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-Pustejovesky 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.

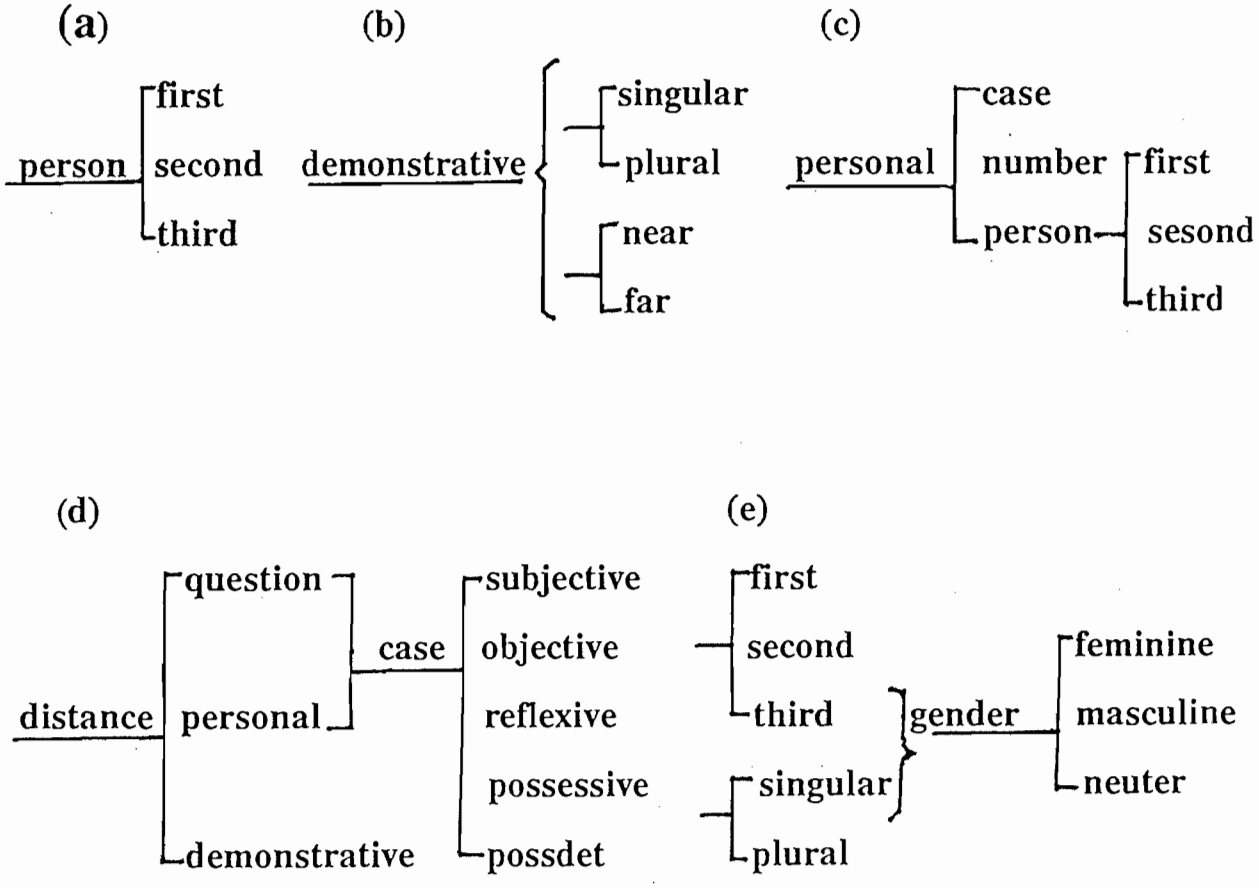

Figure 1 The symbols used in systemic networks

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.

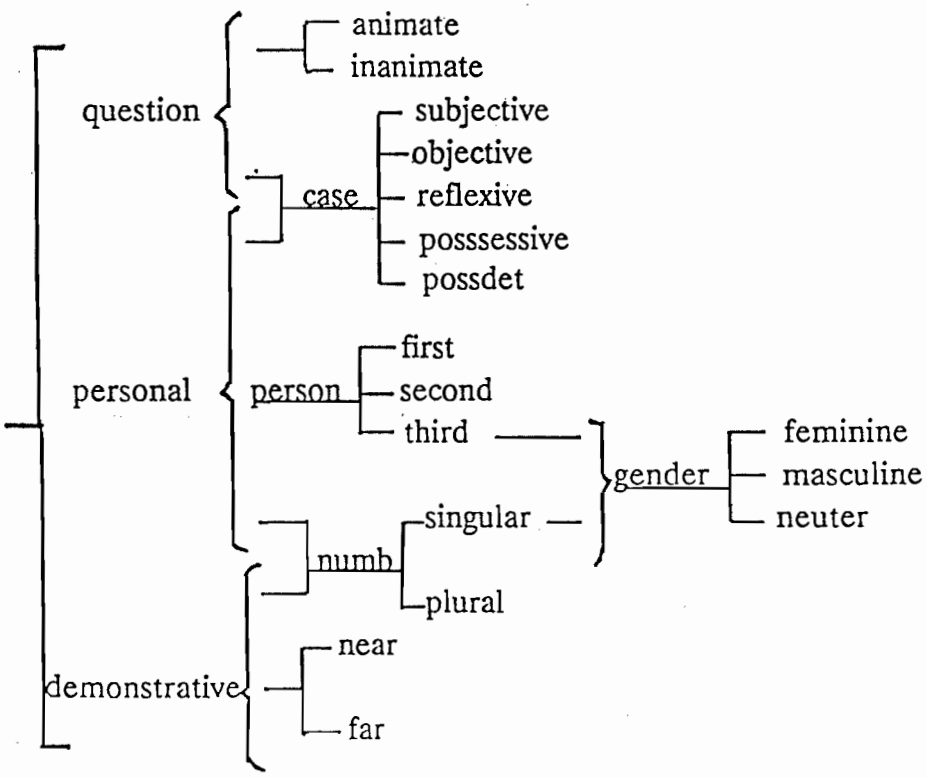

Figure 2 The system network for English pronouns

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.

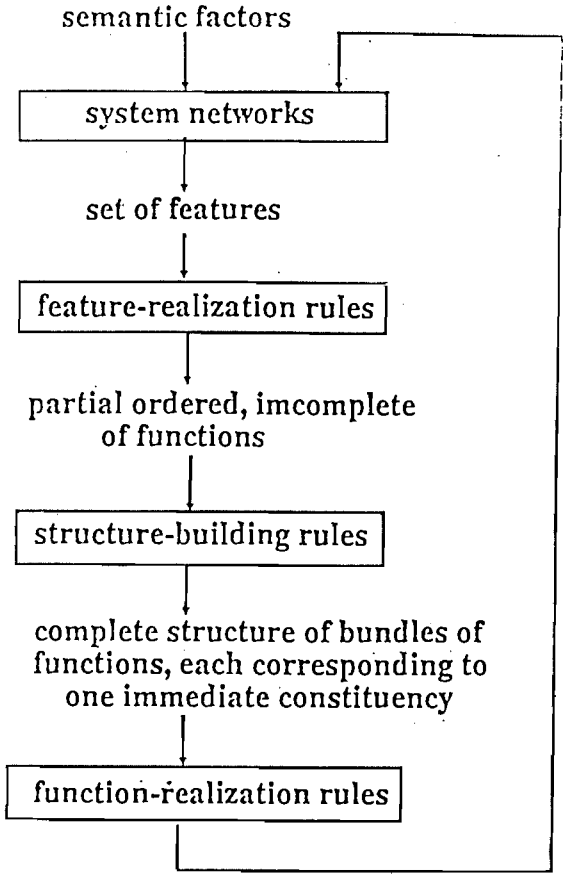

Figure 3 The generative process

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.

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 sub-frames 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

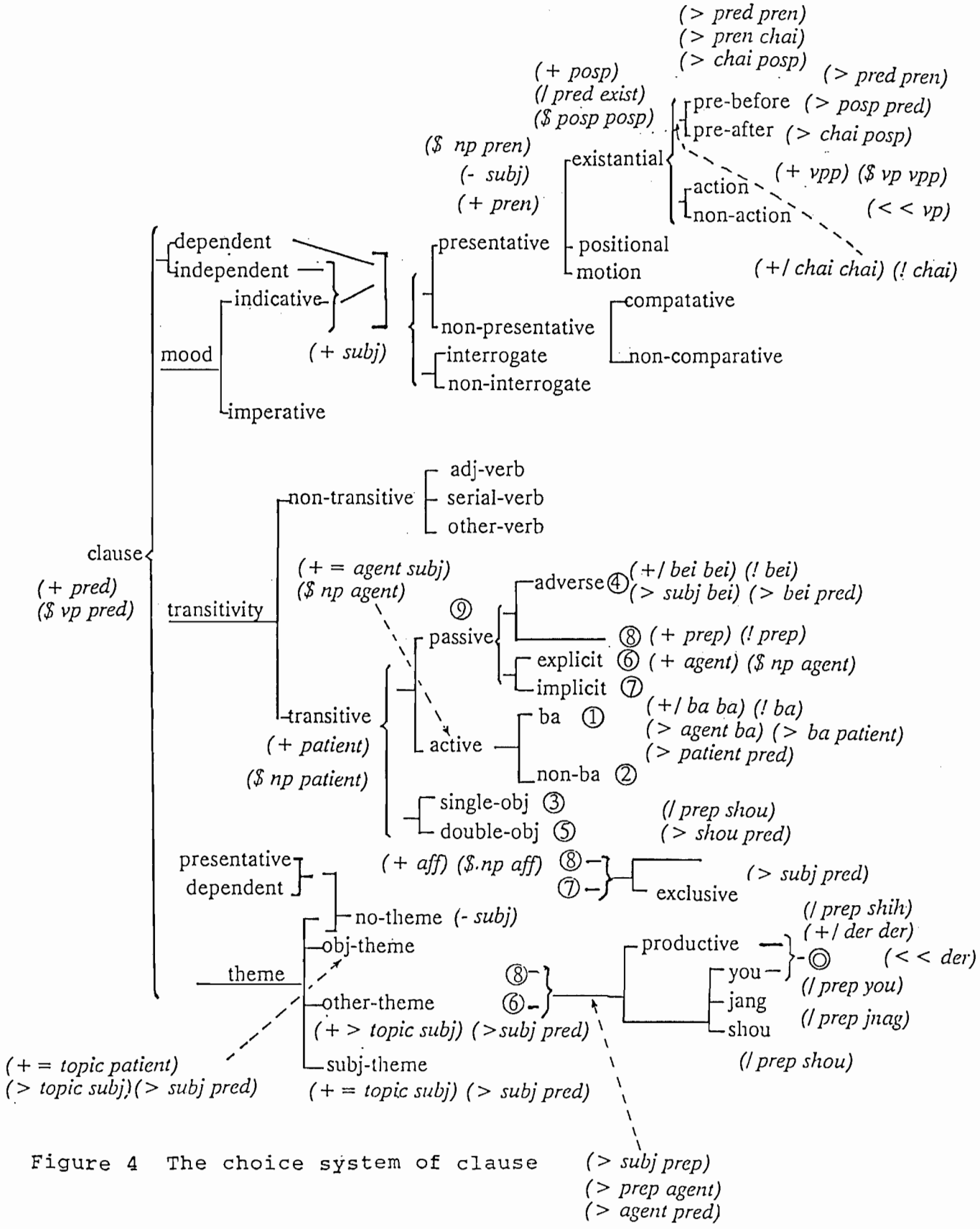

Figure 4 The choice system of clause

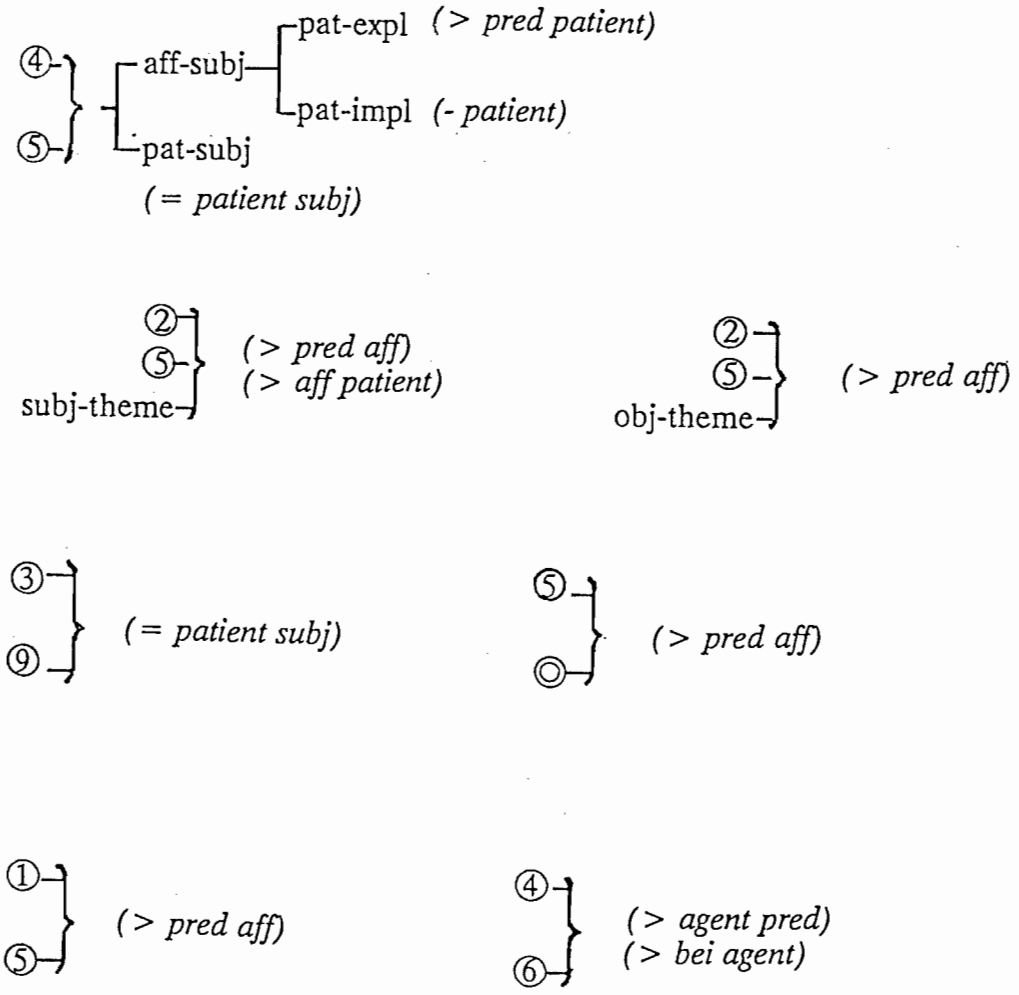

Figure 4 The choice system of clause (continued)

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 indpendent 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 demostrative (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*)

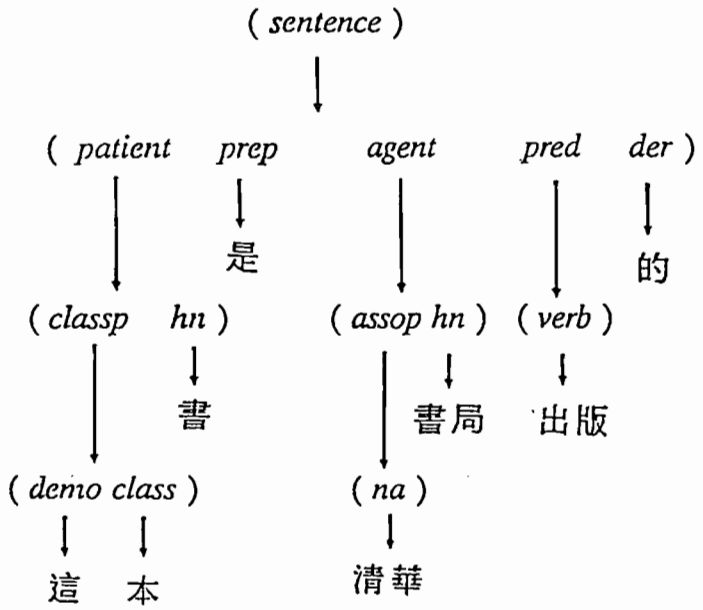

Figure 5 A example (continued)

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 therefor 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-Pustejovesky 1985]
D.D. McDonald and J.D. Pustejovesky, "*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