Utterance Verification Using Garbage Words for a Hospital Appointment System with Speech Interface

Mitsuru Takaoka [#1], Hiromitsu Nishizaki [*2], Yoshihiro Sekiguchi [*3]

[#] *Department of Education Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi, Japan*
4-3-11 Takeda, Kofu-shi, Yamanashi 400-8511 JAPAN
[1] mtaka@alps-lab.org

[*] *Department of Research Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi, Japan*
4-3-11 Takeda, Kofu-shi, Yamanashi 400-8511 JAPAN
[2] hnishi@yamanashi.ac.jp
[3] sekiguti@yamanashi.ac.jp

***Abstract*—On a system that captures spoken dialog, users often use out-of-domain utterances to the system. The speech recognition component in the dialog system cannot correctly recognize such utterances, which causes fatal errors. This paper proposes a method to verify whether utterances are in-domain or out-of-domain. The proposed method trains systems with two language models: one that can accept both in-domain and out-of-domain utterances and the other that can accept only in-domain utterances. These models are installed into two speech recognition systems. A comparison of the recognizers' outputs provides a good verification of utterances. We installed our method in a hospital appointment system and evaluated it. The experimental results showed that the proposed method worked well.**

I. Introduction

Recently, several researchers have reported the development of numerous spoken dialog systems and human-machine interaction systems with speech interface. However, these systems are not widely used by the general public. One reason for this is that these systems cannot respond adequately to irrelevant humans utterances, because the systems accept only rule-based utterances.

While developing a spoken dialog system, we did not simply use the output from an automatic speech recognition (ASR) system, instead ensured addressing the speech recognition problem. Further, when developing a dialog system that can accept spontaneous utterances, we have to consider redundant utterances, which are irrelevant to the system domain. In addition to this consideration, spontaneous speech includes many filled pauses and has various expressions. Therefore, it is difficult for an ASR system to correctly recognize and understand an utterance.

For solving this problem, using a grammar-based language model in an ASR system enables high-performance speech recognition of utterances that fit into the grammar. However, the ASR system with the grammar model cannot accept utterances that do not fit into the grammar. Often, the utterances that do not match the grammar include out-of-vocabulary (OOV) words that are not listed in the ASR dictionary. It is well-known that these OOV words denigrate the usability of spoken dialog systems.

Therefore, to improve the usability of a spoken dialog system, it is very important to correctly reject redundant (out-of-domain) utterances that are not related to the domain of the system.

Many approaches have been proposed to verify in-domain and out-of-domain utterances. Komatani et al. [1] was able to achieve this verification using a weighted finite state transducer (WFST). This method could judge outputs of an ASR system by comparing it to the acceptable grammar. Wilpon et al. [2] proposed an acoustic method. This method trained acoustic models with out-of-domain keywords and non-voice sounds, enabling the model to determine if an utterance is in-domain or out-of-domain. In addition, judgment techniques using support vector machines (SVMs) [3] and confidence measures attached to each word of speech recognition result [4], [5] were also reported.

However, in these researches, the confidence measure based method that requires sufficient training data for building language and acoustic models that are appropriate for each domain of the systems are accepted. Therefore, it is necessary to prepare the training data by recording simulated speech for acoustic modeling and transcriptions for language modeling. In addition, machine learning methods require positive and negative data for training. As described above, the previous research proposals were costly to implement, creating an obstacle for developing a spoken dialog system.

Thus, this paper proposes an utterance verification method that determines if the utterance is in-domain or out-of-domain by incorporating irrelevant words into a language model. These words are called "garbage words" in this paper.

Furthermore, our method uses two ASR systems that improve the performance of the verification process. An utterance verification method using two ASR systems has been proposed by other researchers [6]. Hockey et al. used ASR systems with a grammar-based language model and a class-based statistical language model. Our proposed method has the characteristics of the ASR systems, both of supporting a grammar-based language model and using garbage words as input.

The method we proposed required building a language model for rejecting out-of-domain utterances. However, this effort is very simple and did not add significantly to the cost of building the model. Furthermore, an advantage of the proposed method is that it works well in a grammar-based language model. A grammar-based language model is easily built from utterance patterns, which are acceptable to a spoken dialog

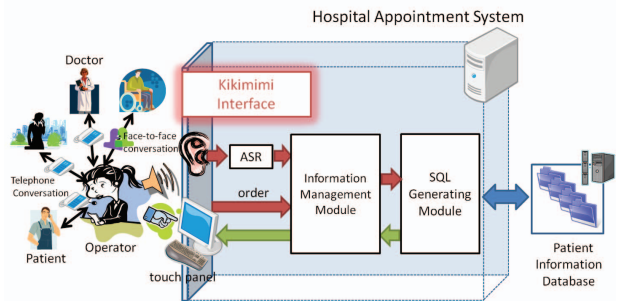

Fig. 1. Module configuration diagram of the hospital appointment system.

system, rather than a statistical language model.

We used two ASR systems: one that has a grammar-based language model with garbage words that can handle out-of-domain utterances, and the other that has a grammar-based language model that can accept only in-domain utterances. By comparing the resulting outputs from both the ASR systems, we can reliably verify whether the utterances are in-domain or out-of-domain.

Our method was tested on a hospital appointment system with a speech recognition interface, which is one kind of spoken dialog system. This system has been installed in a real hospital and it is used for daily activities, such as booking medical treatments. The experimental results showed that our method worked well for verifying utterances: the acceptance rate of in-domain utterances was 91.2% and the rejection rate of out-of-domain utterances was 91.3%.

II. Hospital Appointment System

A. Module Composition

The configuration of the hospital appointment system is shown in Fig. 1. The system is composed of the following four modules:

1) Patient information database
2) Kikimimi interface [7] and ASR system
3) Information management module
4) SQL generating module

1) The Patient Information Database: We used PostgreSQL as the database server. The types of patient information stored in the database are as follows:

- Patient's ID
- Patient's name
- Pronunciation of the name
- Birthday
- Medical staff member's name
- Disease name
- History of medical care
- Appointment date and time

It is important to note that the system is used only in the department of rehabilitation. Hence, the items stored in the database of this department might be different from that stored in the database of another department or hospital.

2) The Kikimimi interface: The Kikimimi interface is a core module of the system. It was implemented using a speech recognition system. It captures an utterance from the operator, and then based on the result of the ASR component, it generates in the SQL generating module the SQL code that is needed to access to the database.

Whenever a patient asks the operator to book the next medical treatment, the operator confirms the patient's name and date and time of the appointment. For example, a patient says "My name is Mitsuru Takaoka," and then the operator repeats the name "You are Mr. Takaoka, aren't you?" The Kikimimi interface captures the voice of the operator. The list of patients' names, which includes "Takaoka," is displayed on the touch panel. The operator can see detailed information about Mr. Takaoka when he or she touches that name in the list.

If patient's name does not display on the list due to ASR errors, the operator does not need to panic. He or she simply inputs the patient's name using the software keyboard or the physical keyboard. In addition, the Kikimimi interface is used as a normal speech interface; in other words, the operator can talk directly to the system.

3) The Information management module and the SQL generating module: These modules control the operation of the appointment system. The main functions of these modules are post-processing of the output from the ASR system, changing the image displayed by the graphical user interface (GUI), decoding touch operations from the operator, and displaying the results of the SQL query to the database based on the ASR output.

The SQL generating module generates SQL to query the database.

B. What can the appointment system do?

The system can implement the process as follows:

- Search for patient information
- Add patient information to the database
- Change patient information
- Delete patient information
- Book medical treatment for a patient (creating, changing, or deleting an appointment)

When the operator searches for the patient's information, he or she inputs the patient's name to the system using the Kikimimi interface, the software keyboard, or the physical keyboard. Moreover, the operator can change or delete the patient's information using a touch operation or a keyboard.

The operator books medical treatments using touch operations on the Kikimimi interface after searching for the patient's name. The Kikimimi interface can extract date and time information from the words uttered by the operator.

In this manner, the operator can easily perform booking operations.

The system has primarily two modes. One is the patient information search mode (main screen), and the other is the medical treatment booking mode. Figs. 2 and 3 show examples of the patient information search mode screen and the medical treatment booking mode screen, respectively.

The process of booking treatment for a patient can be explained as follows. First, using the patient information search mode screen shown in Fig. 2, the operator searches the patient's name using the Kikimimi interface, software keyboard, or physical keyboard. If the operator has had a dialog

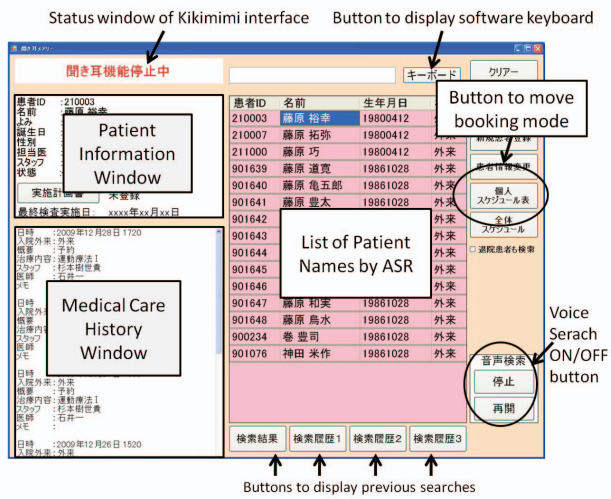

Fig. 2. Screen of patient's name search (main screen).

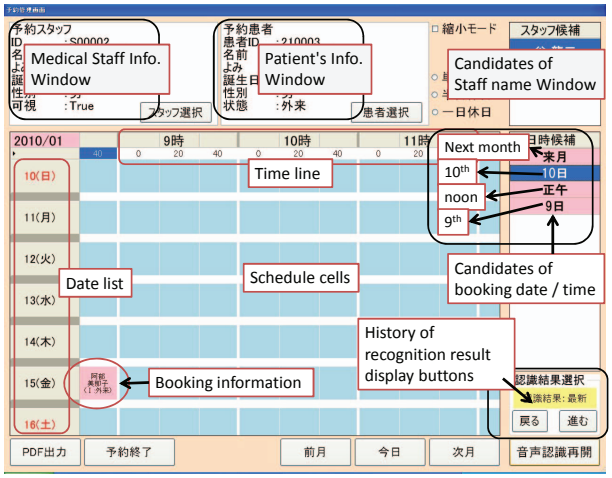

Fig. 3. Screen used for medical treatment booking.

with the patient, a list of patient names is displayed on the touch screen panal, because the Kikimimi interface captured the operator's speech and extracted a list of candidates that match or are similar to the patient's name using the ASR system. Then, the operator touches the patient's name on the screen to display the patient's information. If the patient's name is not in the list because of speech recognition errors, the operator inputs the patient's name using a software keyboard.

After selecting the patient, the operator touches the button to change to the main screen to the medical treatment booking mode (Fig. 3). This screen displays booking tables that are prepared for each medical staff member. The booking is finished by touching any cell. The Kikimimi interface is available for the booking mode. For example, when the operator says "Would you like to book at 10 o'clock on 10th of next month?," the keywords "next month," "10th," and "10 o'clock" are displayed on the window that displays choices for booking a date and time. The cell containing "10th next month, 10:00" is easily displayed on the screen by touching these keywords.

C. ASR system

For the ASR system, we used a large vocabulary speech recognition engine, called Julius[8], which is available as open source software and is widely used in Japanese spoken dialog systems.

Julius has two types of recognition dictionaries and language models. When searching for a patient's name, Julius uses the language model and recognition dictionary that are customized to recognize patient names. Otherwise, when booking medical treatments, Julius uses the language model and dictionary that recognize dates and times.

Acoustic models, which are phoneme-based hidden Markov model (HMM) that have been trained using 25-dimension feature vectors[1] of training speech data, are commonly used in the two speech recognition systems.

The dictionary for recognizing a patient's name is automatically generated using the patient's name and its pronunciations that have been stored in the database. Whenever a new patient's information is stored into the database, the dictionary is dynamically rebuilt.

The Kikimimi interface recognizes only the patient's name, medical staff member's name, date, and time. If the operator speaks sentences that are not related to name, date, and time (out-of-domain utterances), then speech recognition errors might occur. To prevent speech recognition errors, we introduced an utterance verification method that can reject out-of-domain utterances.

III. Utterance Verification

A. Outline

Our utterance verification technique uses two ASR systems. One is based on a context free grammar (CFG) language model that includes garbage words and in-domain keywords. This system contains rules that can accept in-domain and out-of-domain utterances. In particular, out-of-domain utterances are checked for acoustic likelihood by the ASR system using a rule that is based on a sequence of garbage words (it is called a "garbage rule" in this paper). The other ASR system also uses CFG-based language model; however, it can accept only in-domain utterances.

Fig. 4 shows our idea of utterance verification using two ASR systems. If an in-domain utterance is inputted to the ASR systems, both the ASR systems might output results that have a higher rate of correct transcriptions. However, because one ASR system uses rules with garbage words, a handful of words in the transcription generated by this system might be different from those generated by the ASR system that does not use garbage rules. Nevertheless, these transcriptions can be similar.

On the other hand, if an out-of-domain utterance is inputted to the ASR systems, the system with the garbage rules outputs a sequence of garbage words; whereas, the system without the garbage rules is forced to use acceptable rules of in-domain utterances to interpret out-of-domain utterances. As a result, the utterance is transcribed to a word sequence, but this has a completely different meaning from the original utterance. It

[1]A feature vector contains 12 dim. of MFCC, 12 dim. of ΔMFCC and Δpower.

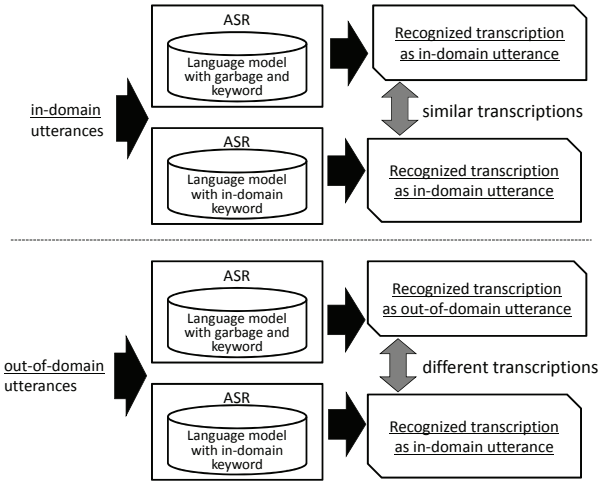

Fig. 4. A concept of the utterance verification.

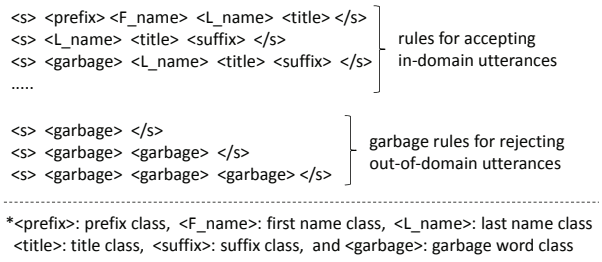

Fig. 5. An example of grammar-based rules.

should be noted that those two sequences are very different from each other, unlike the case of the in-domain utterance. This feature is useful to reject out-of-domain utterances.

B. Language modeling with garbage rules

The language model for recognizing speech uttered by the hospital appointment system operator is very simple. The operator just says a simple confirmation, such as "You are Mr.Takaoka, right?"

Therefore, the CFG-based language modeling should be sensitive enough to recognize these utterances even if a statistical language model, such as N-gram, is not used. The grammar-based model does not require many training sentences for modeling, but some utterance patterns of the operators are needed. These patterns are represented as a sequence of class-based symbols, which contain the first and last names of patients and other data.

We defined a special class named garbage class that correspond to garbage words. As shown in Fig. 5, a rule that accepts out-of-domain utterances is denoted as a sequence consisting of an arbitrary number of garbage words. However, in this paper, we specified a maximum of three garbage classes used by a rule for detecting out-of-domain utterances, because a user's utterances is not very long in duration. A method used to select garbage words is described in the next session.

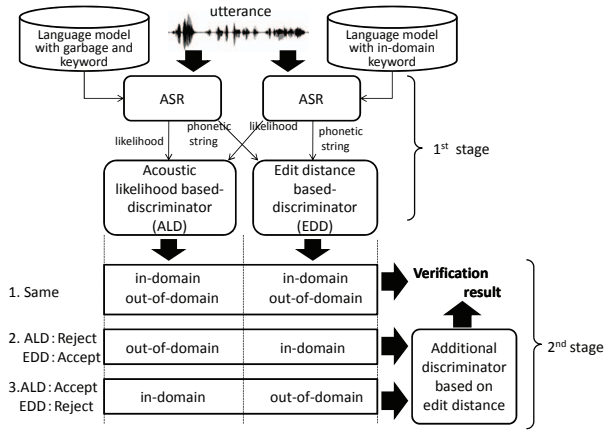

Fig. 6. Verification procedure using two ASR systems.

C. Selecting garbage words

Garbage words should be selected to include as many utterances as possible. Therefore, garbage words that have a wide variety of pronunciations are preferable.

For garbage words, we chose words that are frequently used in daily conversation. The words are selected from transcripts of simulated lectures included in the Corpus of Spontaneous Japanese (CSJ) [9]. The simulated lectures were recorded by several speakers talking about daily happenings.

We registered 286 of the most frequently used words from the CSJ in the garbage class, although we examined various words that are used with the highest frequency in the hospital appointment domain.

D. Verification using two speech recognition systems

Fig. 6 shows an utterance verification procedure using two speech recognition systems. Our utterance verification was performed in two stages. In the first stage, the utterance was recognized by both the ASR systems. And then, two types of discriminators check whether the utterance is in-domain or out-of-domain.

A judgment is made in the next stage based on the results of the previous stage. If the two discriminators output the same judgment, then final verification is clear. However, if there are different judgments from the two discriminators, an additional discrimination is performed to decide the final verification.

1) Acoustic likelihood-based discriminator: One of the discriminators is based on acoustic likelihoods derived from the two ASR systems. The acoustic likelihood-based discriminator (ALD) is expressed by the following equations:

$$\mathrm{Score}_1 = \log p(X|W_{gb+key}) \quad (1)$$

$$\mathrm{Score}_2 = \log p(X|W_{key}) \quad (2)$$

$$\begin{cases} Score_1 - Score_2 \geq T_S & \text{(out-of-domain)} \\ < T_S & \text{(in-domain)} \end{cases} \quad (3)$$

where $Score_1$ is an acoustic likelihood from an ASR system with garbage rules and $Score_2$ is an acoustic likelihood from the other system without garbage rules. If the difference

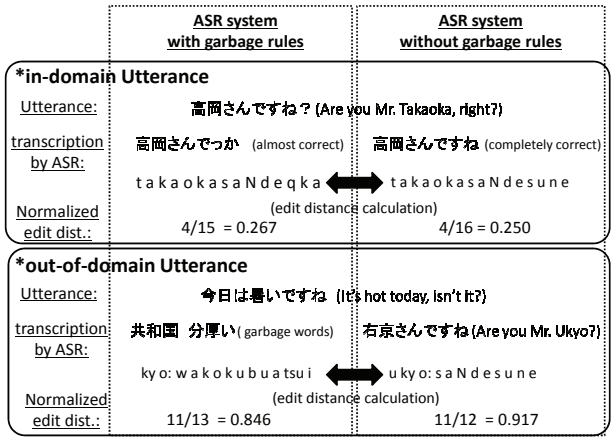

Fig. 7. Example of an edit distance based discrimination.

between $Score_1$ and $Score_2$ is less than the threshold T_S, the inputted utterance is identified as in-domain. Otherwise, if the difference is equal to or greater than T_S, the utterance is identified as out-of-domain.

2) Edit distance-based discriminator: The edit (Levenshtein) distance [10] is a metric used for measuring the amount of difference between two strings. This unit of measurement is widely used in the field of spoken language processing such as spoken term detection (STD) [11]. In this paper, a comparison of these measurements is used to verify the utterance in the other discriminator.

Fig. 7 shows an example of an edit distance-based discriminator (EDD). The transcription of an ASR system is translated into phoneme sequences that equals to its pronunciation. The discrimination is performed by the following rule:

$$\begin{cases} \text{edit distance} > T_L \text{ (out-of-domain)} \\ \leq T_L \text{ (in-domain)} \end{cases} \quad (4)$$

A phoneme-based edit distance is normalized by the length of the phoneme sequence derived from each ASR system. Therefore, two normalized edit distances are obtained, of which this paper adopts the larger one. If a normalized edit distance between two phoneme sequences created by recognizing an inputted utterance is equal to or less than a threshold T_L, then the discriminator judges the utterance as in-domain. On the other hand, if the edit distance is higher than T_L, then the utterance is determined as out-of-domain.

3) Final discrimination: The second stage shown in Fig. 6 judges the final discrimination of the utterance. If the two discriminators output the same judgment in the first stage, it is adopted as the final verification. However, if there are different judgments from the two discriminators, an additional discrimination is performed using new thresholds: $T_{L-accept}$ and $T_{L-reject}$. These thresholds are applied to the edit distance. A discrimination rule is described as follows:

ALD: out-of-domain and EDD: in-domain
in this case, we set a new threshold $T_{L-accept}$ by lowering the threshold T_L. This prevents the rejection of in-domain utterance.

ALD: in-domain and EDD: out-of-domain
in this case, we set the threshold $T_{L-reject}$ by raising T_L. This prevents an out-of-domain utterance from being falsely accepted.

IV. Verification Experiment

A. Experimental setup

We prepared 353 in-domain utterances and 298 out-of-domain utterances as our evaluation data. These utterances were recorded under a situation in which three system operators conversed with a patient on a simulated treatment booking task. The system can accept 2,000 patient names in this task.

Each threshold was set as follows:

- T_S can vary from 0 to 50, changing 5 steps.
- T_L can vary from 0.1 to 0.9, changing 0.1 steps.
- $T_{L-accept}$ is set to $T_L/2$.
- $T_{L-reject}$ is set to $(1-T_L)/2+T_L$.

$T_{L-accept}$ and $T_{L-reject}$ were empirically determined in this paper.

To confirm the effectiveness of our utterance verification, we performed a confidence measure-based verification [12]. This method is very simple. If a confidence measure of an utterance is lower than a threshold, the utterance is rejected as out-of-domain. The confidence measure is calculated by averaging all confidence scores attached to each word. The Julius ASR system can attach a confidence score to each word.

The evaluation metrics used for verification are "acceptance rate" for in-domain utterances and "rejection rate" for out-of-domain utterances. These are calculated by following equations:

$$\text{Acceptance rate} = \frac{\text{\# of accepted in-domain utterances}}{\text{\# of in-domain utterances}} \times 100\ [\%] \quad (5)$$

$$\text{Rejection rate} = \frac{\text{\# of rejected out-of-domain utterances}}{\text{\# of out-of-domain utterances}} \times 100\ [\%] \quad (6)$$

B. Experimental results

Fig. 8 shows the acceptance and rejection rates for the evaluation data using the confidence measure-based utterance verification. When the confidence measure is 0.7, the acceptance rate for in-domain and rejection rate for out-of-domain are 72.8% and 74.2%, respectively.

On the other hand, Figs. 9 and 10 show the evaluation results for in-domain and out-of-domain utterances when the thresholds T_L and T_S vary. T_S is the threshold of the difference between the acoustic likelihoods of the ASR systems, and T_L is the threshold of the edit distance. Therefore, in-domain utterances are likely to be falsely rejected when both the thresholds are set low. Of course, there is a trade-off between acceptance and rejection.

All utterances must be correctly verified to prevent the appointment system from performing false operations. Therefore, when we consider the trade-off, the best acceptance and rejection rates are 91.2% and 91.3%, respectively, under the conditions, of $T_S = 5$ and $T_L = 0.6$.

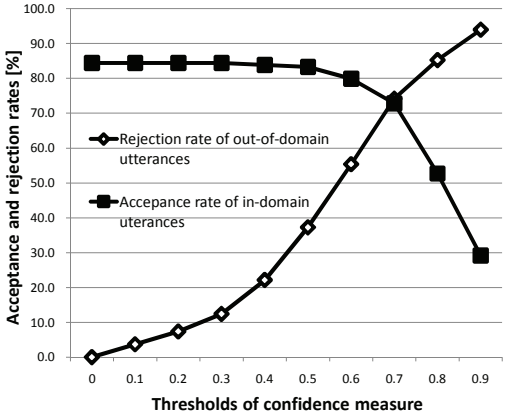

Fig. 8. Acceptance and rejection rates using confidence measure-based verification on the medical treatment booking task. All utterances are rejected when they are lower than the threshold.

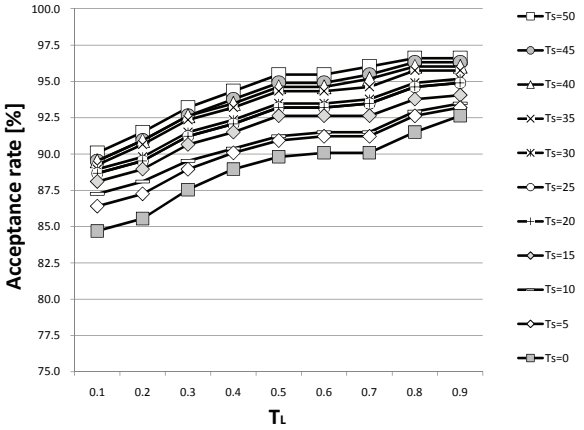

Fig. 9. Acceptance rates for in-domain utterances evaluated by the proposed method on the medical treatment booking task.

Compared with the confidence measure-based verification, these rates were drastically improved. Although our verification method is very simple, it performed utterance verification very accurately in the hospital appointment system that is used for daily activities at a real hospital.

V. Conclusion

In this paper, we proposed an utterance verification method to prevent human-machine interaction system with a speech recognition interface from misinterpreting out-of-domain utterances. Our technique is very simple, easily implemented with a grammar-based language model, and effective for verifying utterances.

Our method uses garbage rules (words) and discriminators based on the output of two ASR systems. The method was installed in a hospital appointment system with the Kikimimi interface and the verification performance was evaluated using an in-domain and out-of-domain utterance set. The experimental results showed that our method worked better than a verification method based on confidence measure.

However, there are some problems in the proposed method: how to select garbage words, how to set the thresholds, and so on. Therefore, in future works, we will address these problems.

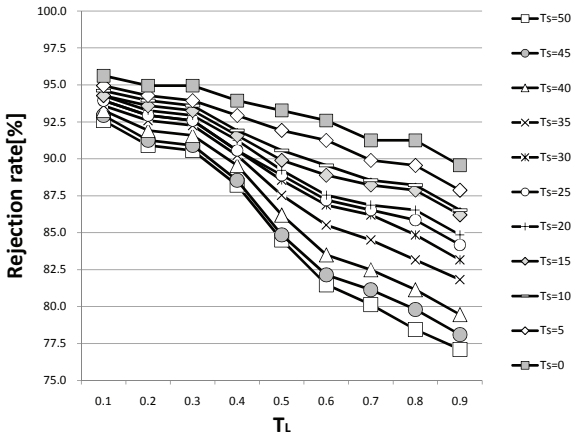

Fig. 10. Rejection rates for out-of-domain utterances evaluated by the proposed method on the medical treatment booking task.

References

[1] K. Komatani, Y. Fukubayashi, S. Ikeda, T. Ogata, and H. G. Okuno, "Selecting help messages by using robust grammar verification for handling out-of-grammar utterances in spoken dialogue systems," *IEICE Transactions on Information Systems*, vol. E93-D, no. 12, pp. 3359–3367, 2010.

[2] J. Wilpon, L. Rabiner, C. Lee, and E. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden markov models," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 38, no. 11, pp. 1870–1878, 1990.

[3] I. Lane, T. Kawahara, T. Matsui, and S. Nakamura, "Out-of-domain utterance detection using classification confidences of multiple topics," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 150–161, 2007.

[4] R. San Segundo, B. Pellom, K. Hacioglu, W. Ward, and J. Pardo, "Confidence measures for spoken dialogue systems," in *Proceedings of the 2010 IEEE Internationl Conference on Acoustics, Speech, and Signal Processing (ICASSP2010)*, vol. 1, pp. 393–396, 2010.

[5] F. Torres, L. Hurtado, F. Garca, E. Sanchis, and E. Segarra, "Error handling in a stochastic dialog system through confidence measures," *Speech Communication*, vol. 45, no. 3, pp. 211–229, 2005.

[6] B. A. Hockey, O. Lemon, E. Campana, L. Hiatt, G. A. J. Hieronymus, A. Gruenstein, and J. Dowding, "Targeted help for spoken dialogue systems: intelligent feedback improves naive users' performance," in *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics (EACL'03)*, pp. 147–154, 2003.

[7] T. Kamihira, H. Nishizaki, Y. Sekiguchi, R. Kurakane, K. Nishizaki, and H. Ikegami, "Development of hospital appointment system with user-friendly speech interface," in *Proceedings of the 2nd Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2010)*, pp. 490–493, 2010.

[8] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine julius," in *Proceedings of the 1st Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC2009)*, 6 pages, 2009.

[9] K. Maekawa, "Corpus of Spontaneous Japanese: Its design and evaluation," in *Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR2003)*, pp. 7–12, 2003.

[10] M. Gilleland. (2011) Levenshtein distance, in three flavors. [Online]. Available: http://www.merriampark.com/ld.htm

[11] S. Natori, H. Nishizaki, and Y. Sekiguchi, "Japanese spoken term detection using syllable transition network derived from multiple speech recognizers' outputs," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, pp. 681–684, 2010.

[12] A. Lee and T. Kawahara, "Real-time word confidence scoring using local posterior probabilities on tree trellis search," in *2004 IEEE Internationl Conference on Acoustics, Speech, and Signal Processing (ICASSP2004)*, vol. 1, pp. 793–796, 2004.