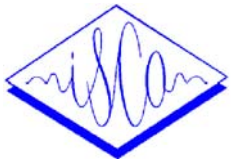

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

Error Recovery for Robust Language Understanding in Spoken Dialogue Systems

Tung-Hui Chiang and Yi-Chung Lin

Advanced Technology Center,
Computer and Communications Laboratories,
Industrial Technology Research Institute,
Chutung, Taiwan 310, R.O.C.
{thchiang,lyc}@atc.ccl.itri.org.tw

10.21437/Eurospeech.1999-443

ABSTRACT

In this paper, we proposed an example-based approach aiming at recovering ill-formed inputs to improve robustness of spoken dialogue systems. In this approach, a treebank, which contains example sentences and their correct parse trees, is used to provide clues for fixing the errors of ill-formed inputs. Particularly, the proposed error recovery method is suitable for spoken dialogue application because of computationally efficiency. In addition, when evaluated in a Mandarine spoken dialogue system, the proposed method has shown to improve the system's understaning rate very significantly.

Keywords: robust parsing, error recovery, spoken dialogue system

1 INTRODUCTION

Spoken language has being viewed as a very important, natural and efficient way to human-machine interfaces. In general, a spoken language system requires a pre-defined grammar to analyze and understand what people said. However, due to speech recognition errors and infrequent language usage, a dialogue system often faces to the sentences with errors in the aspect of the system grammar. To make dialogue systems practical in use, the errors in the ill-fomed sentences should be recovered.

In the literature, the approaches to error recovery are usually formulated to find the fittest parse tree among all alternatives that could be generated by the system grammar 0. However, the system grammar, usually defined for analysis purpose, tends to be over-generated in many cases. That is, such approaches might produce sentences that are syntactically well-formed, but semantically meaningless. Furthermore, the number of structures which can be generated by the system grammar is generally too large to search exhaustedly, some heuristic rules should be applied to reduce computational cost. However, those rules are usually system-specific and not easy to be re-used by other systems.

Motivated by the above concerns, the method proposed in this paper first attempts to improve the robustness of a spoken dialogue system by recovering the ill-formed inputs. In this approach, a treebank which contains example sentences and their correct parse trees is used to provide the clues for fixing the errors of inputs. Since the domains of spoken dialogue systems are usually limited, a small number of examples can offer good coverage on practical language usage. Therefore, the effort of corpus annotation can be minimized. Furthermore, based on real sentences in use, the proposed approach can avoid producing meaningless sentences.

In our approach, the best sentence hypothesis suggested by the speech recognizer is first analyzed according to the system grammar. If the hypothesis is ill-formed, the parser would express the result with a forest of partial parses [2]. Afterwards, the forest is fitted to example parse trees with a dynamic programming procedure. The example tree with minimal distance which is defined as a function of the prori probability of example tree and the number of editing operations [3], is chosen to patch the forest of partial parses to a well-formed full parse. We have implemented the error recovery method in a Mandarine spoken dialogue system. The experiment results have shown that the proposed method can improve the system's understaning rate substantially.

2 LANGUAGE UNDERSTANDING WITH ERROR RECOVERY

Figure 1 gives an overview of our spoken dialogue system. In this system, the speech recognition component transcribes user's utterances into text, and then the language understanding component analyzes the meanings of the text, representing the results using a data structure called semantic frame [4]. Then, the dialogue manager determines the appropriate responses based on both the input semantic frame and the conversational contexts. The responses, including actions of asking

more constraints, confirming user's requests, providing suggestions, etc., are finally presented to the user in graphics, text, and voice by using the language generation component and the text-to-speech synthesizer.

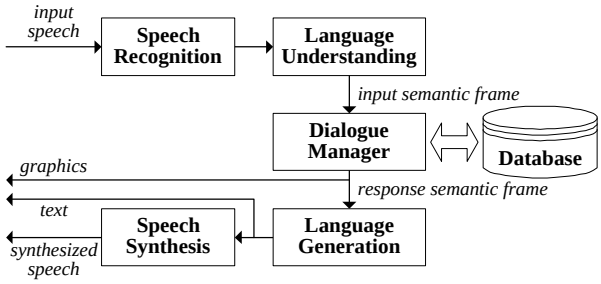

Figure 1. Block diagram of a spoken dialogue system.

In particular, more detailed processing units of the language understanding component are shown in Figure 2. The input word sequence provided by the speech recognizer is tagged with part-of-speech (POS) tags using a standard POS tagger [4]. The structures of the tagged sequences are analyzed by the robust parser. If the input sentences are ungrammatical, the parser, instead of giving the full parses, will provide a forest containing possible partial parses. Afterwards, by consulting the example treebank, the error recovery module would "*predict*" the most appropriate parse tree from which the forest may be derived. Finally, the meanings of user's intention is analyzed by the semantic interpreter to form a semantic frame.

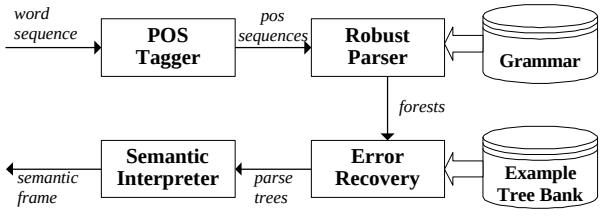

Figure 2. Language understanding component.

2.1 Part-of-speech Tagger

The trigram model of part-of-speech tagging is adopted, where given a word sequence $w_1, w_2, \Lambda\ , w_n$, the likelihood (namely *lexical score*) of choosing $c_1, c_2, \Lambda\ , c_n$ as the corresponding POS sequence is defined as:

$$SC_{lex}(c_1^n \mid w_1^n) = -\ln\left\{\prod_{i=1}^{n} P(c_i \mid c_{i-1}, c_{i-2}) P(w_i \mid c_i)\right\}. \quad (1)$$

All possible POS sequences together with their lexical scores, defined in Equation (1, are passed to the robust parser for structure analysis

2.2 Robust Parser

The CYK parsing algorithm is adopted in our system because it can efficiently parse a sentence into all its possible partial parses. If the input sentence is grammatical, the parser simply use the stochastic context-free grammar approach to choose the preferred full parse. On the other hand, instead of using heuristic rules (such as preferring the longest phrase), we proposed an n-gram assembling model [2] to select the probable forest of partial parses for an ungrammatical sentence. The syntactic score of a forest F parsed from the POS sequence c_1^n is formularized as follows.

$$SC_{syn}(F \mid c_1^n) = -\ln\left\{\prod_{j=1}^{m} P(H_j \mid H_{j-n+1}, \Lambda\ , H_{j-1}) \times \prod_{A \to \alpha \in F} P(\alpha \mid A)\right\} \quad (2)$$

where m denotes the number of partial parses in the forest, H_j denotes the root of the j-th partial parse, and $A \to \alpha$ denotes a grammar rule in the forest. Currently, in out system n is set to 3 (i.e., using a trigram model). The forest with the highest integrated score (defined as sum of the lexical and syntactic scores) would be chosen as the candidate for error recovery processing.

2.3 Error Recovery Unit

The basic idea of the proposed error recovery method is to repair the selected forest to the most similar structure in the example treebank. However, comparison of tree structures is usually computationally expensive. To make error recovery computationally feasible, we use linear matching [3] instead of sturcture matching in our system. The proposed method first projects all structures into string forms, and then finds the best matched projection with the dynamic programming algorithm.

The projection of a forest is defined to be the string of the root nodes of the partial parses in the forest. On the other hand, the projection of a example tree is defined as a string of the constituent nodes of the root node. Then, an example tree will be selected to patch the forest according to the following scoring function.

$$SC_{ER}(E \mid F) \equiv w_{op} \cdot (-\ln P(\Psi(E))) + (1 - w_{op}) \cdot (-dist(\Psi(E), \Psi(F))), \quad (3)$$

where E is an example tree, F is the forest to be patched, $\Psi(E)$ and $\Psi(F)$ denote the projections of E and F respectively, $dist(\cdot,\cdot)$ is a string distance function, and w_{op} is a weighting constant. In our system, the string distance function gives the number of edit operations [3]. The role of the weighting constant w_{op} is to make compromise between the critera of prefering a frequent

example and preferring a small edit distance. The adequate value of w_{op} can be obtained by automatically learning [5]. However, in our current system, w_{op} is set to a small positive value to make sure that the criterion of preferring a small edit distance has higher priority than the criterion of prefering a frequent example. In other words, our system will chose the example with minimum distance. For the example trees of the same distance to the forest, the one with highest prior probability $P(\Psi(E))$ will be selected.

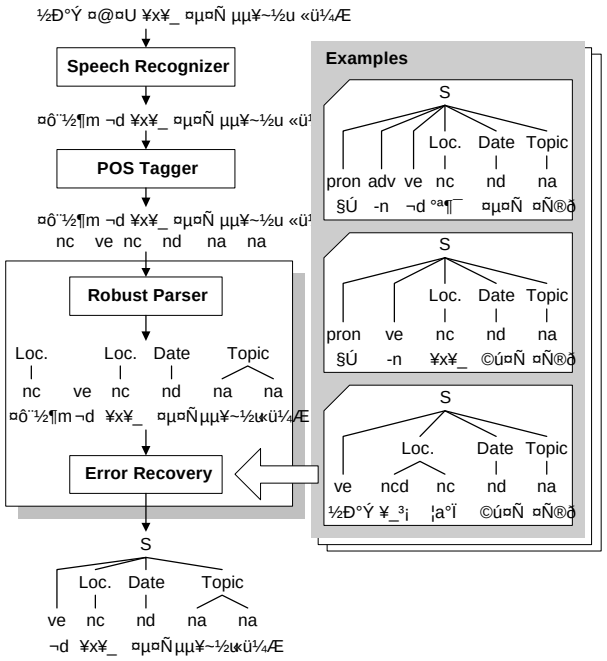

Figure 3. An example of the error recovery process

Figure 3 gives a simplified example for explaining the error recovery process. The input sentence is parsed as a forest whose maximum projection is represented as $\Psi(F)$={**Loc, ve, Loc, Date, Topic**}, where **ve** stands for a verb class (message-query); the non-terminal symbols **Loc**, **Date**, **Topic** denote the concepts of "*location*", "*date*", and "*topic of interest*", respectively. The projections of three example trees (from top to bottom) are $\Psi(E_1)$ ={**pron, adv, ve, Loc, Date, Topic**}, $\Psi(E_2)$={**pron, ve, Loc, Date, Topic**}, $\Psi(E_3)$={**ve, Loc, Date, Topic**}. The distances of the forest's projection to the examples' projections are 2 (one substitution, one insertion), 1 (one substitution) and 1 (one deletion), respectively. Because $\Psi(E_3)$ appears more frequently than $\Psi(E_2)$ in the tree bank, the third example tree is the best choice to patch the forest. In this case, the leftmost partial parse of the forest is deleted, as shown in the bottom of Figure 3.

2.4 Semantic Interpreter

The semantic interpreter [4] first performs structure normalization for the (recovered) parse tree. Next, the interpreter determines user's intention, semantic concepts (called *cases*) of constituents and word senses of content words in the normalized structure. Finally, the results are represented with a frame-based data structure called semantic frame. Interested readers are referred to [4] for details. Here, we show the semantic frame of the sentence " (*I would like to know today's ultraviolet index of Taipei City*)" in Figure 4.

[
Intention: *Verb_Query*
Location: ¥ ¥ ¥ «*(Taipei city)*
Date: ¤ ¤ Ñ*(today)*
Topic: µ ¼ ½ u« ü¼ Æ*(ultraviolet index)*
]

§ Ú ¢Q ¾Ð ¥ ¥ ¤ ¤ Ñ ¤ ¼ ½u ü¼Æ
(*I would like to know today's ultraviolet index of Taipei City.*)

Figure 4. An example of semantic frame

3 EXPERIMENT AND DISCUSSIONS

In our experiment, 239 example sentences are collected and manually parsed to build the treebank. For evaluation, a total of 477 utterances of user requests are used for testing. Among the testing utterances, 96 utterances are transcribed to ill-formed sentences by the speech recognizer. Once the parser identifies an ungrammatical input, it will try to fix it by consulting the treebank. Afterwards, the recovered parse tree is passed to the semantic interpreter to find user's intention. Since user's intention is represented by semantic frame in our system, the performance of the proposed approach is evaluated in term of the accuracy rates of semantic frame and the corresponding slots.

Table 1 lists the performances of language understanding with and without using the proposed error recovery approach. The results show that the proposed error

	Frame accuracy	Slot precision	Slot recall
Without error recovery	30%	72.4%	71.2%
With error recovery	46%	82.3%	78.8%
Error reduction rate	23%	35.9%	26.4%

Table 1. The performances on understanding users' intentions without and with error recovery.

	Number of errors in semantic slots		
	Insertion	Deletion	Substitution
Without error recovery	13	17	51
With error recovery	8	18	32

Table 2. The numbers of different kinds of errors in semantic slots without and with error recovery.

recovery approach can significantly improve the accuracy in understanding users' intentions. When the speech recognizer incorrectly transcribe users' utterances to ill-formed sentences, only 30% of users' intentions can be correctly realized without fixing the speech recognition errors. When the proposed error recovery method is applied, 46% of users' intentions are understood correctly. Improvements are also observed on the precision rate and recall rates of semantic slots.

More detailed experimental results are listed on Table 2. The results show that the numbers of insertion and substitution errors are significantly reduced. But, the number of deletion errors is almost unchanged. This phenomenon is due to the particular relationship between the error types of semantic slots and the error types of recognition errors. The major errors of insertions and substitutions of semantic slots come from the insertion errors caused by speech recognition. However, the deletion errors of semantic slots come mainly from deletion and substitution errors. The following gives a more detailed explanation.

Take the utterance "tell me the weather of Taipei please" as an example. If the speech recognizer inserts a city name "Penghu" at the end of the sentence, it makes the semantic interpreter substitute the semantic slot of location from "Taipei" to "Taipei and Penghu". This kind of errors can be easily fixed just by ignoring the insertion keywords. On the other hand, the deletion errors of semantic slots come mainly from deletion and substitution errors caused by speech recognition. This kind of errors is hard to be recovered because the correct information is missed or distorted after speech recognition. To correct such errors, the speech recognition component and the language understanding component should be tied more tightly. An interface of top-N sentence hypotheses or a word graph between speech recognition and language understanding would be helpful. We plan to study this issue in the near future.

4 CONCLUSIONS

In this paper, we proposed an example-based approach for recovering ill-formed inputs to improve the robustness of spoken dialogue systems. In our approach, a treebank which contains example sentences and their correct parse trees is used to provide the clues for fixing the errors of ill-formed inputs. The proposed error recovery method is particularly suitable for spoken dialogue application because of computationally efficiency. Furthermore, our approach avoids producing unreadable sentences because the examples are real sentences in use. Significant improvements are observed after we integrate the error recovery mechanism into our Mandarin spoken dialogue system.

ACKNOWLEDGEMENT

This paper is a partial result of the projects no. 3SH2100 and 3P11200 conducted by ITRI under sponsorship of the Ministry of Economic Affairs, R.O.C.

REFERENCES

[1] Lee, K. J., C. J. Kweon, J. Seo, and G. C. Kim., "A Robust Parser Based on Syntactic Information," in *Proc. of the 7th Conference of European Chapter of the Association for Computational Linguistics*, pp. 223–228, 1995.

[2] Lin, Y.-C. and K.-Y. Su, "A Level-synchronous Approach to Ill-formed Sentence Parsing," in *Proc. of ROCLING X International Conference*, pp. 89-108, Taipei, Taiwan, 1997.

[3] Ristad, E. S. and P. N. Yianilos, "Learning String-Edit Distance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 5, pp. 522-532, 1998.

[4] Chiang, T.-H. and K.-Y. Su, "Statistical Models for Deep-Structure Disambiguation," in *Proc. of Fourth workshop on Very Large Corpora*, pp. 113-124. 1996.

[5] Chiang, T.-H., Y.-C. Lin and Keh-Yih Su, "On Jointly Learning the Parameters in a Character-Synchronous Integrated Speech and Language Model," *IEEE Trans. On Speech and Audio Processing*, Vol. 4, No.3, pp. 167-189, 1996