Chinese Generation in a Spoken Dialogue Translation System

Hua Wu, Taiyi Huang, Chengqing Zong and Bo Xu
National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing 100080, China
E-mail: {wh, huang, cqzong, xubo}@nlpr.ia.ac.cn

Abstract:

A Chinese generation module in a speech to speech dialogue translation system is presented here. The input of the generation module is the underspecified semantic representation. Its design is strongly influenced by the underspecification of the inputs and the necessity of real-time and robust processing. We design an efficient generation system comprising a task-oriented microplanner and a general surface realization module for Chinese. The microplanner performs the lexical and syntactic choice and makes inferences from the input and domain knowledge. The output of the microplanner is fully instantiated. This enables the surface realizer to traverse the input in a top-down, depth-first fashion, which in turn speeds the whole generation procedure. The surface realizer also combines the template method and deep generation technology in the same formalism. Preliminary results are also presented in this paper.

1. Introduction

In this paper, we will present the core aspects of the generation component of our speech to speech dialogue translation system, the domain of which is hotel reservation. The whole system consists of five modules: speech recognizer, translator, dialogue manager, generator and speech synthesizer. And the system takes the interlingua method in order to achieve multilinguality. Here the interlingua is an underspecified semantic representation (USR). And the target language is Chinese in this paper.

Reiter (Reiter 1995) made a clear distinction between templates and deep generation. The template method is rated as efficient but inflexible, while deep generation method is considered as flexible but inefficient. So the hybrid method to combine both the methods has been adopted in the last few years. Busemann (Busemann 1996) used hybrid method to allow template, canned texts and general rules appearing in one formalism and to tackle the problem of the inefficiency of the grammar-based surface generation system. Pianta (Pianta 1999) used the mixed representation approach to allow the system to choose between deep generation technology and template method.

Our system keeps the surface generation module general for Chinese. At the same time, we can also deal with templates in the input without changing the whole generation process. If the attribute in the feature structure is “template”, then the value must be taken as a word string, which will appear in the output without modification. The surface generation module assumes the input as a predicate-argument structure, which is called intermediate representation here. And any input of it must be first converted into an intermediate representation.

The whole generation process can be modularized further into two separate components: microplanner and syntactic realizer. The microplanner is task-oriented. The input is an USR and the function of it is to plan an utterance on a phrase- or sentence-level. It maps concepts defined in the domain to a functional representation which is used by the syntactic generation components to realize an appropriate surface string for it. The functional description is made of feature structures, the attribute-value pairs. And the functional representation serves as the intermediate representation between the microplanner and the syntactic generator. The intermediate representation is fully instantiated. This enables the surface realizer to traverse the input in a top-down, depth-first fashion to work out a grammatically correct word string for the input, which in turn speeds the whole generation procedure. So our system use a task-oriented microplanner and a general surface realizer. The main advantage is that it is easy to adapt the system to other domains and maintain the flexibility of the system.

In this paper, section 2 gives a brief description of our semantic representation. Section 3 presents our method on the microplanning procedure. Section 4 describes the syntactic generation module. Section 5 presents the preliminary results of our generation system. Section 6 presents discussions and future work.

2. Semantic Representation

The most obvious characteristics of the semantic representation are its independence of peculiarities of any language and its underspecification. But it must capture the speaker's intent. The whole semantic representation has up to four components as shown in figure 1: speaker tag, speech act, topic and arguments.

The speaker tag is either "a" for agent or "c" for customer to indicate who is speaking. The speech act indicates the speaker's intent. The topic expresses the current focus. The arguments indicate other information which is necessary to express the entire meaning of the source sentence.

USR::= speaker: speech_act: topic: argument

Speaker::= a|c

Speech_act ::= give-information | request-information | …

Topic ::= (concept = attribute)^

Argument ::= (concept=attribute)|*

Figure 1 Underspecified Semantic Representation

Both the topic and arguments are made up of attribute-value pairs in functional formalisms. The attribute can be any concept defined in the domain of hotel reservation. The value can be an atomic symbol or recursively an attribute-value pair. The symbol "^" in the topic expression indicate that the expression can appears zero to one time, while The symbol "*" in the argument expression shows that the expression can appears zero to any times. And the attribute-value pairs are order free. Both topic and arguments are optional parts in the USR.

Let us consider a complex semantic expression extracted from our corpus. It is shown in Example 1:

a: give-information: (available = (room = (room- type = double))) : (price = (quantity =200&240,currency=dollor))| (1)

In Example 1, the speech act is *give-information,* which means that the agent is offering information to the customer. The topic indicates there are double rooms. The arguments list the prices of double rooms, which shows that there are two kinds of double rooms available. So the meaning of this representation is " We have two kinds of double rooms which cost 200 and 240 dollars respectively". From the USR, the kinds of rooms are not expressed explicitly in the format. Only from the composite value of the concept "price " can we judge there are two kinds of rooms because the price is different. This is only one example of underspecification, which needs inferences from the input and the domain knowledge.

3. The Microplanner

The input to our microplanner is the underspecified semantic representation. From the above semantic representation, we can see that it is underspecified because it lacks information such as predicate-argument structure, cognitive status of referents, or restrictive/attribute function of semantic properties. Some of the non-specified pieces of information such as predicate/argument structure are essential to generate a correct translation of the source sentence. Fortunately, much of the information which is not explicitly represented can be inferred from default knowledge about the specific domain and the general world knowledge.

The microplanner includes two parts: sentence-level planning and phrase-level planning.

The sentence planner maps the semantic representation into predicate argument structure. And the phrase planner maps the concepts defined in the domain into Chinese phrases.

In order to express rules, we design a format for them. The rules are represented as pattern-constraints-action triples. A pattern is to be matched with part of the input on the sentence level and with the concepts on the phrase level. The constraints describe additional context-dependent requirements to be fulfilled by the input. And the action part describes the predicate argument structure and other information such as mood and sentence type. An example describing a sentence-level rule is shown in Figure 2.

```
((speaker = a ) ( speech_act = give-information )( topic = available ) ( topic_value = room ));
//pattern
(exist(concept, 'price' ));        //constraint
( (cat = clause) ( mood = declarative)
( tense = present)    (voice = active)
  (sentence_type = possessive)
  (predicate ='有')
  (args = (((case = pos)
                (lex = #get(attribute, 'room' )))
              ((case = bel)
                    (cat = de)
                     (modifier =(#get(attribute, 'price' )))
                      (auxiliary = '的'))))
  (!optional: pre_mod = ( time = #get ( attribute,
'time'))));                      //action
```

Figure 2 Example Microplanning Rule

First, we match the pattern part with the input USR. If matched, the constraint is tested. In the example, the concept *price* must exist in the input. The action part describes the whole sentence structure such as predicate argument structure, sentence type, voice, mood. The symbol “#get” in the action part indicates that the value can be obtained by accessing the phrase rules or the dictionary to complete the structure recursively. The “#get” expression has two parameters. The first parameter can be “concept” or “attribute” to indicate to access the dictionary and phrase rues respectively. The second parameter is a concept defined in the domain. In the example, the “#get” expression is used to get the value of the domain concepts *room* and *price* respectively. The symbol “optional:” indicates that the attribute-value pair behind it is optional. If the input has the concept, we fill it.

After the sentence- and phrase-level planning, we must access the Chinese dictionary to get the part-of–speech of the lexicon and other syntactic information. If the input is the representation in Example 1, the result of the microplanning is shown in Figure 3.

```
(cat = clause)
( sentence_type =possessive)
(mood = declarative)
( tense = present)    (voice = active)
(predicate = ( (cat=vcm) (lex ='有')))
(args=((((case = pos)(cat = nct)(lex ='双人间'))
        ((case = bel)
              (cat =de)
              (modifier=((cat=mp)
                   (cardinal =((cat=nc)
                        (n1=((cat=num) (lex=’200’))
                        (n2=((cat=num)(lex="240'))
                   (qtf = ((cat=nct) ( lex ='美元'))))
              (auxiliary =(lex ='的'))))
```

Figure 3 Microplanning Result for Example 1

In the above example, *“cat”* indicates the category of the sentence, phrases or words. *“lex”* denotes the Chinese words. *“case”* describes the semantic roles of the arguments.

Target language generation in dialogue translation systems imposes strong constraints on the whole generation. A prominent problem is the non-welformedness of the input. It forces the generation module to be robust to cope with the erroneous and incomplete input data. In this level, we design some general rules. The input is first to be matched with the specific rules. If there is no rules matched, we access the general rules to match with the input. In this way, although the input is somehow ill-formed, the output still includes the main information of the input. An example is shown in (2). The utterance is supposed for the custom to accept the single room offered by the agent. But the speech act is wrong because the speech act “ok” is only used to indicate that the custom and the agent has agreed on one topic.

c: ok: (room = (room-type = single, quantity=1)): (2)

Although example (2) is ill formed, it includes most information of the source sentence. Our robust generator can produce the sentence shown in (3).

单人间行 (yes, a single room) (3)

4. Syntactic realization

The syntactic realizer proceeds from the microplanning result as shown in Figure 3. The realizer is based on a functional unification formalism.

In this module, we also introduce the template method. If the input includes an attribute-value pair which uses “template” as the attribute, then the value is taken as canned texts or word strings with slots. It will appear in the output without any modification. So we can embed the template into the surface realization without modifying the whole generation procedure. When the hybrid method is used, the input is first matched with the templates defined. If matched, the inputs will go to the surface realizer directly, skipping the microplanning process.

The task of the Chinese realizer is as follows:

- Define the sentence structure
- Provide ordering constraints among the syntactic constituents of the sentence
- Select the functional words

4.1 Intermediate Representation

The intermediate representation(IR) is made up of feature structures. It corresponds to the predicate argument structure. The aim is to normalize the input of the surface realizer. It is of considerable practical benefit to keep the rule basis as independent as possible from external conditions (such as the domain and output of the preceding system).

The intermediate representation 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, and so on. The obligatory arguments are slots of 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. An example is shown in Figure 4. The input is for the sentence “你们现在有单人间吗?” (Do you have single rooms now?). “agrs” and “opt” in Figure 4 represent obligatory arguments and optional arguments respectively.

```
((cat = clause)
( sentence_type =possessive)
(mood =yes-no)
( tense = present)    (voice = active)
(predicate = ( (cat=vcm) (lex ='有')))
(args=(((case = pos)(cat = pron)(lex ='你们'))
        ((case = bel) (cat =nct)(lex=’单人间’))))
(opt=(time=((cat=adv) (lex=’现在’)))))
```

Figure 4 Example Intermediate Representation

4.2 Chinese Realization

In the syntactic generation module, we use the functional unification formalism. At the same time, we make use of the systemic viewpoint of the systemic function 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.

In a spoken dialogue translation system, real-time generation is the basic requirement. As we see from the input as shown in Figure 3, the input to the syntactic generation provides enough information about sentence and phrase structure. Most of the information in the input is instantiated, such as the verb, the subcategorization frame and the phrase members. So the generation engine can traverse the input in a top-down, depth-first fashion using unification algorithm (Elhadad 1992). The whole syntactic generation process is described in Figure 5.

The input is an intermediate representation and the output is Chinese texts. 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. Unlike English, Chinese has no morphological markers for tenses and moods. They are expressed with functional words. Selecting functional words correctly is critical for Chinese generation.

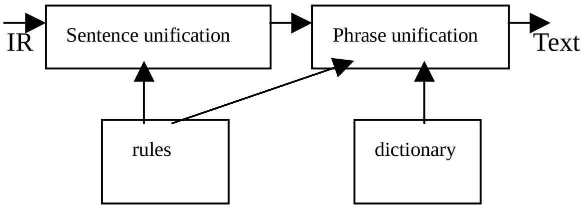

Figure 5 Steps of the Syntactic generator

The whole unification procedure is:

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

5. Results

The current version of the system has been tested on our hotel reservation corpus (Chengqing Zong, 1999). The whole corpus includes about 90 dialogues, annotated by hand with underspecified semantic representation. It contains about 3000 USRs. Now we have 23 speech acts and about 60 concepts in the corpus.

The generation module is tested on all sentences in the corpus. And 90% of the generated sentences are rated as grammatically and semantically correct. The other 10% are rated as wrong because the mood of the sentences is not correct. This is mainly caused by the lack of the dialogue context.

6. Discussion and Future Work

In spoken language translation systems, one problem is the ill-formed input. How to tackle this problem robustly is very important. At the microplanning level, we design some general rules. The input is first to be matched with the specific rules. If there is no rules matched, we access the general rules to match with the input. In this way, although the input is somehow ill-formed, the output includes the main information of the input. And at the surface realization level, we make some relaxation on tests to improve the robustness. E.g. obligatory arguments may be missing in the utterance. This can be caused by ellipsis in sentences such as the utterances "住三天" (stay for three days). We have to accept it as a sentence without the subject because they are acceptable in spoken Chinese and often appear in daily dialogues.

We are planning to further increase the robustness of the system. And if possible, we also hope to adapt our generation system to other domains.

Acknowledgements

The research work described in this paper is supported by the National Natural Science Foundation of China under grant number 69835030 and by the National '863' Hi-Tech Program under grant number 863-306-ZT03-02-2. Thanks also go to several anonymous reviewers for their valuable comments.

Reference:

Stephan Busemann. (1996) Best-first surface realization. In the Eighth International Natural Language Generation Workshop, Sussex, pages 101-110

Michael Elhadad and Jacques Robin. (1992) Controlling Content Realization with Functional Unification Grammars. Aspects of Automated Natural Language Generation. pp89-104

E.Pianta, M.Tovena. (1999) XIG: Generating from Interchange Format Using Mixed Representation. AAAI'99

Ehud Reiter. (1995) NLG vs. Templates. In the Fifth European Workshop on Natural Language Generation , Leiden,

Chengqing Zong, Hua Wu, Taiyi Huang, Bo Xu. (1999) Analysis on Characteristics of Chinese Spoken Language. In the Fifth Natural Language Processing Pacific Rim Symposium, pp358-362