are not denying the co-occurrence restrictions of certain PP to particular nouns. On the contrary, there are certain nouns which do subcategorize PP, e.g. the time noun (represented as *Nt* in our classification) in (9):

(9) It is time for winter vacation

The subcategorized information will be specified for those nouns. Besides, every noun has its own inherent nominal features as exemplified below:

(time time (NOUN (SUBCAT Nt) (NUM SG)) (Nt (F +time) (P event)))

3.1.3 prepositions

Prepositions are the main character in PP attachment. Their syntactic and semantic analyses are the input information to PAT.

Firstly, not all of the prepositions subcategorize noun phrase exclusively. *by*, for example, subcategorizes either a noun phrase (PREP1 in our classification) or a gerundive phrase (PREP6 in our classification).

(10) He went to school by bus

(11) He earned money by writing stories

These syntactic information can help solve the attachment problem to a certain extent as *by* being *PREP6* always modifies verb. They are represented in our lexicon as:

(by by (P (SUBCAT PREP1 PREP6)))

Secondly, for each subcategorization, semantic analysis will be provided in form of case and feature. In fact, it is possible for a preposition to carry different semantic cases. For instance, *by* being *PREP1* may perform the roles of location, instrument, time, and agent, which are exemplified below:

(12) location: He stood by the window

(13) instrument: He went to school by bus

(14) time: He will finish his homework by tomorrow

(15) agent: He was hurt by the dog

Deciding what semantic role(s) a preposition takes depends on many factors. The first is the noun that follows. Thus, the PP in (14) suggests *time* mainly because *tomorrow* is a time noun. The second factor is the semantics of verb in that for the PP *by the window* in (12) to be a location, the verb has to indicate a state, just like *stood*. The last factor is

sentence type The PP in (15) is agentive mainly due to the passive construction in which it appears. The semantic cases, together with the various types of conditions, are represented in lexicon as follows:

**(by by (P (SUBCAT PREP1 PREP6)) (PREP1 (location +state)
(instrument +vehicle) (time +time) (agent +passive)) (PREP6
(event)))**

Of course, there are cases that rely on none of the factors stated above for identification. The case of goal in (16) is always subcategorized by verb, and no further condition is specified.

(16) goal: He put the money in his pocket

3.2 PP Attachment Table (PAT)

As mentioned before, the input information to the PP Attachment Table (PAT) comes from the syntactic and semantic analyses of prepositions. The function of PAT is specifying the attachment conditions idiosyncratic to each preposition. This section aims to explain the function and details of PAT.

Firstly, some of the semantic cases, no matter they are subcategorized or not, always modify verbs, rather than the preceding nouns. They are the cases of goal, instrument, end, commitative etc. Other prepositions, such as *like*, *of*, are usually noun modifiers. Since their presence ensure correct attachment, we devise two markers to show these special properties: *VPP* to those cases which have to be attached to verbs exclusively; *NPP* to those prepositions that only modify nouns. In PAT, they are represented as:

(on (goal (VPP NIL)) ...)

(like (NIL (NPP NIL)) ...)

For the rest that may be attached to verbs and nouns, such as location, both markers are then assigned to them simultaneously. It is this type of PP which constitute the attachment problem. Further conditions are needed to clarify their status. The conditions come chiefly from verbs. Consider the following sentences:

(17) He read a letter from Jack

(18) He bought a car from Jack

The PP *from Jack*, which plays the role of source, is both optional in (17) and (18). Yet, semantics rules out the attachment to the verb *read* in (17), but not to *bought* in (18). It is because *read* is such a non-locomotion verb that it fails to co-occur with a source case. This kind of knowledge is then specified in PAT for *from* so that for it to be a noun modifier, the verb should carry the feature *-locomotion*. These restrictions are represented in PAT as:

(from (source (NPP -locomotion) (VPP NIL)))

The general format for every preposition in PAT is:

(prep_word (case (attachment condition)*)*)*

where *attachment ::= NPP | VPP*

*condition ::= semantics_of_the_ preceding_verb |
 semantics_of_the_ preceding_noun |
 semantics_of_the_noun_after_ prep |
 sentence type*

In conclusion, PAT specifies the attachment conditions for every preposition, including whether a particular case is *VPP* or *NPP*. For those that can be both, further conditions are then provided.

3.3. Parsing Strategy

According to the statistics, 36.4% of English sentences found in the two textbooks include at least one PP, our parser thus takes the problem of PP attachment into serious consideration. Therefore, our discussion of parsing strategy is also subject to PP attachment only. The whole framework can be clearly shown in Figure 1:

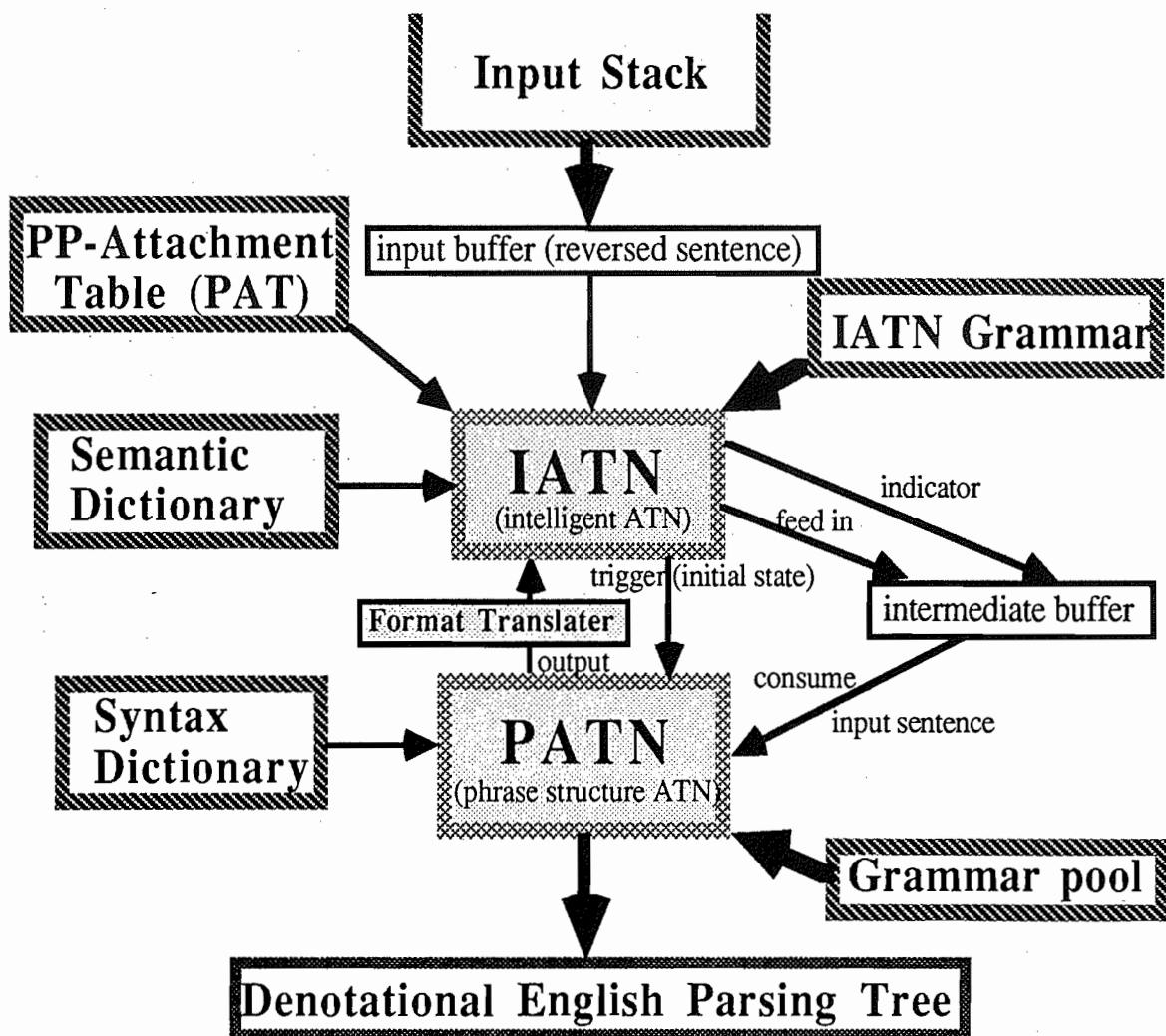


Figure1: the framework of NTUMT system

Simply speaking, our parsing strategy can be thought of as an interaction between two processors -- Intelligent ATN (IATN) and Phrase Structure ATN (PATN). The Input Stack stores all the possible category combinations of the words in the input sentence. It feeds IATN one combination each time. IATN then starts scanning the input combination leftwards, and activating PATN to construct the first bar-level structures (according to the X-bar grammar).

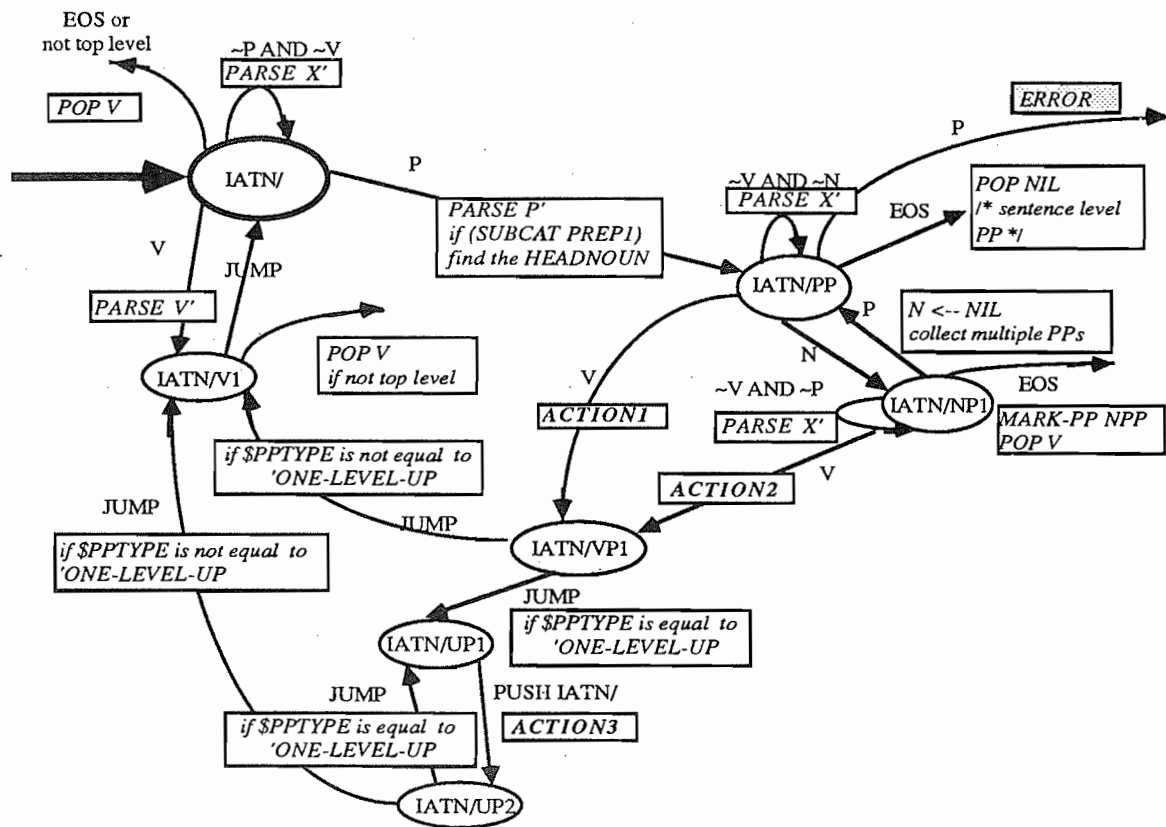
The IATN grammar, which comprises seven states, is capable of solving the problem of PP attachment. They will be discussed individually in the following:

1. The initial state is IATN/, which instructs PATN to build up structures to first bar level. Moreover, whenever a preposition is encountered, it goes to the second state -- IATN/PP.
2. In the state IATN/PP, it tries to find a preceding noun or verb. If none is found, the PP should be the sentence modifier. However, if it meets a noun, the semantic information of the noun will be stored in the register N and then enters into another state IATN/NP1. If it is a verb instead, the function of PP-ATTACHMENT will be called. It searches for the attachment conditions from PAT. The result value, which is either *NPP* or *VPP*, will be added to the PP1/ list in the Intermediate Buffer. Afterwards, it enters IATN/VP1.
3. IATN/NP1 so far includes the information of a prepositional phrase and the noun coming from IATN/PP. When it moves on and finds a preposition, another PP will be grouped together again. The nominal information originally stored in register N will be pushed into the register HEADNOUN, while the prepositional information will be pushed into the register P. In short, the loop -- IATN/PP --> IATN/NP1 --> IATN/PP -- groups as many preposition phrases as possible in the sentence. However, if the

current word is a verb instead, PP-ATTACHMENT will be called. Then, it enters the state of IATN/VP1.

4. In IATN/VP1, in case the verb does not satisfy the attachment conditions, the grammar will go to IATN/UP1 to find another verb of higher-level for attachment. Only after the attachment conditions have been met does it jump to IATN/V1.
5. Under the condition that the PP fails to attach to the verb or the noun of the most proximate clause, IATN/UP1 is then responsible for finding another verb in the higher clause.
6. In IATN/UP2, the information of the verb will be checked against the attachment conditions. If they are still not satisfied, the grammar will go back to IATN/UP1. Thus, IATN/UP1 and IATN/UP2 form a loop until the attachment condition have been satisfied. The grammar then enters the last state-- IATN/V1.
7. The last state is IATN/V1. It either returns the semantic information of the 'qualified' verb to the previous IATN for PP attachment, or goes back to the initial state to process the rest of the words in the sentence.

The complete State Transition Diagram for IATN grammar is shown in Figure 2 below:



NOTES

- ACTION1** (PP-ATTACHMENT V, NIL, P, HEADNOUN)
(PARSE V")
- ACTION2** (PP-ATTACHMENT V, N, P, HEADNOUN)
(PARSE V")
- ACTION3** (COND ((* (PP-ATTACHMENT * NIL P H EADNOUN))
(T (MARK-PP VPP))))

Figure 2: State Transition Diagram for IATN grammar

Finally, several examples are provided in the Appendix to show how IATN and PATN solve the PP attachment problem.

4. Conclusion

By the help of the detailed syntactic and semantic analyses of verbs, nouns and prepositions in lexicon, as well as the attachment conditions in PAT, our parsing strategy can make correct PP attachment for the 338 sentences found in the two English textbooks. For those PPs that are semantically ambiguous, like sentence (1), disambiguation relies on contextual information, which is beyond the ability of our NTUMT system. Further research in this area will be needed.

In fine, since our system just provides one result, the ambiguous prepositional phrase will be assigned to the verb.

Appendix

Example 1: (19) He loved the shirt in the closet

Format V-NP-PP

IATN/ state *	next_state	intermediate buffer after PATN parsing
IATN/ closet	IATN/	((7 7 (NP1/ (ROOT . <closet>) ...)))
IATN/ the	IATN/	((6 6 (DETP1/ (ROOT . <the>) ...) (7 7 ...))
IATN/ in	IATN/PP	((5 7 (PP1/ (ROOT . <in the closet>) (SUBCAT PREP1))))
IATN/PP shirt	IATN/NP1	((4 4 (N (SUBCAT ...)) (5 7 ...))
IATN/NP1 the	IATN/NP1	((3 3 (DETP1/ (ROOT . <the>))) (4 4 ...) (5 7 ...))
IATN/NP1 loved	IATN/VP1	((2 2 (V ...) (3 3 ...) (4 4 ...) (5 7 ... (ATTACH . NPP)))
		/* after IATN call function PP-AGREEMENT */
	==>	(((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))
IATN/VP1 he	IATN/V1	(((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))
IATN/V1 he	IATN/	(((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))
IATN/ he	IATN/	((1 1 (NP1/ (ROOT . <he>) (SUBCAT . Npro))) (2 7 ...))
IATN/ EOS		POP

Example 2 : (20) He met the girl in front of the restaurant at noon

Format V-NP-PP-PP

IATN/ state *		next_state	intermediate buffer after PATN parsing
IATN/	noon	IATN/	((9 9 (NP1/ (ROOT . <noon>) ...)))
IATN/	at	IATN/PP	((8 9 (PP1/ (ROOT . <at noon>) (CASE time)...)))
IATN/PP	restaurant	IATN/NP1	((7 7 (N (SUBCAT ...)) (8 9 ...)) /* N <-- restaurant; HEADNOUN <-- noon; P <-- at*/
IATN/NP1	the	IATN/NP1	((6 6 (DETP1/ (ROOT . <the>)) (7 7 (N (SUBCAT ...)) (8 9 ...)))
IATN/NP1	in_front_of	IATN/PP	((5 5 (P (CASE loc)...)) (6 6 (DETP1/ (ROOT . <the>)) (7 7 (N (SUBCAT ...)) (8 9 ...))) /* N <-- NIL; HEADNOUN <-- (restaurant . noon); P <-- (in_front_of . at) */
IATN/PP	girl	IATN/NP1	((4 4 (N ...)) (5 5 (P ...)) (6 6 (DETP1/ (ROOT . <the>)) (7 7 (N (SUBCAT ...)) (8 9 ...))) /* N <-- girl; HEADNOUN <-- (restaurant . noon); P <-- (in_front_of . at) */
IATN/NP1	the	IATN/NP1	((3 3 (DETP1/ (ROOT . <the>) ...) (4 4 ...) (5 5 ... (6 6 ...) (7 7 ...) (8 9 ...)))
IATN/NP1	met	IATN/VP1	((2 2 (V (ROOT . <meet>) (F))) (3 3 ...) (4 4 ... (5 5 (P (CASE loc) (ATTACH . VPP)...)) (6 6 ... (7 7 ...) (8 9 (PP1/ (ATTACH . VPP)...))) /* after IATN call function PP-AGREEMENT */
		==>	((2 7 (VP1/ (ROOT . <met the girl in_front_of the

restaurant at noon>))))

IATN/VP1 he IATN/V1 ((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))

IATN/V1 he IATN/ ((2 7 (VP1/ (ROOT . <loved the shirt in the closet>))))

IATN/ he IATN/ ((1 1 (NP1/ (ROOT . <he>) (SUBCAT . Npro))) (2 7 ...)

IATN/ EOS POP

Example 3: (21) He saw the money on the desk in the room next to mine

Format V-NP-PP-PP-PP

IATN/ state *	next_state	intermediate buffer after PATN parsing
IATN/ mine	IATN/	((12 12 (NP1/ (ROOT . <mine>) ...)))
IATN/ next_to	IATN/PP	((11 12 (PP1/ (ROOT . <next_to mine>) (CASE loc)...)))
IATN/PP room	IATN/NP1	((10 10 (N (SUBCAT ...)) (11 12 ...)) /* N <-- room; HEADNOUN <-- mine; P <-- next_to*/
IATN/NP1 the	IATN/NP1	((9 9 (DETP1/ (ROOT . <the>))) (10 10...) (11 12 ...)))
IATN/NP1 in	IATN/PP	((8 8 (P (CASE loc)...)) (9 9...) (10 10 ...) (11 12 ...))) /* N <-- NIL; HEADNOUN <-- (room . mine); P <-- (in . next_to) */
IATN/PP desk	IATN/NP1	((7 7 (N ...)) (8 8 (P (CASE loc)...)) (9 9...) (10 10 ...)) (11 12 ...))) /* N <-- desk; HEADNOUN <-- (room . mine); P <-- (in . next_to) */
IATN/NP1 the	IATN/NP1	((6 6 (DETP1/ (ROOT . <the>))) (7 7 ...) (8 8 ...) (9 9 ...) (10 10 ...) (11 12 ...)))
IATN/NP1 on	IATN/PP	((5 5 (P (CASE loc) ...)) (6 6 ...) (7 7 ...) (8 8 (P (CASE loc)...)) (9 9...) (10 10 ...) (11 12 ...))) /* N <-- NIL; HEADNOUN <-- (desk room . mine); P <-- (on in . next_to) */

IATN/PP	money	IATN/NP1	((4 4 (N ...) (5 5 ...) (6 6 ...) (7 7 ...) (8 8 (P ...)) (9 9...) (10 10 ...) (11 12 ...)) /* N <-- money; HEADNOUN <-- (desk room . mine); P <-- (on in . next_to) */
IATN/NP1	the	IATN/NP1	((3 3 (DETP1/ (ROOT . <the>))) (4 4 ...) (5 5 ...) (6 6 ...) (7 7 ...) (8 8 ...) (9 9 ...) (10 10 ...) (11 12 ...))
IATN/NP1	saw	IATN/VP1	((2 2 (V (ROOT . <saw>) (F +perceptual))) (3 3 ...)) (4 4 ...) (5 5 (P (CASE .NPP)...)) (6 6 ...) (7 7 ...) (8 8 (CASE . NPP)...)) (9 9 ...) (10 10 ...) (11 12 (PP1/ (ATTACH . NPP)...))) /* after IATN call function PP-AGREEMENT */ ==> ((2 12 (VP1/ (ROOT . <saw the money on the desk in the room next_to mine>))))
IATN/VP1	he	IATN/V1	((2 12 (VP1/ (ROOT . <saw the money on the desk in the room next_to mine>))))
IATN/V1	he	IATN/	((2 12 (VP1/ (ROOT . < saw the money on the desk in the room next_to mine>))))
IATN/	he	IATN/	((1 1 (NP1/ (ROOT . <he>) (SUBCAT . Npro))) (2 12 ...))
IATN/	EOS	POP	

References

- Bates, M. (1978) "The Theory and Practice of Augmented Transition Network Grammars" Natural Language Communication with Computers (Lecture notes in computer science 63). Berlin: Springer-Verlag, 1978, pp. 191-259
- Fass, D. & Wilks, Y. (1983) "Preference Semantics, III-Formedness, and Metaphor" American Journal of Computational Linguistics, vol 9, NO. 3-4, Jul-Dec 1983, pp. 178-187
- Fodor, J.D. & Frazier, L. (1980) "Is the human sentence parsing mechanism an ATN?" Cognition, vol 8, 1980, pp. 417-459
- Frazier, L. & Fodor, J.D. (1978) "The sausage machine: a new two-stage parsing model" Cognition, 6 (4), Dec 1978, pp. 291-325
- Hirst, G. (1983) "A Foundation for Semantic Interpretation" Proc. of ACL-83, Cambridge, Massachusetts, June 1983, pp. 64-73
- Hirst, G. (1984) "A Semantic Process for Syntactic Disambiguation" Proc. of AAAI-84, Austin, pp. 148-152
- Huang, X. (1987) "XTRA: The Design and Implementation of a Fully Automatic Machine Translation" PhD. Thesis
- Lin, Yia-ping. (1989) "Embedding X' Theory in ATN System to Solve PP Attachment Problem" M.S. Thesis, National Taiwan University.
- Schubert, L.K. "Are There Preference Trade-off in Attachment Decisions?" Proc. of AAAI-86, pp. 601-605
- Shieber, S.M. (1983) "Sentence Disambiguation by a Shift-Reduced Parsing Technique" Proc. of IJCAI-83, Karlsruhe, W.Germany, pp. 699-703

- Wanner, E. (1980) "The ATN and the Sausage Machine: Which one is Balony?" *Cognition*, vol. 8, 1980, pp. 209-225
- Wilks, Y.A., Huang, H.H. & Fass, D.C. (1985) "Syntax, Preference and Right Attachment" *Proc. of IJCAI 85*, los Angeles, California, pp. 779-784
- Woods, W.A. (1969) "Augmented Transition Networks for Natural Language Analysis" *Harvard Computational Laboratory Report No. CS-1*, Harvard University, Cambridge, MA., 1969
- Woods, W.A. (1970) "Transition Networks Grammar for Natural Language Analysis" *Communications of the ACM*. 13 (1970), pp. 591-606
- Woods, W.A. (1973) "An Experimental Parsing System for Transition Network Grammars" *Natural Language Processing*, Randall Rustin, ed., New York: Algorithmics Press, 1973

Systemic Generation of Chinese Sentences

Hwei-Ming Kuo
Jyun-Sheng Chang

Institute of Computer Science
National Tsing Hua University

Abstract

In this paper, we have designed and implemented a generator for Chinese sentences. The generator uses the systemic grammar as the explicit representation of the syntax of Chinese sentences. We have also augmented the generative mechanism of systemic grammar with procedural attachment to make the generator more adaptable to different kinds of input.

1. Introduction

In Section 1, we introduce the general concepts of text generation and systemic grammar. In Section 2, the overall picture of our sentence generator is described. The grammar and the generating process of the generator are discussed in Section 3 and 4.

1.1 Text generation

Text generation is already established as a research area within computational linguistics [Mann 1982]. Up to late 1970's, researchers had tried putting many linguistic theories into sentence generating systems. [Goldman 1975, Grishman 1979, and Shapiro 1979]. These systems can generate more accurate, elegant and readable sentences. But the limitation is that they only convert an isolated chunk of the system's knowledge into an isolated sentence, so their expressive ability is very restricted.

Around 1980, the growing interest in discourse and pragmatics led to development of systems that could produce multi-sentence text [Derr-McKeown 1984, Mann 1984, McDonald-Pustejovsky 1985 and McKeown

1985]. Methodology used in text generation was also the subject of study [Danlos 1984 and Vaughan-McDonald 1986].

1.2 Text generation model

The generally accepted model of text generation consists of mainly the following three phases:

1. *Content determination,*
2. *Text planning,* and
3. *Surface generation.*

A better surface generator usually has three components, each exploiting different kinds of linguistic knowledge: (1) a formal representation of the sentence structure in the language. Several grammar formalisms have been used for surface generation: *Systemic grammar* [Halliday 1976], *Transformational grammar*, *Augmented Transition Network (ATN) grammar* [Woods 1970], and *Functional grammar*. (2) a dictionary containing various information such that proper words for represent concepts and entities conveyed in the messages. (3) a way of doing syntactical and lexical choice.

1.2 Systemic grammar

Systemic grammar, is a linguistic theory developed by M.K.A. Halliday and others at the University of London. Its development is somewhat independent of American generative linguistics and it approaches language structure from a different starting point. Systemic grammar emphasizes the functional organization of a language and tries to answer questions like: what are the functions of language? how does language fulfill these functions? and how does language work? Linguists of this school observes regularities of language patterns people used to achieve some social activities. And hence, they classify the syntactic objects according to the roles which play in interaction, and claim that there exists a relationship between form and meaning of these syntactic objects. The detailed descriptions can be found in [Halliday 1973, 1976 and 1985].

1.2.1 The Choice System

In systemic grammar, the functions of a language are not a haphazard mixture, but can be analyzed as belonging to different systems that operate simultaneously in determining the structure of a sentence. The interdependencies among dimensions and choices can be represented in formal structures known as system networks. A system network is a list of choices representing the options available to the speaker. System network can be written in a simple graphic notation; the basic elements of which are illustrated in Figure 1.

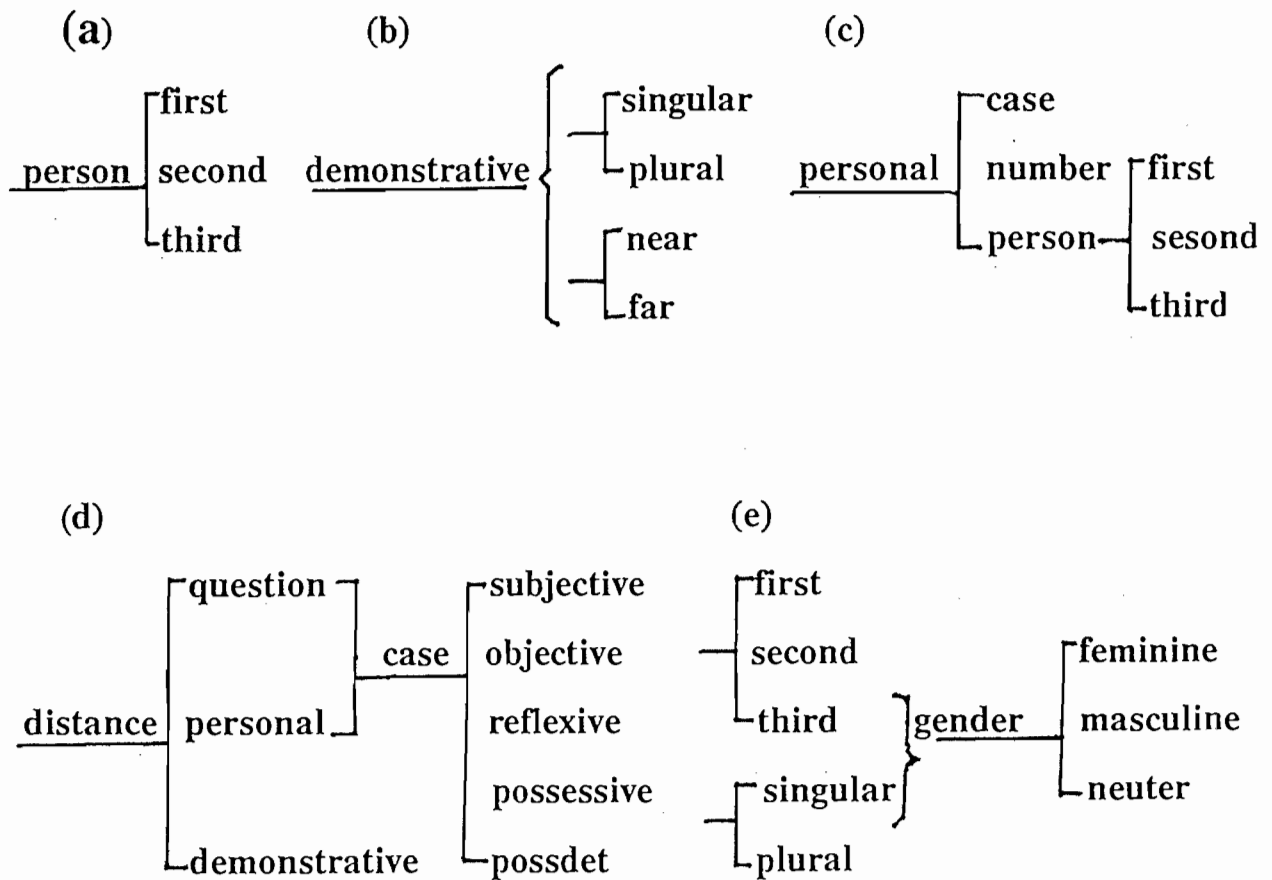


Figure 1 The symbols used in systemic networks

Basically, there are four symbols used in system networks to represent the structures of a choice. They are '[', '{', ']', and '|'. The first two symbols used represent what kind of selection we can make in a choice system. The symbol '[' represents an exclusive choice. For example, in Figure 1(a), we can choose *first*, *second*,

or third. When there are more than one set of co-occurring choices at a point, we use the symbol '{'. In Figure 1(b), four feature combinations are possible: {singular,near}, {singular,far}, {plural,near}, or {plural,far}. The choice system can have a name, which is written above a horizontal line extending to the left of the symbol '[' or '{', such as case in Figure 1(d) and gender in (e)

Each choice system has an entry condition determining whether it is applicable. When the entry condition is a special feature, we directly connect the choice system to the feature as shown in Figure 1(c). When the entry condition is the simultaneous (AND) or alternative (OR) of more than one feature, we use the symbol ']' to indicate an OR relationship, and the symbol '}' to indicate the AND relationship. For example, in Figure 1(d), the choice system case is applicable if either question or personal is selected, while in Figure 1(e), gender is applicable if both third and singular are selected. The elegance and power of this notation can be seen from the pronoun system for English shown in Figure 2.

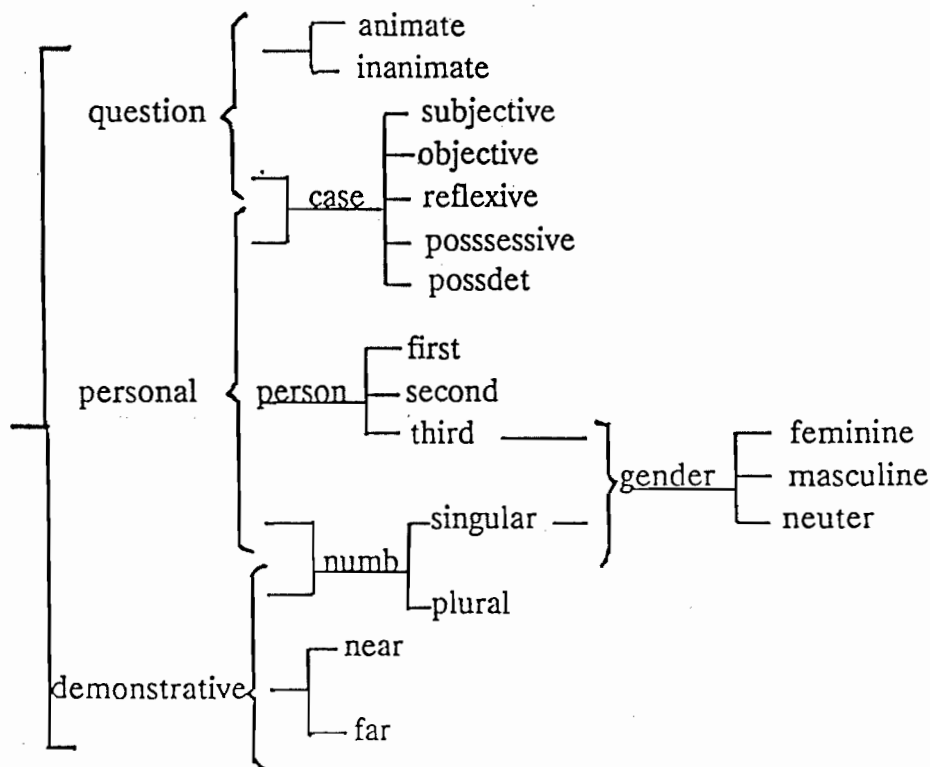


Figure 2 The system network for English pronouns

1.2.2 Realization

To generate surface structure of a clause, some realization rules are attached to the nodes in system network. There are two types of realization rules in systemic grammar. The computational model of language generation is usually decomposed into several strata. The first type of rules are given to prescribe how patterns on one stratum correlate with patterns of other strata. The second type of rules are given for the analysis the relation among patterns within a single stratum. In our work, we implement only rules of the second type. There are three different kinds of second type of rules.

1. *feature-realization rules* -- indicating which functions realize the feature environment that it summarizes.
2. *structure-building rules* -- either specifying how various functions are added to fill out the partial structure generated by feature-realization rules, or prescribing the partial order in which these functions finally appear in a surface sentence.
3. *function-realization rules* -- indicating how the functions should be realized by features of smaller items in the next layer or lexical entries in the dictionary.

The generative process is illustrated in Figure 3.

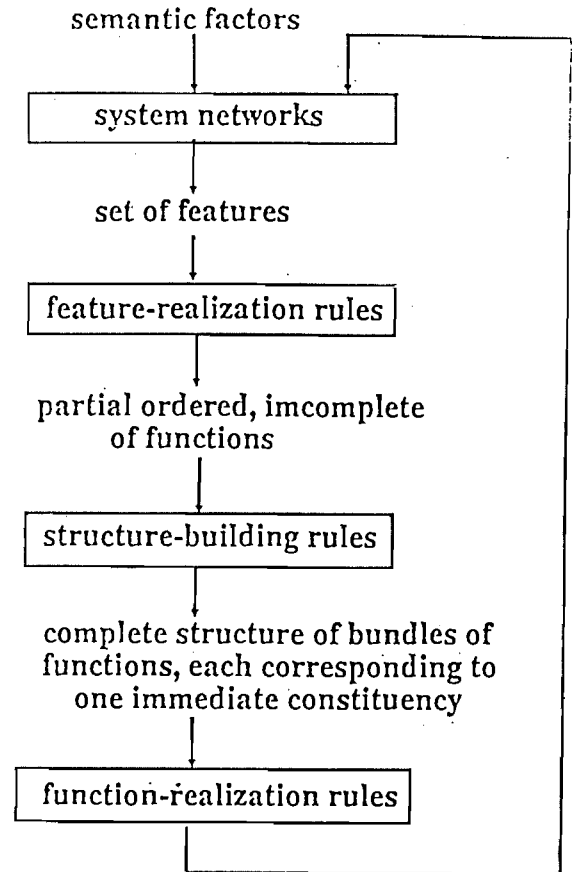


Figure 3 The generative process

2. A Chinese sentence generator

2.1 Form of the input

To prepare for different ways of using the sentence generator, we design its form of input to be as general as possible. When it is to be connected to another system, a simple pre-processor can be included to transform the output generated by the system to this form. We adopt a *frame-like notation* for the input. The frame has three parts -- frame name, a list of features, and an optional list of subframes. The frame name denotes the constituent of the sentence that the systemic network is to generate. The list of features provides the information about the functions that this constituent is intended to perform. The optional subframe list gives the subconstituents that are to be handled by the lower level network. So the subframes have exactly the same structure that we have described. For example, the input of the description of a sentence - "I give him a book," is as follows:

```
(sentence (s-sentence)
  (clause (independent mood indicative transitivity
    transitive active double-obj)
    (agent (np head-noun pronoun (head-noun i)))
    (pred (vp (verb give)))
    (obj-affected (np head-noun pronoun (head-noun he)))
    (patient (np head-noun noun noun-mod
      class-phr (head-noun book))
      (classp (cp number (num one) (class ben))))))
```

The name of top-level frame is *sentence*, and the features in the feature list indicate that we want a simple sentence which is to be realized by the subframe named *clause*. The *clause* is independent, indicative, active, and is composed of predicate, agent, patient, and affected object, all of which are to be realized in term by some other lower level structures. Using recreative definition like this, we are able to express any relationship between components of different level in a sentence. In Section 4, we will present examples to illustrate how the surface generator processes its input.

2.2 Node representation

In order to make the graphic representation of a system network readable to the program, a linear format is necessary. To record the information about the complicated relationship between the nodes in choice networks, we use the following form for nodes:

1. *Name of node* -- Each node has a unique name.
2. *Entry condition* -- The entry condition of a node could be a special feature or the combination of features. In the former case, we put in the feature name directly. In the later case, we use the *and-expression* and the *or-expression* to indicate a simultaneous or alternative condition.
3. *Next nodes* -- There are two kinds of relationship between current node and its successors: *co-occurring* and *exclusive*. We use an *and-expression* and an *amo-expression* to represent them respectively.
4. *Realization rules* -- Various rules are encoded for realizing the feature. Details are given in Section 2.5.
5. *Processing order* - In the input fed to our system, the features chosen are put into an unordered list. But, for efficiency consideration in checking the entry condition, we rearrange the sequence of the features according to the number recorded in this field. The smaller the number, the earlier is the node checked.

Below is an example of a node :

```
(def_node non-transitive
  (entry_conditions transitivity)
  (next_nodes      (amo adj-verb serial-verb other-verb))
  (realization_rules NIL)
  (level 6) )
```

2.3 The grammar

The major sources of linguistic material motivating the development of the grammar used in our sentence generator came from the analysis of Li and Thompson [Li-Thompson 1982], and some functional linguistic theories proposed by Tang [Tang 1985]. Turning these descriptive treatments of Chinese sentences into a formal,

Computational grammatical formalism is the most important part of this paper. A few observations of our own are also included.

As for the grammatical formalism of our sentence generator, we adopt the systemic tradition for the following reasons:

1. It is based on the function of language and emphasizes the mechanism of choice according to the functions. That corresponds closely to the nature of the generation process.
2. The phases before surface generation produce a lot of functional features according to which the systemic grammar is mainly structured.

We will describe the details of the grammar for a subset Chinese sentences in Section 3.

2.4 Control Mechanism

To generate a sentence, the generator first navigates through the choice network and make a proper decision at each choice point according to the input given. At the same time, the system also checks the consistency between the features selected and collects the realization rules of those features if no conflict occurs. After processing the features given in the same level of the input frame, the generator executes the realization rules collected in the order given below:

1. *the feature-realization rules* -- These rules specify how functions are included to realize the features. A confluence of functions is necessary if the same item performs multiple functions. The classification of these functions is also specified as the criteria for the lexical choice.

These rules which are used in our system include:

- | | |
|--------------|---|
| (a) (+ X) | The function X must be present. |
| (b) (= X Y) | The two functions, X and Y, must be conflated.
This means that the two functions will be filled by the same constituent. |
| (c) (+= X Y) | The function X must be present and must be conflated with the function Y. |
| (d) (/ X Y) | The constituent filling the function X must have the characteristic Y. |

(e) (+/ X Y) This rules prescribe both (+ X) and (/ X Y).

2. *the structure-building rules* -- These rules specify the relationship of partial order listed in the rules to construct the total order of the functions.

These rules which are used in our system include:

- (a) (> X Y) The constituent filling the function X must appear before that filling the function Y.
- (b) (>> X) The constituent filling the function X must appear at the last position in the structure.
- (c) (<< X) The constituent filling the function X must appear at the first position in the structure.
- (d) (+> X Y) This rule prescribe both (+ X) and (> X Y).
- (e) (- X) The function X must not be present and all other realization rules related to function X must be cancelled.

3. *the function-realization rules* -- These rules specify proper items in the dictionary for functions that can not be decomposed further. For other functions, the relevant subframes in the input will be extracted and go through step 1-3.

These rules which are used in our system include:

- (a) (! X) The function X must be realized by the item picked out from the dictionary according to its characteristic specified by other rules.
- (b) (\$ X Y) The function Y must be realized by using the subnet whose entry node is X.
- (c) (% X Y) When realizing the function X, the rule Y must be carried over to the subnet.

From the above discussion, one realizes that the generation of a sentence involves a lot of choices and the choices are determined by features. However, in general, a needed feature may not be available. The availability of a feature can be one of the following three cases:

1. The feature is available explicitly in the input. Other phases of the text generator have created this feature.
2. The feature is available implicitly in the input.

3. The feature is not available in the input at all.

To account for these three possibilities, we introduce the so-called *procedural attachment* to the systemic network. A procedure is attached to a choice system if the grammar writer feels that there is a possibility that the feature may not be available in the input. The procedure is intended to produce the decision for the choice system. So when a choice system attached with a procedure is evaluated, the feature involved is checked out in the input. If it is present, then the decision is made. Otherwise, the attached procedure is executed to produce the decision from the information available implicitly in the input. If the attached procedure does not produce a decision because no information is present in the input, then the default in the choice system is selected if one is available. If all of the above fail, then a random choice is made.

We have found out that the idea of procedural attachment is very helpful in handling some special phenomena in Chinese sentences [Tang 1985]. More examples involving procedural attachment are given in Section 4.

2.5 Four functional principles

Owing to the different circumstances and goals of communication, many Chinese sentences with the same cognitive content may have different surface realizations. Tang proposed four principles to explain the role of communicative functions in determining the syntactic structure of the sentence [Tang 1985].

1. the "From Old to New" principle
2. the "From Light to Heavy" principle
3. the "From Low to High" principle
4. the "From Close to Distant" principle

So far, we have implemented the "From Light to Heavy" principle in our system. The reason is that this is the only principle that relies solely on syntactic information only. The other three all have something to do with the thematic, pragmatic and some speaker-related information and it can only be handled properly in a relatively complete system. In our system, we adopt a rather general mechanism to realize the principle of "From Light to Heavy"

, so the other principles could be added to our sentence generator easily. We will describe the mechanism in Section 4.

3. Systemic grammar for Chinese sentences

There are many different levels of detail of grammatical items in a language and properties of them can be expressed in a single all-embracing system network. In our network, there are four levels of detail: sentence, clause, phrase, and word, as shown below. We describe the details in the following sub-sections.

grammatical items {
sentence
clause
phrase
word

In this paper, we only describe the clause system. The discussion of the other systems can be found in [Kuo-Cheng 1989].

3.1 The Clause System

Usually, an English sentence can be analyzed according to different systems, such as mood, transitivity, theme and information. In our clause system, a clause can also be analyzed in the same way. In the mood system, a clause can be classified into *indicative*, *imperative*, *presentative*, *interrogate*, or *comparative*, according the functions that it performs. The relationship between these features is shown in Figure 4.

3.1.1 Presentative Clauses

There are three kinds of presentative clauses: *existential*, *positional*, and *motion*. They use verbs of existence, position and motion respectively to introduce an entity into a discourse. The three examples listed below illustrate these three cases.

抽屜裡有三本書
桌子上放了很多鉛筆
到了一批貨

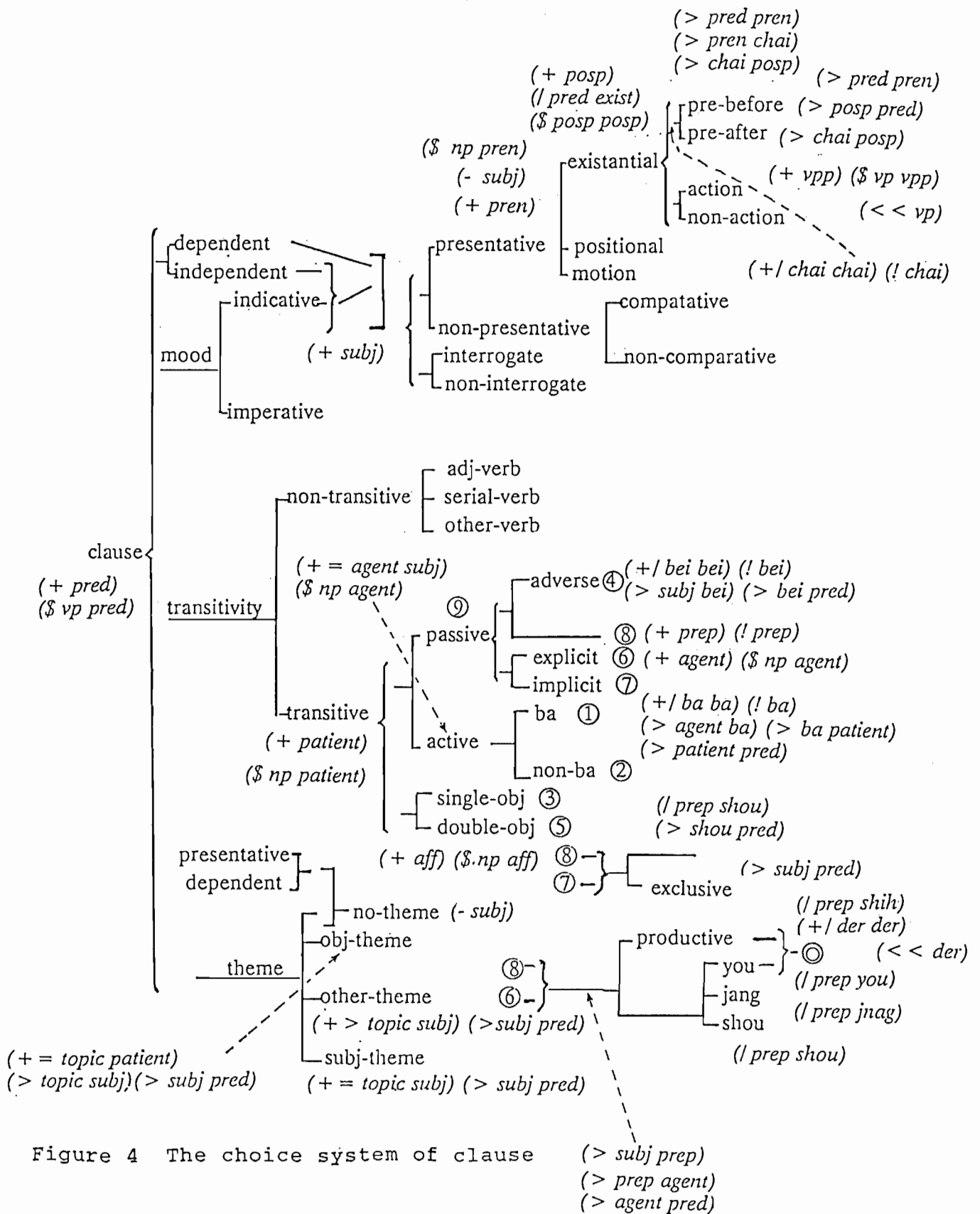


Figure 4 The choice system of clause

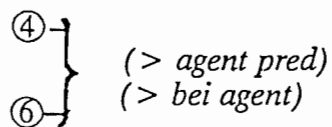
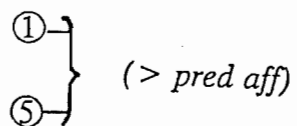
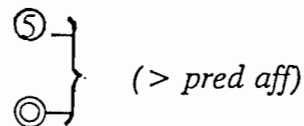
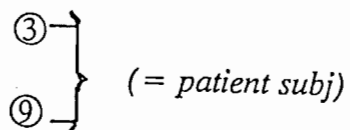
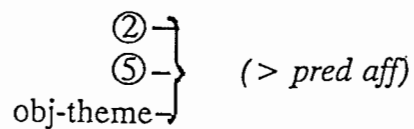
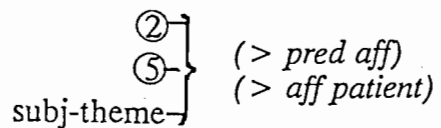
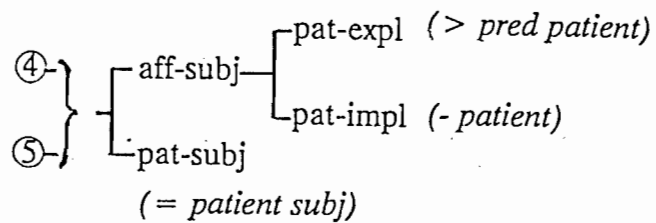


Figure 4 The choice system of clause (continued)

In the existential clauses, the noun phrase presented can appear before or after the *locus*. This decision generally follows the principles "From Old to New" and "From Light to Heavy", as discussed in Section 2.5. We attach a procedure to the network for counting the weight of the presented noun phrase and make a choice according to the weight.

3.1.2 The Transitivity System

In the transitivity system, the choice of *single-obj* or *double-obj* is used to indicate the clause has either one or two participants. Simultaneously, a clause can be *active* or *passive*, indicating either the agent or the patient of an action being the subject of the clause. These are indicated by the *and-link* in the *transitive* node. In the active type, a *ba construction* is used when the verb involved has a disposal favor, and the noun phrase being disposed of is *definite, specific, or generic*. (An action has disposal favor when it involves an object being handled, manipulated, or dealt with.) The choice of *ba construction* is also influenced by the "From Light to Heavy" principle (See Section 2.6).

In the passive type, the *bei construction* is used essentially to express an *adverse* situation, one in which something unfortunate has happened. But the nonadversity usage of the *bei construction* to express the passive meaning of the sentence, has increased in modern Chinese due to the influence of the foreign language, especially English. But many of them are still not acceptable to native speaker of Chinese. In these cases we can use other verbs such as *shou, jang, you*. The agent of the action in the sentence of passive type can be explicit or implicit. The following examples illustrate these phenomena:

1. 小偷偷了他的錢包
2. 他的錢包被小偷偷了
3. 他被小偷偷了錢包
4. 張三誇獎李四
5. 李四受張三誇獎
6. 李四很讓張三誇獎
7. 你來決定這件事
8. 這件事由你決定

If the relationship between the agent and the patient of the clause is *producer-production*, the *shih construction* is usually used in the passive and not adverse circumstance. The following sentences are the examples of *shih construction*.

1. 清華書局出版了這本書
這本書是清華書局出版的
2. 他撰寫了這篇論文
這篇論文是他撰寫的

In the clause of passive type, when the agent is implicit, we may use *shou construction* or put the predicate after the patient directly. The latter only occurs when the class of the agent and the patient are mutually exclusive.

This phenomenon of whether to leave out the word *bei* or *shou* according to the classes of the participants closely parallel to the theory of *semantic preference* proposed by the Wilks [Wilks 1975]. The classes which the participants of a verb can be conveniently recorded in the dictionary. We make this a choice in the systemic network and attach a procedure to the choice to check the exclusiveness between the classes of the participants. A few examples are listed below.

1. 這本書出版了
2. 功課做完了沒有
3. 他的決定受了很大的影響

The first two sentences leave out *bei* and use the *patient-predicate construction*, while the third sentence uses the *shou construction*.

3.1.3 The Theme system

According to the analysis of Li and Thompson [Li and Thompson 1981], most Chinese sentences are *topic-prominent*. The topic of sentence sets a spatial, temporal, or individual framework with which the main predication holds. Except for dependent and presentative clauses, every sentence has a topic. We deal with the topic in the theme system. The topic of a clause could be the subject, the object, or other components of the sentence. The following examples represent these four cases respectively.

1. 前面來了一個人
2. 我送給他一本書
3. 一本書我送給他
4. 這棵樹葉很大

4. The control mechanism

4.1 The Generating Process

1. Set up the environment by reading the dictionary and system network from files and then transfer them into the internal representation.
2. For each sentence that we want to generate, read the frame-like input, $(Name, Feature-list, subframes-list)$, and follow Steps 3-9.
3. Sort the *Feature-list* according to the processing order of each feature.
4. For each feature F in *Feature-list*, do the following steps.
 - 4.1. Make sure that the pre-condition of F stands
 - 4.2. For the next-nodes of F , do the following:
 - (a) If an and-link is encountered, include the features in the expression.
 - (b) if an exclusive-or-link is encountered, do the following:
 - if one of the feature present is in the *Feature-list*, select it, otherwise,
 - if there is a procedure attached to F , use it to select the proper feature, otherwise
 - select the default feature.
5. Collect the realization rules on every feature selected in Step 4.
6. Execute the feature-realization rules and collect the functions included into *Function-list*.
7. Execute the structure-building rules: Find total orders O for functions in *Function-list* complying to the partial order specified by these rules. In general, there might be more than one total order.
8. Execute the function-realization rules. For each function Fn , do the following:
 - 8.1. If the rule is in the form like $(! Fn)$, pick out the item from the dictionary according its characteristic specified by the feature-realization rules.
 - 8.2. If the rule is in the form like $(\$ X Fn)$, search the subframe-list, find a subframe whose name is Fn , go to step 3 with this subframe. If there are special rules related to function Fn , in the form like $(\% Fn X)$, carry the rule X along with the subframe.

9. When all functions in *Function-list* are realized, list them out according to each total order specified in *O*.

4.2 Examples

In this section, we use a presentative sentence to illustrate the generation process of the system. Notice that it is not given in the input whether the object introduced should come before or after verb. This decision follows the "From Light to Heavy" principle. The procedure attached to the node measures the weight of the entity being presented and find that it has a relative clause as modifier. So the procedure choose the *pre-after* feature.

Input :

```
(sentence (s-sentence)
  (clause (mood independent indicative transitivity
    transitive passive single-obj explicit productive)
    (agent (np head-noun noun noun-mod assp-phr (hn bookstore))
      (assop (ap) (na (np head-noun noun (hn proper))))))
    (pred (vp (verb publish)))
    (patient (np head-noun noun noun-mod
      class-phr (hn book))
      (classp (cp demonstrative (demo this)(class ben))))))
```

Generating process :

```
R-rules -- feature-Realization rules
B-rules -- function-Building rules
F-rules -- Function-realization rules
```

```
frame : sentence
level : sentence
R-rules : (+ sentence)
B-rules :
F-rules : ($ clause sentence)
result : ( sentence )
```

Figure 5 A example

frame : clause
level : clause
R-rules : (+ *pred*) (+ *subj*) (+ *patient*) (+ *prep*) (/ *prep shih*) (+ *der*)
 (/ *der der*) (+ *agent*) (= *subj patient*) (+ *topic*) (= *topic subj*)
B-rules : (> *subj prep*) (> *prep agent*) (> *agent pred*) (< < *der*) (> *subj pred*)
F-rules : (\$ *vp pred*) (\$ *np patient*) (! *prep*) (! *der*) (\$ *np agent*)
result : (*patient prep agent pred der*)

frame : patient
level : np
R-rules : (+ *hn*) (/ *hn noun*) (/ *hn book*) (+ *classp*)
B-rules : (< < *hn*) (> *classp hn*)
F-rules : (! *hn*) (\$ *cp classp*)
result : (*classp hn*)

frame : classp
level : cp
R-rules : (+ *class*) (+ *demo*)
B-rules : (> *demo class*)
F-rules : (! *class*) (! *demo*)
result : (*demo class*)

frame : pred
level : vp
R-rules : (+ *verb*) (/ *verb publish*)
B-rules :
F-rules : (! *verb*)
result : (*verb*)

frame : agent
level : np
R-rules : (+ *hn*) (/ *hn bookstore*) (+ *assop*)
B-rules : (< < *hn*) (> *assop hn*)
F-rules : (! *hn*) (\$ *ap assop*)
result : (*assop hn*)

Figure 5 A example (continued)

frame : assop
 level : ap
 R-rules : (+ na)
 B-rules :
 F-rules : (\$ np na)
 result : (na)

frame : na
 level : np
 R-rules : (+ hn) (/ hn proper)
 B-rules : (< < hn)
 F-rules : (! hn)
 result : (hn)

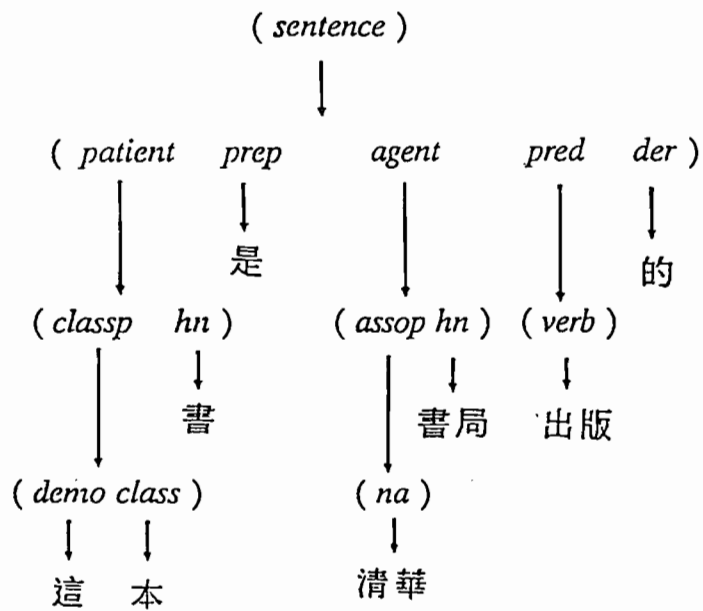


Figure 5 A example (continued)

4.3 Sentences Generated by System

Following are some sentences actually generated by the system.

- 1 他唸書很專心
- 2 上課以前,我先喝一杯茶
- 3 他去香港為的是學廣東話
- 4 因為天黑了,所以我不去
- 5 抽屜裡有三本書
- 6 桌子上放了很多鉛筆
- 7 到了一批貨
- 8 小偷偷了他的錢包
- 9 他的錢包被小偷偷了
- 10 他被小偷偷了錢包
- 11 他的錢包被偷了
- 12 他被偷了
- 13 清華書局出版了這本書
- 14 這本書是清華書局出版的
- 15 這本書出版了
- 16 張三誇獎李四
- 17 李四受張三誇獎
- 18 李四很讓張三誇獎
- 19 那件事影響了他的決定
- 20 他的決定受了那件事影響
- 21 他的決定受了很大的影響
- 22 我送他一本書
- 23 我送一本書給他
- 24 我送他一本研究漢語語法的書
- 25 桌子上有一本書
- 26 有一本書在桌子上
- 27 在桌子上有一本研究漢語語法的書
- 28 房間裡面躺著一隻狗
- 29 有一隻狗在房間裡面躺著
- 30 你來決定這件事
- 31 這件事由你決定

5. Conclusions

In this paper, we have designed and implemented a generator for Chinese sentences. The generator uses the systemic grammar as the explicit representation of the syntax of Chinese sentences. We have also augmented the generative mechanism of systemic grammar with procedural attachment.

The grammar that we have written covers many interesting grammatical phenomena in Chinese sentences. We feel that systemic grammar provides a natural and concise notation for dealing with these phenomena, and can be turned into a generative process easily.

The procedural attachment can be used to facilitate flexible interaction between the sentence generator and other phases of a text generator. One can use attached procedures in the sentence generator to account for uncertainty in the availability of a certain feature. So that other phases of the text generator may have the flexibility of whether to provide this feature or not.

This sentence generator is the first program that generates Chinese sentences using an explicit grammatical formalism. We hope that our generator could be integrated into other systems that produce natural language output in Chinese. We believe the quality of output could be improved using a separate sentence grammar. Besides, our generator could be used as a tool to study many unexplored area in Chinese grammar and the relationship between modules of NLP systems.

6. Future work

1. Extending The Scope of The Grammar

As shown in Section 3, the grammar used in our system does not have a very large scope. We feel that the inclusions of *question*, *comparison*, and *negation* are most urgent.

Besides, some existing parts should also be extended, such as multiple adjectives in noun phrases and the arrangement of various components in verb phrases.

2 Interaction Between Syntax And Morphology

The current grammar of our system concentrated on the syntactic structure of Chinese sentences. Actually, many interesting phenomena in Chinese have something to do with the morphological structure of words. For example, the reduplication of volitional verbs is to signal that the actor is doing something 'a little bit' and the reduplication of adjective make the original meaning of the adjective more vivid. The structure of *verb-object* compound is also a case that we have not dealt with.

These morphological phenomena and their interaction with the syntax must be dealt with in order to enlarge the scope of the grammar. However, it is still not clear how this can be done in systemic grammar.

3 Intonation System

In Chinese, some words within a sentence have very little semantic meaning, but without them, the whole sentence *sounds* odd. For example, the two sentences listed below have the same meaning, but second sentence is *sounds* odd for most people and is seldom used.

李四很讓張三誇獎
李四讓張三誇獎

4. Unification-based sentence generation

There is an alternative to the method we have adopted for the control mechanism. Mellish considered structure-preserving mappings from the description spaces defined by a system network to a Generalized Atomic Formulate (GAF) lattice [Mellish 1988]. The relationship between connected nodes in system network can be viewed as "subsumption." Mellish proposed that logical terms be used to encode the relationship. In the GAF lattice, the greatest lower bound operation is unification, so if the mappings succeed, we can use this operation to make a conjunction, to test the subsumption, and to detect the incompatibility between the features. Unification is a primitive operation in most logic programming systems and is also the basis of many grammatical formalisms. It is therefore a relative well understood operation and can be efficiently implemented.

5. Integration of Sentence Generation To A Complete Text Generation System

References

[Danlos 1984]

L. Danlos, "Conceptual and Linguistic Decision in Generation", Proceedings of the 21st Annual Meeting of the ACL, (COLING 84), pp. 501-504, 1984.

[Derr-McKeown 1984]

M.A. Derr and K.R. McKeown, "Using Focus to Generate Complex and Simple Sentences", Proceedings of the 21st Annual Meeting of the ACL, (COLING 84), pp. 319-326, 1984.

[Goldman 1975]

N.M. Goldman, "Sentence Paraphrasing from a Conceptual Base", CACM 18, pp. 96-106, 1975.

[Grishman 1979]

R. Grishman, "Response Generation in Question-Answering Systems", Proceedings of the 17th Annual Meeting of the ACL, pp. 99-101, 1979.

[Halliday-Hansan 1976]

M.K.A. Halliday and R. Hansan, "Cohesion in English", Longman, London, 1976.

[Kuo 1989]

H.M. Kuo, "A Chinese Sentence Generator Using Systemic Grammar", master thesis, National Tsing Hua University.

[Li-Thomson 1983]

C.N. Li and S.A. Thompson, "The Category 'Auxiliary' in Mandarin", Studies in Chinese Syntax and Semantics, Universe and Scope : Presupposition and Quantification in Chinese, Student book Co., Ltd, 1983.

[Mann 1984]

W.C. Mann, "Discourse Structures for Text Generation", Proceedings of the 21st Annual Meeting of the ACL, (COLING 84), pp. 367-375, 1984.

[Mann 1982]

W.C. Mann, "Applied Computational Linguistics in Perspective: Proceedings of the workshop - Text Generation", AJCL 8, pp. 62-69, 1982.

[McDonald-Pustejovsky 1985]

D.D. McDonald and J.D. Pustejovsky, "A Computational Theory of Prose Style for Natural Language Generation", Proceedings of the 2nd Conference of the European Chapter of the ACL, pp. 187-193, 1985.

[McKeown 1985]

K.R. McKeown "Discourse Strategies for Generating Natural-Language Text", Artificial Intelligence 27, pp. 1-41, 1985.

[Shapiro 1979]

S.C. Shapiro, "Generalized Augmented Transition Network Grammars for Generation from Semantic Networks", Proceedings of the 17th Annual Meeting of the ACL, pp. 25-30, 1979.

[Tang 1985]

T.C. Tang, "Studies on Chinese Morphology and Syntax", Student Book Co., Ltd. 1985.

[Tang 1977]

T.C. Tang, "Studies in Transformational Grammar of Chinese: Volume I: Movement Transformation", Student Book Company, Taipei, 1977.

[Vaughan-McDonald 1986]

M. Vaughan and D.D. McDonald, "A Model of Revision in Natural Language Generation", Proceedings of the 24th Annual Meeting of the ACL, pp. 90-96, 1986.

[Wingrad 1983]

T. Winograd, "Language as Cognitive Process Volume 1: Syntax", Addison-Wesley Publishing Company, 1983

Computer Interpretation of Chinese Declarative Sentences

Based on Situation Semantics

Chun-Hsiao Lee and Hsi-Jian Lee

Department of Computer Science and Information Engineering

National Chiao Tung University, Hsinchu, Taiwan 30050

ABSTRACT

In this paper, we present a method for interpreting Chinese declarative sentences by Head-driven Phrase Structure Grammar (HPSG), which is a unification-based grammatical formalisms with situation semantics as its semantic theory. The primary reasons for using such an approach are that HPSG performs syntactic and semantic analysis in an integrated way and that situation semantics provides a realistic and sound theoretic foundation. There are two kinds of feature structures used in the semantic representations of words, phrases and sentences. The first type of feature structures is the basic type which consists of quantifier, indexed-object, circumstance, and description types. They are used to represent the meanings of lexical signs and unquantified phrasal signs. The second type of feature structure is the complex type, which is are composed of quantified-object types and quantified-circumstance types. They are applied to represent quantified phrasal signs. The process of semantic interpretation is carried out by combining the semantic representations of heads and complements/adjuncts according to their types and then generating a new semantic representation for the larger phrase. A practical system is designed with a set of examples.

1. INTRODUCTION

There are various aspects of natural language processing: syntactic processing, semantic interpretation, discourse interpretation, language generation, knowledge representation, etc. [Allen, 1987]. This paper is mainly concerned with semantic interpretation, which is used to obtain the meaning of a sentence. The primary motivations for semantic interpretation in natural language processing systems are (a). eliminating semantic anomalies, (b). resolving ambiguities, and (c). drawing inferences.

Semantic interpretation is also needed in machine translation systems which translate sentences from a source language to a target language. The more analysis is done, the less human involvement is needed. Taking advantage of the cooperation between linguistics and machine translation systems [Raskin, 1987], linguistic theories are often applied to the system to put semantic interpretation on a theoretical basis and to produce better quality of translation.

Grammar formalisms are developed by linguists to describe the string set, syntax, and semantics of a language [Shieber, 1986]. Examples include transformational grammar, definite-clause grammar, lexical-functional grammar, generalized phrase structure grammar (GPSG) [Gazdar, et al. 1985], head-driven phrase structure grammar, and so on. We consider below how semantics is dealt with by the three formalisms: modular logic grammar, generalized phrase structure grammar, and head-driven phrase structure grammar.

In Modular Logic Grammar (MLG) [McCord, 1987], Logical Form Language (LFL), some kind of second-order predicate calculus, is used as the semantic representation for natural language sentences. The process of semantic interpretation is performed by first recursively interpreting the components in the daughter list of the input syntactic item, reordering them when needed, and lastly combining them by the use of a set of modification rules to obtain the logical form for the sentence. To resolve scoping problems, reshaping operations are done to achieve the desired logical order.

The semantic theory adopted in GPSG is Montague semantics, which uses a model of the world in which linguistic elements, such as nouns and sentences, are assigned denotations (meanings), such as entities and truth-values [Sells, 1985; Hirst, 1987]. Instead of mapping linguistic expressions directly to denotations in a model, Intensional Logic (IL) is used as an intermediate representation language. A natural language sentence is translated into an expression of intensional logic that will be associated with an interpretation in the model.

Situation semantics, which is applied in HPSG, develop a theory of situations that are considered to be components of reality [Barwise and Perry, 1983]. Real situations consist of four primitives: individuals, relations, properties (relations whose arity is one), and space-time locations. Abstract situations (such as situation types, states of affairs, courses of events, and more abstract objects, event-types), which are built up by these primitives, are used to classify and represent real situations. Situation semantics adopts the relation theory of meaning, which takes linguistic meaning as a relation between the types of situations in which utterances are spoken and the types of situations that are described by those utterances. The described situation is the interpretation of an utterance on a particular In HPSG, feature structures of various types are utilized to describe the semantic contents of lexical signs, which provide the information about the primitives of the described situation. A universal principle called the Semantic Principle, accompanied with the Subcategorization Principle, is followed in combining the semantic contents of the head daughter and of the complement daughters to produce the phrasal sign's semantic content.

There are various problems in semantic interpretation, including lexical ambiguities, scoping ambiguities, referential ambiguities, noun-noun modifications, etc. Many issues are appealing to natural language processing researches. The former two problems can be partially solved by our system. Words that have multiple senses, i.e., lexical ambiguities, are the usual source of sentences' semantic ambiguity [Raskin, 1987]. Some of them may be disambiguated by using syntactic analysis. Some of them may be disambiguated by using case structures and selectional restrictions. Some of them may be disambiguated by

lexical association, i.e., word-word interaction. Scoping problems are often introduced into sentences by quantifiers, negations, adverbs, coordinators, etc.

Semantic interpretation is the process of mapping a natural language sentence (or its well-formed part) to its meaning representation or intermediate representation. A sentence's complete meaning representation contains features about lexical meanings, the entities that are referred to, the relations that are explicitly or implicitly specified in the sentence and their arguments, speaker's intention, etc. Knowledge about the context and the world as well as syntactic and semantic knowledge is needed to determine these features [Grosz et al., 1986]. What we are concerned with in this paper is the intermediate representations of the sentences without considering the context.

Since the primary goal of semantic interpretation is to obtain the intermediate meaning representations of natural language sentences, a variety of meaning representation formalisms were proposed in natural language processing systems in the literature. These formalisms represent the meaning of a sentence, which can be used to generate a corresponding sentence in another language in machine translation systems.

In Wilks' Preference Semantics (PS) system [Wilks, 1986], which translates English texts into French, some semantic items are used to represent text items. Word senses are associated with semantic formulas, which are composed of primitive semantic elements. Templates are constructed from formulas for word senses of a sentence as its meaning representation. Paraphrases and case-extraction/common-sense inferences are used to bind templates together in the semantic block that represents a fragmented text.

ABSITY (A Better Semantic Interpreter Than Yours) [Hirst, 1987] takes input from PARAGRAM parser and generates output in a frame representation language, FRAIL, which is used to retrieve and infer knowledge in a knowledge base. In ABSITY, each syntactic category has a type of FRAIL element, making use of the strong typing feature of Montague semantics. LUNAR [Woods, 1986] uses a meaning representation language MRL, a variant of the first-order predicate calculus, as the semantic representation for the meanings of sentences.

In the traditional approach to natural language processing, semantic interpretation is

performed after syntactic analysis and before pragmatic processing. The separation of syntactic analysis and semantic interpretation makes the natural language processing system more modular. But such an approach may produce a lot of syntactic analysis structures that will be judged to be semantically anomalous, resulting in the inefficiency of the system.

The recent approaches to semantic interpretation [Allen, 1987] tend to integrate syntactic and semantic processing. The semantic grammars approach, as in the SOPHIE system [Tennant, 1981], parses sentences according to the semantic categories rather than the syntactic categories of words and phrases. It is easy and efficient in limited domains, but problems occur in making it more general or transportable. In the interleaved approach, as in the SHRDLU system [Tennant, 1981], the semantic interpreter is called immediately when each major syntactic constituent such as a noun phrase is proposed by the syntactic parser. Many syntactically possible constituents that are semantically anomalous can be eliminated by the semantic interpreter as soon as they are proposed by the syntactic parser.

The rule-by-rule approach, as in the ABSITY system [Hirst, 1987], has a set of semantic rules paired with a set of syntactic rules. Each time some syntactic rule is applied to construct a syntactic structure, the semantic interpretation is performed by using the semantic rule to build a semantic representation. The semantic rule is usually specified as part of the annotation on the syntactic grammar rule. Another approach called the semantically driven approach, as in the PS system [Wilks, 1986], carries out semantic interpretation directly on the input using only minimal local syntactic information. More syntactic information will be needed to help the semantic interpretation of complex sentences.

We intend to design a semantic interpretation system that is a part of the Chinese-to-English machine translation system, CEMAT. We wish to interpret the meanings of Chinese sentences, by using one of the current grammar formalisms based on some semantic theory, eliminate semantically anomalous sentences, provide semantic information for other stages (such as word selection and generation) in the machine translation system to improve the quality of translation, and hopefully deal with part of the semantic issues.

This paper consists of four additional sections. Section 2 describes the semantic representations that we take to express the meanings of sentences (and their constituents). The process of semantic interpretation and the combination operations are discussed in Section 3. Section 4 illustrates the implemented system and some examples of interpretation. Conclusions are drawn in the last section.

0

2. SEMANTIC REPRESENTATION

The grammar formalism we adopt is HPSG, whose semantics is based on situation semantics. The primary reason for using HPSG is that the theory performs syntactic and semantic analysis in an integrated way. HPSG can be regarded as a theory of signs; it directly explains the connection between syntactic and semantic phenomena, e.g., subcategorization and semantic roles in the described relation [Pollard and Sag, 1987]. The adopted semantic theory seems to provide a more realistic and reasonable theoretic foundation than other theories that use formal mathematical models of the world, such as Montague semantics. From the viewpoint of design, HPSG makes use of unification so that it can be efficiently implemented by logic programs such as Prolog programs in computers, and the semantic information such as case relation and selectional restrictions can be utilized to describe relations and their roles in situation semantics.

The semantic interpretation system is supposed to take an input sign from the syntactic chart parser, in which the information from the lexicon (including the semantic information of constituents) and the syntactic information proposed by the parser (such as complements and/or adjuncts of the lexical head in a phrase) is specified. It generates the sign with its semantic information as output, when it is semantically valid. The system now can deal with declarative sentences in Chinese

2.1 Semantic Information about Signs

Signs are (partially) described by feature structures which provide phonological, syntactic, and semantic information. Semantic information specified as values for the *sem* attribute will sketch the described situation by the use of individuals and relations in it. The

sem values consist of two attributes: the *cont* attribute which specifies the contribution of a sign to the described situation, and the *inds* attribute which specifies those restricted variables met so far. The outline of a sign's structure looks like:

```
[phon ...,
  syn  ...,
  sem  [cont ...,
        inds ...].
```

The semantic content of a sign can be described by a feature structure of basic types or complex types. The former includes quantifier, indexed-object, circumstance, or description type. The latter consists of quantified-object or quantified-circumstance type. For the roles in the relations, i.e., the ways that things participate in relations (events), which are specified in the content of lexical signs, we adopt the general semantic roles in relations instead of specific roles for each relation in HPSG. This makes the case relation information to be accessed more easily by other modules of the machine translation system. The number of semantic roles ranges from the order of ten (e.g., thirteen in [Winston, 1984]) to the order of thirty (e.g., thirty-four in [Nagao et al., 1986]). Too few roles can not provide enough information to identify an event uniquely, e.g., the inability to distinguish between the instrument case of the word "以" in the sentence "他以書當枕" and the cause case of "以" in "桂林以山水聞名" [Winston, 1984]. Too many roles may result in the similarities of some cases, e.g., the case "*space-from*" and the case "*time-from*" [Nagao et al., 1986]. Based on such consideration, the following semantic roles are proposed: agent, patient, recipient, benefactive, experiencer, company, comparison, instrument, cause, purpose, result, theme, accordance, trajectory, point, source, goal, duration, advantage, inclusion, exclusion, identity, and proposition.

2.2 Basic Types of Feature Structures

The semantic contents of lexical signs and phrasal signs that are not quantified are represented by feature structures of basic types: *quantifier*, *indexed-object*, *circumstance*, and *description*.

2.2.1 Quantifier Type

The semantic content of numbers such as "一" and "十", demonstratives such as "這" and "那", and classifiers such as "個" and "塊" [Li and Thompson, 1981] is a feature structure of type *quantifier*. It is of the form:

```
sem [cont [qua [ATTRIBUTE VALUE]]]
```

where ATTRIBUTE can be one of {num, det, unit}. For example, the lexical sign for the determiner "這些" is:

```
[phon zhei4_xie1,  
syn [loc [head [maj det,  
type demonstrative],  
lex +]],  
sem [cont [qua [det zhei4_xie1]]],  
trans [these]] .
```

2.2.2 Indexed-Object Type

The use of noun phrases in natural languages depends on the context of utterances in general. They usually contribute restricted variables to the semantic content of sentences containing the phrases. The index attribute, *inds*, has as its value (Y) a feature structure of type *index* containing a variable (X) and the restrictions, *rest*, upon the variable. The agreement information includes person and domain hierarchy, *d_hier*; see Section 4. Each indexed-object is assigned an implicit relation name according to the syntactic type of the sign. For example, the common noun involved in sortal properties with the instance role, as in "花" is represented as:

```
[phon hua1,  
syn [loc [head [maj n,  
type individual,
```

```

                                adjuncts    [Poss; Det; Classifier]],
                                subcat [],
                                lex    +]],
sem  [cont  [ind  Y],
      inds  Y:[var X:[per    3rd,
                    d_hier  plant],
      rest [reln  hua1,
            inst  X]]].

```

2.2.3 Circumstance Type

Circumstances are used by HPSG to describe partially possible ways the world might be. They correspond to states of affairs in situation semantics. For verbs and adjectives, feature structures of type *circumstance* are taken as their semantic contents to describe circumstances:

```

sem  [cont  [reln    [E:RELNAME],
              ROLE    V(AGR),
              ...     ...,
              location L],
      inds  [var  E:[PRO],
            rest  [],
            [var  L,
            rest  []] .

```

where AGR specifies the agreement that must be satisfied by the filler of the role in the relation, and PRO indicates the property associated with the sign in the property hierarchy (see Section 4). The variable E, similar to the event variable [McCord, 1987], is used to represent the event (or state) denoted by the relation. The variable L, functioning like the indexed variable used in LFL [McCord, 1987], is utilized to express space-time locations in situation semantics. The following lexical signs illustrate the verb "打" :

```
[phon da3,
```

```

syn  [loc  [head  [maj      v,
                  asp      dur; per; exp,
                  type     vtc,
                  crs      +,
                  adjuncts [adv(manner; y-n; frequency)]]],
      subcat [X2:NP2(acc), X1:NP1(nom)],
      lex    +]],
sem  [cont  [reln      [E:da3],
              agent    X1(d_hier  human),
              patient   X2(d_hier  human),
              location  L],
      inds  [var  E:[prop  action],
              rest  []],
            [var  L,
              rest  []]].

```

2.3.4 Description Type

Another type of feature structures, which is usually associated with adverbs and prepositions, is introduced to describe the event (state) or space-time location that is associated with a certain circumstance. Feature structures of this type, i.e., of *description* type, have the form:

```

sem  [cont  [reln  RELNAME,
              ROLE V(AGR),
              ...   ...,
              DESC X(PRO)],
      inds  []].

```

where DESC can be the *event* attribute if it describes some event (state), or the *located* attribute if it describes some space-time location; and PRO will be the agreement requirement of the property of the described event (state) when DESC is *event*. The

following examples show the lexical sign for "很":

```
[phon hen3,
syn  [loc  [head [maj      adv,
          adjuncts  []],
      subcat [],
      lex  +]],
sem  [cont [reln hen3,
          event X(prop      stative)],
inds  []].
```

2.3. Complex Types of Feature Structures

Quantified noun phrases such as "三朵花" and the larger phrase containing them such as "買三朵花" introduce the problems of quantification and scoping. To take them into consideration, feature structures of complex types are used to represent the semantic contents of quantified phrasal signs. They are constructed from the feature structures of basic types and divided into *quantified-object* type and *quantified-circumstance* type.

2.3.1 Quantified-Object Type

Feature structures of type *quantified-object* are formed by combining feature structures of type *quantifier*, corresponding to classifier/measure phrases [Li and Thompson, 1981], and the ones of type *indexed-object*, corresponding to nouns. For example, the semantic content of noun phrase "兩個人" will be:

```
sem  [cont [qua  [num liang2,
                unit ge5]],
      [ind  Y],
inds  Y:[var X:[per      3rd,
                d_hier  human],
      rest [reln  ren2,
```

inst X]]] .

2.3.2 Quantified-Circumstance Type

A feature structure of the *quantified-circumstance* type consists of two attributes: the *quant* attribute, whose value is a feature structure of type *quantified-object*, and the *scope* attribute, whose value is a feature structure of type *circumstance* or *quantified-circumstance*. The verb phrase "兩個人逃走", for instance, has the following semantic content:

```
sem  [cont  [quant [qua  [num  liang2,
                        unit  ge5]],
        [ind  Y]],
      [scope [reln      [E:tao2_zou3],
                  agent  X:[per      3rd,
                              d_hier  human],
                  location L]],
      inds [var  E:[prop  moving],
            rest [],
            [var  L,
            rest [],
            Y:[var X:[per  3rd,
                      d_hier human],
            rest [reln  ren2,
                  inst  X]]] .
```

3. INTERPRETATION SCHEME

Our system will construct the semantic representation of a sign according to its syntactic information such as its complements and/or adjuncts and its semantic information such as the types of semantic representations of its constituents. Additional relevant information such as agreement features will be unified.

Combination operations combine the semantic representations of the constituents (such as the head, complements, and adjuncts) in some systematic ways to produce the semantic representation of the whole sign. The result of combination is reflected by the values in the *cont* attribute and the *inds* attribute.

3.1 Combining Heads with Complements

In general, the lexical heads except nouns of phrases, such as verbs, adjectives, and prepositions, characterize the described situation with relations that take place in it. The complements of these heads, such as noun phrases, verb phrases, and prepositional phrases, will provide information about the fillers of the roles in the relations described by lexical heads. According to the types of feature structures in the semantic contents of the head and its complement, the following steps of interpretation are taken:

- *Combining the circumstance or description type with the indexed-object type:*

When the semantic content of the head is a feature structure of type *circumstance* (e.g., for verbs) or *description* (e.g., for prepositions) and that of the head's complement is of type *indexed-object* (e.g., for noun phrases), the restricted variable in the complement's content is unified (including agreement information) with the corresponding role in the relation specified by the head. The *inds* values of the head and the complement are collected together. For example, the verb phrase "買花" has the following semantic content:

```

sem  [cont  [reln      [E:mai3],
           agent     X1(d_hier  human),
           patient   X:[per     3rd,
                    d_hier   plant],
           location  L],
inds [var  E:[prop   dative], rest  []],
     [var  L, rest   []],
     Y:[var X:[per   3rd,
              d_hier plant],

```

```
rest [reln hua1,
      inst X]]] .
```

- *Combining the circumstance type with the description type:*

In this case, the head with the content of type *circumstance*, e.g., a verb, subcategorizes for a complement whose content is of type *description* (e.g., a prepositional phrase), and whose content has combined with its constituent's content (e.g., the noun phrase in the prepositional phrase). The operation described in the above paragraph is also applicable to deal with this case. The following examples show the semantic content of the verb phrase "把花給":

```
sem  [cont [reln      [E:gei3],
                  agent X1(d_hier  human),
                  recipient X2(d_hier animate),
                  patient X:[per    3rd,
                             d_hier  plant],
                  location L],
      inds [var  E:[prop  dative],
            rest [],
            [var  L, rest []],
            Y:[var X:[per  3rd,
                       d_hier  plant],
              rest [reln hua1,
                    inst X]]] .
```

- *Combining the circumstance type with the quantified-object type:*

When the head's content is of type *circumstance* (or *quantified-circumstance*) and the associated complement's content is of type *quantified-object*, e.g., a quantified noun phrase, a feature structure of type *quantified-circumstance* is built from them as the

semantic representation of the whole phrase. The corresponding variables are unified. The *inds* values are also collected. For instance, from the verb "喜歡" and the noun phrase "一個人", the semantic content of the verb phrase "喜歡一個人" can be constructed as:

```

sem  [cont  [quant [qua  [num  yi1,
                        unit  ge5]],
        [ind  Y]],
      [scope [reln      [E:xi3_huan1],
        experiencer X1(d_hier  human),
        patient     X:[per     3rd,
                        d_hier  human],
        location    L],
inds  [var  E:[prop  mood],
      rest  []],
      [var  L, rest[]],
      Y:[var X:[per  3rd,
                d_hier  human],
        rest [reln  ren2,
              inst  X]]] .

```

- *Combining the circumstance or description type with the circumstance type:*

When we want to combine a head having a content of type *circumstance* or *description* with a complement having a content of type *circumstance*, we fill the role in the relation described by the head with the semantic content of the complement and collect indices. For example, the verb "打算" is combined with the complement "逃走" resulting in the semantic content of "打算逃走":

```

sem  [cont  [reln      [E1:da3_suan4],
        agent        X1 (d_hier  human),

```

```

proposition [reln      [E2:tao2_zou3],
             agent     V1(d_hier  animal),
             location  L2:[prop  moving],
             location  L1],
inds [var  E1:[prop  feeling], rest  []],
     [var  L1,   rest  []],
     [var  E2:[prop  moving],   rest  []],
     [var  L2,   rest  []] .

```

3.2 Combining Heads with Adjuncts

Different actions of interpretation are taken to deal with the heads and their adjuncts in Chinese in which the adjuncts of nouns may be adjectives, classifier/measure phrases, associative phrases, and relative clauses [Li and Thompson, 1981]; and verbs' adjuncts can be prepositional phrases, adverb phrases or verb phrases.

- *Combining the quantifier type with the quantifier type:*

When the semantic contents of the head and its adjunct are both feature structures of type *quantifier* (e.g., in a classifier/measure phrase), we just take the set union of the *qua* values as the new semantic content. For example, the content of "三塊" is produced as:

```

sem  [cont [qua [num san1,
               unit kuai4]]] .

```

- *Combining the indexed-object type with the quantifier type:*

In this case, we form a feature structure of type *quantified-object* by joining the head's content which is of type *index-object* with the adjunct's content which is of type *quantifier*. The content of "三塊蛋糕" is represented as follows:

```

sem  [cont [qua [num san1,

```

```

        unit kuai4]],
[ind Y],
inds Y:[var X:[per 3rd,
            d_hier food],
rest [reln dan4_gao1,
inst X]]] .

```

• *Combining the indexed-object type with the indexed-object type:*

The associative phrase introduces an adjunct having the content of type *indexed-object* to a head with the content of the same type. A relation with the name *associate* is created to relate the two restricted variables which appear in the contents. This relation will be added to the restrictions upon the variable specified in the head's content. At last indices-collecting is performed. For instance, from the contents of "我" and "蛋糕", we have the content of "我的蛋糕":

```

sem [cont [ind Y1],
inds Y1:[var X1:[per 3rd,
                d_hier food],
rest [reln dan4_gao1,
inst X1],
[reln associate,
associated X1:[per 3rd,
                d_hier food],
associative X2:[per 1st,
                d_hier human]]],
Y2:[var X2:[per 1st,
            d_hier human],
rest [reln referring,

```

```

referred    X2,
referent    speaker]]].

```

- *Combining the indexed-object type with the circumstance type or combining the circumstance type with the description type:*

This is the case where the head such as a noun is to be combined with an adjunct such as an adjective or a relative clause; or the case where the head like a verb is to be combined with an adjunct like an adverb or a prepositional phrase. During the process of interpretation, the restricted variable specified in the head's content or the previous restriction upon the variable in the content of type *circumstance* is unified with the corresponding role in the relation specified in the adjunct's content, as well as agreement information. Then the relation is asserted as a new restriction upon the variable. The *inds* values are collected. For example, when the predicative adjective "漂亮" is combined with the adjunct "很", the predicative adjective phrase "很漂亮" has the content:

```

sem  [cont  [reln      [E:piao4_liang4],
              patient  X1(d_hier  concrete),
              location  L],
      inds  [var  E:[prop  stative],
            rest  [reln  hen3,
                  event E:[prop  stative]]],
            [var  L,
            rest  []]] .

```

- *Combining the circumstance type with the circumstance type:*

When the semantic contents of the head and the adjunct are both of type *circumstance*, we just add the relation specified in the adjunct's content to the restrictions upon the variable specified in the head's content, and collect indices together. This kind of combination is used to handle serial verb constructions in Chinese [Li and Thompson, 1981]. For

instance, the content of "買票進去" (taking "進去" as the head) is:

```

sem  [cont  [reln      [E2:jin4_qu4],
                agent   V1(d_hier  animal),
                location  L2],
inds  [var  E2:[prop   moving],
        rest [reln      [E1:mai3],
                agent   X1(d_hier  human),
                patient  X:[per     3rd,
                               d_hier  amusement],
                location  L1]],
        [var  L2,  rest  []],
        [var  E1:[prop  dative],  rest  []],
        [var  L1,  rest  []],
        Y:[var  X:[per     3rd,
                    d_hier amusement],
            rest [reln  piao4, inst  X]]] .

```

3.3 Interpretation Process

The whole process of semantic interpretation is that given a syntactically analyzed sign in which the head and its associated complements and adjuncts have been specified, the head's content is first successively combined with each complement from the more oblique complement to the less oblique one, and then successively combined with every adjunct. In each time the combination operations are taken according to the principles given in the previous section, and the results of interpretation are passed to the next combination stage.

For example, the interpretation process of the sentence "李四常上台北" will begin with the main verb "上" and takes nouns "台北" and "李四" as complements, the adverb "常" as an adjunct. After combining "上" with "台北", "李四" and "常", the sentence "李四常上台北" has the following forms:

```

sem  [cont  [reln      [E:shang4],
              agent   X1:[per   3rd,
                        d_hier  human],
              goal    X2:[per   3rd,
                        d_hier  space],
              location L],
inds [var  E:[prop   active],
      rest [reln  chang2,
            event E:[prop   active]]],
      [var  L,
      rest []],
      Y2:[var   X2:[per   3rd,
                  d_hier  space],
          rest  [reln    naming,
                named   X2,
                name    tai2_bei3]],
      Y1:[var   X1:[per   3rd,
                  d_hier  human],
          rest  [reln    naming,
                named   X1,
                name    li3_si4]]] .

```

4. IMPLEMENTATION AND DISCUSSION

The semantic interpretation system is implemented on the Quintus Prolog system under VMS that is installed on a VAX 780 computer. Some data structures are defined for representations, and interpretation rules are written as Prolog programs. Examples will be given to show the results of semantic interpretation.

The type hierarchy expresses knowledge about the structure of the things that it

describes. Knowledge of this kind is useful for describing the relationships between different things, e.g., what kind of objects that can play a certain role in some relation, or what kind of relations that a particular relation can describe (modify) it.

Two type hierarchies are utilized by the semantic interpretation system. The domain hierarchy is used to classify the objects, to which nouns correspond. Nouns may represent objects that are of type CONCRETE, including subtypes HUMAN, PLANT, NATURE, etc., or of type ABSTRACT, which are divided into TIME, SPACE, and CONCEPT types. Verbs are classified by another type hierarchy, the property hierarchy, according to the relations are described by them. They may describe states (i.e., of type STATIVE) or describe events (i.e., of type ACTIVE).

The information about type hierarchies is included in the agreement information of restricted variables which are associated with nouns and verbs in their semantic contents. Variables are unified with other restricted variables or feature structures only when their corresponding agreement information can be unified together. After successful unification, they all have the same values (variables or feature structures) with the same agreement information; otherwise, the unification fails on incompatible values for some features.

For simplification and succinctness, only those parts that are related with semantic interpretation are specified in a sign during implementation. A feature structure is represented by a list with the feature name as the first element and the feature value as the second one. The variable and its agreement information are also put in a list. For example, the sign for "李四" is represented by the following list:

```
[li3_si4,
 [sem, [[cont, [ind, Y],
 [inds, [Y, [var, [X, [[per, 3], [d_hier, human]]],
 [rest, [[reln, naming],
 [named, X],
 [name, li3_si4]]]]]]]]].
```

The semantic interpretation system inputs a list representing the sign that is supposed to be provided by the parser, and outputs a list that represents the input sign including its semantic representation. If the sign is semantically ill-formed, the system rejects it and informs the parser.

According to the types of feature structures appearing in semantic contents, various interpretation procedures are fired to combine the semantic information about the heads with that about the complements/adjuncts so as to build new semantic representations. The sentences listed below can be interpreted by our system currently.

- 1.李四常上台北。
- 2.李四把筆給我。
- 3.李四給我一支筆。
- 4.他去理髮。
- 5.學校來了兩個人。
- 6.他很漂亮。
- 7.李四替媽媽買紅花。
- 8.他一定不來。
- 9.他不一定來。
- 10.他是個有錢人。
- 11.喜歡你的是我。
- 12.門打開了。
- 13.門外有一座高山°
- 14.開發中國家的人民的生活水準普遍不高。
- 15.我昨天買票進去看電影。
- 16.他來的時候,我已經做完功課了。

As for the scoping problem, the portion of a sentence that follows some element such as an adverb is in the scope of that element [Li and Thompson, 1981]. Thus the sentence "他一定不來" is interpreted as:

sem [cont [reln [E:lai2],

```

agent      X:[per      3rd,
             d_hier    human],
locaiton   L],
inds [var  E:[prop    active],
      rest [reln      yi2_ding4,
            event     [reln  bu4,
                       event E:[prop    active]]:[prop active]]],
[var  L,
rest  []],
Y:[var X:[per      3rd,
        d_hier    human],
   rest [reln      referring,
        referred  X,
        referent  spoken]]] ,

```

where the adverb "一定" has "不" in its scope, while the sentence "他不一定來" has the semantic representation:

```

sem  [cont [reln      [E:lai2],
              agent   X:[per      3rd,
                        d_hier    human],
              locaiton L],
inds [var  E:[prop    active],
      rest [reln      bu4,
            event     [reln  yi2_ding4,
                       event E:[prop    active]]:[prop active]]],
[var  L,   rest  []],
Y:[var X:[per      3rd,
        d_hier    human],
   rest [reln      referring,

```

referred X,
referent spoken]]],

where the adverb "不" includes "一定" in its scope.

5. CONCLUSIONS

In the work that we developed, Chinese sentences are interpreted as feature structures by the use of various interpretation rules. These feature structures sketch the situations, which are described by the sentences, in terms of relations, space-time locations, and individuals that appear in the described situations. The interpretation rules, following HPSG, are devoted to combining the semantic information about heads with the one about complements/adjuncts. The implemented system can eliminate semantically anomalous sentences by means of unification on agreement information, partially interpret the sentences and handle some semantic issues.

Our semantic interpretation system allows partial semantic analysis of sentences. Since the system is compositional, the meaning of the whole is systematically and incrementally constructed from the meanings of the parts. When a partial syntactic analysis of a sentence (e.g., a verb phrase) is obtained, we can form the semantic representation of that part if such a representation is semantically valid. For example, from the verb phrase "買花" in the sentence "李四買花", we know the information about the filler (i.e., "花") of the patient role in the relation "買". The determination of such partial semantic analysis does not have to be postponed until the whole syntactic analysis of the sentence is completed.

Some lexical ambiguities are resolved by having a distinct sign for each word sense of the ambiguous word. For example, the "call" sense of the word "叫", as in the sentence "我叫他", is assigned to the lexical sign that subcategorizes for a noun phrase filling the patient role in the relation "*call_jiao4*". And the "cause" sense of "叫", as in the sentence "這件事情叫我很難過", appears in the sign which needs a clause as the complement to fill the proposition role in the relation "*cause_jiao4*". According to the complements to be combined, the correct word sense is selected.

This paper has only proposed a preliminary application of HPSG, which is based on

situation semantics, to the semantic interpretation of Chinese declarative sentences. Further researches would be concerned with anaphoric reference involving the discourse context, with other syntactic constructions involving other parts of speech, with inference involving world knowledge, and so on.

REFERENCES

- Allen, J. *Natural Language Understanding*. Menlo Park, California: Benjamin/Cummings, 1987.
- Barwise, J. and J. Perry. *Situations and Attitudes*. Cambridge, MA: MIT Press, 1983.
- Gazdar, G., E. Klein, G.K. Pullum and I.A. Sag. *Generalized Phrase Structure Grammar*. Oxford: Basil Blackwell, 1985.
- Hirst, G. *Semantic Interpretation and the Resolution of Ambiguity*. Cambridge: Cambridge U. Press, 1987.
- Li, C.N. and S.A. Thompson. *Mandarin Chinese: A Functional Reference Grammar*. Berkeley and L.A.: University of California Press, 1981.
- McCord, M. "Natural language processing in Prolog," in A. Walker (ed.). *Knowledge Systems and Prolog*. Reading, MA: Addison-Wesley, 1987.
- Nagao, M., J. Tsujii and J. Nakamura. "Machine translation from Japanese into English," *Proc. IEEE*, vol 74, 993-1012, 1986.
- Pollard, C. and I.A. Sag. *Information-based Syntax and Semantics: Volume I, Fundamentals*. Stanford: Center for the Study of Language and Information, 1987.
- Raskin, V. "Linguistics and natural language processing," in S. Nirenburg (ed.). *Machine Translation: Theoretical and Methodological issues*. Cambridge: Cambridge U. Press, 1987.
- Sells, P. *Lectures on Contemporary Syntactic Theories*. Stanford: Center for the Study of Language and Information, 1985.
- Shieber, S.M. *Introduction to Unification-based Approaches to Grammar*. Stanford: Center for the Study of Language and Information, 1986.
- Tennant, H. *Natural Language Processing*. New York: Petrocelli Books, 1981.
- Wilks, Y. "An intelligent analyzer and understander of English," in B.J. Grosz, K.S. Jones and B.L. Webber. *Readings in Natural Language Processing*. Palo Alto, CA: Morgan Kaufmann, 1986.
- Winston, P.H. *Artificial Intelligence*. Reading, MA: Addison-Wesley, 1984.
- Woods, W.A. "Semantics and quantification in natural language question answering," in B.J. Grosz, K.S. Jones and B.L. Webber. *Readings in Natural Language Processing*. Palo Alto, CA: Morgan Kaufmann, 1986.

THE VERB-COMPLEMENT (V-R) COMPOUNDS IN MANDARIN CHINESE

Fu-Wen Lin

Computing Center, Academia Sinica

V-R compounding is a rich source of new verbs in Mandarin Chinese. It presents a puzzle all along in the interpretation and construction of subcategorization frames for lacking of a rule-governed process to deal with how the function of a V-R compound is related to the functions of its constituents. This work aims at investigating the restrictions on stem collocation and the construction of subcategorization frames of V-R compounds in terms of lexicalized semantic and grammatical information of the verbs which are juxtaposed to form the compounds.

1. INTRODUCTION

Mandarin Chinese employs compounding as a major device to augment its lexicon. There is a group of compounds often referred to as the VERB-COMPLEMENT (V-R) compounds which are structurally $[V_1-V_2]_V$ [1] in general. They are the focus of this work.

1.1. THE AIM OF THIS WORK

Based on past works, it seems to be very difficult to establish a rule-governed interpretation process of the V-R compounds, because the subcategorization frames of the V-R compounds are not formed by simply concatenating the frames of the V and the R. Nor do the meanings of the V-Rs seem to be straightforwardly compositional. Thus, though this compounding type is very productive and the bulk of the lexical items formed by this word-formation are transparent, they always present a puzzle in the interpretation and construction of subcategorization frames. This paper investigates whether the restrictions on stem collocation and the construction of subcategorization frames of V-R compounds can be predicted from

lexicalized semantic and grammatical information of the verbs which are juxtaposed to form the compounds.

1.2. AN OVERVIEW: DESCRIPTIVE ACCOUNTS

About the V-R compounds, there have already been a lot of works devoted to explore their syntactic and semantic functions. The following is a summary of some earlier works on which this work bases.

Consulting Chao(1968:435-480), we classify the Rs into five types according to their functions:

Types	Functions of the complement and examples
(1) Resultative	Describing the state of the subject or object after the completion of the action described by the V member; such as <div style="margin-left: 40px;"> <p>ji -<u>tsen</u> le dijian, "擊沉了敵艦" attack sink ASP enemy warship 'have sunk the enemy warship'</p> <p>chr-<u>guang</u> le fan, "吃光了飯" eat exhaust ASP rice 'have eaten all the rice'</p> <p>chr-<u>bau</u> le fan; "吃飽了飯" eat full ASP rice 'have had enough food'</p> </div>
(2) Phase	Expressing phase (aspect) of action of the V member; such as <div style="margin-left: 40px;"> <p>tsai -<u>jau</u>, "猜著" guess hit-the-mark 'have guessed just right'</p> <p>peng-<u>dau</u>, "碰到" meet reach 'have met'</p> </div>

yu -jian, "遇見"
meet see
'have met'

tzuo-wan, "做完"
do finish
'have finished doing'

chr-guo, "吃過"
eat pass
'have finished dinner'

- (3) Intensifying Intensifying the state described by the V member; such as

mei -ji le, "美極了"
beautiful extremely CRS
'extremely beautiful'

huai-tou le. "壞透了"
bad thoroughly CRS
'thoroughly bad'

re -sz le; "熱死了"
hot die CRS
'be hot to death'

- (4) Potential [2] There are three subtypes of Rs whose central meanings are to express potentiality:

- (a) Dummy potential complements; such as

tzuo(-bu)-liau, "做不了"
do not finish
'not be able to do'

tzuo-(bu)-lai; "做不來"
do not come
'not be able to make'

- (b) Minimal potential complement; such as

ch(-bu)-de; "吃不得"
eat not obtain
'inedible'

- (c) Lexical potential complements; a limited number of potential complements compounds occur either mostly or exclusively in potential form with idiomatic meanings; such as

lai -de (/ -bu)-ji, "來得(/不)及"
come obtain not reach
'can(not) come so as to reach'

*lai -ji; "來及"
come reach

(5) Directional There are four subtypes:

- (a) Indicating motion toward or away from the speaker ---- lai "來", chiu "去"; such as

sung-chiu, "送去"
send go
'send away'

- (b) Referring to the variety of path to which the theme moves; there are nine verbs involved [3]:

shang "上", 'ascend, -up'	shia "下", 'descend, -down'
jin "進", 'enter, -in'	chu "出", 'exit, -out'
chi "起", 'rise, -up'	huei "回", 'return, -back'
guo "過", 'pass, -over'	kai "開", 'open, -away, apart'
lung "攏"; 'gather, -together'	

such as in

tzou-kai; "走開"
walk open
'walk away'

- (c) Double complements which are formed with a type (b) followed by a type (a) complement, such as

reng -guo -chui; "扔過去"
throw pass go
'threw over there'

- (d) Verbs of motion which can not form double complements with type (a), such as

peng -dau le, "碰倒了"
collide fall ASP
'have knocked down'

the others like fan 'turn over' "翻", san 'scatter' "散", etc.

Examining the above classification, we observe that members of the lexical classes except type (1) are quite limited and their predicative functions are fixed. The Rs belonging to the types (2), (3) and (4) all predicate the situation of the event described by the Vs; and those belonging to the type (5) regularly describe, literally or metaphorically, the state of the object, if V is transitive, or the subject of V, if V is intransitive. As for the type (1), there does not seem to be a clear-cut overall generalization; such as

(1) ta he -tzuei le jiu, "他喝醉了酒"
he drink drunk ASP wine
'He is drunk'

(2) ta guan-tzuei le liz; "他灌醉了李四"
he pour drunk ASP LIZ
'He got LIZ drunk'

In (1), TZUEI 'drunk' "醉" predicates the subject TA 'he' "他", but in (2), it predicates the object LIZ 'LIZ' "李四", though

the Vs are transitive in both cases.

Mainland Chinese linguists have investigated V-R compounds in terms of the predicative functions of the R members in a sentence and their possible alternative constructions. They probed into these linguistic phenomena by means of the transitivity of the concatenated members and their corresponding compounds. The compendium of their classification is illustrated in the following tabular form [4]:

Predicative types	Transitivity			Examples
	V-R	V	R	
0 P	t	t	i	wusung <u>da</u> - <u>sz</u> le lauhu Wusung hit die ASP tiger 'Wusung have killed the tiger.' "武松打死了老虎" ta <u>chang-huai</u> sangtz le he sing bad throat ASP 'His throat got hurt for singing.' "他唱壞了嗓子"
	t	i	i	ta <u>die</u> - <u>duan</u> le tui he fall break ASP leg 'His leg broke by a fall.' "他跌斷了腿" ta <u>ji</u> - <u>hung</u> le lian he worry red ASP face 'His face got red for worrying.' "他急紅了臉"
S P	t	t	t	ta <u>shiu</u> - <u>huei</u> le tzoulu he learn comprehend ASP walk 'He is able to walk after learning.' "他學會了走路"

t t i ta he -tzuei le jiou
he drink drunk ASP wine
'He is drunk.'

"他喝醉了酒"

t i t ta tzou-jin le jiaushr
he walk enter ASP classroom
'He walk into the classroom.'

"他走進了教室"

i i i ta ji -ku le
she worry cry ASP
'She cried for worrying.'

"她急哭了"

shiauli jang-pang le
Shiauli grow fat ASP
'Shiauli gained weight.'

"小李長胖了"

V P t t i ta miau-juen le batz
he gaze accurate ASP target
'He has aimed at the target.'

"他瞄準了靶子"

i i i ta lai -wan le
he come late ASP
'He came lately.'

"他來晚了"

A i t i tamen da -chi -lai le
they fight rise come ASP
'They begin to fight.'

"他們打起來了"

t t i tamen chang-chi ge lai le.
they sing rise song come ASP
"They begin to sing."

"他們唱起歌來了"

i i i ta ku -chi -lai le
he cry rise come ASP
'He begins to cry.'

"他哭起來了"

V-R compounds can be unambiguously decomposed into their constituent morphemes; but, how to predict the subcategorization frames of the compounds from their constituent morphemes?

2. THE LEXICALIZED PROPERTIES AND THE COMPOUNDS

In this work, we take an approach different from our predecessors. We try to factor semantic properties conflated in verbs in order to classify the verbs to discover the restrictions governing the stem collocation and the rule constructing the subcategorization frames of the compound verbs.

Different languages have different strategies of representing meaning incorporation; Mandarin Chinese employs compounding as indicated by the contrast between English (3) and Chinese (4):

- (3) a. He walked into the house.
b. He entered the house.

- (4) a. ta tzou-jin le nejianwutz. "他走進了那間屋子"
he walk enter ASP that house
'He walked into the house.'
- b. ta jin le nejianwutz. "他進了那間屋子"
he enter ASP that house
'He entered that house.'

In English, the predicate WALK involves a preposition to express the semantic property of PATH, while the same property is conflated in the lexical entry of ENTER. In Chinese, TSOU 'walk' "走" is juxtaposed with JIN 'enter' "進" which is a verb. Now, let's examine more closely the following examples:

die-puo "跌破", shuai-puo "摔破",

*diao-puo "掉破", *luo-puo "落破", *dau-puo "倒破";

The lexical items ---- tie "跌", shuai "摔", diao "掉", luo "落", dau "倒" ---- all roughly mean 'fall' or 'drop' and are free morphemes. However, they have different morphosyntactic characteristics with regard to the formation of V-R compounds. What determines the stem collocation properties? Let's shift to the subcategorization frames of this compounding type, investigating the followings:

(5) ta he -guang le jiou. "他喝光了酒"
 he drink exhaust ASP wine
 'He have drunk all the wine.'

'he < AG, TH > + 'guang < arg >' [5]

----> 'he-guang < AG, TH >'
 |
 arg

(6) ta he -tzuei le jiou. "他喝醉了酒"
 he drink drunk ASP wine
 'He is drunk.'

'he < AG, TH > + 'tzuei < arg >'

----> 'he-tzuei < AG, TH >'
 |
 arg

(7) ta (yung jiou) guan-tzuei le lisz. "他(用酒)灌醉了李四"
 he use wine pour drunk ASP Lisz
 'He got Lisz drunk.'

'guan < AG, EXP, (INST) >' + 'tzuei < arg >'

----> 'guan-tzuei < AG, EXP, (INST) >'

|
arg

The Vs as well as the Rs seem to be able to influence the control relation of the relevant arguments. What factors lead to that result? We assume that certain semantic properties conflated from both the Vs and Rs attribute to the determination of these phenomena. Consequently, we will discuss the lexicalized properties and the classification of verbs, the restrictions on stem collocations, and the rule for subcategorization.

2.1. THE LEXICALIZED PROPERTIES AND THE CLASSIFICATION OF VERBS

The theory of lexicalization and the names of the majority of the semantic categories in this proposal are mainly drawn from Talmy (1975, 1985). The definitions of the cited terms adopted here are as follows:

PATH : The respect in which one object is considered as moving or located to another object;

MANNER: Referring to a subsidiary action or state that a THEME manifests cocurrently with its main action or state;

CAUSE : The basic reference is the same as MANNER except that the subsidiary action or state is manifested by an AGENT or INSTRUMENT.

Additionally, there are two terms ---- MOVE and BEL(a mnemonic for 'be-located') ---- which specify two motional states of a motion situation. Now, let's get into the classification of verbs.

The verbs specifying simple motion situations:

(1) MOVE: dung 'to move' "動";

Invloving metaphoric extensions of MOVE:

a. Unaccusative [6]:

i. Sentence required for the participant:

hun "昏", bau "飽", tzuei "醉", sz "死",
'daze' 'full' 'drunk' 'death'
shing "醒", huei "會";
'awake' 'knowing'

ii. Sentence not required for the participant:

huai "壞", tsuo "錯", puo "破", tzang "髒",
'bad' 'wrong' 'broken' 'dirty'
chi "齊", jiang "僵", bai "敗", luan "亂",
'even' 'stiff' 'fail' 'messy'

b. Unergative

ni "膩", yan "厭", fan "煩", lei "累",
'bored' 'sick off' 'annoyed' 'tired'
pa "怕", sheng "勝", ying "贏", shu "輸";
'fear' 'win' 'win' 'lose'

(2) BEL : tzai 'to be at' "在";

Involving metaphoric extensions of BEL:

bai "白", gau "高", dan "淡", tsu "粗",
'white' 'high' 'light' 'coarse'
shin "新", nan "難", tian "甜", ganjing "乾淨";
'new' 'difficult' 'sweet' 'clean'

The motion+PATH-specifying verbs:

lai "來", chiu "去", shang "上", shia "下",
'come' 'go' 'ascend' 'descend'
jin "進", chu "出", dau "到", dau "倒",
'enter' 'exit' 'arrive' 'topple',
guo "過", chi "起", diau "掉", tzou "走",
'pass' 'rise' 'fall' 'walk'
huei "回", lung "攏", kai "開", san "散",
'return' 'together' 'open' 'scatter'

luo "落", zhuei "墜", kua "垮";
'fall' 'fall' 'collapse'

The motion+MANNER-specifying verbs:

(1) MOVE+MANNER

a. Self-agentive situations

tzou "走", pau "跑", tiau "跳", fei "飛";
'walk' 'run' 'jump' 'fly'

b. Undergoer situations

die "跌", guan "滾", liou "流", fu "浮";
'fall' 'roll' 'flow' 'float'

(2) BEL+MANNER

a. Self-agentive situations

jian "站", duen "蹲", tan "躺", tzuo "坐",
'stand' 'squat' 'lie down' 'sit'
shuei "睡";
'sleep'

b. [TH] BEL all over [GO/LOC]

man "滿", bian "遍";
'full' 'all over'

The motion+CAUSE-specifying verbs:

a. [AG] CAUSE [GO] to MOVE

da "打", yi "移", tuei "推", la "拉",
'hit' 'remove' 'push' 'pull'
ban "搬", dau "倒";
'remove' 'pour'

b. [AG: affected] CAUSE [TH/GO] to MOVE

chr "吃", he "喝", ting "聽", shuei "學"
'eat' 'drink' 'listen' 'learn'

- c. [TH] MOVE into a status by [AG] MOVE it in a specific environment

shai "曬", hung "烘", jin "浸", shiun "燻";
'shine' 'toast' 'soak' 'smoke'

- e. [AG] CAUSE [TH] to MOVE into existence

shie "寫", tzuo "做", wa "挖", gai "蓋";
'write' 'do' 'dig' 'build'

- f. [AG] CAUSE [TH] to BEL

fang "放", tian "填", sai "塞", tie "貼",
'put' 'fill' 'fill' 'paste'
gua "掛", bai "擺", ge "擱";
'hang' 'place' 'lay'

- g. [AG1] INDUCE [AG2: affected/Exp] to MOVE

guan "灌", wei "餵", jiau "教", tsau "吵",
'pour' 'feed' 'teach' 'hubbub'
shia "嚇", dou "逗";
'scare' 'tease'

The above classification is mainly based on the properties entailed by the meanings of verbs. We will show how the classification predicts V-R compounding results in the next section.

2.2. INTERNAL TO THE COMPOUNDS:

THE RESTRICTIONS ON STEM COLLOCATIONS

Structurally, the bulk of the V-R compounds are formed from simplex lexical elements; though themselves are lexical items, they can not enter the word formation process to construct the further V-R compounds in a recursive fashion like the modifier-head compounds[7]. Semantically, both concatenated members of the

compounds have predicative functions; and, their meanings are generally that the second member describes the state of the subject, the object, or the event after the completion of the action described by the first member. Generally speaking, their constituent morphemes are easily identifiable and the knowledge of the meanings of constituent morphemes is sufficient for native speakers to interpret the compounds when they are encountered in context.

How contextual information determines the compatibility of two morphemes involved in V-R compounds is not our concern here. We focus on the inherent meanings of the concatenated members. The possible combinations of V-R compounds with causative-resultative reading:

V2	MOVE	BEL	+PATH	MOVE+MANNER	BE+MANNER	+CAUSE
V1						

MOVE	+	+	+			
BEL	+	+	+			
+PATH			+			
MOVE+MANNER	+	+	+		+	
BE+MANNER	+	+	+		+	
+CAUSE	+	+	+		+	

The restrictions that we induce from the above combinations are described as the following:

Re.1: V:[+PATH] can not be the V1 except when V2 is also [+PATH].

Re.2: V:[+CAUSE] and V:[MOVE+MANNER] can not be V2.

Re.3: *[V1:MOVE/BEL - V2:[BE+MANNER]]_V

By the restrictions we can explain the following contrast:

die-puo, shuai-puo, *diau-puo, *luo-puo, *dau-puo
跌 破 摔 破 掉 破 落 破 倒 破

The motion-PATH-specifying verbs can not be the first member in a V-R compounds with causative-resultative reading in Chinese lexical system.

2.3. EXTERNAL TO THE COMPOUNDS: THE RULES FOR SUBCATEGORIZATION

Superficially, the morpholexical process which produces the derived subcategorization frames of the compounds from the concatenated verb stems seem to be highly irregular; and the control relations between the relevant arguments do not show any significant regularity. In this section, we try to probe into these phenomena in the view of intrinsic meanings of the concatenated members. The rule of constructing the subcategorization frames is as follows:

Given two verbs V1 and V2 as the V member and the R member of a V-R compound respectively;

IF : The argument structure of V2 is < arg, (LOC)[8] >
THEN: IF : V2 is an unergative verb
THEN: The arg of V2 is controlled by AGENT of V1
OTHERWISE: IF : The arg should be [+sentient] entailed by the meaning of V2
THEN: The arg of V2 is controlled by the EXPERIENCER or AFFECTED AGENT of V1
OTHERWISE: The arg of V2 is the THEME or GOAL of the V-R compound [V1-V2]_V.

From the above rule, the choice of controller is predictable. For instance:

IF: V2 is an unergative verb.

- (a) ta chr-ni le. "他吃膩了"
he eat bored ASP
'He is tired of that food.'

'chr < AG, TH >' + 'ni < arg >'

----> 'chr-ni < AG, TH >'

|
arg

- (b) ta shiu-fan le. "他學煩了"
he learn tired ASP
'He is tired of learning.'

'shiu < AG, TH >' + 'fan < arg >'

----> 'shiu-fan < AG, GO >'

|
arg

- (c) ta jiau -lei le. "他教累了"
he teach tired ASP
'He is tired of teaching.'

'jiau < AG, EXP, TH >' + 'lei < arg >'

----> 'jiau-lei < AG, EXP, TH >'

|
arg

OTHERWISE:

IF: the arg of V2 should be [+sentient] entailed by the meaning of V2.

- (a) ta chr-bau (fan) le "他吃飽(飯)了"
he eat full (rice) ASP
'He has had enough food.'

'chr < AG, TH >' + 'bau < arg >'
affected

----> 'chr-bau < AG, TH >'

|
arg

(b) ta shiue -huei le "他學會了"
 he learn knowing ASP
 'He learned it.'

'shiue < AG, GO >' + 'huei < arg >'
 affected

----> 'shiue-huei < AG, GO >'
 |
 arg

(c) ta (yung jiou) guan-tzuei liz le "他(用酒)灌醉李四了"
 he use wine pour drunk Lisz ASP
 'He got Lisz drunk.'

'guan < AG, EXP, INST >' + 'tzuei < arg >'

----> 'guan-tzuei < AG, EXP, INST >'
 |
 arg

OTHERWISE:

(a) ta ku -shr le shoupa. "他哭溼了手帕"
 he cry wet ASP handkerchief
 'He cried so much that the handkerchief got wet.'

'ku < AG >' + 'shr < arg >'

----> 'ku-shr < AG, < TH > >'
 causer |
 arg

(b) ta han -ya le shangtz. "他喊啞了嗓子"
 he yell toarse ASP throat
 'He has a toarse voice because of yelling.'

'han < AG >' + 'ya < arg >'

----> 'han-ya < AG, < TH > >'
 undergoer |
 arg

(c) henduoren e -sz le "很多人餓死了"
 many people hungry die ASP
 'Many people have been starved to death.'

'e < TH >' + 'sz < arg >'

----> 'e-sz < TH >'
 |
 arg

(d) ta tzou-jin le jiaushr "他走進了教室"
 he walk enter ASP classroom
 'He walked into the classroom.'

'tzou < TH >' + 'jin < arg, LOC >'

----> 'tzou-jin < TH, LOC >'
 |
 arg

(e) ta ban -dung le nekuaidashrtou "他搬動了那塊大石頭"
 he remove move ASP that big stone
 'He have reomved that big stone.'

'ban < AG, GO >' + 'dung < arg >'

----> 'ban-dung < AG, GO >'
 |
 arg

(f) ta ba yitz tuei-jin keting "他把椅子推進客廳"
 he BA chair pull enter parlour
 'He pushed the chair into parlour.'

'tuei < AG, GO >' + 'jin < arg, LOC >'

----> 'tuei-jin < AG, GO, LOC >'
 |
 arg

So far, everything is not simple, but neither is it as messy as we thought at the very beginning.

3. CONCLUDING REMARKS

The V-R compounding is a rich source of new verbs in Mandarin Chinese. It is quite a problem all along for lacking of a rule-governed process to deal with how the function of a V-R compound is related to the functions of its constituents, though its internal structure is rather simple. This work tries to probe into the problem by means of lexical decomposition. We do get some instructive results, but there are, we think, some tasks needing more effort: HOW MANY and WHAT primitives, like MANNER, PATH, etc., will we need to capture the details of the lexical information involved competence? WHAT about the syntactic realization of the arguments of the compounds, do the meanings conflated in a lexicon determine the syntactic representations to the arguments the lexicon subcategorizes?

NOTES

1. The name 'V(erb)' we use includes adjectives; since Chinese adjectives can function as predicatives without linking verb, we adopt the view of treating them as a class of verbs (Chen & Huang (1989)).
2. Most potential complement compounds are formed by infixion of de "得" to separable V-R compounds.
3. Chao(1968) considers that the majority of compounds with the complements of this type have idiomatic meanings, such as

ai -shang le ta, "愛上了他"
love ascend ASP him
'to have fallen in love with him'

chou -shang le yin, "抽上了癮"
smoke ascend ASP addition
'to become addicted to smoking'

shuo-kai le, "說開了"
say open ASP
'call a spade'

fa -chi yi ge yundung. "發起一個運動"
issue rise one CL movement
'to initiate a movement'

These complements are pervasive in Chinese Lexical system; such as

(1) Aspectual usages:

-shang 'ascend' "上" ---- to start and continue; such as

shihuan-shang, "喜歡上"
like ascend
'to become fond of'

kan -shang, "看上"
look ascend
'to take a fancy on'

-chi 'rise' "起" ---- to start; such as

chang-chi ge lai le "唱起歌了"
sing rise song come ASP
'to begin to sing'

(2) Abstract usages:

nau -kai le, "鬧開了"
fight open ASP
'to have come to an open conflict'

chau -fan lian le; "吵翻臉了"
quarrel turn-over face ASP
'to have turned hostile for quarreling'

If treating these compounds as idioms, we could lose some information. So, we suggest to capture the predicative functions of the metaphoric usages of these complements by rules instead of listing idioms.

4. The construction and content of this form is largely based on Fan Shiau 范曉 (1987).

Annotations for the symbols used in the form:

A: Aspect

S: Subject

O: Object

P: Predicate

R: The COMPLEMENT member of a V-R compound

V: The VERB member of a V-R compound

t: transitive

i: intransitive

5. AG(ent): the object which performs the action;
TH(eme): the object which undergoes change of location or state, of which location or state is described;
GO(al) : the object to which an action or event is directed, literally or metaphorically;
INST(rument) : the object which is used for accomplishing an action;

EXP(eriencer): the object which perceives an action or undergoes an emotion or an intellectual state;
 LOC(ative) : the place in which an event takes place or of which a state is described.

6. The term UNACCUSATIVE verb was first proposed in Relational Grammar by Perlmutter(1978) to refer to a subclass of intransitive verbs whose subjects seem to be patientlike; the another term UNERGATIVE verb is used to refer to the other subclass of intransitive verbs whose subjects are agentlike.
7. For instance: jiun-shr-wei-yuan-huei "軍事委員會" has the internal structure ---- $[[N-N]_N - [[N-N]_N - N]_N]_N$, where all of the intermediate stages are of the same structure ---- $[N-N]_N$.
8. If this V2 is [+PATH].

REFERENCES

- Chao, Y. R. 1968. *A Grammar of Spoken Chinese*. University of California Press. Berkeley, Los Angeles, London.
- Chen, K.-J. and C.-R. Huang. 1989. *Word Classifications and Grammatical Representations in Chinese*. ms. Academia Sinica, Taipei.
- Croft, W. A. 1986. *Categories and Relations in Syntax: the Clause-Level Organization of Information*. Dissertation. Stanford.
- Dowty, D. 1988. *Thematic Proto-Roles, Subject Selection, and Lexical Semantic Defaults*. 1987 LSA Colloquium Paper.
- Jackendoff, R. 1983. *Semantics and Cognition*. MIT Press, Cambridge, Massachusetts.
- Jackendoff, R. 1987. *The Status of Thematic Relation in Linguistic Theory*. *Linguistic Inquiry* 19.3: 369-411.
- Perlmutter, D. M. 1978. *Impersonal Passives and the Unaccusative Hypothesis*. *Proceeding of the fourth BLS*: 157-189. Berkeley Linguistics Society.
- Saksena, A. 1980. *The Affected Agent*. *Language* 56.4: 812-826.
- Talmy, L. 1975. *Semantics and Syntax of Motion*. In J. Kimball, ed., *Syntax and Semantics* 4. New York, Academic Press.
- Talmy, L. 1985. *Lexicalization Patterns: Semantics Structure in Lexical Forms*. In T. Shopen, ed., *Language Typology and Syntactic Description* 3. Cambridge University Press.
- Talmy, L. 1988. *The Relation of Grammar to Cognition*. In Brygida Rudzka-Ostyn ed., *Current Issues in Linguistic Theory* 50: *Topics in Cognitive Linguistics*.
- Tang, T.-C. Charles 1988. *On the Notion "Possible Verbs of Chinese"*. *Tsing Hua Journal of Chinese Studies, New Series* XVIII, No.1, 43-69. Hsinchu, Taiwan.

李臨定，1980．動補格句式．中國語文 1980年 第二期：93-102．

馬希文，1987．與動結式動詞有關的某些句式．中國語文 1987年 第六期：
424-441．

范曉，1987．V-R 及其所構成的句式．語言研究集刊 第一輯．復旦大學出版社：230-247．

張嘉賓，1984．動補結構與其賓語之間的語意、語法關係．求是學刊 1984年 第一期：26-30．

A Quantitative Comparison Between an LR Parser and an ATN Interpreter

Chao-Lin Liu and Keh-Yih Su

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

Abstract

We have known that a bottom-up parser will parse an English sentence faster than a top-down parser does. Nevertheless, there is still no report on how much faster it is. To quantitatively compare these two parsers, an LR parser and an ATN interpreter are built and are used as bottom-up and top-down parsers respectively. These two parsers are currently widely used in the computational linguistics community. From the tests we have proceeded, we find that the average parsing speed of the LR parser is tens of times faster than that of the ATN interpreter.

Introduction

LR and ATN parsers are currently two of the most widely used parsers for natural languages processing. In this paper, we will compare their parsing speed quantitatively. Since the speed is the main topic of concern, not the grammar formalism, the grammars used in both parsers are kept the same through the whole comparison.

In the design of translators of programming languages, bottom-up translation has been one of the well-known strategies. Among the parsers based on bottom-up translation techniques, LR parser [AHO 86] is the most popular one. A lot of translators, like C compiler and YACC in UNIX, take advantage of an LR parser to translate the input files.

Although the LR parser has been a good parser for the translation of programming languages. It can not be used to parse natural languages, like English, directly. The point is that the traditional LR parser does not accept grammars that are ambiguous, while the grammars for natural languages are usually ambiguous. But, due to its success in parsing programming languages, researchers [SU 85, TOMI 85, HSU 86] augmented it to accept ambiguous grammars and give it the power to handle linguistic problems, so that similar techniques can be used to parse natural languages.

On the other hand, another famous formalism called Augmented Transition Networks (ATN) [WOOD 70] based on top-down translation techniques were designed to parse natural languages. ATN formalism is derived from the Recursive Transition Networks (RTN) which is in return derived from the Basic Transition Networks (BTN). Both RTN and BTN are not adequate to parse the natural languages due to some shortcomings of them [BATE 78]. In addition to the intrinsic features of RTN and BTN, the arcs of an ATN are associated with actions and conditional checks. This augmentation gives an ATN the computational power of a Turing machine [FINI 83].

The major difference between ATN and augmented LR parsers is the way by which they form a larger node from their constituents. An ATN parser is essentially a top-down parser which adopts hypothesis-driven paradigm [MARC 80]. On the other hand, the LR parsers are bottom-up parsers, and thus are data-driven. Due to this, ATN is expected to parse English slower than LR parser does. However, we are curious about how much slower the ATN is when compared with the LR parser. In this paper, we will describe the tests we have conducted on our LR parser and ATN interpreter and compare their parsing speed quantitatively.

The Environment

Both parsers under testing are currently implemented on the SUN 3/160C workstation. The LR parser is written in C language while the ATN interpreter is written in SUN Common Lisp Version 2.0. Before the comparison, the LR parser is compiled into the executable object codes of the SUN 3/160C, and the ATN interpreter is compiled into binary codes that are executable by the Lisp interpreter of the SUN 3/160C. A simple Benchmark shows that the

compiled Lisp codes have almost identical execution speed to that of a compiled C program, therefore, we shall ignore the possible speed-up effects introduced by the differences between these two languages. The tests are proceeded on the SUN 3/160C workstation which, declared by the SUN microsystems, INC., is a 2 MIPS machine.

The reason that we implemented an ATN interpreter instead of an ATN compiler is that the former is far easier to implement than the latter is. According to Finin's experience [FINI 83], the parsing speed of the former is about 5 times slower than the latter's. Knowing about this, we do not bother to implement an ATN compiler to compare an ATN and LR parser. On the contrary, we only have to build an ATN interpreter to compare them. Besides, the reason to choose Common Lisp instead of C as the implementation language of ATN interpreter is that ATN interpreters are typically implemented in Lisp. As described in the last paragraph, the compiled Lisp has almost identical execution time to that of a C program, hence, the differences in these two languages should not have significant effects on the comparison. Finally, the arcs and actions of the ATN interpreter are implemented as suggested in [BATE 78, CHRI 83, FINI 83].

Under this environment, we will compare the parsing speeds of a typical LR parser and a typical ATN interpreter.

The Parameters Controlled and Compared

To compare these two parsers, we must use the same grammar, the same lexicon, and the same set of sentences as test data. The grammar we used is the second test grammar in appendix F of [TOMI 85], as shown below:

```

S : NP VP PP PP ;
  : NP VP PP ;
  : PP NP VP ;
  : NP VP ;
  : S conj S .
NP : NP conj NP ;
    : NP1 that S ;
    : NP1 S ;
    : NP1 .
PP : PP conj PP ;
    : prep NP .
NP1 : ADJM NPO PP PP ;
      : ADJM NPO PP ;
      : ADJM NPO ;
      : NPO PP ;
      : NPO ;
      : NPO PP PP .
ADJM : adj ;
      : adj ADJM ;
      : ADVM adj ;
      : ADJM conj ADJM .

```


NPO : NM ;
 : ADJM NM ;
 : art NM ;
 : art ADJM NM .
 NM : n ;
 : n NM .
 ADVM : adv ADVM ;
 : adv ;
 : ADVM conj ADVM .
 VP : VC NP ;
 : VP conj VP ;
 : VC .
 VC : aux v ;
 : v .

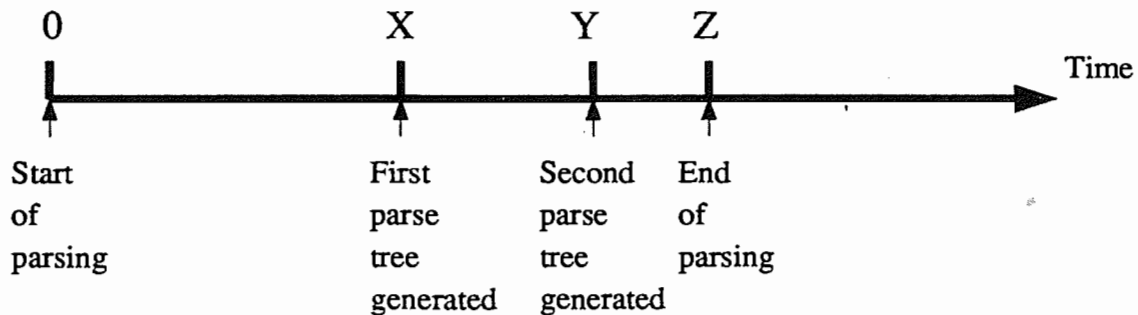
In the grammar listed above, the capital symbols represent the non-terminal symbols while the other symbols represent terminal symbols. Written in the form of ATN grammar, this grammar contains 45 states and 67 arcs. And, the lexicon we used is a small one which currently contains only 61 words.

39 sentences are used in the tests. They can be divided into several groups according to their word counts and number of ambiguities. These two parameters will affect the parsing time needed to parse the sentences. In the next section, tests will be taken to see how they affect the parsing speed of the parsers.

Two parsing times will be compared in the tests, they are :

- (1) First Parse tree Time (FPT) : the period of time from the beginning of parsing to the time the first legal parse tree is generated.
- (2) Average Parse tree Time (APT) : the period of time from the beginning of parsing to the time the last parse tree is generated divided by the number of parse trees generated.

For example: Suppose that sentence A has two ambiguities, and when A is parsed the timings are recorded as shown below:



then $FPT = X$ and $APT = Y/2$.

Tests and Discussions

In this section, we will call the ATN parser 'ATNP' and the LR parser 'LRP' for short. In each of the tests, the test sentences and the results are discussed.

Test 1: Test for sentences with the same number of ambiguities but different number of words.

In this test, the number of ambiguities of the sentences is the same but the number of words of the sentences is different. Five sentences are used in this test, as shown below. The sentences listed below have 5, 10, 15, 20, and 25 words respectively, and all of them have 2 syntactic ambiguities with respect to the test grammar. Because the vocabulary for the test version of the parsers is limited, these sentences may be semantically nonsense.

- (1) Cruelly cruelly cruel computer loves.
- (2) I require the beautiful beautiful beautiful computer in the apple.
- (3) This good maintenance will show where shows the tree at the station where in plane.
- (4) The angry angry angry woman that the good good good man loves kills the green green green apple in plane.
- (5) The angrily angrily angrily angry woman that the man loves kills the angrily angrily angrily angry woman that the man loves in the green apple.

The results are shown in Figure 1, Figure 2, and Figure 3 on the next pages. Figure 1 and Figure 2 are the timings of the ATNP and LRP. The X-coordinates of both figures represent the number of the parse tree generated. The '0', on the X-coordinate, represents the beginning of parsing. The 'END', on the X-coordinate, represents the end of parsing. The Y-coordinates of both figures represent the CPU time¹ taken by the parsers to parse the sentences. Each line in the figures represents the timings for one sentence. For examples, as shown in Figure 1, it takes ATNP about 8.5 seconds CPU time to generate the first parse tree for sentence (4) and 9.7 seconds CPU time to generate both parse trees for sentence (4). The curves for ATNP have larger initial slopes, but the slopes become flatten after the first parse tree is acquired. The curves for the LRP, on the other hand, does not show this feature. That is, most of the parsing time taken by the ATNP to parse a sentence is used to find the first parse tree while the LRP is not.

¹ CPU time = User CPU time + System CPU time of process, and henceforth.

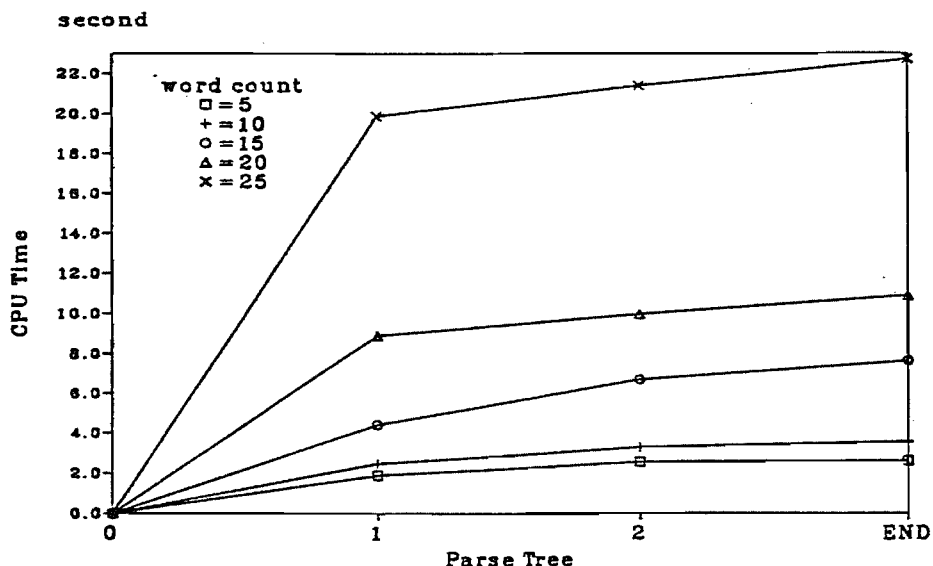


Figure 1 : ATNP timings for sentences with the same number of ambiguities

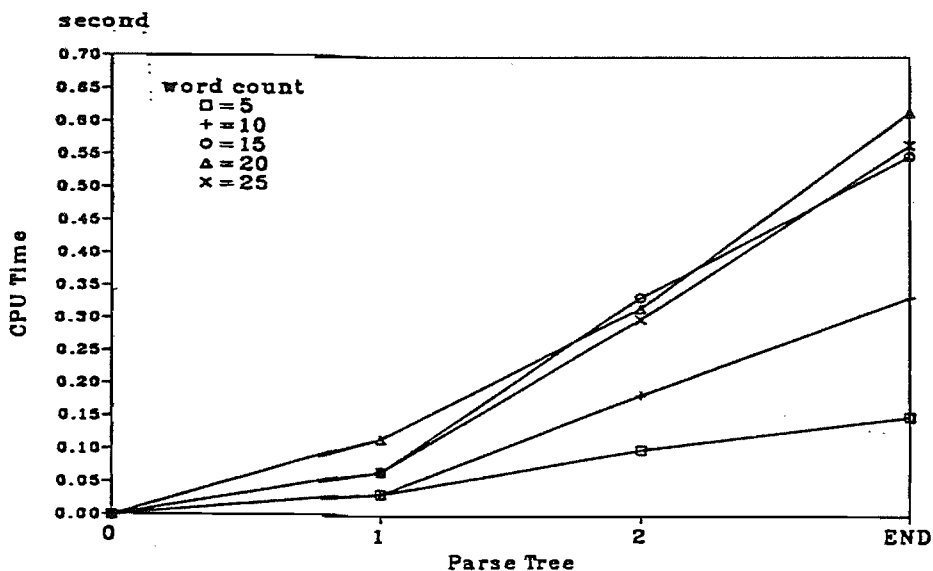


Figure 2 : LRP timings for sentences with the same number of ambiguities

In Figure 3 on the next page, the X-coordinate represents the word count of the sentences and the Y-coordinate represents the time ratio of the ATNP and LRP. The line with square markers is the time ratio of the times taken by the ATNP and LRP to generate the *first* parse tree of the sentences, that is, the ratio of FPT of ATNP and LRP. Similarly, the line with plus markers is the time ratio of the times taken by the ATNP and LRP to generate *both* parse trees of the sentences. Finally, the line with circle markers is the time ratio of the times taken by the whole parsing process. From this figure, we find a number of interesting facts:

- (1) For FPT's, the ATNP is slower than the LRP for a factor of at least 55.
- (2) To generate both parse trees, the ATNP is slower than the LRP for a factor of at least 20.

- (3) From point (1) and (2) above, the speed-up or reduction in time becomes less drastic after the first parse tree is acquired. This is because most information acquired in the analysis of the first parse tree is retained and is available to successive analyses during backtracking.

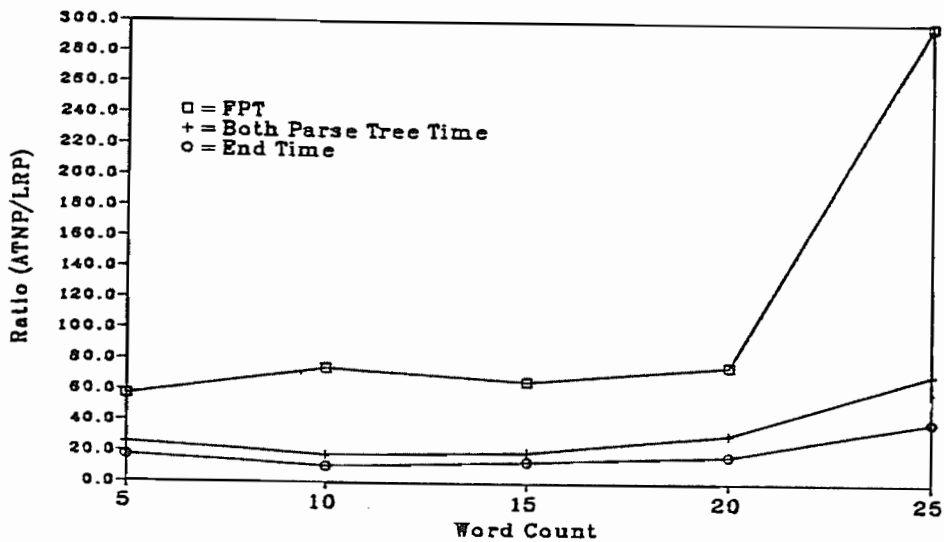


Figure 3 : Ratios of timings (ATNP/LRP)

The first observation suggests that one can benefit from an LR parser significantly if the first parse tree is the one to be used as output. This is usually the case for a system with well-defined scoring mechanism.

Test 2: Test for sentences with different number of ambiguities.

In this test, 34 sentences are used. All of them have 10 or 11 words. And the number of sentences having 1, 2, 4, 5, 6, 8, 10, and 12 ambiguities are 9, 5, 7, 3, 2, 5, 2, and 1 respectively. To save space, we will not list the sentences here.

The results are shown in Figure 4 and table 1 on the next page. In Figure 4, the Y-coordinate represents time ratio of the ATNP and the LRP, and the X-coordinate represents the number of ambiguities for the test sentences. The line with square markers is for FPT while the line with plus markers is for APT.

All timings used to calculate the ratios in Figure 4 are the average ones. In other words, the FPT's or APT's for a given number of ambiguities, say 2 ambiguities, are averaged before their ratio are computed. For example, from Figure 4, we know that the ratio of the average FPT is about 60 for sentences having 2 ambiguities. From Figure 4, we also find that :

- (1) The more ambiguities the sentences have, the larger the FPT ratio is. And the ratio is at least 35.
- (2) The more ambiguities the sentences have, the lesser the APT ratio is. And the ratio is at most 35.

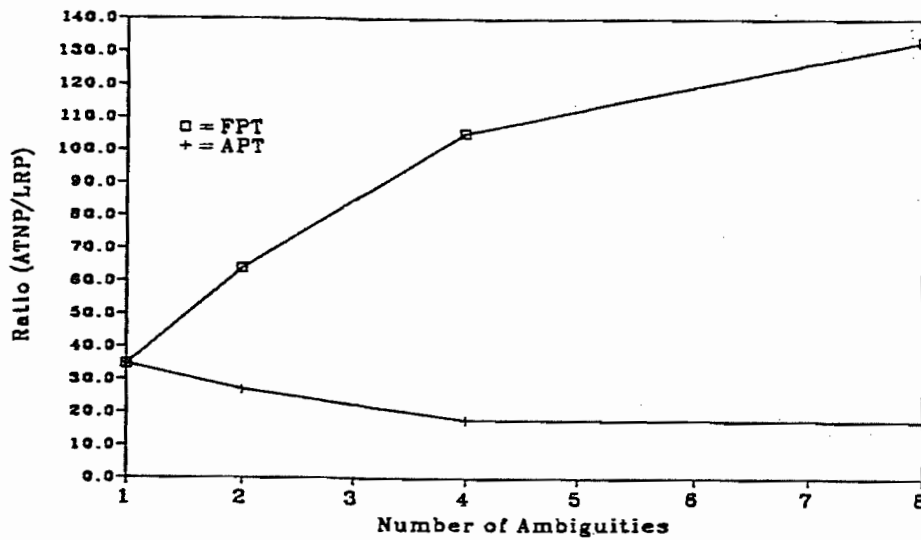


Figure 4 : Ratios of FPT and APT (ATNP/LRP)

#AMBIGUITY	PARSER	ATNP		LRP	
	STATISTICS	FPT	APT	FPT	APT
1	STD	0.51	0.51	0.020	0.020
	AVG	1.99	1.99	0.057	0.057
	STD/AVG	0.26	0.26	0.34	0.34
2	STD	3.62	2.45	0.065	0.093
	AVG	6.18	4.04	0.097	0.150
	STD/AVG	0.59	0.61	0.67	0.62
4	STD	1.71	0.51	0.015	0.007
	AVG	4.51	2.15	0.043	0.121
	STD/AVG	0.38	0.24	0.35	0.06
8	STD	2.05	0.85	0.032	0.020
	AVG	8.45	2.59	0.063	0.149
	STD/AVG	0.24	0.33	0.51	0.14
TOTAL	STD	3.77	1.31	0.037	0.056
	AVG	5.67	2.60	0.063	0.124
	STD/AVG	0.67	0.50	0.59	0.45

Table 1 : Statistics of the second test. (Unit : second CPU time)

In Table 1, we list some statistics about the FPT's and APT's in this test. The number in the leftmost column represents the number of ambiguities of the test sentences. The row labeled 'TOTAL' represents all of the 34 test sentences. In each row, 'STD' and 'AVG' stand for the *standard deviation* and *average* value, and 'STD/AVG' stands for the ratio of standard deviation to the average. Using statistics in this table, we can derive some other statistics.

For example, to find the difference of the average FPT's of these 34 sentences. From this table, we find that the average FPT of the ATNP is 5.67 second CPU time and LRP is 0.063 second CPU time. From this table, we find that :

- (1) For the 34 sentences, the FPT ratio of ATNP and LRP is 90.43, that is, in average, the ATNP is 90 times slower than the LRP in generating the first parse tree of the sentences.
- (2) For the 34 sentences, the APT ratio of ATNP and LRP is 20.99, that is, in average, the ATNP is 21 times slower than the LRP in generating all the parse trees of sentences.
- (3) For the 34 sentences, the 'STD/AVG' of FPT of both ATNP and LRP are about 0.60. This means that the variance, in the sense of percentage changes, of the time needed to generate the first parse tree by both parsers is about the same.

From the tests discussed above, we see that ATNP is slower than the LRP. We think that the major reasons for this phenomenon are :

- (1) Top-down parsing is intrinsically inferior to bottom-up for natural languages like English. In other words, the characteristics of English makes it more desirable to use a data-driven parser instead of a hypothesis-driven one.
- (2) The state transition of LRP is explicitly coded in the parsing table while the ATNP is not. So, in the course of parsing, the LRP does not have to recompute the next possible state transition while the ATNP does have to.
- (3) The Lisp in itself is slower than the C language. Although it is easier to implement an ATN interpreter in Lisp. This factor, however, is consider less significant when both parser/interpreter are compiled, because our preliminary Benchmark shows that they have almost identical execution time for the Benchmark.

Furthermore, using simple toy grammar such as the one we used in the tests, the ATN interpreter has been so slow in the parsing speed compared with the LR parser . We expect that their difference in parsing speed will be even greater with more complicated grammar, say a grammar for a machine translation system.

Conclusion

In this paper, we have shown that the average parsing speed of a typical ATN interpreter is tens of times slower than that of the LR parser. And we have discussed briefly why the ATN interpreter is slower than the LR parser. Although an ATN parser can be easily constructed, it may not be practical, as far as parsing efficiency is considered, in constructing a large system which has a complicated grammar.

Acknowledgement

We are grateful to Professor I-Pen Lin at the Department of Computer Science and Information Engineering of National Taiwan University for his helpful recommendation and suggestions on the implementation of the ATN interpreter.

References

- [AHO 86] Aho, A. V., R. Sethi and J. D. Ullman, *Compilers : Principles, Techniques and Tools*, Addison-Wesley Publishing Company, Reading, MA, 1986.
- [BATE 78] Bates, M., "The Theory and Practice of Augmented Transition Network Grammars," in *Natural Language Communication with Computers*, pp. 191–259, Leonard Bolc (ed.), Springer Verlag, 1978.
- [CHRI 83] Christaller, T., "An ATN Programming Environment," in Bolc, L. (ed.) *The Design of Interpreters, Compilers, and Editors for Augmented Transition Networks*, pp. 71–148, Springer Verlag, 1983.
- [FINI 83] Finin, T. W., "The Planes Interpreter and Compiler for Augmented Transition Network Grammars," in Bolc, L. (ed.) *The Design of Interpreters, Compilers, and Editors for Augmented Transition Networks*, pp. 1–69, Springer Verlag, 1983.
- [HSU 86] Hsu, H.-H. and K.-Y. Su, "A Bottom-up Parser in the Machine Translation System with the Essence of ATN," *Proc. Int. Computer Symposium (ICS) 1986*, Vol. 1, pp. 166–173, Tainan, Taiwan, R.O.C., 1986.
- [MARC 80] Marcus, M.P., *A Theory of Syntactic Recognition for Natural Language*, MIT Press, London, U.K., 1980.
- [SU 85] Su, K.-Y., H.-H. Hsu, and M.-L. Hsiao, "Development of an English-to-Chinese Machine Translation System," *Proc. Natl. Computer Symposium (NCS) 1985*, Vol. 2, pp. 997–1001, Kao-Hsiung, Tainwan, R.O.C., 1985.
- [TOMI 85] Tomita, M., *An Efficient Context-Free Parsing Algorithm for Natural Languages and its Applications*, Ph. D. Dissertation, Carnegie-Mellon University, 1985.
- [WOOD 70] Woods, W. A., "Transition Network Grammars for Natural Language Analysis," *Communication of ACM*, pp. 591–606, October 1970.

Parsing English Conjunctions And Comparatives

Using The Wait-And-See Strategy

Rey-Long Liu and Von-Wun Soo

Institute of Computer Science

National Tsing-Hua University

ABSTRACT

The major problems in parsing conjunction and comparative English sentences are ambiguities of the scoping and the ellipsis. For a correct parsing, the parser must use not only the syntax but also the semantic information of these sentences. However, as Chiang et. al. [1] pointed out, the semantic information of these sentences can only be obtained after these sentences have been parsed. It is also the reason why a syntax-directed parsing strategy without collecting adequate semantics of input sentences needs to backtrack each time when it makes incorrect assumptions during parsing.

The Wait-And-See strategy, introduced by Marcus [2], is based on the "determinism hypothesis" which claims that the natural language can be parsed by a computationally simple mechanism without backtracking. In this paper, we show a method using the Wait-And-See strategy to parse conjunctions and comparatives simultaneously. In order to enhance the efficiency and correctness of the parser, several mechanisms such as bottom-up preparsing, suspension, and pattern matching are implemented. The bottom-up preparsing looks up the dictionary and recognizes isolated sentence fragments which can be determined without ambiguities. Suspension allows the parser to suspend temporarily at ambiguous points and continue to parse the rest of the sentence until it obtains necessary information to resolve the ambiguities. Pattern matching uses the concept of symmetry to detect missing components (the ellipses) in the two conjuncted or compared sentence fragments.

1. Introduction

When parsing sentences with conjunction and/or comparative words, it is possible to make incorrect assumptions at some decision points. Ambiguities of scoping and ellipsis are the major problems in parsing conjunctions and comparatives. Scoping

problems occur when a parser has no adequate information to detect the boundaries of constituents, while ellipsis problems occur when a parser has no adequate information to determine the missing components.

For solving the scoping ambiguities, Kosy [6] proposed a Wait-And-See strategy to parse conjunctions deterministically. Rules are written separately to handle the detection of the boundaries of constituents (segmentation rules) and the valid attachment of constituents (recombination rules) respectively. Segmentation operations are separated from and always proceed the recombination operations. This parser can parse many complex sentences efficiently. However, it has difficulty when parsing sentence:

John gives Mary the pen that I give you and Bob gives the man who smiles in the classroom an apple.

In order to detect the boundary of the NP *the man who smiles in the classroom*, it needs to use the recombination operation to "recombine" the clause *who smiles in the classroom*. However, this type of interleaving operations is not allowed in their parsing method. Thus when the recombination operation proceeds, it will not have adequate information to determine the scope of the conjunction word *and*.

For solving the ellipsis ambiguities, Huang [8] presented an algorithm to resolve the ambiguities of ellipses including Gapping, Right Node Raising, Reduced Conjunctions. However, the scoping ambiguities remained unaddressed. Kwasny [3] treated conjunctions as ellipses (ungrammatical forms) and handled them with a pattern matching method. When a conjunction word is seen, patterns are generated dynamically from already identified elements and matched against the remaining segments of an input sentence. This treatment reduces the size of grammar rules and handles the ellipsis ambiguity problem very well. However, the scoping ambiguity problem still remains unsolved. For example, when parsing sentence:

John gives Mary the pen that I give you and Bob gives Jane a pen.,

a simple pattern matching can not determine which grammatical constituents are actually conjuncted by the conjunction.

Huang [8], Ryan [9], and Chiang et. al. [1] had analyzed many sentences of different conjunction and comparative types. Chiang et. al. [1] also implemented an ATN parser for parsing such kind of sentences. Their parser requires preparing the basic terms (including noun phrases, verbs, conjunction words, and prepositions) which reduces the reconstruction of basic terms when backtracking. And while parsing basic terms, the parser collects semantic information of the sentence for later construction. Thus, the efficiency is promoted. However, as they pointed out, there is still one drawback in their parser, --- it cannot deal with the sentence which has both conjunction and comparative words. This is because the ATNs for conjunction and the ATNs for comparatives are written independently. We must write other ATNs to handle a sentence with both conjunction and comparative words. However, this could cause too much overhead.

For solving scoping and ellipsis problems and parsing conjunctions and comparatives simultaneously and deterministically, we implemented an efficient parser based on the Wait-And-See strategy.

2. The Wait-And-See Strategy

The Wait-And-See strategy, introduced by Marcus [2], is based on the "determinism hypothesis" which says that a natural language can be parsed by a computationally simple mechanism without backtracking.

A Wait-And-See Parser (WASP) has a production system architecture, whose grammar and parsing heuristics are expressed in terms of rules which are composed of condition and action parts. Two major data structures, defined by Marcus [2], are:

1. active node stack: a pushdown stack of incomplete constituents,

2. lookahead buffer: a small constituent buffer containing constituents which are complete, but whose higher grammatical function is as yet uncertain.

In general, the rules in a WASP are partitioned into rule packets. Each rule packet contains rules which are particularly for the configuration of the top of the node stack. For example, if the top of the node stack is a VP, the corresponding rule packet for the VP is activated. However, the selection of which rule to fire may depend on the contents of the lookahead buffer and the node stack. Readers who are not familiar with the Wait-And-See strategy are referred to a chapter in Allen's book [10].

Since a WASP partitions its knowledge base into independent parts, it has the merits of modularity. We can extend easily to handle more complex type of sentences, and introduce heuristics for each part of knowledge individually to take care of different types of sentences. However, there are still some tasks to be made to improve efficiency --- including bottom-up preparsing, suspension, and pattern matching which are to be discussed in detail in section 3, section 4, and section 5 respectively.

3. Bottom-up Preparsing

According to Winston [7], a WASP requires preparsing the NPs in the original input sentence. In general, simple NPs can be preparsed deterministically, but not a complex NP.

We introduce the bottom-up feature of parsing to promote the efficiency of the parser. In fact, the bottom-up preparsing looks up the dictionary and performs a simple type of pattern matching to recognize isolated sentence fragments which can be determined without any ambiguities.

There are four types of grammatical constituents to be preparsed:

a. word types:

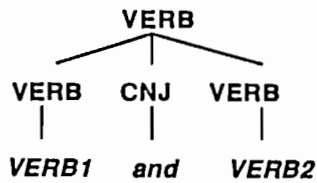
e. g. VERB, NOUN, PREPOSITION, ..., etc.

b. simple NPs:

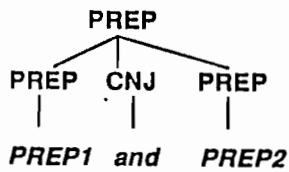
e. g. [DET] (ADJ)* [NOUN].

c. simple conjunctions of words with the same types:

e. g. For a pattern like "VERB1 and VERB2" where VERB1 and VERB2 share the same verb type, we treat it as a VERB and the following tree is constructed:



Similarly, for the pattern like "PREP1 and PREP2" where both PREP1 and PREP2 are prepositions, we combine two conjuncted prepositions into one without any ambiguities:



d. idioms

e. g. "take care of" may be treated as a VERB.

For example, if the input sentence is:

I meet and take care of the patient at and through the night.,

the result after bottom-up preparsing will be:

((NP I)

(VERB (VERB meet) (CNJ and) (VERB (take care of))

(PREP (PREP at) (CNJ and) (PREP through))

(NP (DET the)(NP night)))

Preparsing obtains a lot of important information for our WASP. This will contribute greatly to a correct parsing.

4. Suspension

Hayes et. al. [4] used the concept of parsing suspension to the problem of interjection, restart, and implicit termination in spoken and written languages. The main purpose of its parsing suspension is to provide a flexible way to ignore the input mismatch. In our problem domain, the suspension used here is quite different from that in Hayes et. al. [4]. In order to parse a sentence deterministically without backtracking, a simple lookahead (lookaheading simple words) might not be sufficient. What a parser needs to "lookahead" may be grammatical constituents (e.g. VPs, PPs,... etc) which could only be obtained by "parsing". The parsing suspension mechanisms will be suitable for not only the conjunction and comparative sentences but also for cases where a grammatical constituent lookahead is needed (such as the trace assignment problem mentioned in Cheung [5]). Three types of suspensions are implemented in our WASP:

1. Suspension for scoping ambiguity.
2. Suspension for ellipses before the conjunction words.
3. suspension for pattern formation and for subsequent pattern matching. The first two types of suspensions are discussed in this section and the third type is discussed in the next section.

For parsing conjunctions, ambiguous point might occur in two conjuncted NPs. There are two reasons for suspending the binding of the conjuncted NPs. The first one is that an NP may have two roles in a sentence --- either the subject or the object, but

never both. For example, consider the example:

The pen that I give you and Bob gives Jane costs five dollars. (1)

When a parser encounters *you and Bob*, it does not yet have adequate information to determine the role of the NP *Bob* presumably the parser scans the sentence from left to right. Fig.1 shows the parse tree of this sentence.

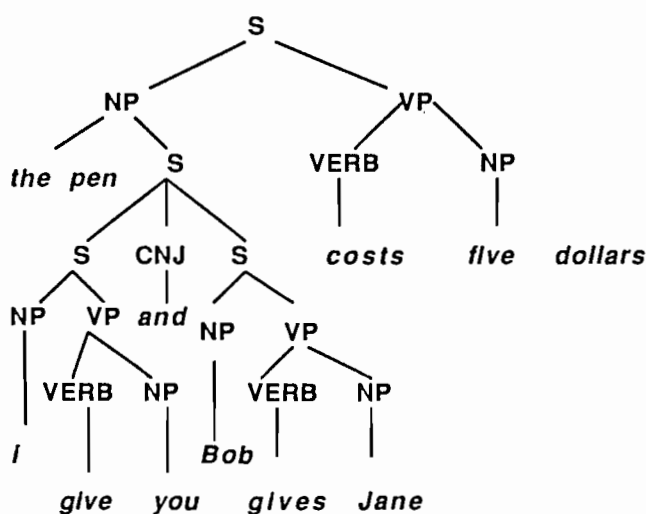


Fig. 1. The parse tree of the sentence:
 "The pen that I give you and Bob gives Jane costs five dollars."

The second reason is that even if the role of an NP is determined, the binding may be still ambiguous. Consider the following examples:

John gives Mary the pen that I give you and Bob gives Jane a pen. (2)

and

John gives Mary the pen that I give you and Bob gives Jane. (3)

Although the NP *Bob* in both sentence is a subject, the presence of the NP *a pen* determines the binding of two conjuncted sentences. In sentence (2) the sentence *Bob gives Jane a pen* should be conjuncted with the major sentence *John gives Mary the pen ...*, while in sentence (3), the sentence *Bob gives Jane* is conjuncted with *I give*

you, and then the whole conjuncted sentence will serve as a clause. The parse tree for sentence (2) and sentence (3) are shown in Fig.2 and Fig.3 respectively.

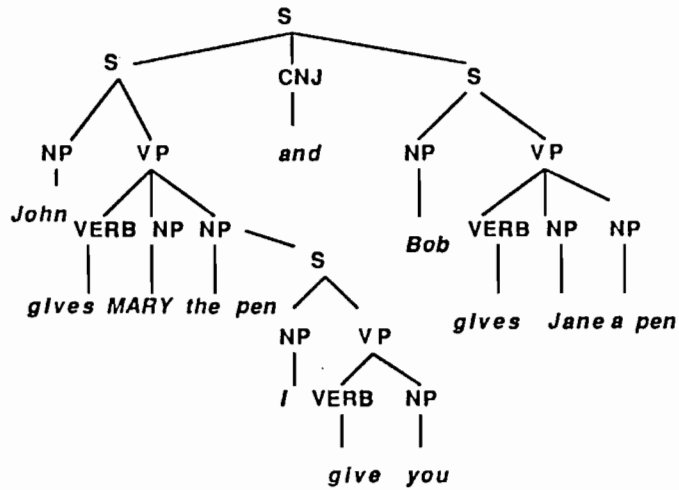


Fig. 2. The parse tree of the sentence:
"John gives Mary the pen that I give you and Bob gives Jane a pen."

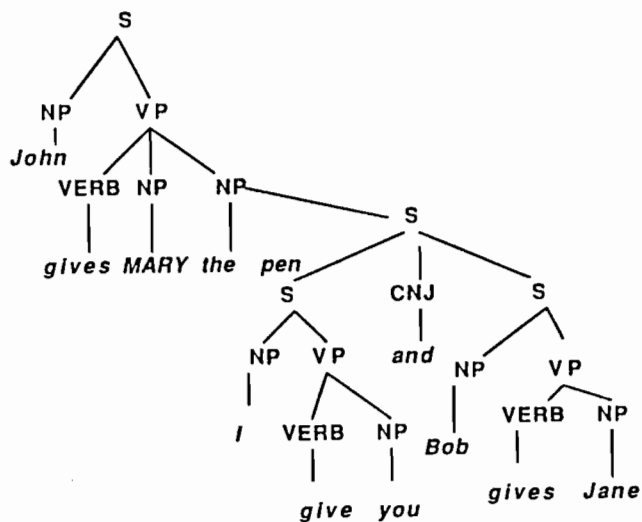
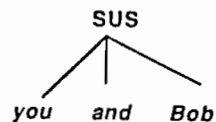


Fig. 3. The parse tree for the sentence:
"John gives Mary the pen that I give you and Bob gives Jane."

Thus, a parser must collect adequate information to determine the roles of these NPs and ways of binding conjuncted grammatical constituents. It might be necessary for a parser to lookahead. However, what it needs to lookahead may be a grammatical constituent (e.g. a VP, S, ...) rather than words. So, it is necessary to suspend the parsing in order to lookahead for a needed grammatical constituent. Consider this example:

John gives Mary the pen that I give you and Bob gives the man who smiles.

When a parser encounters the conjuncted NPs --- *you and Bob*, it is necessary to determine the grammatical role (subject or object) of the NP *Bob* and the way of binding. In order to make a correct decision, it is necessary to collect more information from the input following *Bob*. So our WASP pushes a suspension node (SUS) containing the ambiguous part *you and Bob* onto the node stack:



The parsing will continue from the word immediately following *Bob*, i.e. the verb *gives*. After getting the grammatical constituent (in this case it is a VP) following the suspension node, our WASP may have a clear view about the sentence structure to make a correct binding for NPs in the suspension node. In this case, the NP *Bob* should be a subject of a sentence which is conjuncted with the sentence *I give you*. And this solves the scoping ambiguity problem of the conjunction. A complete parse tree is shown in Fig.4.

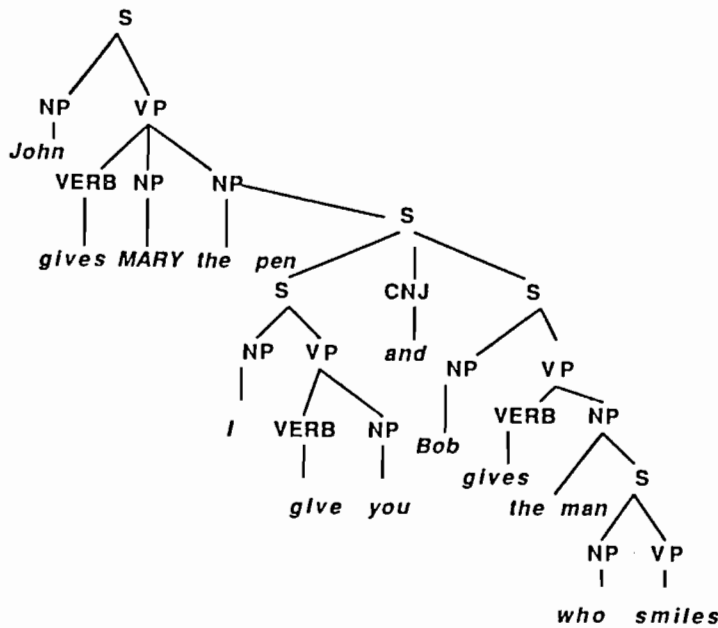


Fig. 4. The parse tree for the sentence:
 "John gives Mary the pen that I give you and Bob
 gives the man who smiles".

The second type of suspension is used to solve the ambiguity problem of the ellipses which occurs before the conjunction word. The missing constituents might be found only when the constituents after the conjunction word have been parsed. Thus a suspension is introduced here. Consider the example:

The man kicked and the woman played the ball.

Since the verb *kicked* is transitive, there must be a missing NP before the conjunction word *and*. The parser suspends this ambiguity here and continues to parse the components after the conjunction word. After parsing the constituent after the conjunction word (in this case, it is an S) the suspension is resumed, and the missed component (in this case, it is the NP *the ball*) can be found and copied.

It should be noted that our parser acts in a one-pass and backtrack-free manner regardless the introduced suspension mechanism. And since there is no work done in vain during parsing, the way of parsing is very efficient.

5. Pattern matching

The "symmetric property" of conjunctions and comparatives is an important feature that can be used to parse these sentences. The symmetric property means that any two conjoined or compared constituents (NPs, PPs, VPs, or S) will have similar syntactical structures. Thus when handling ellipses in these sentences, the syntactical patterns of these two constituents may be compared (matched) to determine the ellipses. This is a basic approach for parsing conjunctions and/or comparatives.

Consider the example:

I ate an apple and John a hotdog.

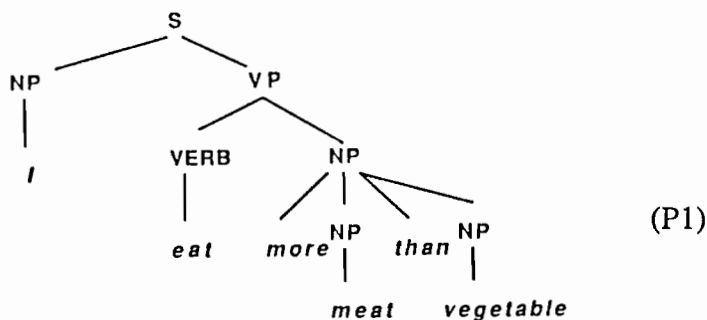
By comparing the syntactical structures before and after *and*, the parser can easily find the ellipses in this sentence, and treat this sentence as:

I ate an apple and John ate a hotdog.

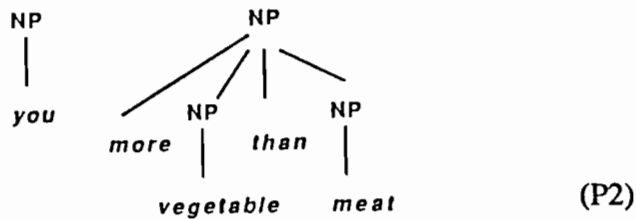
When there is an incomplete syntactical structure (e.g. a VP which is lack of an object, a PP without an NP,... etc.) and a conjunction or a comparative word, the pattern matching is necessary to "fill the gap" of these syntactical structure. For example:

I eat more meat than vegetable and you more vegetable than meat.

When the parsing process proceeds to the conjunction word *and*, a parser has parsed a complete sentence *I eat more meat than vegetable*, and will have the following partial parse tree:



However, the pattern following the conjunction word *and* is an incomplete one:



These two patterns P1 and P2 must be matched and combined to get the whole complete sentence. The parse tree is shown in Fig.5.

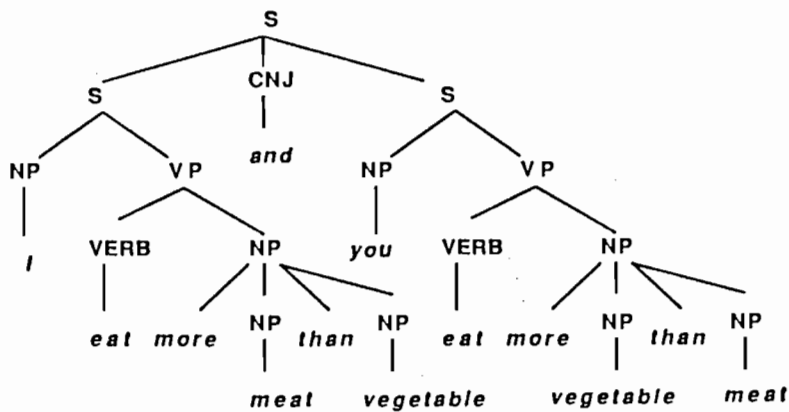


Fig. 5. The parse tree for sentence:
*"I eat more meat than vegetable and you
 more vegetable than meat."*

The question is: *when and how can a parser form the patterns?* It is obvious that only when patterns are parsed, can a WASP perform pattern matching to solve the ellipsis ambiguity problem. This means that the parser should lookahead in a way similar to the suspension action mentioned in the above section. There are three rules

for constructing patterns:

a. If the current node is a suspension node (SUS) and the next input token is a simple NP (directly obtained from preparsing), try to extend the NP to its largest scope, and then attach it to the suspension node.

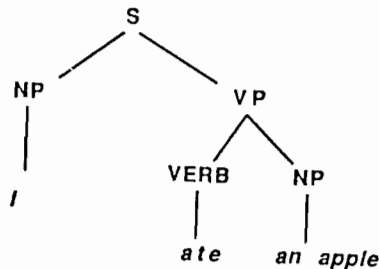
b. If the current node is a suspension node (SUS) and the next input token is a PREP, try to build a complete PP, and then attach it to the suspension node.

c. If the current node is a suspension node (SUS) and the next input token is VERB, the pattern is now formed in the SUS, and the pattern matching is followed.

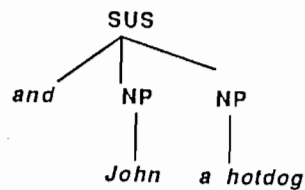
For example, consider the sentence mentioned above:

I ate an apple and John a hotdog.,

the partial parse tree before suspension node is:



and the suspension node is:



Thus, a pattern matching is needed and the verb *ate* is copied. Fig.6 shows the complete parse tree.

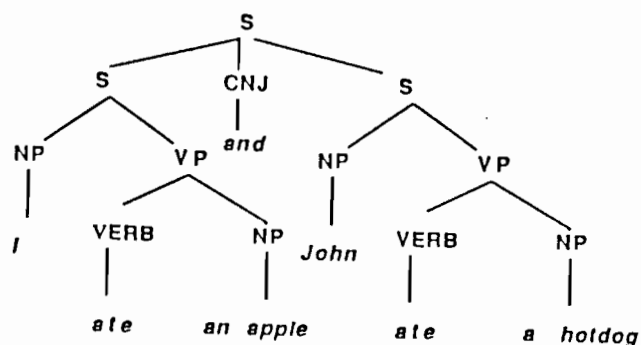


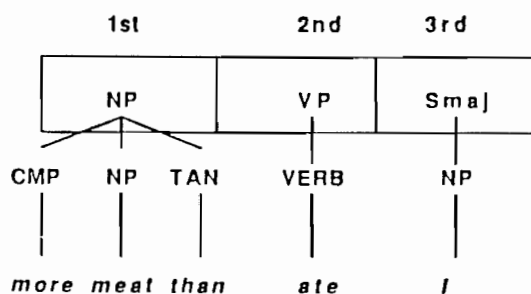
Fig. 6. The parse tree for sentence: "I ate an apple and John ate a hotdog."

Consider a more complex example with both conjunction and comparative words:

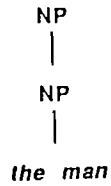
I ate more meat than the man who gave Mary a pen and John a hotdog.

Our WASP will proceed the following steps:

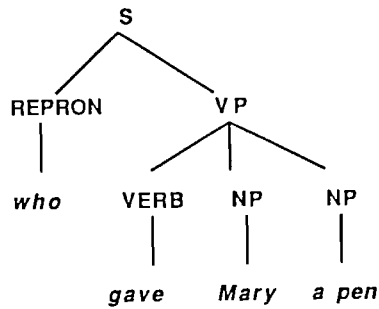
a. When comparative word *more* is encountered, by lookaheading the NP *meat*, our parser concludes that there is a larger NP consisting of comparative words. Thus it tries to build the larger NP. The node stack looks like:



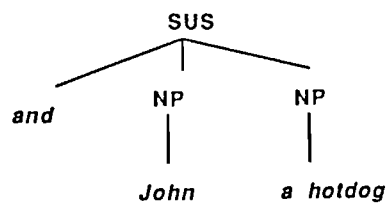
b. When the NP *the man* and the relative pronoun *who* are encountered, Our WASP tries to build a new NP. An NP is pushed, and the top of the node stack is:



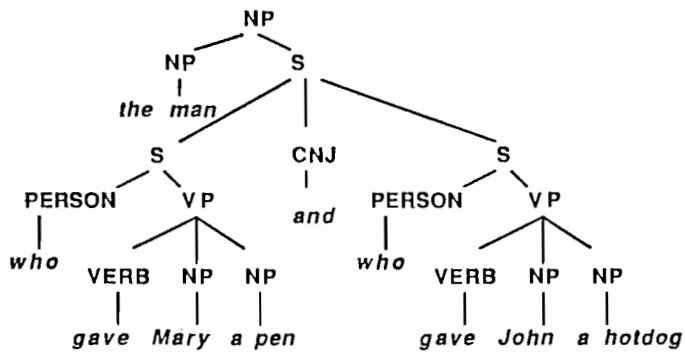
c. When the conjunction word *and* is encountered, the top of the node stack is:



And a suspension node should be constructed as before:



d. Then pattern matching is needed, and the *gave* is copied. And a complete clause is constructed:



e. After the NP *the man who gave ...* is parsed, it can be matched either with *meat* or with *I*. Since *I* and *the man who gave ...* have the same word type --- PERSON, it is better to match these two NPs. Thus, our parser will successfully parse this sentence. The complete parse tree is:

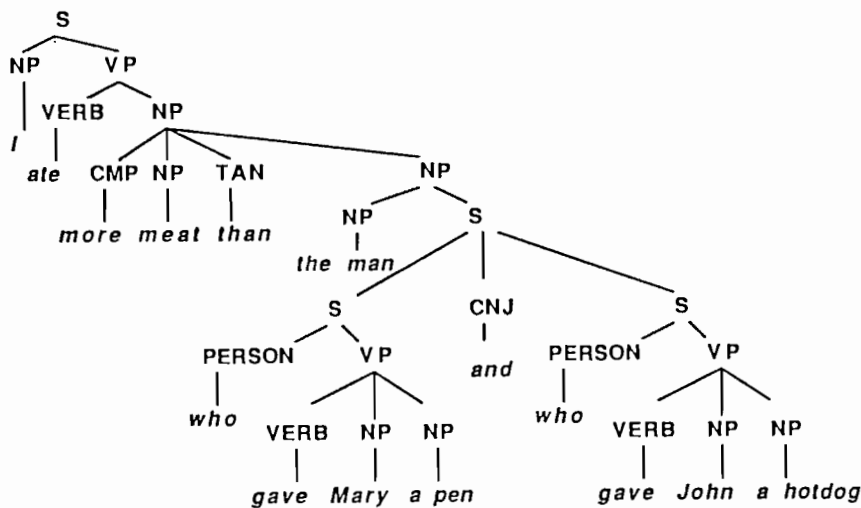


Fig. 7. The parse tree for sentence:
*"I ate more meat than the man who gave Mary a pen
 and John a hotdog."*

6. Implementation

The design of our parser has taken into consideration of the sentences which allow comparatives and conjunctions to appear simultaneously at any grammatical constituents. To pay for this capability, additional rules are needed. However, the effort is relative minute.

Our system is currently implemented in LISP and runs under the GCLISP interpreter system on a PC386. There are currently about 100 rules in the rule packets. In Appendix, we illustrate up to 42 sentences to test different patterns of comparative and conjunction sentences. The run time for each sentence is also recorded. Almost all the sentences can be successfully parsed within 300 msec. However, the ambiguities of the attachment of the prepositional phrases can sometimes cause problems. Most of the cases, we found, require more semantic information than actually assumed in our implementation.

Future extension of our work requires a sound and complete dictionary, a better preparsing mechanism to take care of a variety of idioms. How to incorporate more semantic features into the system to guide correct parsing is also an important direction of our research.

7. Conclusion

For parsing English sentences with comparatives and conjunctions, we are concerned with the efficiency and extensibility of a parser. Therefore, we adopt the Wait-and-See strategy to eliminate the backtracking that is a key factor affecting the efficiency. In addition, we introduce such mechanisms as preparsing, suspension, and pattern matching to further promote the power of the parser. The bottom-up preparsing promotes the efficiency by simplifying the subsequent tasks of parsing; the parsing

suspension allows to collect information for guiding a backtrack-free parsing and resolves the scoping ambiguities; and the pattern matching resolves the ellipsis ambiguities in the conjunctions and comparatives. Since our WASP is designed in a highly modular and uniform manner, its extensibility is high.

8. Acknowledgement

The authors would like to acknowledge valuable comments from anonymous referees.

9. References

1. Whay-Loong Chiang, Jiunn-Jen Chen, and I-peng Lin, *Generalized Augmented Transition Network for English-Chinese Machine Translation System*, ICS, 1988.
2. Mitchell Marcus, *A Computational Account of Some Constraints on Language*, Theoretical Issues in Natural Language Proceeding-2 D. Waltz, ed., 236-246, Urbana-Champaign, Association for Computational Linguistics, 1978.
3. Stan C. Kwasny and Norman K. Sondheimer, *Relaxation Techniques for Parsing Grammatically Ill-formed Input in Natural Language Understanding System*, American Journal of Computational Linguistics, Volume 7, Number 2, 1988.
4. Philip J. Hayes and George V. Mouradian, *Flexible Parsing*, American Journal of Computational Linguistics, Volume 7, Number 4, 1981.
5. B. Cheung and K. P. Chow, *Universal Feature Instantiation Principles And Wait-And-See Strategy*, ICS, 1988.
6. Donald W. Kosy, *Parsing Conjunctions Deterministically*, Proceedings of the 24th ACL Conference, 78-83, 1986.

7. Patrick H. Winston, *Artificial Intelligence*, 2nd edition, Addison-Wesley Publishing Company, 1984.
8. Xiuming Huang, *Dealing With Conjunctions In A Machine Translation Environment*, Proceedings of COLING 84, Stanford, 243-246.
9. Karen Ryan, *Corepresentational Grammar and Parsing English Comparatives*, Proceedings of the 19th ACL conference, Stanford, 13-18, 1981.
10. James Allen, *Natural Language Understanding*, Chapter 6, The Benjamin/Cummings Publishing Company, Inc., 1987.

Appendix: Table of test sentences successfully parsed.

A. SENTENCE WITH CONJUNCTIONS:	Run Time (sec)
Part 1. SENTENCES WITH SCOPING PROBLEMS	
The story that John told Mary and Bob give the man who was crying a hint.	0.27
The story that John told Mary and Bob told you is a good story.	0.27
The story that John told Mary and Bob is a good story.	0.22
Henry repeated the story that John told Mary and Bob told you.	0.27
Henry repeated the story that John told Mary and Bob told John his opinion.	0.28
The pen that I give you and Bob gives Jane costs five dollars.	0.28
John gives Mary the pen that I give you and Bob gives Jane a pen.	0.27
John gives Mary the pen that I give you and Bob gives Jane.	0.28
John gives Mary the pen that I give you and Bob gives the man who smiles.	0.33
I ate meat and vegetable in the store.	0.16
I played a football and John ate the dinner.	0.17
I give the man who gives Mary and Bob a paper a hint.	0.22
Part 2. SENTENCES WITH ELLIPSIS PROBLEMS	
The man kicked and the woman played the ball.	0.22
John drove the car through and completely demolished a window.	0.22
John played tennis and Jack football.	0.16
I give Mary an apple and John a hotdog.	0.17
I ate an apple and John a hotdog.	0.16
I ate and kicked and the man who are crying ate an apple.	0.27
I give Mary an apple and John a hotdog and an apple.	0.16
I give Mary an apple and John a hotdog and an apple is eaten.	0.22
I played the ball in the store and tennis in the school.	0.22
I ate the dinner slowly and Mary quickly.	0.16
The man kicked the child and ate the dinner.	0.16
I played a football and John ate the dinner.	0.17
I gave the pen to Mary and John to Bob.	0.22
Bob gave the pen to Mary in the store and John in the school.	0.27
Bob gave the pen to Mary in the store and John to Bob in the school.	0.28
I gave the pen to Mary and the apple to Bob.	0.22
The man who gave John an apple and Mary a hotdog kicked the ball.	0.22
B. SENTENCES WITH COMPARATIVES:	
John reads more than most students.	0.16
You run faster than I.	0.11
John has learned more words than Jane.	0.16
John eats more meat than vegetable.	0.17
John reads more than most students do.	0.16
Taller people than I gave the apples to Mary.	0.16
John ate more apple than Mary gave him.	0.16
I give the man taller than you an apple.	0.17
C. SENTENCES WITH BOTH CONJUNCTIONS AND COMPARATIVES:	
John and Bob run faster than Mary and Jane.	0.16
I ate more vegetable and fruit than meat and hotdog.	0.16
John reads more than most students who are crying and I.	0.22
John eats more meat than vegetable and Jane more vegetable than meat.	0.27
I ate more meat than the man who gives Mary a pen and John a hotdog.	0.27

A Unification-based Approach to Mandarin Questions

Yu-Ling Shiu* & Chu-Ren Huang**

* BTC R&D Center
2F, No. 28, R&D Road II
Science-Based Industrial Park
Hsinchu, Taiwan, R.O.C.

** Institute of History and Philology
Academia Sinica
Nankang, Taipei, R.O.C.

ABSTRACT

This paper provides unification-based GPSG and LFG analyses of Mandarin questions. First, we briefly introduce four kinds of Mandarin questions, namely, **WH-questions**, **A-not-A questions**, **disjunctive questions**, and **particle questions**. Their different interrogative messages are adequately encoded with different feature-value pairs. Then, the compatibility of these interrogative information in simple sentence is investigated. Both GPSG and LFG can provide straightforward account for their mutual exclusiveness. Finally, the scope of percolation of Mandarin interrogative information is examined. It is suggested that the matrix verb of a complex sentence is responsible for the scope of interrogative information in its complement sentence. According to our observations, Mandarin verbs should be divided into at least three classes. We provide preliminary analyses of this topic. The GPSG analysis relies on the **Foot Feature Principle (FFP)** and the LFG analysis relies on **functional uncertainty**. It is shown that the transmitting of Mandarin interrogative information can also be adequately accounted for in GPSG and LFG.

0. Introduction

In contrast to a purely formal concern of whether a string is generatable by the grammar of a certain language, recently an informational approach to linguistic phenomena presents linguists' renewed perspective of regarding language as a system for encoding and transmitting ideas (see Kay (1986)). This approach requires grammar formalisms representing how language convey information. Such requirement is accomplished by associating strings with their informational domain of well-structured set of feature-value pairs. Grammar formalisms derived from this design choice are capable of encoding various kinds of information, which is especially important in the research community of natural language understanding and generation. Thus, in this paper we attempt to study Mandarin questions from an informational point of view.

Traditionally, Mandarin questions are divided into four main types, namely, **WH-questions**, **A-not-A questions**, **disjunctive questions** and **particle questions**.¹ Unlike English, which always involves Subject-Aux inversion or WH-word fronting in question formation, Mandarin Chinese does not have any characteristic syntactic constructions to mark interrogatives. Except for intonation,² which is beyond our syntactic consideration in this paper, declarative and interrogative counterparts in Mandarin may just differ in the existence of a crucial element, such as a **WH-word**, an **A-NOT-A construction**, a **disjunctive conjunction**, or an **interrogative sentential clitic**. This is illustrated as follows:³

- (1) *Yijing pa lauhu.*
Yijing fear tiger
' Yijing is afraid of tiger. '
- (2) *Shei pa lauhu ?* (WH question)
Who fear tiger
' Who is afraid of tigers ? '
- (3) *Yijing pa-bu-pa lauhu ?* (A-NOT-A question)
Yijing fear-not-fear tiger
' Is Yijing afraid of tigers or not ? '
- (4) *Yijing pa lauhu haishr pa shrtz ?* (Disjunctive question)
Yijing fear tigers or fear lions
' Is Yijing afraid of tigers or afraid of lions ? '

¹ This classification is adopted mainly from Tang (1981), in which tag questions are not regarded as a separate type. Discussions of tag questions can be found in Tang (1981: 20-21) and Li & Tompson (1981: 546).

² It is always possible to turn a Mandarin statement into a question by using a rising intonation.

³ The Romanization system adopted in this paper is Mandarin Phonetic Symbols II (MPS II), which is formally announced by the Ministry of Education R.O.C. in 1986.

(5) *Yijing pa lauhu ma ?*
Yijing fear tiger MA
' Is Yijing afraid of tigers ? '

(Particle question)

Different kinds of interrogative elements may co-occur within a sentence, and their conditions on compatibility and environments of their co-occurrences seem rather intriguing. In addition, different kinds of interrogative elements encode different kinds of interrogative information and have different kinds of semantic implications. Taking the informational approach, we provide a systematic and straightforward solution to this problem and a preliminary study of the encoding and transmitting of Mandarin interrogative information. In particular, the compatibility nature and the scope of percolation of these interrogative information will be carefully investigated. Since the flow of information is much more explicitly formulated in unification-based formalisms, and **Generalized Phrase Structure Grammar (GPSG)** and **Lexical Functional Grammar (LFG)** are two of the linguistically best-established frameworks using this approach, we will adopt them in subsequent discussions.⁴ Accounts in either frameworks are independently motivated. Their mutual compatibility and validity, however, lend support to Shieber's (1986, 87) advocacy of unification as an underlying grammar formalism.

I. The Encoding of Mandarin Interrogative Information

1.1. A GPSG Analysis

As mentioned previously, Mandarin questions are marked solely by the existence of interrogative elements. In GPSG, this phenomena may raise problems on semantic interpretation. Adopting the basic concept of Montague Grammar, syntax and semantics in GPSG are separate but parallel components, in which every syntactic structure is directly paired with a semantic interpretation. Since Mandarin declaratives and interrogatives do not differ in their syntactic structures, their semantic denotations could also be indistinguishable. As a consequence, syntactic specifications which are semantically interpreted have to be introduced to encode different kinds of interrogative information.⁵

A. WH Questions

In Mandarin, WH-questions are formed by simply replacing the elements questioned with appropriate WH-words. Thus, the presence of a WH-word is the sole marker of a WH-question. Since syntactic categories in GPSG are taken to be sets of feature-value pairs and each pair encodes a piece of linguistically significant information, a feature-value pair [QTYPE

⁴ Readers are referred to Sells (1985) for a general overview of the GPSG and LFG frameworks, to Gazdar et al. (1985) for the most complete description of GPSG, and to Bresnan (1982) for a collection of important LFG literatures. In-depth discussion of unification can be found in Shieber (1986), Sag et al. (1986), and works cited therein.

⁵ For more detailed discussion on how syntax and semantics interact in GPSG, please see Gazdar et al. (1985: 182-244).

WH] is hence postulated to encode the interrogative messages of WH-words. Accordingly, the typical WH-word *shei* will be listed in lexicon as shown in (6):

(6) < *shei*, [N +], [V -], [QTYPE WH], ...] ... >

One point worth noting is that the interrogative information is crucially related to sentence type. Thus, although the interrogative specifications are encoded in the lexical entry of WH-words, they must be semantically interpreted at a sentential level. A natural solution to this problem in GPSG is to assign the feature QTYPE to the class of FOOT features. In GPSG, features, according to their percolation properties, are divided into three classes; namely, HEAD features, FOOT features, and LOCAL features. Foot features distributions obey the Foot Feature Principle (FFP) :

(7) FOOT Feature Principle (FFP) :

The FOOT feature specifications that are instantiated on a mother category in a tree must be identical to the unification of the instantiated FOOT feature specifications in all of its daughter categories.

(Gazdar et al. (1985: 82))

The basic operation underlying FFP is unification. Based on such mechanism, specifications will be "passed up" from a phrasal daughter to a mother. Thus, interrogative information in GPSG can be locally specified in lexicon, while be checked and percolated (if unification is successful) unbounded up the tree.

B. A-NOT-A Questions

Traditionally, an A-NOT-A question is considered as the result of identical elements deletion from a full coordinate structure which is formed by an affirmative sentence and its negative counterpart. However, this analysis is not appropriate here because there are no transformations in GPSG at all. An alternative approach is to regard a A-NOT-A question as involving a morphological copying process. Thus, we assume that the whole A-NOT-A construction, after some kind of morphological process, encodes a specification [QTYPE A-NOT-A].

C. Disjunctive Questions

Most linguistic articles analyze A-NOT-A questions on a par with disjunctive questions. Both of them explicitly present the respondent with a choice of some possible answers. But syntactically, disjunctive questions have less restrictions on their conjuncts.⁶ Thus, in GPSG, we must assume the disjunctive conjunction *haishr* independently bears a kind of interrogative information [QTYPE DJ] in its lexicon. The lexical entry of *haishr* is given below:

(8) < *haishr*, [..., [QTYPE DJ], ...] ... >

⁶ The conjuncts of an A-NOT-A question must be an affirmative predicate (or predicate phrase) and its negative counterpart. That is, the number of them is limited to two, and the syntactic category of them must be a predicate. But disjunctive questions do not have such restrictions on their conjuncts.

D. Particle Questions

According to Shiu (1989), *ma* is the most typical interrogative sentential clitic in Mandarin,⁷ and it functions to turn a statement into a yes-no question. So, the lexicon of *ma* is presented in (9):

(9) < *ma*, [[CLIT MA], [QTYPE YN], ...] ... >

1.2. An LFG Analysis

In LFG, since semantic interpretation is derived from the attribute-value matrix representations of f-structures, we also have to properly introduce different feature-value pairs to encode interrogative information. Here, we also assume that the presence of the feature QTYPE marks a sentence as a question and the value of this feature further specifies which kind of question the sentence is. Thus, Mandarin interrogative elements are represented in lexicon as (10):

(10) Lexicon

<i>ma</i>	CLIT	(↑LAST) = + (↑QTYPE) = YN
<i>pa-bu-pa</i>	V	(↑PRED) = 'FEAR<(↑SUBJ)(↑OBJ)>' (↑QTYPE) = A-NOT-A
<i>shei</i>	N	(↑QTYPE) = WH (↑QTYPE) = body bottom (↑PRED) = 'PRO' (↑HUMAN) = +
<i>haishr</i>	CONJ	(↑QTYPE) = DJ (↑QTYPE) = body bottom

Again, this interrogative feature QTYPE should be interpreted at the matrix level in f-structure. But instead of general feature percolation principles as in GPSG, the LFG mechanism of functional equations explicitly specify how the functional information contained in lexicon or on a node in c-structure participates in f-structure. That is, the flow of information in LFG is governed by independent functional equations. The lexical entry of *ma* has been discussed in Shiu (1989). The treatment of A-NOT-A construction is similar to that of GPSG. We assume the whole A-NOT-A construction is the output of a morphological process and encodes an equation '(↑QTYPE)= A-NOT-A'. The WH word *shei* and the disjunctive conjunction *haishr* encode an equation (↑QTYPE)=WH and (↑QTYPE)=DJ respectively.

⁷ Zwicky (1985) has investigated the grammatical status of clitics and particles. It is suggested that 'clitic' is a theoretical construct which belongs to a level between 'word' and 'affix', while 'particle' is a redundant cover term which should be eliminated. Following this line of approach, Huang (1985) explicitly points out that Mandarin sentential particles are indeed sentential clitics.

The equation '(↑QTYPE)=body bottom' encoded on both of them indicates a device of **functional uncertainty** (proposed in Kaplan & Zaenen (in press)), which will be discussed in detail in section IV.

Given the above GPSG and LFG analyses, every kind of Mandarin interrogative information can be adequately encoded and appropriately interpreted. These analyses will be further supported in the next two sections.

II. The Compatibility Nature of Mandarin Interrogative Information

In this section, we will briefly discuss how the interrogative information in Mandarin interacts within simple sentences. Let us consider the following sentences:

- *(11) *Shei pa-bu-pa lauhu ?*
(WH word & A-NOT-A construction)

- *(12) *Shei pa lauhu haishr pa shrtz ?*
(WH word & disjunctive conjunction)

- *(13) *Yijing pa-bu-pa lauhu haishr shrtz ?*
(A-NOT-A construction & disjunctive conjunction)

- *(14) *Shei pa lauhu ma ?*
(WH word & sentential clitic *ma*)

- *(15) *Yijing pa-bu-pa lauhu ma ?*
(A-NOT-A construction & sentential clitic *ma*)

- *(16) *Yijing pa lauhu haishr shrtz ma ?*
(disjunctive conjunction & sentential clitic *ma*)

From the above sentences, we can conclude that different kinds of interrogative elements cannot co-occur within simple sentences. Based on the analyses proposed in the previous section, we will provide adequate and straightforward accounts for this phenomenon.

2.1. A GPSG Analysis

Notice that syntactic categories in GPSG are partial functions from features to values. Defining categories this way has a natural consequence that no well-formed syntactic category may have different specifications for the same feature. Thus, the mutual exclusiveness of different kinds of interrogative information can be accounted for in GPSG by assuming each kind of interrogative element encodes one kind of specification of the feature QTYPE. Summarizing our encoding of Mandarin interrogative information in GPSG, the feature QTYPE and the set of its possible values are indicated below:

(17) feature value range feature class

QTYPE { YN, WH, DJ, A-NOT-A } **FOOT**

According to this analysis, the grammaticality of (11)-(16) can be nicely captured by FFP and unification. Owing to FFP, different kinds of interrogative specifications in a sentence will all percolate up to the matrix node and result in feature clash. Thus, all these sentences are ruled out as ungrammatical because of failure of unification.

2.2. An LFG Analysis

Taking a similar approach to GPSG, we attribute all kinds of interrogative information to the feature QTYPE. The encoding of this feature in different kinds of questions is summarized below:

(18) **Lexicon**

<i>ma</i>	CLIT	(↑QTYPE)= YN
<i>pa-bu-pa</i>	V	(↑QTYPE)= A-NOT-A
<i>shei</i>	N	(↑QTYPE)= WH (↑QTYPE)= body bottom
<i>haishr</i>	CONJ	(↑QTYPE)= DJ (↑QTYPE)= body bottom

So, the LFG account of the grammaticality of (11)-(16) is similar to that of GPSG in that they both resort to unification.

Thus, it is suggested that the seemingly complicated phenomena of the compatibility of Mandarin interrogative information can be straightforwardly accounted for with our analyses in unification-based formalisms.

III. The Scope of Percolation of Mandarin Interrogative Information

With an understanding of the compatibility of Mandarin interrogative information, we will further examine their behaviors within Mandarin complex sentences.

Consider the pair of contrasting sentences below:

(19) *Tamen shiwang [shei pa lauhu] ?*
they hope who fear tigers
' Who do they hope is afraid of tigers ? '

(20) *Tamen tauluen [shei pa lauhu].*
they discuss who fear tigers
' They discuss the topic that who is afraid of tigers. '

Although both sentences contain a WH-word *shei* 'who', yet (19) must be interpreted as a direct question, and (20) must be interpreted as a statement taking an indirect question. The difference between (19) and (20) reveals an interesting phenomenon concerning the scope of percolation of Mandarin interrogative information. Again, we will discuss this topic within the GPSG and LFG frameworks.

3.1. A GPSG Analysis

As mentioned previously, the FFP in GPSG requires that all the FOOT feature specifications instantiated on a daughter be instantiated on its mother in any given local tree. Since our proposed interrogative features are all FOOT features, without additional stipulations, the interrogative messages should be passed to the top matrix node, rather than be limited in the embedded clause. But this prediction is contradictory to the empirical fact shown in (20).

According to Grimshaw (1979), it is suggested that the matrix verb of a sentence is responsible for the scope of interrogative information in its complement sentence.⁸ Different kinds of verbs will result in different kinds of percolation of information. This idea has been widely adopted among researchers on interrogatives. Here, we will following this line of approach and make a crucial use of the feature SUBCAT in our GPSG analysis.⁹ In this section, we just take verbs *shiwang* 'hope', *tauluen* 'discuss' and *irdau* 'know' as illustrative samples. Three ID rules are postulated as shown in (21):¹⁰

(21) a. VP → V[11], S – [QTYPE A-NOT-A]

b. VP → V[12], S[QTYPE]

c. VP → V[13], S([QTYPE])

First, let us discuss the verb *shiwang*. We assume it is listed in lexicon as (22):

(22) < *shiwang*, [N –], [V +], [SUBCAT 11], ...] HOPE' >

Consider the following sentences:

(23) *Tamen shiwang [Yijing pa lauhu].*

They hope Yijing fear tigers

' They hope that Yijing is afraid of tigers. '

*(24) *Tamen shiwang [Yijing pa lauhu ma] ?*

[QTYPE YN]

⁸ For ease of description, we use the term 'verbs' to stand for predicates in Mandarin.

⁹ The use of the feature SUBCAT is an important mechanism in GPSG whereby the relevant subclasses of a preterminal symbol can be matched with the ID rules that introduce it.

¹⁰ V[11] is just an abbreviation for V[SUBCAT 11].

(25) *Tamen shiwang [Yijing pa lauhu] ma ?*
[QTYPE YN]

(26) *Tamen shiwang [shei pa lauhu] ?*
[QTYPE WH]

(27) *Tamen shiwang [Yijing pa lauhu haishr pa shrtz] ?*
[QTYPE DJ]

*(28) *Tamen shiwang [Yijing pa-bu-pa lauhu] ?*
[QTYPE A-NOT-A]

*(29) *Shei shiwang [Yijing pa lauhu] ma ?*
[QTYPE WH] [QTYPE MA]

*(30) *Shei shiwang [Yijing pa-bu-pa lauhu] ?*
[QTYPE WH] [QTYPE A-NOT-A]

*(31) *Shei shiwang [Yijing pa lauhu haishr pa shrtz] ?*
[QTYPE WH] [QTYPE DJ]

(23) shows that *shiwang* can take a statement as its complement. The contrasting pair (24) and (25) show that the interrogative sentential clitic *ma* can only attach to a matrix sentence instead of an embedded sentence. This phenomenon has been discussed and accounted for in Shiu (1989: 33-41).¹¹ With the GPSG analyses proposed in Shiu (1989), *ma* will always function to form a direct question, and the specification [QTYPE YN] will be always interpreted at the level of matrix sentence. (26) and (27) show that although the [QTYPE WH] and [QTYPE DJ] specifications are introduced in the embedded sentences, they will percolate up to the matrix sentences by FFP, and make the whole sentences interpreted as direct questions. However, it is shown in (28) that [QTYPE A-NOT-A] cannot appear in the complement of *shiwang*. This fact can be nicely captured because *shiwang* is introduced by ID rule (20)a, in which the specification (– [QTYPE A-NOT-A]) is explicitly stipulated, and thus complements containing [QTYPE A-NOT-A] will be ruled out because of feature clash. (28) shows that if the matrix sentence has encoded one kind of interrogative message, the attachment of *ma* will cause unification of incompatible information and thus (29) is ungrammatical. Finally, in (30)-(31), both the matrix sentences and embedded sentences bear some kind of interrogative information. In these cases, except [QTYPE A-NOT-A],

¹¹ The GPSG analyses of *ma* proposed in Shiu (1989) are summarized below:

(i) lexicon

< *ma*, [[CLIT MA], [+ LAST], [QTYPE YN],...] >

(ii) ID rule

S' → S, [CLIT α]*

(iii) LP statement

x < [+ LAST]

other interrogative specifications encoded in embedded sentences will percolate up to matrix sentences and merge with the ones encoded in matrix sentences. Since a feature can have only one value, the grammaticalities of (30)-(31) will also be nicely accounted for.

Next, consider the verb *tauluen*. We assume this verb is listed in lexicon as (32):

(32) < *tauluen*, [N -], [V +], [SUBCAT 12], ...] DISCUSS' >

Let us consider the following sentences:

*(33) *Tamen (tzai) tauluen [Yijing pa lauhu]*.
They (be Ving) discuss Yijing fear tigers

*(34) *Tamen (tzai) tauluen [Yijing pa lauhu] ma ?*
[QTYPE YN]

(35) *Tamen (tzai) tauluen [shei pa lauhu]*.
[QTYPE WH]

(36) *Tamen (tzai) tauluen [Yijing pa lauhu haishr pa shrtz]*.
[QTYPE DJ]

(37) *Tamen (tzai) tauluen [Yijing pa-bu-pa lauhu]*.
[QTYPE A-NOT-A]

(38) *Shei (tzai) tauluen [Yijing pa-bu-pa lauhu] ?*
[QTYPE WH] [QTYPE A-NOT-A]

(39) *Shei (tzai) tauluen [Yijing pa lauhu haishr pa shrtz]*.
[QTYPE WH] [QTYPE DJ]

It is worth noting that *tauluen* obligatorily takes a question as its complement, as exemplified in (33)-(37). This can be achieved by the SUBCAT feature of *tauluen* and the ID rule in (21)b. As indicated earlier, the percolation of FOOT features in GPSG is manipulated by the FFP. But notice that the FFP governs only instantiated FOOT feature specifications.¹² Since the FOOT feature QTYPE in ID rule (21)b is inherited rather than instantiated, its behavior is not regulated by the FFP. As a consequence, all the QTYPE specifications encoded in embedded sentences will not be passed up to matrix sentences but rather be terminated within the embedded sentences. Thus, (35)-(37) are interpreted as indirect questions instead of direct questions. Further, (38)-(39) are not counterexamples to the proposals in previous section because no interrogative information will flow up from the embedded sentences and incompatible specifications do not co-occur in any categories in matrix sentences.

Last, let's turn to the verb *jrdau*. Its lexicon is shown in (40).

(40) < *jrdau*, [N -], [V +], [SUBCAT 13], ...] KNOW' >

¹² Readers are referred to (7) for the definition of the FFP.

We need to account for the following sentences with *jrdau*:

(41) *Tamen jrdau [Yijing pa lauhu].*
 They know Yijing fear tigers
 ' They know that Yijing is afraid of tigers. "

(42) *Tamen jrdau [shei pa lauhu].*
 [QTYPE WH]

(43) *Tamen jrdau [Yijing pa lauhu haishr pa shrtz].*
 [QTYPE DJ]

(44) *Tamen jrdau [Yijing pa-bu-pa lauhu].*
 [QTYPE A-NOT-A]

(45) *Shei jrdau [Yijing pa-bu-pa lauhu] ?*
 [QTYPE WH] [QTYPE A-NOT-A]

(46) *Shei jrdau [Yijing pa lauhu haishr pa shrtz]?*
 [QTYPE WH] [QTYPE DJ]

The verb *jrdau* can take either a statement or an indirect question as its complement. Thus we introduce it by (21)c, in which an optional QTYPE feature is specified. When *jrdau* takes a statement as its complement, the feature QTYPE is absent, but when it takes a question as its complement, the feature QTYPE is present. Thus, the grammaticality of (42)-(46) is accounted for in a way as we just discussed with *taulu*.

Generally speaking, all the Mandarin verbs can be divided into these three classes, therefore the scope of percolation of Mandarin interrogative information is successfully accounted for in GPSG.

3.2. An LFG Analysis

Recall the LFG treatment of interrogative markers in previous section. We repeat the lexicon of these interrogative markers in (47):

(47) **Lexicon**

<i>ma</i>	CLIT	(↑LAST) = + (↑QTYPE) = YN
<i>pa-bu-pa</i>	V	(↑PRED) = 'FEAR<(↑SUBJ)(↑OBJ)>' (↑QTYPE) = A-NOT-A

<i>shei</i>	N	(↑QTYPE) = WH (↑QTYPE) = body bottom (↑PRED) = 'PRO' (↑HUMAN) = +
<i>haishr</i>	CONJ	(↑QTYPE) = DJ (↑QTYPE) = body bottom

Notice that both *shei* and *haishr* lexically encode an equation '(↑QTYPE)=body bottom'. This equation indicates a functional uncertainty device which is recently developed in LFG. The mechanism of functional uncertainty, explicated in Kaplan & Zaenen (in press), is originally proposed to account for long-distance dependencies in natural languages, such as topicalization and English WH questions. The basic idea of this mechanism is that long-distance dependencies are in fact functionally conditioned, and this kind of relation should be captured by a direct link between functions rather than through the mediation of local dependencies.¹³

The general rule of functional dependencies is formally expressed in Kaplan & Zaenen (in press), as shown in (48):

$$(48) \text{ S}' \text{ — } \Omega \qquad \Sigma$$

$$\qquad (\uparrow \text{DF}) = \downarrow \qquad \uparrow = \downarrow$$

$$\qquad (\uparrow \text{DF}) = (\uparrow \text{body bottom})$$

[where Ω is a maximal phrasal category, Σ is some sentential category, DF is taken from the set of discourse functions (TOPIC, FOCUS, etc.), and body must be a regular expression.¹⁴]

The equation $(\uparrow \text{DF}) = (\uparrow \text{body bottom})$ in (48) is a functional uncertainty path in which any language can impose its own specific conditions on the functions of the body and the bottom only if the body is a regular expression.

This approach to long-distance dependencies is well supported by the study of Icelandic, English, and Japanese data. Huang et al. (1989), based on Mandarin topicalization and relative clauses, also suggests that functional uncertainty can provide an elegant solution to long-distance dependencies in Mandarin. In this paper, we use a reverse kind of functional uncertainty in resolving the percolation of interrogative information.

Mandarin interrogatives in fact do not involve overt long-distance dependencies. Unlike English WH questions, no gap-filler pairs can be found in any type of Mandarin questions. But as pointed out earlier, in some cases the existence of an interrogative element will turn the whole sentence into a direct question regardless of how deeply embedded the bearer of interrogation is. Thus, some bears of interrogation should be able to link to a f-structure

¹³ The COMP to COMP movements in Transformational Grammar (TG) and the SLASH feature in GPSG are devices which try to account for long-distance dependencies through the mediation of local dependencies.

¹⁴ A regular expression involves only the use of the Kleene closure operator, designated by '*', or the positive Kleene closure operator, designated by '+', on sets.

many layers up and theoretically there is no limit to the distance of such linking. In LFG, functional uncertainty is the mechanism to capture this kind of unbounded relation. But notice that there are two basic differences between the ordinary long-distance dependencies, such as topicalization, and the dependencies discussed in this section. First, as we have pointed out, Mandarin questions do not involve the so-called gap-filler relations, thus the functional uncertainty equations for them are not to specify the associations between the gap functions and the filler functions, but to ensure the interrogative feature QTYPE to be interpreted at the right places at f-structure. Second, Mandarin questions are characterized by the existence of bears of interrogation, but these interrogative elements do not occupy a specific position at surface structure, such as the sentence initial clause-external position for topic, therefore it is not appropriate to encode the functional uncertainty equations at c-structure rules such as (48). On the contrary, intuitively the functional uncertainty equations for Mandarin questions should be encoded in the lexicon of interrogative markers. Since the interrogative sentential clitic *ma* never occur in embedded sentences, no functional uncertainty path should be posed on it. As for the A-NOT-A construction, it is observed that its interrogative information never percolates to higher sentences, so no functional uncertainty path on this construction is necessary.¹⁵ However, WH questions and disjunctive questions are not interpreted wholly locally. For example, consider the following sentences:

(49) *Dashiung jiuede tamen shiwang shei pa lauhu ?*

Dashiung feel they hope who fear tigers

' Who does Dashiung feel that they hope is afraid of tigers ? '

(50) *Dashiung jiuede tamen shiwang Yijing pa lauhu haishr pa shrtz ?*

Dashiung feel they hope Yijing fear tigers or fear lions

' Does Dashiung feel that they hope Yijing is afraid of tigers or is afraid of lions ? '

Though the WH word *shei* and the disjunctive conjunction *haishr* are encoded in embedded sentences, they turn the whole matrix sentences into direct questions. This phenomenon prompts us to propose a reverse kind of functional uncertainty equations which are encoded in the lexicon of WH words and *haishr* and can characterize the unbounded upward association between interrogative specifications. The general form of such equations is given in (51):

(51) (\uparrow QTYPE) = (body bottom)

According to our observation, the **bottom** of the uncertainty path is the feature QTYPE, and the **body** of the path is a regular expression of the metavariable ' \uparrow '. The metavariable ' \uparrow ' refers to the grammatical function represented by the mother node. Since the grammatical functions in LFG form a finite set, the **body** defined in this way is still a regular set.

¹⁵ An apparent exception concerns a particular set of verbs, such as *tsai*'guess', and *shiang*'think', etc. Tang (1981,1983) call them "the semantically bleached verbs". These verbs cannot form A-NOT-A constructions, but if their complement sentences containing A-NOT-A constructions, the whole sentences are interpreted as direct questions. However, this type of verbs exhibit several other syntactic idiosyncrasies, such as their non-co-occurrence with aspect markers, their inability of constructing condensed answers by itself, etc. Since properties of this kind of verbs are not clear to us at this moment, the analyses of them are left open in this paper.

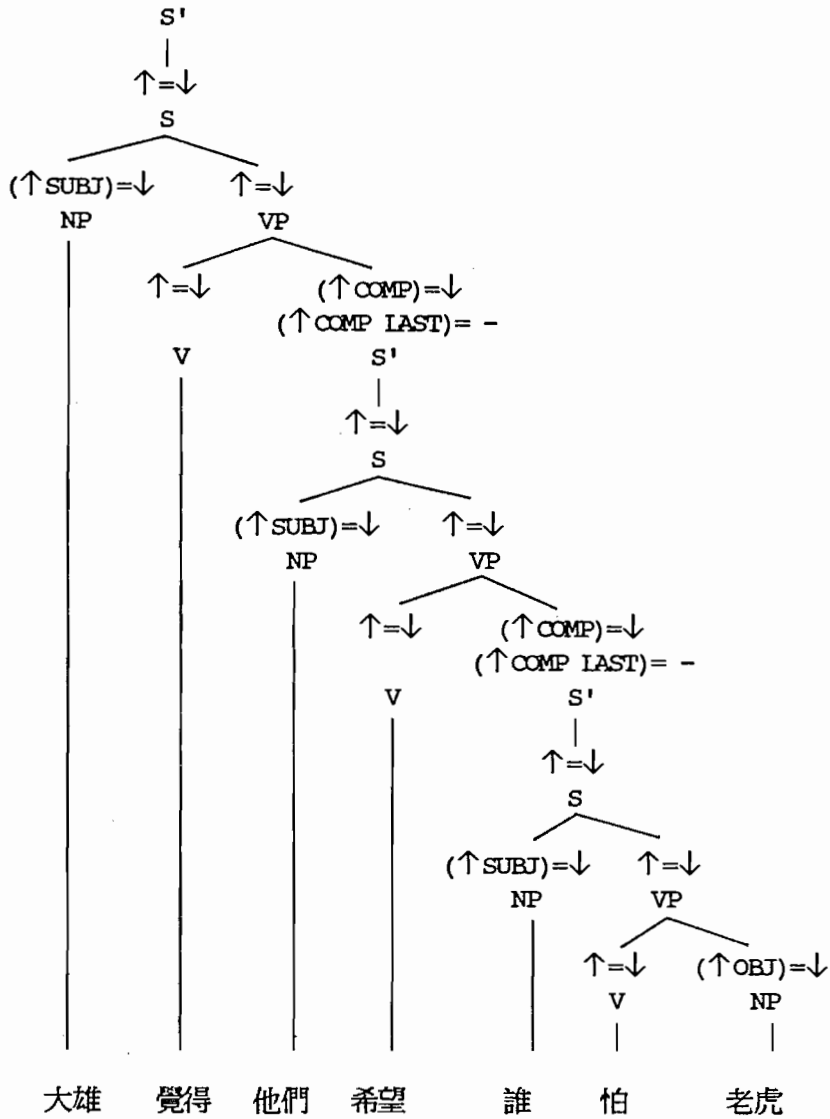
No particular difficulty will arise in solving the verification problem and the satisfiability problem of this kind of functional uncertainty.¹⁶ Thus, the unbounded nature of Mandarin WH questions and disjunctive questions can be specified by the uncertainty equation given in (52):

$$(52) (\uparrow QTYPE) = (\{ \uparrow \} * QTYPE)$$

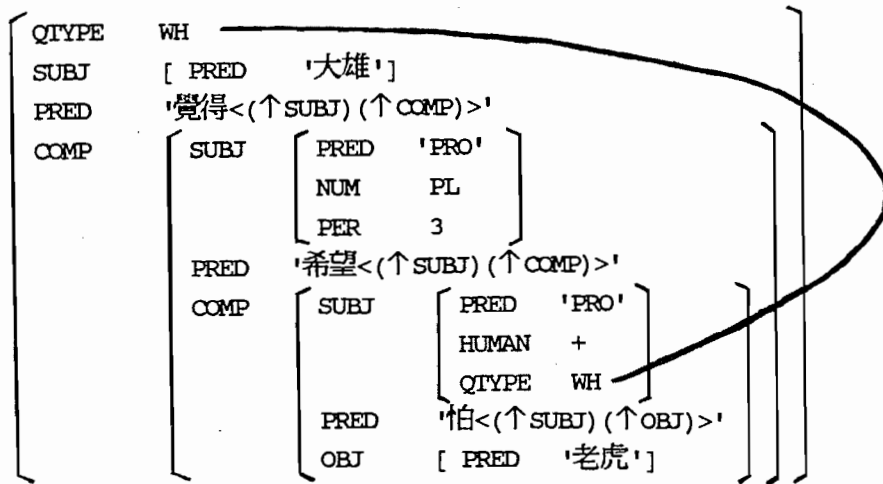
Under this approach of Mandarin interrogative information, the WH question in (49) and the disjunctive question in (50) will have correct c-structure and f-structure pairs as shown in (53) and (54) respectively.

¹⁶ An efficient algorithm for the verification and the satisfiability of functional uncertainty is proposed in Kaplan & Maxwell (1988 a).

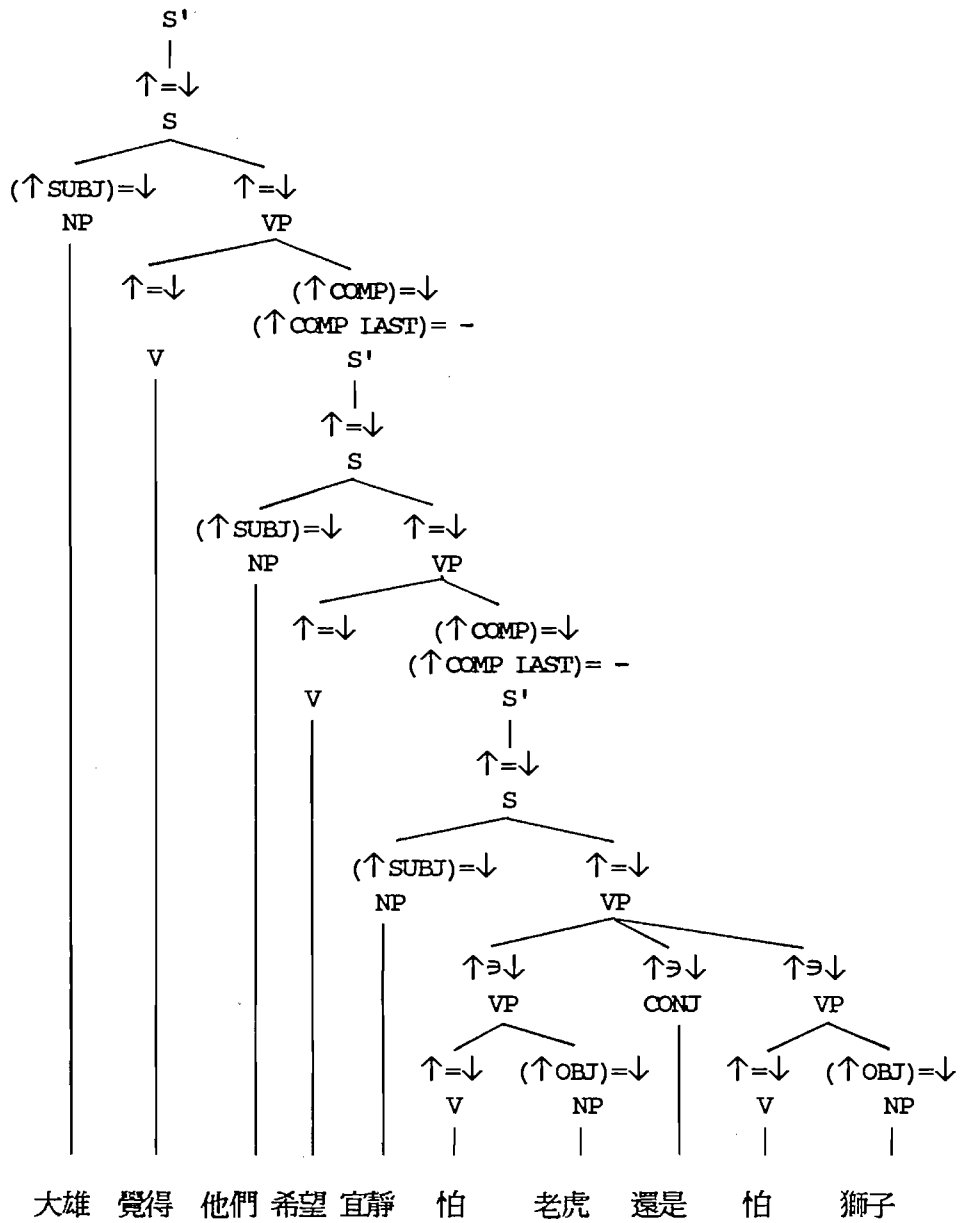
(53) (for (49))
a. c-structure



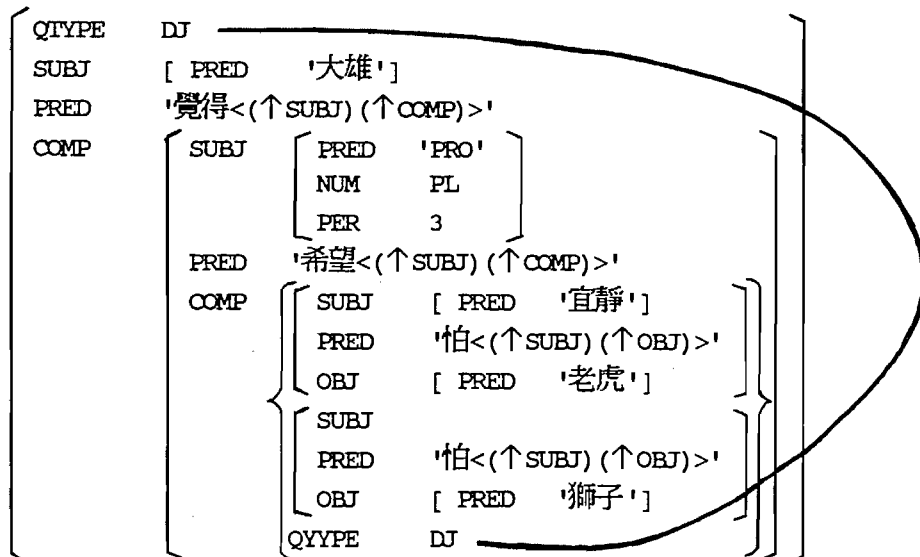
b. f-structure



(54) (for (50))
 a. c-structure



b. f-structure



However, the LFG analysis of Mandarin interrogative information above might appear to be still too general. As mentioned earlier, Mandarin verbs may impose their specific requirements on the sentence types of their complement sentences. Thus, the unbounded linking we proposed in (52) should be subject to the conditions encoded on verbs. Based on the same data as presented in section II A, we assume the verbs *shiwang*, *tauluen*, and *jrdau* encode different kinds of constraints as shown below:

(55) *shiwang* – (| COMP QTYPE)

(56) *tauluen* (| COMP QTYPE)

(57) *jrdau* ((| COMP QTYPE))

(55) states that the feature QTYPE cannot be present at the complement function of the verb *shiwang*. Thus, the interrogative specifications encoded in embedded sentences must be linked to higher f-structures. On the other hand, the verb *tauluen* encodes an existential constraint which will ensure the presence of the feature QTYPE at the function of its complement sentence. Thus, *tauluen* must take a question as its complement, and this question is an indirect question because the feature QTYPE is just interpreted at embedded level. As for the verb *jrdau*, it can take either a statement or an indirect question as complement. Hence, an optional constraint is imposed on it.

In conclusion, we have successfully shown that the scope of interrogative information can also be adequately managed in LFG.

IV. Conclusion

This paper investigates the interrogative information of Mandarin questions. It is suggested that the compatibility and the scope of percolation of different kinds of interrogative information can be adequately and straightforwardly accounted for in GPSG and LFG. The GPSG analysis relies on the Foot Feature Principle (FFP) and the LFG analysis on functional uncertainty. However, from the comparative study we presented in this paper, readers may have noticed that the analyses in GPSG and LFG are quite similar. One important reason for their similarity is that they are both unification-based formalisms. They agree with each other in taking feature-value pairs as their basic linguistic objects and in adopting unification as their basic operation. Owing to their similarity, we are able to extract and compare the main concepts in them. Further, it is also easier to adopt ideas from the other theories to solve problems in their own. These merits of unification-based grammar formalisms have led many researchers to adopt this approach in their theoretical models as well as in their computational implementations.¹⁷ Owing to the brevity of this paper, we just provide a preliminary unification-based study for Mandarin questions, but promising results on this topic can be expected along this line of research.

¹⁷ Unification-based formalisms of theory type consist of LFG, GPSG, HPSG (Head-driven Phrase Structure Grammar), etc. and of tool type consist of PATR-II, FUG (Functional Unification Grammar), and DCG (Definite-Clause Grammar), etc.

References

- Alleton, Viviane (1981). "Final Particles and Expression of Modality in Mandarin Chinese." Journal of Chinese Linguistics 9, 91-115.
- Bresnan, Joan (1982) Ed. The Mental Representation of Grammatical Relations. Cambridge : MIT Press.
- Chao, Yuen-Ren (1968). A Grammar of Spoken Chinese. Berkely : University of California Press.
- Chinese Knowledge Information Processing (CKIP, 中文詞知識庫計劃) (1988). 國語的詞類分析(修訂版). 中研院計算中心: 台北 南港.
- Chu, Chauncy C. (1983). "Ambiguities in Mandarin Verb Phrases — Cases with *Le* and *Shi-De*." Presented at Linguistic Conference on East Asian Languages. Univ. of Southern California.
- (1987). "Semantics and Pragmatics of Modality in Mandarin." Manuscript. Univ. of Florida.
- Falk, Yehuda N. (1983). "Constituency, Word Order, and Phrase Structure Rules." Linguistic Analysis 11, 331-360.
- Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag (1985). Generalized Phrase Structure Grammar. Oxford : Blackwell.
- Grimshaw, Jane (1979). "Complement Selection and Lexicon." Linguistic Inquiry 10, 279-326.
- Halvorsen, Per-Kristian (1982). "Lexical-Functional Grammar and Order-Free Semantic Composition." COLING 82, 115-120.
- Huang, Cheng-Teh James (1982 a). "Move WH in a Language without WH Movement." Linguistic Review 1, 41-80.
- (1982 b). Logical Relations in Chinese and the Theory of Grammar. MIT. dissertation.
- Huang, Chu-Ren (1985). "Chinese Sentential Particles : A Study of Cliticization." LSA Annual Meeting, Dec. 1985.
- (1987). Mandarin Chinese NP de : A Comparative Study of Current Grammatical Theories. Cornell Univ. Dissertation. Published as Special Publications No. 93. Taipei : Institute of History and Philology, Academia Sinica.
- (1988 a). "A Unification-Based LFG Analysis of Lexical Discontinuity." The Fourth International Workshop on East Asia Languages and Linguistics, Paris, June 1988.
- (黃居仁) (1988 b). " 聯併(Unification): 語法理論與剖析." Proceedings of ROCLING I, 29-54. Academia Sinica.
- (1988 c). " 再析國語「領屬主語」結構 — 概化詞組結構語法(GPSG)及詞彙功能語法(LFG)之比較研究." 漢學研究, 第六卷 第二期, 109-134.
- (1988 d). "Towards a Morphosyntactic Account of Taiwanese Question Particle *Kam*." To appear in 史語所集刊 第五十九本.
- , Keh-Jann Chen, Wan-Pei Chen, & Tzu-Ying Hu (1989). "Resolution of Long-Distance Dependencies in Mandarin Chinese with an Algorithm Based on Functional Uncertainty." To appear in Proceedings of 1989 International Conference on Computer Processing of Chinese and Oriental Languages.

- Kaplan, Ronald M. & John T. Maxwell (1988 a). "An Algorithm for Functional Uncertainty." COLING 88, 297-302.
- , & John T. Maxwell (1988 b). "Constituent Coordination in Lexical-Functional Grammar." COLING 88, 303-305.
- , & Annie Zaenen (in press). "Long-Distance Dependencies, Constituent Structure, and Functional Uncertainty." To appear in Alternative Conceptions of Phrase Structure. Eds. M. Baltin and A. Kroch. Chicago : University of Chicago Press.
- Kay, Martin (1985). "Parsing Functional Unification Grammar." In Natural Language Parsing : Psychological, Computational, and Theoretical Perspectives. Eds. David Dowty, Lauri Karttunen, and Arnold M. Zwicky. 251-278. Cambridge Univ. Press.
- Klavans, Judith (1982). Some Problems in a Theory of Clitics. Dissertation. Univ. College London. [Reproduced by IULC]
- (1985). "The Independence of Syntax and Phonology in Cliticization." Language 61, 95-120.
- Li, Charles N. and Sandra A. Thompson (1981). Mandarin Chinese : A Functional Reference Grammar. Berkeley : Univ. of California Press.
- Lu, Jian-Ming (陸儉明) (1982). "由'非疑問形式+呢'造成的疑問句." 中國語文 6, 435-438.
- (1984). "關於現代漢語裡的疑問語氣詞." 中國語文 5, 331-337.
- Nevis, Joel A. (1985). "A Syntactic Account of Second Position Clitics." In Proceedings of the Second Eastern States Conference on Linguistics. Eds. S. Choi, D. Danvitt, W. Janis, T. Mcloy, and Z.-S. Zhang. 186-197. Columbus, Ohio State Univ.
- Sag, Ivan A., Gerald Gazdar, Thomas Wasow, and Steven Weisler (1984). "Coordination and How to Distinguish Categories." CSLI Report No. CSLI-84-4. Also in Natural Language and Linguistic Theory 3.
- , Ronald Kaplan, Lauri Karttunen, Martin Kay, Carl Pollard, Stuart Shieber, and Annie Zaenen (1986). "Unification and Grammatical Theory." Proceedings of the Fifth West Coast Conference on Formal Linguistics. Stanford : Stanford Linguistic Association.
- Sells, Peter (1985). Lectures on Contemporary Syntactic Theories. CSLI Lecture Notes, No. 3, Stanford : CSLI, Stanford Univ.
- Sheu, Ying-Yu (1987). "Chinese Morphosyntax." Paper presented at the Twentieth International Conference on Sino-Tibetan Languages and Linguistics. Vancouver, Canada.
- Shieber, Stuart M. (1986). An Introduction to Unification-based Approaches to Grammar. CSLI Lecture Notes, No. 4, Stanford : CSLI, Stanford Univ.
- (1987). "Separating Linguistic Analyses from Linguistic Theories." In Linguistic Theory and Computer Applications. Eds. P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, & P. Bennett, 1-36. New York : Academic Press.
- Shiu, Yu-Ling Una (1989). "Mandarin Sentential Particles and Particle Questions." Unpublished thesis. National Tsing Hua Univ.
- & Chu-Ren Huang (1988). "Unification-based Analysis and Parsing Strategy of Mandarin Particle Questions." Proceedings of the International Computer Symposium, 38-43.
- Tang, Ting-Chi (湯廷池) (1981). "國語疑問句的研究." 師大學報 26, 219-277. also in Tang (1988 a), 241-311.

- (1983). "國語疑問句研究續論." 師大學報 29, 381-436. also in Tang(1988 a), 313-399.
- (1984). "國語裡「移動 α 」的邏輯形式規律." 教學與研究 6, 79-114. also in Tang (1988 a), 401-448.
- (1988 a). 漢語詞法句法論集. 台北, 台灣學生書局.
- (1988 b). "普遍語法與漢英對比分析." 第二屆世界華語文教學研討會論文集, 1309-1334.
- (1989) "普遍語法與英漢對比分析 第一章「X標橫理論」與詞組結構." Manuscript. Tsing Hua Univ.
- Wescoat, Michael T. (1987). "Practical Instructions for Working with the Formalism of Lexical Functional Grammar." Manuscript. Stanford Univ.
- Zwicky, Arnold M. (1985). "Clitics and Particles." Language 61, No. 2, 283-305.
- Zwicky, Arnold M., and Geoffrey K. Pullum (1983). "Cliticization vs. Inflection : English N'T." Language 59, No. 3, 502-513.

Smoothing Statistic Databases in a Machine Translation System

Keh-Yih Su*, Mei-Hui Su** and Li-Mei Kuan**

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

**BTC R&D Center
28 R&D Road II, 2F
Science-based Industrial Park
Hsinchu, Taiwan, R.O.C.

ABSTRACT

In a Machine Translation (MT) system, it is necessary to be able to determine the most likely structure among the ambiguities. This can be accomplished by using the probability as a selection basis for the well-formedness of each structure. However, this method requires a very large set of training data for the probabilistic database in order to obtain an acceptable degree of selection appropriateness.

In ArchTran English-Chinese Machine Translation System, a probability-based approach to automatizing the structure selection process is adopted. Although this method performs satisfactorily for structures already in the database, it performs rather poorly for structures not in the database. This is the problem with a sparse database. Therefore, in this paper, we propose to improve the prediction power of the database by a technique called **Database Smoothing**. Briefly, there are two smoothing methods that can be adopted. The first method is to employ a flattening constant to smooth the empty probability cells of the database. The second method is to incorporate additional information from another database into the one to be smoothed. We have conducted a simulation on the smoothed database and an improvement of 13.1 percent is observed for the open test samples. This is very encouraging because it shows improvements can be achieved for all database applications that employ a smoothed probabilistic model.

MOTIVATION

In a Machine Translation (MT) system, it is natural to have more than one interpretation for most input sentences. These ambiguous interpretations are attributable not only to the over-generative grammar adopted by the system but also to the inherent characteristics of the source language. Since the main purpose of a MT system is to produce a single appropriate interpretation for an input sentence in order to reduce the work for post editor, it is therefore desirable that the system provides a fast and competent mechanism to single out the correct interpretation.

In order to minimize the time spent on selecting the correct parse trees, we constructed several statistical databases (SDBS) as the means to automatize the tree selecting process [SU 88]. These databases contain the tree structures that are successfully parsed and selected by the linguist. With these databases, the well-formedness, in terms of score, of every ambiguous parse tree can be calculated for an input sentence. Afterwards, the parse tree with the highest score is selected as the preferred interpretation over all the other ambiguities.

We reported an experiment in regard to SDBS's prediction accuracy in [SU 88]. We found that with the database size of 1468 sentences, the accuracy rate for the close test can reach as high as 85%. However, the result is less accurate for the open test. The reason for this difference is because the training data for the database is not large enough. Consequently, the variety of sub-structures that can be found in the database is not extensive enough. Because of this, even if the structure is correct if its sub-structures do not match any corresponding entry in the database, its likelihood probability approaches zero. This is a serious problem for using database that is sparse in a MT system. In this paper, we propose to adopt the database smoothing technique that maintains the high accuracy rate for the structures already in the database and improves the prediction accuracy for structures outside the database.

There are two general approaches in smoothing a sparse database for improving the selection result of an open test. The first method is to smooth the cells of a database by a small flattening constant [FIEN 72]. The second method is to include information from a database that might not perform as well as the database to be smoothed but is less sparse. In the later sections, the approaches adopted for database smoothing will be presented.

Aside from structure selection, database smoothing can also be extended to other database applications. For instance, the truncation parsing mechanism in ArchTran also employs a probability database to direct the parsing of the input sentences. Information from this database is used to predict whether a path will eventually succeed or not. If a path receives a low prediction value, it will be truncated and the time will be saved. Similar to structure

selection, the truncation mechanism will also fall short of its function if its database is sparse and database smoothing is not used. Therefore, it is obvious that database smoothing is required for improving the reliability of the applications that use databases.

In the following sections, we will briefly discuss how the well-formedness of a structure is measured; how the databases in ArchTran are constructed and their shortcomings. Then, the mechanism of database smoothing will be described, followed by the result of our testing on the smoothed database. Last but not least, we will discuss some limiting factors that will affect the result of the database smoothing.

SCORE

The degree of well-formedness of a structure can be measured in terms of the syntactic well-formedness ($SCORE_{syn}$), the semantic well-formedness ($SCORE_{sem}$) and the lexical well-formedness ($SCORE_{lex}$) of the structure [SU 88]. According to [SU 88], for a structure X , its score can be reduced to $SCORE(X) = SCORE_{syn}(X) * SCORE_{sem}(X) * SCORE_{lex}(X)$. So for a sentence with more than one ambiguous structure, the most appropriate structure should be the one with the highest score. Since the semantic score and the lexical score have similar formulation as the syntactic score, they will not be discussed here.

The syntactic score of a structure can be generalized as the product of the conditional probability of its reduction sequences. Take the syntax tree in Fig.1 as an example. In this tree, n and v are the lexical categories of the input words, and S , NP and VP are the grammatical symbols. For this tree, written in the form of context sensitive rules with one right lookahead and one left context symbol, the reduction sequences of a LR derivation are : $(\phi n v \Rightarrow \phi NP v)$, $(NP v \phi \Rightarrow NP VP \phi)$, and $(\phi NP VP \phi \Rightarrow \phi S \phi)$, where ϕ is the null symbol.

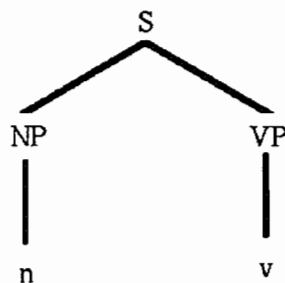


Fig. 1 A syntax tree

For these reduction sequences, the conditional probabilities are : $P(NP/ \phi n v)$, $P(VP/ NP v \phi)$ and $P(S/ \phi NP VP \phi)$, respectively. From [SU 88], the syntactic score for S is

$$SCORE_{syn}(S) = P(S|\phi NP VP \phi) * P(VP|NP v \phi) * P(NP|\phi n v).$$

Based on the structural well-formedness defined above, we can construct probability databases for selecting the most likely structure among the ambiguities for an input sentence.

DATABASES AND THEIR SPARSE DATA PROBLEM

In this section, we will first briefly describe the databases constructed for structure selection in ArchTran. Next, the sparse data problem of these databases will be discussed and the possible solutions will be presented.

Currently, we have ten independent databases that store the conditional probabilities of different types of reduction sequence. They are : L3, L2R1, L2, L1R2, L1R1, L1, R3, R2, R1 and N (no context information), where the numbers following L and R designate the number of left context symbols and right lookahead symbols referenced. These databases differ in that they incorporate different scopes of context information during their construction. For example, the L2R1 database is constructed with two left context symbols and a right lookahead symbol.

The problem with using a probability-based approach to select the most appropriate structure is that it can not do well for structure that is outside the scope of the database. This sparse data problem which can be decomposed into two parts. The first is the proliferation of empty cells (every possible reduction sequence occupies a cell in a database) because the training sample is small relative to all possible reduction sequence in the analysis grammar. The second is a special instance of the sparse data problem [JELI 80] when the samples in a set of databases are not large enough. As a result, some databases will be more reliable but have less statistics support, while other databases, are less reliable but have more samples to produce significant statistics. Under such circumstances, one database may perform better in some cases but less favorably in other cases.

The empty cell problem will affect the prediction performance of the database when most cells in the database are essentially empty. And the effect is that most of the cell queries will be zero during structure selection. Since the probability estimation of small values will not reflect the true probabilistic model, it could not be trusted as noted in [NADA 85]. Therefore, these cells must be filled. The most obvious solution is to enter as much sampling data into the SDBS as possible. But this is a very time-consuming long-term task whose affect is not

immediately felt. The reason is that the man power needed to find those correct sampling structures that will completely cover all possible derivations of an analysis grammar for a natural language is simply too enormous to even consider. A more feasible alternative is to adopt the flattening constant method suggested in [FIEN 72]. A more detailed description of this method will be presented in the next section.

Next, we will address the second aspect of the sparse data problem. The performance of different databases differs because the context reference in building a database also serves as a constraining factor in building the entries and in matching the sub-structures of a parse tree during structure selection. For example, the L2R1 database might support the linguistic model more accurately than the L1R1 database, but the variance of L2R1 is larger than L1R1. Therefore, L2R1 has less statistics for the open test samples and the prediction on of these samples is lowered. The following example will demonstrate this problem more concretely.

$$L_2 L_1 A R_1 \rightarrow L_2 L_1 B R_1$$

Let the above equation be a sub-structure included in the SDBS, where A is the symbol that reduces to B ; L_2 and L_1 are the left context symbols; and R_1 is the right context symbol. Then, there will be an entry of $L_2 L_1 A R_1 \rightarrow L_2 L_1 B R_1$ in the L2R1 database. At the same time, there will be an entry of $L_1 A R_1 \rightarrow L_1 B R_1$ in the L1R1 database. If a given sub-structure to be matched is $L'_2 L_1 A R_1 \rightarrow L'_2 L_1 B R_1$, this will not match any entry in L2R1 but it will match $L_1 A R_1 \rightarrow L_1 B R_1$ in L1R1. This means with L1R1, this sub-structure will have a value for its likelihood but not so with the L2R1. This shows that with a small training sample, L1R1 has more matchable entries than L2R1. In other words, for a structure outside a database's training data, it is more likely to obtain some usable information from database that is not as context-sensitive.

Following this logic, we can claim that if there is a database with no context restriction, any sub-structure will be most likely to match some entry in this database. But from [SU 88], it is shown that the accuracy rate for less restrictive database is lowered for selecting structures already in the database. The reason for this is the context-sensitiveness of the natural language. As the context information is discarded, the prediction power deteriorates. Therefore, switching a database to a less restrictive one (i.e. from L2R1 into L1R1) will not improve the selection result in general.

There are two ways to resolve this problem. The first method, is to enter as much sampling data into the SDBS as possible. Again, the required man power and time are the

limiting factors for adopting this method. The second method, an extension of an existing technique in signal processing [LEE 88], is to smooth the database with information from another database that is less context restricted. This technique of the database smoothing will be discussed in the next section.

SMOOTHING

To compensate for the inadequacy of not well-trained database in selecting structures outside database, we are adopting two methods of database smoothing to improve the prediction accuracy of a database.

The first is to smooth the databases with a flattening constants. In order to explore the extent of empty cells in the database, we did a tentative check. With 182 English sentences from the open test sample, all the ambiguity structures are broken down into database queries. And the result is tabulated in the following table.

Number of Queries	Databases	L2R1	N
Total Queries		53019	53019
Total Empty Cells Queried		20329	6401

Table 1. Database queries of the open test sentences

It is obvious from the table above that most cell queries from sentences in the open test sample are empty and therefore flattening constant is needed. The inclusion of flattening constant can be summarized in three simple steps. If we let the flattening constant be α , then the steps for smoothing entries with empty cells are as follows :

- [1] For every empty cell, let the cell value be equal to α .
- [2] For every non-empty cell, increment the cell value by α .
- [3] For every cell, calculate the probability of each cell by cell value / total occurrences in the entry.

From [FIEN 72], we choose to set α equal to 1/2 and we will demonstrate these steps with the following example. Let two original entries in a database be that shown in Fig. 2.

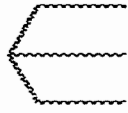
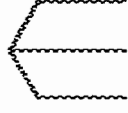
	<u>Left Context</u>	<u>Current State</u>	<u>Lookahead</u>		<u>Reduced To</u>	<u>Occurrences</u>	<u>Probability</u>
1.	L2 L1	S	R1		A	500	500/501
					B	1	1/501
					C	0	0
					Total Occurrences = 501		
2.	L2' L1'	S'	R1'		A'	1	1/1
					B'	0	0
					C'	0	0
					Total Occurrences = 1		

Fig. 2 Two Entries in L2R1 Database

In the above figure, each entry consists of three cells (or the number of possible reduction sequences) and each cell is followed by its number of occurrences and its conditional probability.

From this example, an additional problem of using a simple probability model can be observed. In Fig.2, the first reduction cell of the first entry has a probability value of 500/501 and the first reduction cell of the second entry has a value of 1/1. Consider the number of occurrences, it is obvious that the first instance of the first entry should be more likely than the first instance of the second entry. But the values of 500/501 vs. 1/1 do not reflect this observation. We will see that with the flattening constant added, this will be remedied. In the following figure, the entries are modified with the flattening constant.


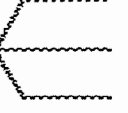
	<u>Left Context</u>	<u>Current State</u>	<u>Lookahead</u>		<u>Reduced To</u>	<u>Occurrences</u>	<u>Probability</u>
1.	L2 L1	S	R1		A	500+1/2	1001/1005
					B	1+1/2	3/1005
					C	1/2	1/1005
					Total Occurrences = 502+1/2		
2.	L2' L1'	S'	R1'		A'	1+1/2	3/5
					B'	1/2	1/3
					C'	1/2	1/3
					Total Occurrences = 2+1/2		

Fig. 3 Two Entries in L2R1 Database with α

Now, the empty cells of these entries are filled with values relative to the total number of occurrences of the entry. It should be noted that, the original value of 500/501 is replaced

by 1001/1005 and the original value of 1/1 is replaced by 3/5. This new set of values now reflects their real relative probability state.

The second smoothing method is to smooth the database with another database that is less sparse. So, the score of a tree is not the conditional probability calculated from just a single database. Instead, it is the interpolated conditional probability calculated from several databases.

In order to acquire a modeling for our databases, we devised a reward function y that rates how well the correct structures are selected. The reward function is such that after all the ambiguities of a sentence are ranked by the score from a database, if the correct structure falls at the first place, a reward of 5 is added. If the correct structure falls at the second place, a reward of 2 is added. For any place beyond, no reward is added. Now, we can show how different databases perform with this reward function. In the following figure, the open test sentences are grouped according to the percentage of empty cells they have queried. The numbers in the square brackets are the number of sentences in each group. For each group, the average reward is found and plotted against the group.

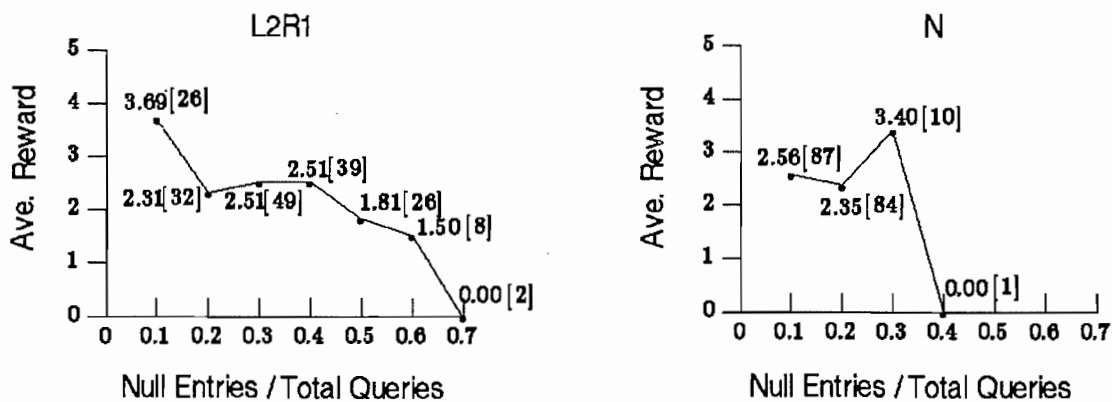


Fig. 4 Ave. Reward vs. Null Entries

From the figure above, it can be seen that the performance of L2R1 database deteriorates as the percentage of empty cells increases. But it is actually the opposite for the N database. Therefore, if we smooth the L2R1 database with the N database, the prediction of those sentences whose database queries are mostly zero will improve.

If we let P_s be the interpolated sum and P_i be the conditional probability calculated from the i th database, then $P_s=c_1P_1+c_2P_2$, with P_1 from L2R1 and P_2 from N. The coefficients are subject to $c_1+c_2=1$. The reason L2R1 database is selected as the one to be smoothed is because it exhibits the highest prediction rate for the close test. As for the N database, the reason why it is selected for smoothing is because it has most entries.

The P_s equation can be further modified with additional weighting functions. The reason for these functions is that the trustworthiness of a probability should be dependent on the total occurrences of all cells within the same entry. Therefore, the new equation is $P_s=c_1h_1(x)P_1+c_2h_2(x)P_2$, where $h_1(x)$ and $h_2(x)$ are the weighting functions such that $x=n/t$ (n is the number of total occurrences for this entry; t is the number of cells in this entry). The need for the weighting functions can be justified from the curve in Fig.5.

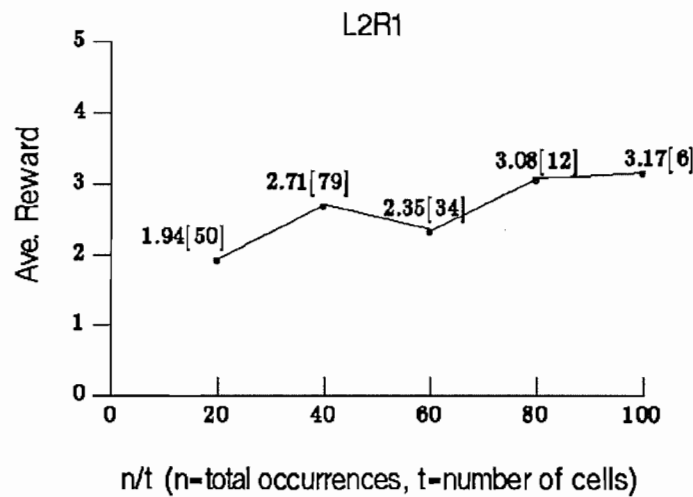


Fig. 5 Ave. Reward vs. n/t

In Fig. 5, the test sentences are divided into several groups according to the average of total occurrences divided by the number of cells of each database query (n/t). Afterwards, each group is plotted against the the average reward value of the group. From the curve, it can be seen that as the n/t value increases, the corresponding average reward increases. Therefore, there is a direct link between the accuracy of a probability and its n/t . With this curve, we can define the weighting function for L as $h_1(x)=a(1-e^{-xb})+1$, where a and b are tunable variables for matching the current state of the database. For simplicity's sake, the weighting function for N database is set to 1. The final equation for P_s is as follows :

$$P_s = c_1 h(x) P_1 + c_2 P_2,$$

Where $c_1 + c_2 = 1$,

$$h(x) = a \left(1 - e^{-bx}\right) + 1, \quad x = \frac{n}{t},$$

$n = \text{number of occurrences}$, $t = \text{number of cells}$,

a and b are tuning variables,

P_1 and P_2 are probabilities from two different databases.

In the following section, we will present the result of the simulation we conducted for testing the new P_s equation derived above.

TEST

We conducted a simulation with 182 sentences as the open test samples. For these test sentences, the reward value for using L2R1 database is 428 and for using N database is 418. The purpose of the simulation is to find out to what degree the reward value increases for the smoothed database.

During the simulation, we encountered two problems. First, our original databases did not record the empty cells because they will take up too much space. So, we have to expand the databases to include the empty cells for adding the flattening constant. But, it is simply impossible to generate all possible sub-structure of an over-generative analysis grammar for a natural language. As a result, we resort to expand the databases with just the ambiguous structures we have collected in the past. The second problem we encountered is that the reward function y does not have an analytic formula. So, all we can do is to observe the improvement of y as c_1 and c_2 make small deviations. Note that the reward function is not the same as the smoothed score function. The reward function is a measuring function of how well the smoothed score function is, that is, how well it predicts the correct structure of an input sentence.

Now, it is the question of finding a best set of a , b , c_1 , and c_2 for the smoothed database such that the reward value is the greatest. This can be seen as an optimization problem for the nonlinear reward function with certain constraints. We have devised an iteration method for finding these coefficients. Briefly, with some pre-selected values for a and b , we start with a set of initial coefficients, $C^0 = [c_1, c_2]$. The next set of coefficients are found by shifting each coefficient slightly in a direction such that the reward function increases, $C^{i+1} = C^i + \Delta C^i$. This iteration process continues until an optimal value is found for the reward function.

In the simulation, we selected several sets of initial values for a , b , c_1 , and c_2 . The results after several iterations are tabulated in the following table.

Data Sets \ Inputs & Results	a	b	c1	c2	Final Reward	Improvement (%)
1	0	~	0.8	0.2	475	11%
2	1	1	0.8	0.2	476	11.2%
3	1	1	0.6	0.4	483	12.8%
4	1	2	0.9	0.1	482	12.6%
5	1	100	0.7	0.3	475	11%
6	100	1	0.8	0.2	484	13.1%

Table 2. Open test results of the smoothed L2R1 database with different sets of inputs

As can be seen, the highest value we have achieved so far is 484. Compared with 428, it is an improvement of 13.1 percent. We also conducted a close test which consists of 50 sentences on the smoothed database. The open test results are tabulated in Table 3 with entries corresponding to the data sets in Table 2.

Data Sets \ Results	Reward Value	Deterioration (%)
1	208	3%
2	210	2.5%
3	207	3.7%
4	210	2.5%
5	208	3%
6	213	0.9%

Table 3 Close test results on smoothed L2R1 database

Comparing the results in Table 3 with the reward value of 215 for the original L2R1 database, it is obvious that the result of the close test has not deteriorate much. All in all, the result of the open test is very encouraging with the few points we tried. In the future, we would like to conduct a more extensive search for a even better set of values.

LIMITING FACTORS IN DATABASE SMOOTHING

In this section, we would like to discuss three factors that might influence the outcome of a smoothed database.

First, if the number of iterations is not large enough in looking for c_i s, it is questionable whether or not we have arrived at the best choice of all maximums. The embedded problem is that the analytic reward function is not known and its stability is dependent on the training sample of the databases. But there is an additional action that can be taken to minimize the effect of this problem. One can take some coefficient vectors that are more distant from the current maximum and start other searching iterations. When different end results are compared, if the current point is still the maximum then it can be certain that it is a relatively good maximum.

Second, if the test sentence sample is not large or random enough, then not every sentence type outside the database is compiled into the sample. As a consequence, the prediction power might not have improved for some sentences outside the database. Ideally, if it is possible to compile every possible sentence structure into the test sample, then a nearly perfect database can be constructed.

Third, if the test sample for the smoothing mechanism is too small then the variance in the smoothed database will be so large that it will affect the selection of structures that are within the database. Therefore, it is better to do the smoothing iteration with a test sentence sample consists of sentences from both inside and outside the database.

These factors are intended to serve as a reminder when employing the technique of database smoothing.

CONCLUSION

In a MT system, it is a time-consuming task to manually select the correct interpretation for a sentence among all generated ambiguities. Therefore, the idea of employing a statistic database as a tool to automatizing the structure selection evolves. But when the database has a small training sample, its prediction accuracy is not good enough for the open test. In this paper, we proposed to overcome this deficiency with the technique of database smoothing. This includes the adding of a flattening constant and the incorporating of additional information from another database.

We have conducted an open test of 182 sentences on the smoothed database. The result of a few trial tests shows an improvement of 13.1 percent. This encouraging result has prompted a more extensive testing planned in the near future.

ACKNOWLEDGEMENT

We are indebted to Prof. Anne Chao for her valuable suggestions. In addition, we would like to thank Jing-Shin Chiang for his helpful comments on this paper.

REFERENCE

- [FIEN 72] Fienberg, S.E and P.W. Holland, "On the Choice of Flattening Constants for Estimating Multinomial Probabilities," *Journal of Multivariate Analysis*, Vol. 2, PP. 127–134, 1972.
- [JELI 80] Jelinek, F. and R.L. Mercer, "Interpolated Estimation of Markov Source Parameters from Sparse Data," In E.S. Gelsema and L.N. Kanal (eds.) : *Pattern Recognition in Practice*, North-Holland Publishing Company, Amsterdam, Netherlands, PP. 381–397, 1980.
- [LEE 88] Lee Kai-Fu, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System," Doctoral thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1988.
- [NADA 87] Nada, A., "On Turing's Formula for Word Probabilities," *IEEE Transaction on Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, No. 6, PP. 1414–1416, Dec. 1985.
- [SU 88] Su K.Y. and J.S. Chang, "Semantic and Syntactic Aspects of Score Function," *Proceedings of the 12th international conference on Computational Linguistic*, Budapest, Hungary, PP. 642-644, 1988.

The Syntactic Projection Problem and the Comparative Syntax of Locative Inversion

by

Joan Bresnan

Stanford University,
Xerox Palo Alto Research Center

March 1, 1990

The fundamental problem this study addresses is how to predict the syntactic properties of verbs from information about their meaning and use—the ‘syntactic projection problem’. That a universal solution to this problem must exist has been argued from nature of language acquisition (Pinker (1989)). But the most compelling evidence for universal projection principles comes from comparative syntax, where languages that may differ genetically, areally, and typologically can be shown to instantiate the same principles for projecting verbal meaning and use into syntactic structures. Such a case is examined here.

English, a West Germanic language spoken in England and its former colonies, is genetically and areally unrelated to Chicheŵa, a Bantu language spoken in East Central Africa. The two languages also differ typologically, English belonging to a group of languages that employ case and government to express syntactic relations, and Chicheŵa belonging to a group that employs noun class and agreement instead (Bresnan and Mchombo (1987)). Despite these differences, English and Chicheŵa show remarkable correspondences in the properties of locative inversion, a syntactic unaccusative alternation studied in Chicheŵa by Bresnan and Kanerva (1989):

I will show that at the level of argument structure and function, English and Chicheŵa are subject to the same principles of syntactic projection, from which the unaccusativity or inversion phenomenon arises (following Bresnan and Kanerva (1989)).

1 Argument Structure

Locative inversion verbs in English and Chicheŵa have remarkably close correspondences at the level of argument structure.

1 ARGUMENT STRUCTURE

1.1 Intransitivity

In English, locative inversion occurs only with intransitive verbs, such as *be*, *sit* and *come*:

- (1) a. A lamp was in the corner.
- b. My friend Rose was sitting among the guests.
- c. The tax collector came back to the village.

Each of the examples alternates with a locative inverted form that shares the same thematic role structure:

- (2) a. In the corner was a lamp.
- b. Among the guests was sitting my friend Rose.
- c. Back to the village came the tax collector.

Note the characteristic preposing of the locative phrases and concomitant postposing of the subjects in (2a–c). This does not occur with transitive verbs such as *seat*, *find*, and *place*:

- (3) a. My friend Rose seated my mother among the guests of honor.
- b. *Among the guests of honor seated my mother my friend Rose.
- c. *Among the guests of honor seated my friend Rose my mother.
- (4) a. The locals can find lemon grass in the valley.
- b. *In the valley can find lemon grass the locals.
- c. *In the valley can find the locals lemon grass.
- (5) a. Susan has placed a menorah on the table.
- b. *On the table has placed a menorah Susan.
- c. *On the table has placed Susan a menorah.

1 ARGUMENT STRUCTURE

The same is true in Chicheŵa (Bresnan and Kanerva (1989)). Intransitive verbs such as *-li* 'be', *khala* 'sit' and *bwera* 'come' allow locative inversion. Example (6) is representative:¹

- (6) a. A-lendô-wo a-na-bwér-á ku-mu-dzi.
2-visitor-2 those 2 SB-REC PST-come-IND 17-3-village
'Those visitors came to the village.' (B-K (2b))
- b. Ku-mu-dzi ku-na-bwér-á a-lendô-wo.
17-3-village 17 SB-REC PST-come-IND 2-visitor-2 those
'To the village came those visitors.' (B-K (1b))

Transitive verbs such as *pěza* 'find', *thamangitsa* 'chase', and *tumiza* 'send' disallow locative inversion, as example (7) illustrates:²

- (7) a. Mâyi a-na-péz-á mw-aná kú-dâmbo.
1A mother 1 SB-REC PST-find-IND 1-child 17-5 swamp
'The mother found the child in the swamp.' (B-K (44a))
- b. *Ku-dâmbo ku-na-péz-á mâyi mw-ăna.
17-5 swamp 17 SB-REC PST-find-IND 1A mother 1-child
Lit.: 'In the swamp found the mother the child.' (B-K (44b))

1.2 Split Intransitivity and Passives

While locative inversion in English applies only to intransitive verbs, it does not apply to *all* intransitive verbs (Postal (1977, 147)). Intransitive verbs split as to whether they allow it (Levin (1986)):

- (8) a. Among the guests was sitting my friend Rose.
b. *Among the guests was knitting my friend Rose.
- (9) a. Onto the ground had fallen a few leaves.
b. *Onto the ground had spit a few sailors.

¹Chicheŵa examples taken from Bresnan and Kanerva (1989) are indicated by "B-K" followed by the example number in that work. In the glosses, roman numerals denote the 18 gender classes; the locative gender classes are 16, 17, and 18.

²Bresnan and Kanerva (1989) note that the result is ungrammatical whether the inverted subject precedes or follows the direct object in such examples.

1 ARGUMENT STRUCTURE

- (10) a. Into the hole jumped the rabbit.
b. *Into the hole excreted the rabbit.
- (11) a. Toward me lurched a drunk.
b. *Toward me looked a drunk.
- (12) a. On the corner stood a woman.
b. *On the corner smoked a woman.

Furthermore, locative inversion is possible with *passivized* transitive verbs. For example, the transitive verbs *seat*, *find* and *place* illustrated above all allow locative inversion when passivized. Note, however, that there is a restriction against the expression of the passive *by* phrase.

- (13) a. Among the guests of honor was seated my mother (?* by my friend Rose).
b. In the valley can be found lemon grass (?* by the locals).
c. On the table has been placed a menorah (?* by Susan).

Other examples of locative inversion with passives are the following:

- (14) a. To Louise was given the gift of optimism.
b. To a French research team has been attributed the discovery of a new virus.
c. In the package with your Saturday ticket are included a free hotdog, a BART coupon, and an Oakland A's sunvisor.
d. In this pot is being cooked a live lobster.

Exactly the same is true of Chicheŵa (Bresnan and Kanerva (1989)). The intransitive split is illustrated in (15) and the passive case in (16):

- (15) a. Ku-mu-dzi kw-a-khal-á nkhalambá zó-kha.
17-3-village 17SB-PERF-remain-IND 10elder 10-only
'In the village have remained only old people.' (B-K (50a))

- b. *Ku-mu-dzi kú-ma-lúk-á nkhalambá zó-kha.
 17-3-village 17 SB-PRS HAB-weave-IND 10 elder 10-only
 Lit.: 'In the village weave only old people.' (B-K (50b))
- (16) a. Ku-dâmbo ku-na-péz-édw-á mw-ána (?? ndí mâyi).
 17-5 swamp 17 SB-REC PST-find-PASS-IND 1-child (by 1A mother)
 'In the swamp was found the child (??by the mother).' (B-K (51b))
- b. M-nkhâli mw-a-phik-idw-á chákúdyá.
 18-9 cooking pot 18 SB-PERF-cook-PASS-IND 7 food
 'In the pot has been cooked food.' (B-K (54d))

Note that in Chicheŵa exactly as in English, there is restriction against the expression of the passive agent by a *ndí* 'by' phrase.

1.3 Locative Arguments and Theme Subjects

What characterizes the examples that allow locative inversion? It cannot be just those examples that have intransitive verbs with a locative argument. Consider the verb *shoot*, which takes a locative path argument and has two intransitive uses, illustrated in (17b,c):

- (17) a. A marksman shot a bullet through the wedding band.
 b. A marksmen shot through the wedding band.
 c. A bullet shot through the wedding band.

Although there is potential ambiguity in these examples, the intended reading of (17a) is that the marksman shot a projectile through the wedding band, while in (17b) the bullet is the projectile that passes through the wedding band. Locative inversion is clearly preferable with the latter:

- (18) a. ?*Through the wedding band shot a marksman.
 b. Through the wedding band shot a bullet.

What seems to characterize the locative inversion examples in both English and Chicheŵa is the interpretability of the subject as the argument of which the location, change of location, or direction expressed by the locative

argument is predicated—a *theme* in the sense of Gruber (1965) and Jackendoff (1972; 1976; 1987).³ This is precisely what distinguishes the uninverting (18a) from the inverting (18b). The marksman is not passing through the wedding band in shooting, so the subject designating this participant does not invert. But the bullet is passing through the wedding band, and this is the referent of the inverting subject.

The theme subject generalization clearly holds true of the locative inversion examples given earlier. Verbs like *sit*, *stand*, *fall*, and *lurch* predicate locations or change of locations of their subjects. In the case of motional activity verbs such as *jump* in (4), *fly*, or *run*, the subject is an agent in that it causes or controls the action, but it is also a theme in that it undergoes a change of location. In the case of the transitive verbs like *seat*, *find*, *place*, location or change of location is predicated of a theme object, not a subject, and locative inversion is not possible. When these verbs are passivized, however, the transitive object argument, which corresponds to the theme, is realized as a subject, and locative inversion becomes possible. And in the case of the dative examples (8a,b), location can be understood in an abstract sense.⁴

Theme subjects are necessary for locative inversion, but not sufficient: the verb must have a locative argument which is predicated of the theme. Tan (forthcoming) cites the contrast between the locative argument in an example like (19a), where the location is predicated of the rocks, and the locative adjunct in (19b), where the location is *not* predicated of the Rockies. Only the former allows locative inversion with passives (20):

- (19) a. Men placed the rocks in the helicopter.
 b. Men watched the Rockies in the helicopter.
- (20) a. In the helicopter were placed the rocks.
 b. *In the helicopter were watched the Rockies.

And even though it is the men who are in the helicopter in (19b), the subject designating them cannot invert:

³This generalization is observed for locative inversion in English by Levin (1986) and for Chicheŵa by Bresnan and Kanerva (1989).

⁴Pinker (1989) argues convincingly for a semantic difference between the dative expressed with *to*, which is abstractly locational, and the dative expressed with the double NP construction, which is possessional.

(21) *In the helicopter watched men.

This is because a locative argument, not an adjunct, is required for locative inversion.

We see, then, that the verbs that undergo locative inversion in English and Chicheŵa have a distinctive argument structure, in which the verb predicates of the subject a location, change of location, or direction expressed by the locative argument. I schematize this conclusion as follows:

(22) < th loc >
 |
 s

2 Presentational Focus

Not only the argument structure, but the discourse functions of locative inversion in English and Chicheŵa have remarkable correspondences. In both languages locative inversion has a special discourse function of *presentational focus* (Hetzron (1971; 1975), Bolinger (1971; 1977), Rochemont (1984)), in which the referent of the inverted subject is introduced on the scene. One effect of presentational focus is illustrated in (23), where (B) seems an odd response to (A):

(23) A: I'm looking for my friend Rose.

B: #Among the guests of honor was sitting Rose.

C: Rose was sitting among the guests of honor.

(B) seems odd because it seems to depend on a scene having been set that includes guests of honor, which (A) does not provide, and because Rose, having just been mentioned in (A), cannot be introduced on the scene naturally in (B). The uninverted form (C) is a more natural response. This effect is exactly analogous in Chicheŵa (Bresnan and Kanerva (1989, ex. (75))).

2.1 Pronominal Restriction

Next, there is the pronominal restriction: although the postposed subject may be definite or indefinite, it cannot be an anaphoric pronoun, as Rochemont (1984) observed:

(24) *Rose_i? Among the guests of honor was sitting she_i/her_i.

The reason appears to be that anaphora is pragmatically inconsistent with presentation. The ill-formedness of (24) cannot be attributed solely to a restriction against inverted pronouns, because the deictic use of the English pronoun is acceptable with locative inversion, again as observed by Rochemont (1984):

(25) Among the guests of honor was sitting HER [pointing].

Exactly the same restriction appears in Chicheŵa (Bresnan and Kanerva (1989)), as the following example illustrates. The pronoun used in (26) is nondeictic.

(26) *Ku-mu-dzi ku-na-bwér-á iwo.
 17-3-village 17 SB-REC PST-come-IND III PL PRON
 Lit.: 'To the village came they/them.' (B-K (76))

2.2 Contrastive Focus

The inverted subject is not only presented on the scene, but as Bresnan and Kanerva (1989) point out, it is focussed relative to the locative. This is brought out by the following contrast. In (27a) the locative is highly marked as a focus of contrast for the final *not* phrase, while the inverted subject is fine (27b):⁵

- (27) a. ??On the wall hung paintings, but not on the door.
 b. On the wall hung paintings, but not photographs.

In the uninverted forms, both the locative and the subject can be foci of contrast for the final *not* phrase:

- (28) a. Paintings hung on the wall, but not on the door.
 b. Paintings hung on the wall, not photographs.

⁵It is necessary to exclude the "repair" intonation from (27a), in which the utterance is repeated to correct a preceding statement. An example of repair is the following interchange between speakers A and B. A: On the door hung paintings. B: No! On the WALL hung paintings, not: "On the DOOR hung paintings." With the repair intonation, it is unnatural to continue with "... but not ..."

Exactly the same holds in Chicheŵa. The inverted forms are in (29) and the corresponding uninverted forms are in (30):

- (29) a. *Ku-mu-dzi ku-na-bwér-á mi-kângo ósatí kú-chi-tsíme.
 17-3-village 17 SB-REC PST-come-IND 4-lion not 17-7-well
 Lit.: 'To the village came lions, not to the well.' (B-K (80b))
- b. Ku-mu-dzi ku-na-bwér-á mi-kângo ósatí njovu.
 17-3-village 17 SB-REC PST-come-IND 4-lion not 10 elephant
 'To the village came lions, not elephants.' (B-K (80a))
- (30) a. Mi-kângo i-na-bwér-á ku-mudzi ósatí kú-chi-tsíme.
 4-lion 4 SB-REC PST-come-IND 17-3-village not 17-7-well
 'Lions came to the village, not to the well.' (B-K (79b))
- b. Mi-kângo i-na-bwér-á ku-mu-dzi ósatí njovu.
 4-lion 4 SB-REC PST-come-IND 17-3-village not 10 elephant
 'Lions came to the village, not elephants.' (B-K (79a))

These correspondences between two unrelated languages suggest that general principles of grammar underlie the alternation, and moreover, that these principles must relate the argument structure to the discourse function.

3 The Syntactic Projection Theory

Why is the distinctive theme-location argument structure associated with locative inversion? The answer proposed by both Levin (1986; 1987) for English and by Bresnan and Kanerva (1989) for Chicheŵa starts from the observation that the semantic role of theme (and patientive roles in general) universally alternates between syntactic subject and object. As Bresnan and Kanerva (1989) observe: "Cross-linguistically, the theme or patient is canonically expressed as either subject or object: (i) the subject in syntactically ergative languages (Kibrik (1985), Mel'čuk (1988)), (ii) the object in syntactically active languages . . . , and (iii) the transitive object and intransitive subject in syntactically accusative languages." In both English and Chicheŵa, the theme is the syntactic object of an active transitive verb and the syntactic subject of the passive transitive verb. And in both languages, intransitive verbs like *be*, *sit*, and *come* have the theme as the subject, but allow it to appear in the syntactic object position in locative inversion. This is, in essence, the unaccusative hypothesis.

The other semantic roles are syntactically constrained as well. Again as Bresnan and Kanerva (1989) observe: "Thus, cross-linguistically, the agent is canonically *not* encoded as object: in syntactically accusative languages it is the canonical subject, and in syntactically ergative languages it is a thematically restricted, nonobjective function (Dixon (1979), Wierzbicka (1981), Mel'čuk (1988))." Concerning the locative role, Bresnan and Kanerva (1989) state: "Finally, there is cross-linguistic evidence that locative arguments alternate between oblique and subject; particularly in existential sentences, locatives often appear with the basic word order and other properties of subjects (Kuno (1971), Clark (1978))."

3.1 Decomposition of Syntactic Functions

To distill these pervasive cross-linguistic generalizations into a formal theory of syntactic alternations in grammar, Bresnan and Kanerva (1989) postulate that the grammatical functions of subject, object, and oblique are constituted of more primitive elements, just as phonemes are constituted of more primitive distinctive features in phonological theory.⁶ Such primitives explain the existence of natural classes of functions, which share subsets of primitive elements.

Subject and object are hypothesized to have the primitive property of being semantically unrestricted—that is, capable of being associated with different semantic roles (and even having no semantic roles, as with expletive subjects and objects). This property is designated $[-r]$. On the other hand, objects are hypothesized to have the primitive property of complementing transitive predicators such as verbs and adpositions, and not complementing intransitive predicators such as basic nouns and adjectives. This property is designated $[+o]$. Obliques are restricted in the semantic roles they may express, hence $[+r]$, and they are nonobjectlike (complementing basic nouns and adjectives), hence $[-o]$. A consequence of this scheme is that there should be two kinds of syntactic objects, unrestricted and restricted. By definition, it is only the unrestricted objects that can alternate with subjects, and the restricted objects must have fixed semantic roles, like obliques.

⁶A similar proposal is made by Simpson (1983).

$$(31) \quad \begin{array}{cc} \begin{bmatrix} -r \\ -o \end{bmatrix} & \text{SUBJ} & \begin{bmatrix} +r \\ -o \end{bmatrix} & \text{OBL}_\theta \\ \begin{bmatrix} -r \\ +o \end{bmatrix} & \text{OBJ} & \begin{bmatrix} +r \\ +o \end{bmatrix} & \text{OBJ}_\theta \end{array}$$

(Note that OBL_θ abbreviates multiple oblique functions, one for each semantic role θ : OBL_{go} , OBL_{instr} , etc. In just the same way, OBJ_θ abbreviates restricted objects that are individuated thematically.)

This classification gives the following natural classes of syntactic functions:

$$(32) \quad \begin{array}{cc} [-r] = \text{SUBJ, OBJ} & [-o] = \text{SUBJ, OBL}_\theta \\ [+r] = \text{OBJ}_\theta, \text{OBL}_\theta & [+o] = \text{OBJ, OBJ}_\theta \end{array}$$

If we assume that the negative feature values are unmarked, we can also derive the following markedness hierarchy of the syntactic functions:

$$(33) \text{ markedness hierarchy:} \quad s > \begin{array}{c} \circ \\ \text{OBL}_\theta \end{array} > o_\theta$$

The subject is the least marked function; the restricted object is the most highly marked. In fact, many languages lack restricted objects altogether.

3.2 Syntactic Underspecification of Argument Roles

Under these assumptions, alternations between natural classes of syntactic functions are characterized by *underspecification*, rather than (lexical or syntactic) transformation. Thus, the typological generalizations described above are distilled into the following formal principles, which partially specify the syntactic functions of agent, theme, and location roles on the basis of the intrinsic meanings of the roles:

(34) Intrinsic classifications (IC):	agent:	$\begin{array}{c} ag \\ \\ [-o] \end{array}$
	theme:	$\begin{array}{c} th/pt \\ \\ [-r] \end{array}$
	locative:	$\begin{array}{c} loc \\ \\ [-o] \end{array}$

3.3 Hierarchical Argument Structure

Further specific properties of the syntactic function associated with a role—whether it is a subject or object, for example—derive from the argument structure of the verb. An argument structure consists of the lexical roles of a verb, their intrinsic syntactic classifications, and an ordering that represents the relative prominence of the roles. An important hypothesis in morphosyntax is that this relative prominence is not arbitrary, but semantically determined, the most prominent roles being those of the more causally active participants in events. This is the essential import of the ‘thematic hierarchy’, according to which (in the version assumed here) roles descend in prominence from agent through beneficiary, goal (recipient) and experiencer, instrumental, patient and theme, to location:⁷

$$(35) \quad ag > ben > go/exp > ins > pt/th > loc$$

Thus *sit* and *seat* have the respective argument structures:

$$\begin{array}{ccc} < th & loc & > & < ag & th & loc & > \\ & [-r] & [-o] & & [-o] & [-r] & [-o] \end{array}$$

In each argument structure the roles descend in prominence from left to right. The most prominent semantic role of a predicate is designated $\hat{\theta}$. Hence, $\hat{\theta}$ of *sit* is *th*, while $\hat{\theta}$ of *seat* is *ag*.

These hierarchically ordered argument structures, together with the intrinsic classifications, play a role in our theory that is analogous to the D-structure representations of syntactic movement theories of unaccusativity

⁷See Bresnan and Kanerva (in press) for references and discussion of alternative hierarchies.

and passivization (Burzio (1986), Chomsky (1986)). Like D-structures, these argument structures impose syntactically relevant prominence relations on underlying lexical semantic structures. Like D-structures, they allow us to define both the 'internal argument' and 'external argument' of syntactic movement theories of unaccusativity, though we define these lexically:⁸

- (36) 'Internal argument' role: θ
 $\quad \quad \quad \quad \quad \quad |$
 $\quad \quad \quad \quad \quad \quad [-r]$
- 'External argument' role: $\hat{\theta}$
 $\quad \quad \quad \quad \quad \quad |$
 $\quad \quad \quad \quad \quad \quad [-o]$

But unlike D-structures these argument structures also allow us to define the concept of 'logical subject', which plays an important role in grammar as well. The logical subject is $\hat{\theta}$, regardless of whether it is an internal or external argument role, or neither. Finally, unlike D-structures these argument structures are not defined over the syntactic vocabulary of constituent structure representation (NP, VP, PP, etc.), and so the principles that relate them to surface arrangements of syntactic functions differ substantively from restrictions on movement transformations.

3.4 Morpholexical Operations on Argument Structure

Argument structures can be altered by morpholexical operations, which add, suppress, or bind roles. For example, the Passive suppresses the highest role (the logical subject) of a verb:

- (37) Passive: $\hat{\theta}$
 $\quad \quad \quad \quad \quad \quad |$
 $\quad \quad \quad \quad \quad \quad \emptyset$

Suppression means simply that the role is syntactically unexpressed; it nevertheless remains the $\hat{\theta}$ in the argument structure of a passive verb. The agent phrase can be indirectly expressed as an optional, thematically bound adjunct (Bresnan (1978), Grimshaw (1988), Jackendoff (1987)). Examples of

⁸This observation is due to Zaenen (1988)) The characteristic structural difference between internal and external arguments with respect to the VP is derived below.

morphological operations which add and bind roles are the causative (Mohan (1988), Alsina (1989)) and the applicative (Alsina and Mchombo (1988)).

3.5 Default Syntactic Specifications

Default syntactic specifications apply finally, after any and all morphological operations. These (in the syntactic accusative language type) make the highest role unrestricted and lower roles restricted, by default.

- (38) a. $\hat{\theta}$
 $\quad \quad \quad |$
 $\quad \quad \quad [-r]$
- b. θ
 $\quad \quad \quad |$
 $\quad \quad \quad [+r]$

Defaults (39a,b) are ordered by the elsewhere condition; the default with the more restricted environment applies first.

A very general constraint on all function specifications is that they must preserve information: they can only add features, not delete or change them. This is called the monotonicity constraint. Thus, roles that are intrinsically classified $[-r]$ will not undergo default (38b), and may continue to alternate between subject and object, subject to the final well-formedness conditions.

3.6 Well-formedness Conditions

Finally, there are two well-formedness conditions on the specified argument structures resulting from the preceding principles, which are called 'lexical forms':⁹

- (39) (i) *The subject condition*: Every (verbal) lexical form must have a subject;
- (ii) *Function-argument biuniqueness*: Each expressed lexical role must be associated with a unique function, and conversely.

⁹Bresnan and Kanerva (1989) observe: "The generality of the subject condition (due to Baker (1983)) is open to question, because many languages have constructions in which there is no overt subject (see, e.g., Cole et al. (1989), Durie (1985a; 1987a)). It remains unclear whether these cases involve an empty nonlogical subject, as proposed by Baker (1983), or whether the subject condition itself is language-dependent."

4 Why Locative Inversion Occurs

The defaults (39i,ii) have the effect of always making the external argument the subject, and making the internal argument the subject only when there is no external argument. To see why this is so, consider first the active transitive verb *seat*, which has the three roles agent, theme, and location:

(40)	<i>seat</i>	<	<i>ag</i>	<i>th</i>	<i>loc</i>	>
intrinsic:			[-o]	[-r]	[-o]	
defaults:			[-r]		[+r]	
			S	O/S	OBL _{loc}	
w.f.:			S	O	OBL _{loc}	

The agent, being both $\hat{\theta}$ and intrinsically classified [-o], is the external argument, and it becomes the default subject. This forces the unrestricted theme (the internal argument) to become the object, by function-argument biuniqueness. The locative is oblique by default. This accounts for examples like (3a) *My friend Rose seated my mother among the guests of honor.*

Next consider the intransitive verb *sit*, which has the two roles theme and location:

(41)	<i>sit</i>	<	<i>th</i>	<i>loc</i>	>
intrinsic:			[-r]	[-o]	
defaults:				[+r]	
			O/S	OBL _{loc}	
w.f.:			S	OBL _{loc}	

Here there is no external argument. The theme, which is the internal argument, is $\hat{\theta}$. The theme can be either subject or object, but the defaults again make the location an oblique, so the theme must become subject to satisfy the well-formedness condition that every lexical form have a subject. This accounts for examples like (1b) *My friend Rose was sitting among the guests.*

Now consider the passive verb *seated* which shares the same role structure as the active verb. Passivization suppresses the *ag* role, which is the external argument, so that the derived argument structure resembles the

two-role verb *sit* above, and the defaults and well-formedness conditions apply in the same way.¹⁰ The internal argument becomes the subject to satisfy the subject condition.

(42)	<i>seat</i>	<	<i>ag</i>	<i>th</i>	<i>loc</i>	>
intrinsic:			[-o]	[-r]	[-o]	
passive:	<i>seated</i>		∅			
defaults:					[+r]	
				O/S	OBL _{loc}	
w.f.:				S	OBL _{loc}	

This accounts for examples like *My mother was seated among the guests of honor (by my friend Rose)*.

Thus by the defaults above, the external argument becomes the subject, and when there is no external argument, the internal argument does.

But now consider the requirements of presentational focus. In presentational focus, a scene is set and a referent is introduced on the scene to become the new focus of attention. In the core cases, a scene is naturally expressed as a location, and the referent as something of which location is predicated—hence, a theme. This imposes a natural selection of the < *thloc* > argument structure. As we have just seen, the unmarked syntactic realization of these arguments would have the theme become the subject and the location, an oblique. But a pervasive functional generalization across languages is that the subject is the unmarked discourse topic, and this would often conflict with the presentational focussing of the theme argument, for the same reason that pronominal anaphora conflicts with it. Given that the theme is unrestricted, however, there is a way to solve this problem: make the location the subject, for it is the more topical argument. The well-formedness conditions will then force the theme to be realized as an object, and the object is the focussable syntactic function *par excellence*. But this solution has two essential limitations: first, it is conditioned by the special environment of presentational focus; second, it will always fail in the presence of an active agent in the argument structure, for the active agent (being the external argument) becomes the grammatical subject, and blocks any other subject.

¹⁰Since $\hat{\theta}$ is unexpressed, the effect of the $\hat{\theta}$ default, specifying the *ag* as [-r], is vacuous and not shown.

This idea is formally incorporated in our theory in the following 'focus subject default' postulated by Bresnan and Kanerva (1989) as an addition to the defaults previously given (38a,b). As before, these defaults are ordered by the elsewhere condition; hence (43) must precede the final default that makes all theta roles [+r] (38b). Its effect, then, is to make the *loc* role the subject (or alternatively, to introduce an expletive subject in the same context).

(43) Focus subject default:

$$\begin{array}{c} [f] \quad loc \\ | \\ [-r]/expl \end{array}$$

The feature [f] refers to the presentational focus attribute(s), and *expl* denotes an expletive subject which may appear as an alternative to the classification of *loc* as [-r]. In English, this expletive is what is known as 'presentational *there*' (Aissen (1975)).¹¹

They further propose that the distribution of the focus feature [f] is a parameter of variation across languages. In Chicheŵa it is subject to the constraint given in (44), which states that only the theme argument can bear the [f] feature, and only when it is the highest expressed role:

(44) Focus parameter:

$$\begin{array}{c} < \quad th/pt \\ | \\ [f] \end{array}$$

The same parameter is selected in the grammar of English.

With these additions to the theory, locative inversion in English now falls into place. Consider how it arises with the intransitive verb *sit*:

(45)

	<i>sit</i>	<	<i>th</i>	<i>loc</i>	>
intrinsic:			[-r]	[-o]	
focus:			[f]		
defaults:				[-r]	
			o/s	s	
w.f.:			o	s	

¹¹The provision for an expletive subject is a parameter of variation which is not taken in Chicheŵa.

The theme is the highest expressed role, and when it is presentationally focussed, the focus subject default is applicable, making the locative the subject. By the well-formedness conditions, the theme becomes the object. This accounts for examples like (2b) *Among the guests was sitting my friend Rose*.

In contrast to *sit*, an intransitive verb like *spit* has an agent rather than a theme as the highest role. The locative is not predicated of the agent, which thus lacks themelike properties and receives only the agentlike intrinsic classification. Hence the agent is an external argument role, and since it must become the subject, locative inversion could never arise (because there can only be one subject):

(46)	<i>spit</i>	<	<i>ag</i>	>	<i>loc</i>	>
intrinsic:			[-o]		[-o]	
focus:			*[f]			
defaults:			[-r]		[+r]	
			-----		-----	
			S		OBL _{loc}	

This account for examples like (9b) **Onto the ground had spit a few sailors*. Thus the split intransitivity of locative inversion falls out of this theory.

Motional verbs like *creep*, *jump* are thematically ambivalent (Bresnan and Kanerva (1989)): their highest role is both an agent because it is in control of the activity, and a theme because it undergoes a change of location. These verbs can receive in principle either the theme ([-r]) or the agent ([-o]) classification of $\hat{\theta}$, and will undergo locative inversion with the theme [-r] classification. In this way the theory accounts for both 'active' and 'stative' types of locative inversion in English (Aissen (1975)).

The effect of passivization on locative inversion falls out as well. With an active transitive verb like *seat*, locative inversion can never arise because the external argument role will become the subject. But under passivization this role is suppressed, locative inversion can occur:

(47)	<i>seat</i>	<	<i>ag</i>	<i>th</i>	<i>loc</i>	>
intrinsic:			[-o]	[-r]	[-o]	
passive:	<i>seated</i>		∅			
focus:				[f]		
defaults:					[-r]	
			o/s	s		
w.f.:			o	s		

This accounts for examples like (13a) *Among the guests of honor was seated my mother.*

The *by*-phrase restriction on locative inversion with passives can also be explained, assuming that the *by*-phrase adjunct binds the $\hat{\theta}$ role, and thereby serves indirectly to express it. The focus parameter (44) will thus be inapplicable to such a passive argument structure, where the theme is not the highest expressed role:

$$\textit{seated} \langle \textit{ag}_i \textit{th} \textit{loc} \rangle \textit{by} \langle \theta_i \rangle$$

This accounts for the ill-formed variant of (13a) with the passive *by*-phrase: *?*Among the guests of honor was seated my mother by my friend Rose.*

Thus the theory explains why locative inversion fails to occur with transitive verbs, why it splits among intransitives and occurs with passives, why it prohibits the passive *by*-phrase, why passive verbs with non-theme subjects disallow it, and why it occurs in the marked context of presentational focus. Given our theory and the focus parameter (44), these properties necessarily cluster together, and their presence as a group in both English and Chicheŵa is not accidental.

The theory also derives the salient structural difference between internal and external arguments—their asymmetry with respect to the VP. It follows from the projection theory that external arguments are always subjects, while internal arguments may be subject or objects. In the \bar{X} theory of Bresnan (1982) the syntactic categories are defined in terms of syntactic functions. By definition, the VP is the phrase structure category that is both predicative (i.e. cannot dominate a subject NP) and potentially transitive (i.e. can dominate object NPs). It follows that if a language has a VP, the external argument must appear in a position external to the VP, while the internal argument may appear either VP-internally or VP-externally.

5 References

- Aissen, Judith (1975) "Presentational-There Insertion: A Cyclic Root Transformation" in Robin E. Grossman, L. James San, and Timothy J. Vance, eds., *Papers from the Eleventh Regional Meeting Chicago Linguistic Society April 18-20, 1975, Chicago, Illinois*, Chicago Linguistic Society, Chicago, Illinois, pp. 1-14.
- Alsina, A. (1989) "Causatives," ms., Department of Linguistics, Stanford University.
- Alsina, A. and S. A. Mchombo (1989) "Object Asymmetries in the Chicheŵa Applicative Construction," ms., Departments of Linguistics, Stanford University and the University of California, Berkeley.
- Bolinger, D. (1971) "A Further Note on the Nominal in the Progressive," *Linguistic Inquiry* 2, 584-586.
- Bolinger, D. (1977) *Meaning and Form*, Longman Group Ltd., London.
- Bresnan, J. (1978) "A Realistic Transformational Grammar," in M. Halle, J. Bresnan, and G. Miller, eds., *Linguistic Theory and Psychological Reality*, The MIT Press, Cambridge, pp. 1-59.
- Bresnan, J. (1982) "On Control and Complementation," *Linguistic Inquiry* 13, 343-434.
- Bresnan, J. and J. M. Kanerva (1989) "Locative Inversion in Chicheŵa: A Case Study of Factorization in Grammar," *Linguistic Inquiry* 20.1, 1-50. Also forthcoming in E. Wehrli and T. Stowell, eds., *Syntax and Semantics 24: Syntax and the Lexicon*, Academic Press, New York.
- Bresnan, J., and J. Kanerva (to appear) "The Thematic Hierarchy and Locative Inversion in UG. A Reply to Paul Schachter's Comments," in E. Wehrli and T. Stowell, eds., *Syntax and Semantics 24: Syntax and the Lexicon*, Academic Press, New York.
- Bresnan, J. and S. A. Mchombo (1987) "Topic, Pronoun, and Agreement in Chicheŵa," *Language* 63, 741-782.
- Bresnan, J. and L. Moshi (to appear) "Object Asymmetries in Comparative Bantu Syntax," *Linguistic Inquiry* 21.2.
- Burzio, L. (1986) *Italian Syntax: A Government-Binding Approach*, Reidel, Dordrecht.

5 REFERENCES

- Chomsky, N. (1985) *Knowledge of Language: Its Nature, Origin, and Use*, Praeger, New York.
- Grimshaw, J. (1988) "Adjuncts and Argument Structure," *Lexicon Project Working Paper #21 and Occasional Paper #36*, The Center for Cognitive Science, MIT.
- Gruber, J. S. (1965) *Studies in Lexical Relations*, Doctoral dissertation, MIT, Cambridge, Massachusetts.
- Hetzron, R. (1971) "Presentative Function and Presentative Movement," *Studies in African Linguistics*, Supplement 2, Oct, 79-105.
- Hetzron, R. (1975) "The Presentative Movement, or Why the Ideal Word Order is VSOP," in Charles N. Li, ed., *Word Order and Word Order Change*, University of Texas Press, Austin, 345-388.
- Jackendoff, R. (1972) *Semantic Interpretation in Generative Grammar*, MIT Press, Cambridge, Massachusetts.
- Jackendoff, R. (1976) "Toward an Explanatory Semantic Representation," *Linguistic Inquiry* 7, 89-150.
- Jackendoff, R. (1987) "The Status of Thematic Relations in Linguistic Theory," *Linguistic Inquiry* 18, 369-411.
- Levin, Lorraine (1986) *Operations on Lexical Forms: Unaccusative Rules in Germanic Languages*, MIT Ph.D. dissertation.
- Mohanan, T. (1988) "Causatives in Malayalam," ms., Department of Linguistics, Stanford University.
- Pinker, S. (1989) *Learnability and Cognition: The Acquisition of Argument Structure*, The MIT Press.
- Postal, Paul (1977) "About a 'Nonargument' for Raising," *Linguistic Inquiry* 8.1, 141-154.
- Rochemont, M. S. (1986) *Focus in Generative Grammar*, John Benjamins Publishing Company, Amsterdam.
- Simpson, J. (1983) *Topics in Walpiri Morphology and Syntax*, doctoral dissertation, MIT.
- Tan, Fu (forthcoming) *The Notion of Subject in Chinese*, Department of Linguistics, Stanford University.

5 REFERENCES

- Zaenen, A. (1988) "Unaccusativity in Dutch: An Integrated Approach,"
to appear in J. Pustejovsky, ed., *Semantics and the Lexicon*, Kluwer
Academic Publishers.