Introducing Utterance Verification in Spoken Dialogue System to Improve Dynamic Help Generation for Novice Users

Kazunori Komatani Yuichiro Fukubayashi Tetsuya Ogata Hiroshi G. Okuno

Graduate School of Informatics
Kyoto University
Yoshida-Hommachi, Sakyo, Kyoto 606-8501, Japan
{komatani,fukubaya,ogata,okuno}@kuis.kyoto-u.ac.jp

Abstract

A method is presented that helps novice users understand the language expressions that a system can accept, even from unacceptable utterances made that may contain automatic speech recognition errors. We have developed a method that dynamically generates help messages, which can avoid further unacceptable utterances from being made, by estimating a users' knowledge from their utterances. To improve the accuracy of the estimation, we developed a method to estimate a user's knowledge from utterance verification results. This method estimates whether a user knows an utterance pattern that the system considers acceptable, and suppresses useless help messages from being generated.

1 Introduction

We have developed a user friendly spoken dialogue system, even for novice users, that generates help messages dynamically (Fukubayashi et al., 2006). Since novice users do not necessarily know the language expressions that can be accepted by a system, help messages need to be generated to instruct them of acceptable expressions. Such messages can be generated by estimating each user's knowledge of the system through their interactions with the system.

Users often make out-of-vocabulary or out-of-grammar utterances. This is unavoidable because of the characteristics of speech, that is, speech interfaces do not provide enough affordance (Norman, 1988). A graphical user interface (GUI) provides users with a clear representation of the kind of input required by the system; however, users have difficulty in understanding the input required when speech interfaces are used. Unfortunately, the range of language expressions a spoken dialogue system can handle is inherently limited. Even when a statistical language model is used in automatic speech recognition (ASR) and large numbers of expressions can be handled, patterns of language expressions are limited in language understanding (LU) or dialogue management (DM) components. This problem is compounded when novice users do not know what utterances can be accepted by a system. This is the very situation in which help messages should be generated, but ASR results for this type of utterance are unreliable because the utterances are often considered unacceptable. Even from such erroneous ASR results, systems have to estimate a user's knowledge accurately.

We addressed this problem by introducing an utterance verification technique. Since utterance verification does not use ASR results but uses acoustic scores of ASR, information about a user's utterances can be obtained, even from utterances that are considered unacceptable. By using its result, we can measure how close an utterance is to the grammar of a system.

Several studies have focused on generating help messages (Gorrell et al., 2002) (Hockey et al., 2003). Since they did not consider changes in the user's knowledge during the dialogue, the same help messages were generated when the same speech recognition results were obtained. Furthermore, these

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 202–205,
Antwerp, September 2007.

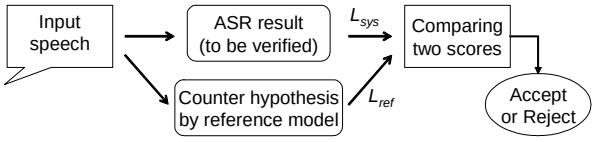

Figure 1: Overview of utterance verification

studies used ASR results from a "secondary" statistical language model when the primary grammar-based ASR failed. We ensured a user's knowledge state was updated appropriately, even when their utterances did not perfectly match utterances expected by a system developer, by detecting them as out-of-grammar utterances.

2 Generating Dynamic Help Messages Using Utterance Verification

2.1 Utterance Verification Using Differences in Acoustic Likelihoods

Utterance verification is generally performed by comparing log-scaled scores between an ASR output to be verified and a counter hypothesis based on a reference model. Same acoustic models were used in both recognizers. An outline of this process is shown in Figure 1. We denote the acoustic likelihood of the reference recognizer as $L_{\rm ref}$, the acoustic likelihood of the target-domain recognizer as $L_{\rm sys}$, the duration of the utterance as T (sec.), and the threshold as θ_{score}. The verification is assessed by using the following equation:

$$\begin{cases} S = (L_{\rm ref} - L_{\rm sys})/T & < \quad \theta_{score} \quad (Accept) \\ & \geq \quad \theta_{score} \quad (Reject) \end{cases} \tag{1}$$

The difference in the scores between the two recognizers indicates how close the user's utterance is to the system's grammar, which provides different information from conventional confidence measures (CMs) that are calculated for each word (Komatani and Kawahara, 2000).

Various studies have investigated the different reference models used in utterance verification (Sukkar et al., 1995; Kawahara et al., 1998). We used a simple utterance verification method in which the difference between log-scaled acoustic scores of the two recognizers is calculated. This is because we are now focusing on how utterance verification results can be used in spoken dialogue systems. The utterance verification method itself can be replaced if more accurate methods become available.

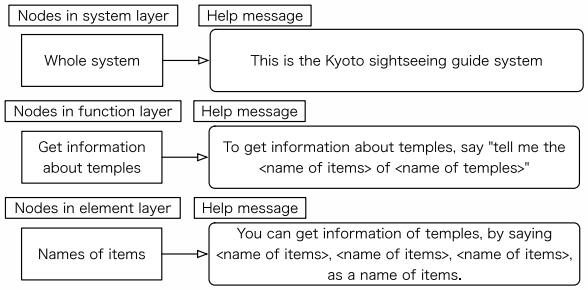

Figure 3: Example of help messages for each node

2.2 Generating Dynamic Help Messages

We have developed a method to generate help messages that fills the gap between a user's knowledge and the actual structure considered acceptable by the system. A detailed explanation of an algorithm we developed has been presented in our previous paper (Fukubayashi et al., 2006). The following is a concise explanation of that algorithm.

A *domain concept tree* was designed to represent a concept structure of a system, which represents the hierarchical layers of the target domain. This tree consists of four layers: "system", "function", "element", and "content word". The domain concept tree of the Kyoto sightseeing guide system is shown in Figure 2 as an example. *Known degrees* of each node in the domain concept tree are estimated. The degree represents how well a user knows a concept corresponding to the node. Known degrees are updated after each user's utterance; for example, a known degree of a node in the content word layer is increased if the content word is contained in an ASR result, and the effect is propagated to the known degrees of its ancestors. Lastly, a help message is generated after searching for a node having the lowest known degree. The message is generated by using templates, as shown in Figure 3.

The domain concept tree was updated on the basis of the ASR results from the user's utterances and the generated help messages. A user's knowledge, however, must be updated correctly, even when the content words in the user's utterances contain ASR

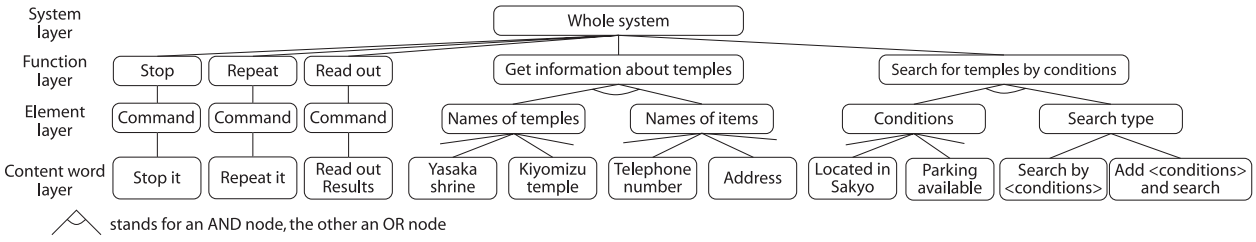

Figure 2: Domain concept tree of Kyoto sightseeing guide system

errors. For example, if a user says "Please tell me *an approach* to Yoshida Shrine." Even if *an approach* is unknown to the system, the system should be able to estimate that this user knows the utterance pattern considered acceptable but that they do not know the content words considered acceptable by the system.

2.3 Updating Domain Concept Tree Using Utterance Verification Results

Two loss functions are defined as:

$$cost_1 = (FA + SErr)/2$$
$$cost_2 = (FA + SErr + (1 - Acc))/3$$

Ratio FA (false acceptance) is the ratio of incorrectly accepted utterances that should be rejected, and $SErr$ (slot error) is the ratio of correct utterances that are not accepted (Komatani and Kawahara, 2000). We also calculated the accuracy of the language understanding by using the following equation: $Acc = (N - D - S - I)/N$, where N is the number of correct content words, and D, S and I are the numbers of deletion, substitution, and insertion errors, respectively. The accuracy of utterance verification is represented as $cost_1$, and $cost_2$ takes the accuracy of the language understanding results into consideration.

We defined two thresholds, θ_1 and θ_2, to minimize $cost_1$ and $cost_2$, respectively. Therefore, thresholds θ_1 and θ_2 focus on whether a whole utterance is in-grammar and whether content words in an utterance are correct. As a result, user utterances can be classified into one of the following three categories:

1. $S < \theta_1$: in-grammar and correct language understanding result,
2. $\theta_1 \leq S < \theta_2$: in-grammar but incorrect language understanding result, and
3. $S \geq \theta_2$: out-of-grammar and incorrect language understanding result.

The known degrees can be updated based on the above classification. When $S < \theta_1$, known degrees are ordinarily updated on the basis of the content words in the ASR results. Utterances whose S is greater than θ_2 are normally rejected. When $\theta_1 \leq S < \theta_2$, this utterance is estimated to be an in-grammar utterance, but its language understanding result seems to be incorrect. That is, the utterance seems to match to the system's grammar, even though it may contain incorrect content words. Then, the known degrees of the nodes in the function layer increase. This update allows the system to acquire information as to the user's knowledge regarding the system's grammar for the domain concept tree, even for utterances whose content words are not correctly recognized, and consequently suppresses unnecessary help messages from being generated regarding grammars.

3 Experimental Evaluation

We used dialogue data collected from users when they operated the Kyoto sightseeing guide system (Fukubayashi et al., 2006) in our evaluation. The dialogue data consists of 1,518 utterances from 12 subjects, none of which had previously used the system. Therefore, many user utterances were outside the range considered acceptable by the system and caused many ASR errors.

We used a grammar-based ASR engine, Julian[1], that has a vocabulary of 673 words. The average accuracy of the ASR was 42.9%. As a reference model for utterance verification, we used the outputs from a speech recognizer, Julius, which is based on statistical language models. Its language model was trained using newspaper articles and has a vocabulary of 20,000 words. The same acoustic model was

[1] http://julius.sourceforge.jp/

Table 1: Classification of utterances when setting θ_1 to 75 and θ_2 to 125

Correct answer of UV	LU results	$S < \theta_1$	$\theta_1 \leq S < \theta_2$	$S \geq \theta_2$
Accept	Correct (100%)	$454^{\dagger}$	$50^{(\star)}$	8
Accept	Some errors (<100%)	84	$29^{(\star\star)}$	34
Accept	No output	28	$13^{(\star\star\star)}$	23
Reject	Some errors (insertion error)	158	104	$185^{\ddagger}$
Reject	No output (correct rejection)	166	86	98

UV: utterance verification, LU: language understanding

used in both recognizers.

Loss functions $cost_1$ and $cost_2$ were minimized when θ_1 was 125 and θ_2 was 75. We counted the number of utterances in each category. The results are listed in Table 1. We compared the results from Table 1 with the results when the utterances with $S \geq \theta_2$ were simply rejected in which the performance was optimized by considering both the utterance verification and language understanding results. A value denoted by $^{(\star\star)}$ in Table 1 represents utterances that were incorrectly accepted despite some errors being contained in their language understanding results, and a value denoted by $^{(\star\star\star)}$ represents utterances that were rejected because no language understanding result was obtained. Therefore, the system obtained new information indicating that a user knows about the expression considered acceptable by the system, from 42 utterances which are denoted by $^{(\star\star)}$ and $^{(\star\star\star)}$. This enables the system to correctly update the domain concept tree at the function layer, even when correct language understanding results are not obtained.

In this case, correct language understanding results will be incorrectly rejected for 50 utterances denoted by $^{(\star)}$. Therefore, the performance needs to be improved. One reason for the inadequate performance is that the utterance verification algorithm used was very simple: only the differences in acoustic scores between the two recognizers were used. The performance of the classification is currently being improved, especially for short utterances whose differences in acoustic scores were not large enough, by considering other features.

4 Conclusion

We developed a method to update a user's knowledge that uses results from utterance verification, even when correct ASR results are not obtained. By using the utterance verification results, the system can estimate whether a user knows utterance patterns and can increase known degrees in the domain concept tree accordingly, which results in suppressing help messages from being generated regarding utterance patterns. Our future work includes improving the classification accuracy and using actual dialogues in experimental evaluations.

References

Yuichiro Fukubayashi, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. 2006. Dynamic help generation by estimating user's mental model in spoken dialogue systems. In *Proc. INTERSPEECH*.

Genevieve Gorrell, Ian Lewin, and Manny Rayner. 2002. Adding intelligent help to mixed-initiative spoken dialogue systems. In *Proc. ICSLP*.

Beth A. Hockey, Oliver Lemon, Ellen Campana, Laura Hiatt, Gregory Aist, James Hieronymus, Alexander Gruenstein, and John Dowding. 2003. Targeted help for spoken dialogue systems: intelligent feedback improves naive users' performance. In *Proc. EACL*.

Tatsuya Kawahara, Kentaro Ishizuka, Shuji Doshita, and Chin-Hui Lee. 1998. Speaking-style dependent filler phrase model for key-phrase detection and verification. In *Proc. ICSLP*, pages 3253–3256.

Kazunori Komatani and Tatsuya Kawahara. 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. COLING*, pages 467–473.

Donald A. Norman. 1988. *The Psychology of Everyday Things*. Basic Books.

Rafid A. Sukkar, Anand R. Sethur, Mazin G. Rahim, and Chin-Hui Lee. 1995. Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training. In *Proc. IEEE-ICASSP*.