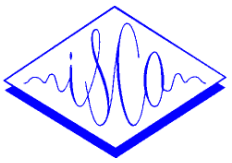

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

A GENERATION SYSTEM FOR CHINESE TEXTS

Hua WU, Taiyi HUANG, Bo XU
National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing
E-mail: { wh, huang, xubo } @nlpr.ia.ac.cn

ABSTRACT[1]

A domain-independent, reusable, general text generation system for Chinese is presented in this paper. This system combines the template method and generation technology in a single formalism, which enables the system to maintain both flexibility and efficiency. At the same time, in order to maintain the generator's independence of application domains, An upper model is designed to interface the different application and the general generation system, which enables the system's adaptability to different application domains. The upper model is a kind of semantic hierarchy. It is organized according to the semantic relationship between predicates and their arguments, nouns and their modifiers. Tests show that the generation system embodies good adaptability to different application domains and good performances.

1. INTRODUCTION

Natural language generation (NLG), an important branch of natural language processing, is the area that investigates how computer programs can be built to produce natural language text from a computer-internal representation. The whole natural language generation procedure can be modularized into three components: content determination, sentence planning and surface realization.

In this paper, we introduce our Chinese text generation system, which only concerned with surface realization. This system makes use of both the systemic functional grammar and the functional unification formalism [1]. And it also combines the template method and the generation technology in the same formalism. The main advantage is that the generation system maintains both the flexibility and efficiency. Meantime, we consider maintaining the generator's independence of application domains. An upper model is designed to interface the different application and the general generation system, which enables the system's adaptability to different application domains.

The second section introduces the surface generation system. The third section discusses the upper model that interfaces different application and the general Chinese text generation system. The last section presents the tests on the surface generation system and the conclusion.

[1] The research work described in this paper is supported by the National Natural Science Foundation of China under grant number 69835030, the National '863' Hi-Tech Program under grant number 863-306-ZT03-02-2 and the National Key Fundamental Research Program (the 973 Program) of China under the grant G1998030504

2. SURFACE REALIZER

The task of the surface realizer is to convert results of the sentence planning into natural language text. In order to maintain the surface realizer's independence of the output of the preceding system, we define an intermediate representation. Any input to the surface realizer must be first converted into the intermediate representation.

2.1 Intermediate Representation

The intermediate representation (IR) is made up of many feature stru ctures. In fact, It is an extended predicate argument structure. It includes three parts: predicate information, obligatory arguments and optional arguments. The predicate information describes the top-level information in a clause including the main verb, the mood, the voice, the tense and so on. The obligatory arguments are roles that must be filled in a clause for it to be complete. And the optional arguments specify the location, the time, the purpose of the event etc. They are optional because they do not affect the completeness of a clause. The BNF definition is shown in Figure 1.

```
IR={feature-structure}*
Feature-structure = (Attribute    Value)
Attribute = Syntax | Semantic
Attribute = process | mood | voice | tense | predicate |
            template | …
Value = Chinese-word | symbol | feature-structure
```

Figure 1: Intermediate Representation

2.2 Chinese Realizer

The tasks of the surface realizer are:

- Define the sentence and the phrase structure
- Define the order of the constituents in a sentence and in phrases.
- Add functional words

In the surface realizer, Two factors determine our selection. First, the generation process is from function to structure. Second, the feature structure can be easily extended to express

templates. So we select systemic functional grammar as the surface grammar. The grammar is represented in a functional formalism and implemented with the unification algorithm.

2.2.1 The Grammar System

In the surface generation module, we use the functional unification formalism. And at the same time, we make use of the systemic functional grammar. The rule system is made up of many sub-systems such as transitivity system, mood system, tense system and voice system. The input must depend on all of these systems to make different level decisions. Now we only focus on the transitivity system because of space limit.

The transitivity system consists of seven processes. The first six processes are similar to those Halliday proposed [2]. They are material process, mental process, behavior process, relation process, verbal process and attribute process. We add the compound process according to the characteristics of Chinese. It can be further classified into two sub process systems: the parallel process (there are two parallel verbs in a single sentence) and the pivotal process. For example, The sentence "**他**(he)**乘**(take)**飞机**(plane)**去**(go)**北京**(Beijing)**了**" belongs to the parallel process because there are two verbs "**乘**" and "**去**" in the sentence and they are parallel to each other. The sentence "**我们**(we)**选**(elect)**约翰**(John)**当**(act as)**班长**(monitor)" belongs to the pivotal process because John acts as both the object of the verb "**选**" and the subject of the verb "**当**". The whole transitivity system is shown in Figure 2.

The seven processes only capture the common characteristics of Chinese. There are many particular phenomena that can not be included in the processes. In some case, although two verbs belong to the same process, the semantics of their participants may be different. For example, "kick" and "eat" are two verbs, both of which belong to the material and require two participants. But the semantic category of the object of *eat* must belong to food, while that of *kick* must belong to physical object. This kind of knowledge can only be represented in the subcategorization frame of the particular verb. So we represent the common structures of verbs in the processes and individual knowledge in the subcategorization frame of the particular verb.

2.2.2 Template and NLG technology

Template methods and NLG technology are both used in generating natural language. Each of them has its pros and cons. The template method is rated as efficient but inflexible, while the NLG technology is considered as flexible but inefficiency. So the mixed or hybrid method to combine the template method and the NLG technology has been developed. Busemann [3] used hybrid method to allow template, canned texts and general rules appearing in the same formalism and to tackle the problem of the inefficiency of the grammar-based surface generation system. E.Pianta [4] used the mixed representation approach to allow the system to choose between NLG technology and template method.

Our system keeps the surface generation module general for Chinese. At the same time, it can also combine template method and generation technology in the same formalism to tackle the characteristics of the spoken Chinese and to speed up the generation procedure. The surface realizer deals with the spoken Chinese language. And it is used in the real application tasks. In spoken Chinese, there are many fixed expressions such as those that express thanks and apologies. So we use the template method to generate the fixed expressions and use the NLG technology to generate complex sentences. We can also deal with template (here we use template to refer to both template and canned texts) in the input without changing the whole generation process. If the attribute in the functional description is "template", then the value must be taken as canned texts or strings with slots.

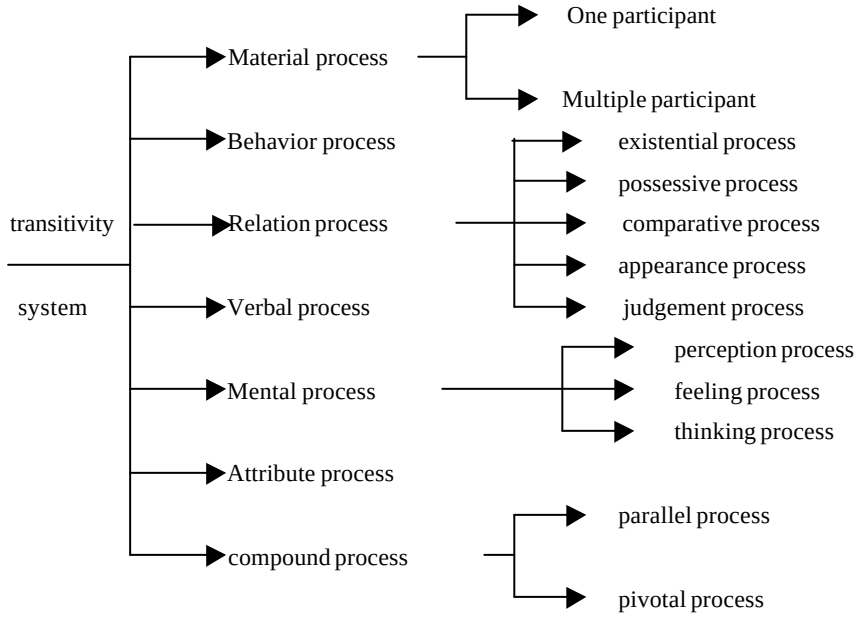

Figure 2: The Chinese transitivity system

After the template is inserted, there are three different structures in the surface realizer: the template with slots, canned texts and the rule-based system. We take the word "template" as the attribute item and the canned string or strings with slots as the value item according to the function structure. Then the structure (template string) forms a standard feature equation. If we use "pattern" to indicate the final order of the constituents in a sentence, then we can express the template structure easily as shown in (1):

(pattern template) (1)

Now we have combined the template method and NLG technology in the same formalism without any modification.

Let's consider one example taken from the travel information retrieval task. When the customer asked if there are tours to northwest of China and we have no such tours, our system always replies as "**去西北，对不起，我们目前没有这样的服务**"(To north-west. I am sorry but we currently have no such service). In this case, we use the template method. It is shown as follows:

((cat clause)
(template **去** <place>, **对不起，我们目前没有这样的服务。**))

cat stands for the syntactic category. <*place*> is a slot standing for one variable of scenic spots. From the example, we can see that the template method is very efficient in dialogues.

2.2.3 The Realization Procedure

The input to the syntactic generation provides enough information about sentence and phrase structure. Most of information such as the main verb, thesubcategorization frame, the sentence type is defined in the input. So we can traverse the input in a top-down, depth-first fashion.

In our functional formalism, the unification algorithm is the basic operation between the input and the grammar. When two feature equations are unified, both must be tested if they are compatible with each other. Here we extend the basic unification algorithm to be suitable for our system. The definition of the unification is as follows:

Definition 1: Unification Algorithm ($\cup$)

1. Both a and b are the atom values of the feature item, then
   ① $a \cup b = a$ if $a = b$;
   ② $a \cup b = b$ if $a \neq b$ and a subsumes b in the hierarchy;
   ③ $a \cup b = a$ if $a \neq b$ and b subsumes a in the hierarchy;
   ④ or else, $a \cup b = \phi$ (ϕ is an empty set).

2. Both α and β are complex feature structures and f is a feature item and v is a value, then
   ① f = v belongs to $\alpha \cup \beta$ if $\alpha(f) = v$ and $\beta(f)$ is not defined yet;
   ② f = v belongs to $\alpha \cup \beta$ if $\beta(f) = v$ and $\alpha(f)$ is not defined yet;
   ③ f =(v1 $\cup$ v2) belongs to $\alpha \cup \beta$ if $\alpha(f) = v1$, $\beta(f) = v2$, v1 and v2 are not in conflict with each other. Or else, $\alpha \cup \beta = \phi$.

The whole unification procedure is as follows:

- Unify the input with the sentence part of the grammar.
- Identify the constitutes inside the input
- Unify the constituents with the phrase part of the grammar recursively in a top-down, depth-first fashion.

From the above procedure, we can see the whole surface generation process is composed of two phases: sentence unification and phrase unification. The sentence unification phase defines the sentence structure and orders the components among the sentence. The phrase unification phase defines the phrase structure, orders the components inside the phrases and adds the function words.

3. THE UPPER MODEL

For a domain-independent, reusable, general text generation system, it is critical to interface with different domain applications. It is difficult for us to consider what domains in which it will be used when we design a general text generation system. How can we relate the domain concepts into their linguistic form in the output? One way is to design the application domain according to the generation system. But it is usually infeasible because, in this way, the domain designer must be familiar with the whole generation process. So we must seek for an interface which can interact with different domain knowledge and the linguistic form in the generation system. This interface is called as upper model here.

We organize the upper model in a hierarchical structure and classify it into three top levels: object, quality and process. They correspond to noun, adjective and verb in the linguistic concept. Each level can be further classified into sub types. For example, the object level can be classified into physical object, abstract object, temporal object and spatial object. Quality can be further classified into size, color, feature, shape, and so on. Process can be classified into seven sub classes as described in Figure 2 of section 2.2.1. The classification of the object is made according to the semantic relationship between the predicate and its arguments, while the classification of the quality is made according to the semantic relationship between nouns and their modifiers.

The effect of the upper model can be seen from the relationship between the upper model and the domain knowledge, and the relationship between the upper model and the surface realizer. First, let us look at the relationship between the upper model and the domain knowledge. Any concepts in different domains can be subsumed under the upper model types. Let us consider two examples in two different domains: one is retrieved from the travel route domain. The inputs are open-class lexicon items with some attributes including"**我**"(I), "**星期五**"(Friday), "**去**"(go), "**北京**"(Beijing). The other is retrieved from the room reservation domain. The inputs are "**我们**"(we), "**有**"(have), "**便宜**"(cheap), "**双人间**"(double room). The relationship is shown in Figure 3. In the travel route domain, "**我**", "**星期五**", "**去**" and "**北京**" belong to person, time, move and route respectively. The domain concepts including time, move and route are subsumed under temporal object, material process and spatial object respectively. In this way, we can determine the type of every lexicon in one sentence, without considering their relationship with the upper model directly.

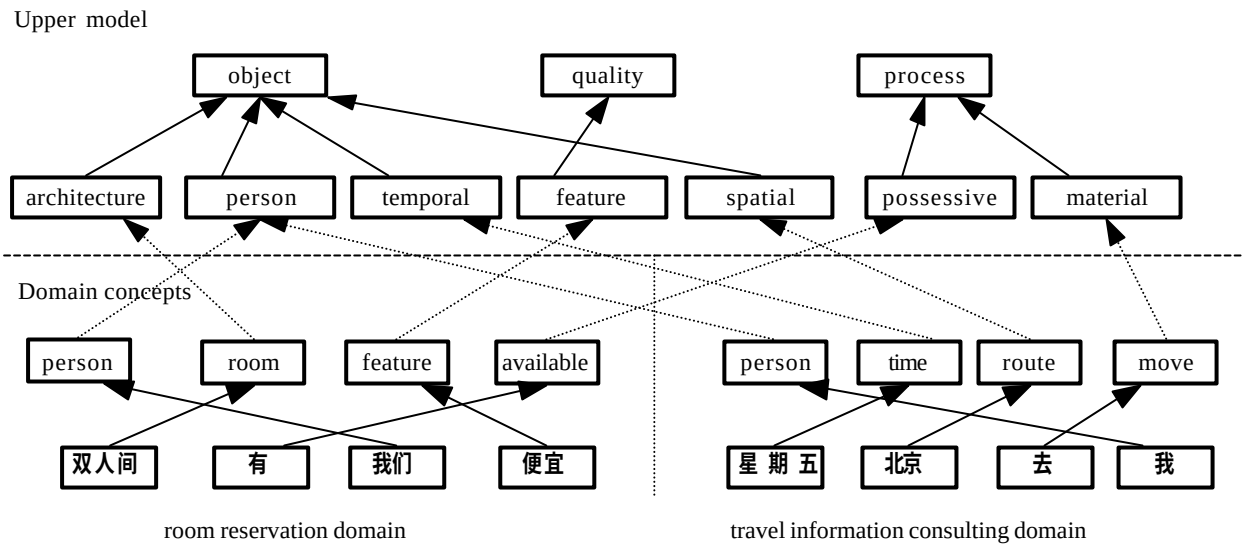

Figure 3: different domain concepts under the upper model

Secondly, the relationship between the upper model and the generation system can be seen from the input to generation system. Each input to the generation system is an intermediate representation. The focus of the input is the process type of the predicate and the types of its arguments. From the type each word belongs to, the grammar can determine the sentence type, the order among these constituents of the sentence and functional words. Therefore, the surface realizer generates a correct word string for it. Let's consider the two examples mentioned above. Their inputs to the generation system are shown in Figure 4. And the generated outputs are "**我星期五将要去北京**"(I will go to Beijing on Friday) and "**我们有便宜的双人间**"(We have cheap double rooms).

((process (type material) (lex "**去**"))
(tense future)
(Agent ((type person))
(person first)
(number single)))
Range ((type place) (lex "**北京**")))
(time ((type temporal) (lex "**星期五**")))

Input for travel domain: *I will go to Beijing on Friday.*

((process (type possessive)(lex "**有**"))
(possessor ((type person))
(person first)
(number plural)))
(possessed (((type object) (lex **双人间**))
((type size)(lex "**便宜**")))))

Input for hotel reservation domain: *We have cheap double rooms.*

Figure 4: Two inputs into the generation system

4. EXPERIMENTS AND CONCLUSION

Now our surface realizer has been tested in two systems. One is the dialogue system that provides travel information for customers. The other is a speech translation system whose domain is hotel reservation. In the dialogue system, our task of the realizer is response generation. In the speech translation system, the surface realizer is used to translate the source language expressed in the interlingua into Chinese. Tests show that the system embodies good adaptability to different application domains and good performances. 90% of the generated sentences are rated as grammatically and semantically correct [5].

The success in these two different systems should be ascribed to two factors. First, the input to the surface system is not affected by the preceding output and the domain because an upper model is used to interface the application domain and the general generation system. Second, we combined the template method and generation technology. This makes the system both efficient and flexible, which can tackle the spoken phenomena easily.

REFERENCES:

1. Michael Elhadad and Jacques Robin. Controlling Content Realization with Functional Unification Grammars. *Aspects of Automated Natural Language Generation.* pp89-104, 1992.
2. Zhuanglin Hu, Yongsheng Zhu, Delu Zhang. A Survey of Systemic Functional Grammar. *Hunan Education Publisher.* 1988. (in Chinese)
3. Stephan Busemann. Best-first surface realization. *In eighth International Natural Language Generation Workshop.* Sussex, pp101-110, 1996.
4. E.Pianta, M.Tovena. XIG: Generating from Interchange Format Using Mixed Representation. *AAAI' 99*
5. Hua Wu, Taiyi Huang, Chengqing Zong, Bo Xu. Chinese Generation in a Spoken Dialogue Translation System. *COLING*, Saarbrucken, Germany, 2000