Approach to Spoken Chinese Paraphrasing Based on Feature Extraction

Chengqing Zong[+]*, Yujie Zhang*,
Kazuhide Yamamoto*, Masashi Sakamoto* and Satoshi Shirai*
[+]National Laboratory of Pattern Recognition, Institute of Automation, CAS
P. O. Box 2728, Beijing 100080, China
cqzong@nlpr.ia.ac.cn
*ATR Spoken Language Translation Research Laboratories
2-2-2 Hikaridai Seika-cho, Soraku-gun, Kyoto 619-0288, Japan
{yujie.zhang, masashi.sakamoto, satoshi.shirai}@atr.co.jp

Abstract

This paper presents an approach to spoken Chinese language paraphrasing based on feature extraction and techniques of language generation. In this approach, an input utterance is first analyzed in terms of phrase structure, dependency of chunks, etc., by using multiple methods. Then, the main features of the input utterance are extracted, and the extraction results are represented by a frame. Finally, other possible expressions of the input are generated based on the analysis results by different methods. Preliminary results are shown in the paper.

1 Introduction

Although many approaches have been proposed to cope with spoken language phenomena and many strategies for translation have been developed, spoken language translation (SLT) systems still suffer from performance limitations. One of the key problems involves deciding how to robustly parse the input utterances. If we examine the techniques employed by human interpreters, we can see that paraphrasing is unavoidable at times. When an interpreter is unable to directly translate an utterance due to an ill-formed expression or an even worse problem, he or she may have to paraphrase the utterance into other expressions in his/her mind before translating the utterance.

To cope with this effect, Yamamoto *et al.* (2001) proposed the Sandglass SLT paradigm. This paradigm separates the complicated parsing procedure in the Sandglass system from the translation module and explains the meaning of an input utterance in the source language itself. Accordingly, it is possible to employ a simple transfer to convert the paraphrased input utterance into the target language. The main goal of the paraphrasing module is to make it easy to get correct translation, especially for complicated utterances.

In this paper, we present an approach to Chinese utterance paraphrasing based on feature extraction. In Section 2, the related works on paraphrasing are briefly reviewed, and the problems and our countermeasures are introduced. In Section 3, the implementation of an experimental system is described in detail. In Section 4, the experimental results are shown. Finally, Section 5 gives concluding remarks.

2 Problems in Paraphrasing and Our Countermeasures

2.1 Related Works on Paraphrasing

Chandrasekar *et al.* (1996) presented ways to simplify long and complicated sentences by a Finite State Grammar (FSG) based approach and the Supertagging (DSM) model. In their method, punctuation marks and relative pronouns are necessary to define a set of rules that map from the given sentence patterns to simpler sentences patterns. Unfortunately, in SLT systems there are no punctuation marks to use because all of the sentences that the system's paraphraser processes are from the system's speech recognizer, which does not generally provide punctuation marks. Furthermore, the Chinese language does not use relative pronouns to indicate articulation points.

Dras (1997) introduced several methods to represent paraphrases by using synchronous TAGs. These methods, however, closely depend on the synchronous TAGs and require a fairly well parsed syntactic structure of the input. But in SLT systems, it is usually very difficult to parse

an utterance into an adequate syntactic structure, especially when the input contains noisy words.

Boguslavsky *et al.* (2000) introduced synonymous paraphrasing of sentences, but did not address structure rewriting. McKeown (1983) described a paraphraser for a natural language question-answering system (CO-OP), but the system was only syntactic based. In addition, the method was found to have limitations in spoken language paraphrasing. Sato (1999) and Kondo *et al.* (2001) described methods to paraphrase Japanese technical papers' titles and simple Japanese sentences, respectively.

All of the research works mentioned above have provided us with very beneficial cues for paraphrasing the spoken Chinese language. Unfortunately, before we started our work, there was no reported work that specially addressed the Chinese language paraphrasing.

2.2 Problems and Countermeasures

In a paraphrasing system (see Fig. 1), the input and output should comply with the following policy:

The same language: the output should use the same language as the input.

The same semantics: the output should have (almost) the same meanings as the input.

Simplification: in general, the output should have simple and well-formed expressions, especially when the input is ill-formed.

Accordingly, the paraphraser has to tackle at least the following two problems:

- ✧ How to determine the meaning of the input utterance.
- ✧ How to express this meaning by using other expressions that are simple and well-formed.

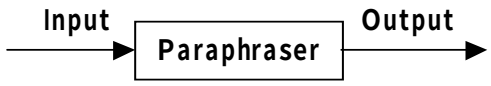

Figure 1. A Paraphraser

Like to develop a machine translation system, several candidate approaches can be used. The pattern based approach is one choice. This method is easy to realize and the speed is high, but the generality of the method is often limited. If the types of input utterances vary greatly and there are many unseen types of utterances, the performance of a system typically degrades. The statistical approach is another choice. In this method, paraphrasing is treated as just a procedure of translation based on the statistical approach. The only difference is that the paraphrasing is done within the same language rather than a translation between two different languages. Unfortunately, the statistical method needs very large-scale tagged corpora. Specifically, it is not practical to employ a costly statistic based paraphraser to rewrite input in a real-time SLT system.

Based on the analysis above, we proposed an approach to paraphrasing Chinese utterances based on feature extraction. The main ideas of the approach may be described as follows:

1) Segment the complex utterances into simple parts. Each part is separately paraphrased by using the following steps.
2) Jointly parse each separated part by multiple analyzers including a phrase parser, chunk dependency analyzer, and special chunk recognizer.
3) Extract the main features of the analyzing part, including the expression type (interrogative or declarative expressions, etc.), syntactic features, semantic features and so on.
4) Generate other possible expressions of each part by using different methods.

2.3 Why We Use the Approach

Our approach based on feature extraction is mainly grounded on the following points:

a) Almost all analysis results in the approach are useful and beneficial for the subsequent translation module in an SLT system. For example, the expression type, syntactic structures, and the relations among the chunks, are all necessary pieces of information for the translation module. This means that the paraphraser could not only provide alternative expressions of an input utterance to the translation module but also reduce much of the translation module's analysis work.

b) The possible expressions are generated under the guidance of analysis results. To a certain extent, the expressions may be generated with high correctness and well-formed structure.

c) The approach is not limited by any conditions. That is, the input utterances may be any possible expressions. This ensures that the approach is capable of processing the spoken

Chinese language.

d) The expressions are generated by different methods. The input is not only paraphrased based on the parsing results but also based on the other features. This helps the input to be paraphrased correctly even if the input is parsed incorrectly.

3 System Implementation

Based on the ideas presented above, we have implemented an experimental system for paraphrasing Chinese utterances in the domain of hotel reservation. This section describes the key points in implementation of the experimental system in detail.

3.1 Analyzer

In our approach, the analyzer includes six modules: (1) time chunk recognizer implemented by an FST (Finite State Transducer); (2) phrase parser employing PCFG (Probabilistic Context Free Grammar) rules; (3) chunk dependency analyzer achieved by an FST; (4) utterance type analyzer based on recognition rules; (5) keyword spotting analyzer achieved by an FST; and (6) tense analyzer also achieved by an FST. In this sub-section, we describe (1), (2) and (3).

• **Time Phrase and Quantifier Recognizer**

In the spoken Chinese language used in the hotel reservation domain, time phrases and quantifiers appear very frequently. According to our statistical results of 64,800 utterances in the domain of hotel reservation, 21.98% of utterances contain quantifiers or time phrases. Furthermore, the time phrases and quantifiers may act as different constituents in different contexts, such as an adverbial adjunct, object, or predicate. Accordingly, the time phrases and quantifiers of each input are recognized first by our analyzer before parsing.

The time phrase here mainly refers to a number related time expression, such as '下午三点半 (3:30 in the afternoon)' or '7月16号以前 (before July 16)'. Other time expressions are recognized by the phrase parser.

The number related time expression is recognized by an FST(*time*), which accepts only the following three types of words: (a) temporal noun (NT); (b) cardinal number (CD); and (c) the classifier (Mt). The set of the three types of words is signed as W_a. When a $Word_i \in W_a$ appears in the utterance under analysis, the FST(*time*) starts to work. In the case of $Word_i \notin W_a$, the FST(*time*) is stopped and the time chunk is marked.

Figure 2. FST(*time*)

According to our experimental results, the FST(*time*) processing of the time phrases and quantifiers was 65.6% completely correct, 17.2% partially correct but with no error, and 16.1% not processed. The error ratio was 1.1%, and the accuracy of the parser improved 6.5% by using the FST(*time*) (see Section 4).

• **Phrase Parser**

After the recognition of the time phrases, the input is segmented into n+1 parts by n (n is an integer and $n \geq 0$) time phrases. Each part is parsed by employing PCFG rules. Although there are large differences between the spoken Chinese language and the written Chinese language, we think these differences are mainly reflected at the sentence level, e.g., different orders of constituents containing redundant words in spoken Chinese expressions. Phrase construction in spoken Chinese and that in written Chinese follow the same policy. Accordingly, the PCFG rules employed in our system are directly extracted from the Penn Chinese Treebank (Xia, 2000). All of the rules comply with the condition of $\sum_i P(LHS \rightarrow \alpha_i)=1$. For example:

NN NN → NP, 1.00
MSP VP → VP, 0.94
MSP VP → NP, 0.06

In our system, the target of the parser is to recognize phrases rather than whole sentences.

• **Chunk Dependency Analyzer**

The chunk dependency is analyzed by the FST(*chunk*), which treats a predicate as the center of an analyzed part. The dependency between the predicate and other chunks are divided into nine types as shown in *Table* 1.

Since the task of the paraphraser is not translation, the dependency is not divided to produce fine details. Some chunks are distinguished by their positions, e.g., far/near adverbial adjunct and pre-/post-SPV.

In the system, the predicate is recognized first, and the recognizer gives the most plausible candidate as the predicate. The dependencies between the predicate and other chunks are analyzed by the following algorithm.

Step 1. If there is only one predicate candidate, search for the subject and adverbial adjunct at the left of the predicate candidate and then determine the complement, object, quantifier, etc., at the right of the candidate predicate.

Step 2. If there are n (n>1) predicate candidates VP_i (i =1 .. n), perform the following operations:

1) *Determine the subject and adverbial adjunct at the left of VP_1;*
2) *If VP_1 cannot take an object, treat the part after VP_1 as another processing unit, signed as PART-X;*
3) *If VP_1 is allowed to take an object, determine the object, complement, quantifier, etc., after VP_1 but before VP_2 and treat other parts as another processing unit PART-X;*
4) *For others cases: (a) VP_1 may take two objects; (b) VP_1 may take a clause as its object; (c) VP_1 may take a noun as its object but the noun (pivot word) can act as the agent of another following verb; and (d) VP_1 is the judgment verb. Determine the possible sentence according to different situations and treat the remainder of the input as another processing unit PART-X.*

Step 3. Treat PART-X as the input and repeat Step 1 and Step 2 until all chunks have been analyzed.

Step 4. Record all possible dependencies and fill the Frame (see Sub-section 3.2).

Table 1. Dependency Relations

Marks	***Types***
SUB	Subject
Q-NUM	Quantifier
COMP	Complement
D-OBJ	Direct object
I-OBJ	Indirect object
ADV	Adverbial adjunct
SPV	Sequential predicates[1]
PW	Pivot word[2]
CPW	Complement of pivot word

[1] For example, 他拿了钥匙上楼了。(Having taken the keys, he is going upstairs.)

[2] For example, 我选他当主席。 (I elect him to be chairman.)

3.2 Frame Representation

According to the description above, an input utterance is analyzed into *n* (*n*≥1) parts, and each part is mapped into a frame.

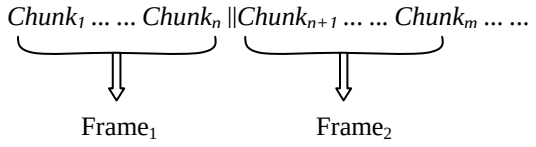

Figure 3. Chunks to Frame

A Frame consists of two parts, which we call Head and Body. The Head records the main features of the analyzed part, including the part's type, keywords, tense, and attribute. Type here refers to: (1) interrogative, (2) declarative, (3) greeting, or (4) simple reply. The attribute indicates the role that the part plays in the entire input. It may be a condition marked by Chinese words '如果 (if)', '要是 (suppose)', etc., or a reason clause marked by the Chinese words like '因为 (because)'.

Frame: HEAD: Type {Interrogative/…}
: Keywords {$Word_1$/Positon, …}
: Tense {Present/Past/…}
: Attribute {Condition/…}
BODY: Subject;
: $Adverbial_1$ / $Adverbial_2$ …;
: Predicate;
: Object1 → Sub-Body;
: Object2;
: Quantity;
: Complement;
: PW;
: CPW;
: SPV_1 / SPV_2;

Figure 4. Structure of Frame

PW, CPW, and SPV have the same meanings as in *Table* 1. If the object is a simple clause, the object is represented by a sub-Body that has the same structure as the main-Body. Some of the slots in the Frame may be null.

3.3 Generating Expressions

Based on the Frame representation, the possible expressions are generated by different methods. If the input is just a simple reply or a greeting phrase, like '好的 (OK) or '没问题 (no

problem)', there is no generation. Otherwise, the expressions are generated by using the following methods.

Method 1: Change the positions of adverbials. For example, the input: 昨天他在北京订了个单人间 (Yesterday he reserved a single room in Beijing.) (*I*-1) is parsed, and the phrases '昨天 (yesterday)' and '在北京 (in Beijing)' are respectively recognized as two adverbials. The positions of these two adverbials are changed in the output expressions as shown below:

(a) 在北京他昨天订了个单人间;
(b) 他昨天在北京订了个单人间;
(c) 昨天在北京他订了个单人间;
(d) 在北京昨天他订了个单人间.

Method 2: Generate the expressions by using the Head information in the Frame and also explain each constituent separately. For example, after the analysis of the input *I*-1, the Head information of the Frame is determined as:

<Type> = Declarative Expression;
<Keywords> = {他/1, 订/4, 单人间/7};
<Tense> = Past;
<Attribute> = Null;

According to the Head information, the expression is generated as: 他订了单人间. Then, the non-null constituents in the Frame are individually explained as follows:

(a) 他在北京订了单人间;
(b) 昨天他订了单人间;
(c) 他订了一个单人间.

Method 3: Change the interrogative expression by using fixed patterns. For example, an interrogative expression like '有没有X' may be changed into '有X吗', and '可以/能X吗' may be changed into '可不可以/可以不可以X' or '能不能X'. Here, 'X' is any word, phrase, or even a simple clause.

Method 4: Generate expressions by using phrase based patterns. Here, we assume that the input has already been parsed into phrases, so that phrase based patterns may be extracted to generate other expressions of the input, e.g., from the sentence '我想要个大点儿的房间 (I want a bigger room.)', we may extract the following patterns:

我想要个 VA 的 NP
⇒ 最好给我个 VA 的 NP
⇒ 能不能给我个VA 的 NP
⇒ 可以给我个VA 的 NP吗

4 Experimental Results

At present, the experimental system employs 244 PCFG rules, 43 sentence type recognition rules, 14 rules for complicated utterance segmentation at the shallow level, and a dictionary with 6,500 Chinese words extracted from 64,480 utterances. Some test results are presented in this section.

4.1 Parsing Results

Here, we mainly test the effect of the time phrase and quantifier recognizer on the phrase parser by using 54 utterances, in which 89 time phrases or numerals are contained. *Table* 2 shows the parsing results both when the time and numeral phrases are pre-processed (Case 1) and not pre-processed (Case 2).

Table 2. Parsing Results

	Case 1	Case 2
Output Phrase	164	148
Correct	147	123
Correct Ratio	89.6%	83.1%

From the table we can see that the parsing accuracy improved 6.5% by using the time and numeral phrase recognizer.

4.2 Dependency Analysis Results

The dependency analyzer was tested by using 100 utterances, which include 107 simple sentences. The analysis results are divided into two types. The first type is the completely correct results. If any relation is analyzed incorrectly, the result belongs to the second type. The test results show that 61 simple sentences were analyzed complete correctly. This number amounts to about 57% of the total input. On the other hand, 46 simple sentences were analyzed incorrectly, which was about 43% of the total input. The wrong results could be attributed to three reasons: (a) wrong parsing results, (b) wrong word segmentation, and (c) wrong dependency analysis. *Table* 3 gives the distribution of the three error types. The worse parsing result is clearly the main cause of incorrect dependency analysis.

Table 3. Errors Distribution

	(a)	(b)	(c)
Number	38	4	4
Ratio (%)	82.6	8.7	8.7

4.3 Paraphrasing Results

The entire paraphrasing system was tested by using the same 100 input utterances mentioned above. Sixty simple sentences were not paraphrased, and 47 simple sentences were paraphrased into 90 expressions. That is, one simple sentence was paraphrased into 1.91 possible expressions on average. The paraphrased results are divided into three types: (A) the results are correct and well expressed, (B) the results are understandable and acceptable, and (C) the results are wrong. *Table* 4 presents the three types of paraphrased results.

Table 4. Paraphrasing Results

	(A)	(B)	(C)
Number	56	14	20
Ratio (%)	62.2	15.6	22.2

Table 4 shows that about 77.8% of the paraphrased results were good or acceptable. The wrong results were mainly caused by dependency analysis errors.

5 Conclusion

Chinese paraphrasing is a new research effect, although many of the techniques employed in our approach are not new. Although the performance of our experimental system is not yet satisfactory, the preliminary results have given us confidence in our development of a practical paraphraser for SLT systems. We believe that the ideas proposed in this paper are beneficial not only for paraphrasing tasks but also for robust spoken language understanding, information extraction, and other related tasks.

However, the approach faces many complicated problems, including robust parsing, dependency analysis, and natural language generation. The following two problems remain for further research: (1) How to judge whether the generated results are simpler and better formed than the input at the very least, they should not be worse than the input; and (2) How to rank the generated results and then output the 'best' result to the transfer (translation) module.

6 Acknowledgements

The authors specially thank professor Shiwen Yu, professor Fuji Ren, Dr. Kiyonori Ohtake, and Ms. Lan Yao for their very useful help.

References

Boguslavsky, Igor, Nadezhda Frid et al. 2000. Creating a Universal Networking Language Module within an Advanced NLP System. *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, 83-89.

Chandrasekar, R., Christine Doran, and B. Srinivas. 1996. Motivations and Methods for Text Simplification. *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, 1041-1044.

Dras, Mark. 1997. Representing Paraphrasing Using Synchronous TAGs. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, 516-518.

Kondo, Keiko, Satoshi Sato, and Manabu Okumura. 2001. Paraphrasing by Case Alternation (in Japanese). *Information Processing Society of Japan (IPSJ) Journal*. 42(3), 465-477.

Mckeown, Kathleen R. 1983. Paraphrasing Questions Using Given and New Information. *American Journal of Computational Linguistics*, 9(1): 1-10.

Sato, Satoshi. 1999. Automatic Paraphrase of Technical Papers' Titles (in Japanese). *Information Processing Society of Japan (IPSJ) Journal*. 40(7): 2937-2945.

Xia, Fei. 2000. The Part-of-Speech Tagging Guidelines for the Penn Chinese Treebank (3.0). http://www.ldc.upenn.edu/ctb/.

Yamamoto, Kazuhide, Satoshi Shirai, Masashi Sakamoto, and Yujie Zhang. 2001. SANDGLASS: Twin Paraphrasing Spoken Language Translation. *Proceedings of the 19th International Conference on Computer Processing of Oriental Language (ICCPOL)*, 154-159.

Zhou, Ming. 1999. J-Beijing Chinese-Japanese Machine Translation System (in Chinese). *Proceedings of National 5th Joint Symposium of Computational Linguistics*, 312-319.

Zhou, Qiang. 1999. The Chunk Parsing Algorithm for Chinese Language (in Chinese). *Proceedings of National 5th Joint Symposium of Computational Linguistics*, 242-247.

Zong, Chengqing, Yujie Zhang, Kazuhide Yamamoto, Masashi Sakamoto and Satoshi Shirai. Paraphrasing Chinese Utterances in Spoken Language Translation System (in Chinese). To appear in proceedings of *International Conference of Chinese Computing*. Nov., 2001. Singapore.