Regular Paper

Spoken Term Detection Using Phoneme Transition Network from Multiple Speech Recognizers' Outputs

Satoshi Natori[1,a) Yuto Furuya[1,b) Hiromitsu Nishizaki[1,c) Yoshihiro Sekiguchi[1,d)

Received: May 30, 2012, Accepted: November 2, 2012

Abstract: Spoken Term Detection (STD) that considers the out-of-vocabulary (OOV) problem has generated significant interest in the field of spoken document processing. This study describes STD with false detection control using phoneme transition networks (PTNs) derived from the outputs of multiple speech recognizers. PTNs are similar to subword-based confusion networks (CNs), which are originally derived from a single speech recognizer. Since PTN-formed index is based on the outputs of multiple speech recognizers, it is robust to recognition errors. Therefore, PTN should also be robust to recognition errors in an STD task, when compared to the CN-formed index from a single speech recognition system. Our PTN-formed index was evaluated on a test collection. The experiment showed that the PTN-based approach effectively detected OOV terms, and improved the F-measure value from 0.370 to 0.639 when compared with a baseline approach. Furthermore, we applied two false detection control parameters, one is based on the majority voting scheme. The other is a measure of the ambiguity of CN, to the calculation of detection score. By introducing these parameters, the performance of STD was found to be better (0.736 for the F-measure value) than that without any parameters (0.639).

Keywords: majority voting, multiple speech recognizers, network-based indexing, spoken term detection

1. Introduction

Recently, the number of information technology environments in which numerous audio and multimedia archives such as video archives and digital libraries can be easily used has increased. In particular, there is a rapidly increasing number of archived spoken documents such as broadcast programs, spoken lectures, and meeting recordings, with some of them being accessible through the Internet. Although there is an increasing need to retrieve such spoken information, there are currently no effective retrieval techniques to meet these needs. Therefore, the development of the technology for retrieving such information has become increasingly important.

The National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency hosted the Text REtrieval Conference (TREC) Spoken Document Retrieval (SDR) track in the second half of the 1990s, and many studies on SDR of English and Mandarin broadcast news documents were presented [1]. The TREC-SDR is an ad-hoc retrieval task that retrieves spoken documents, which are highly relevant to a user query. In 2006, NIST initiated the Spoken Term Detection (STD) project with a pilot evaluation and workshop [2]. STD intends to detect the positions of target spoken terms from audio archives.

STD requires automatic speech recognition for speech-to-text conversion. Therefore, STD is difficult with respect to searching for terms in a vocabulary-free framework because search terms are unknown before using the speech recognizer. Many studies [3], [4] that address STD tasks have been proposed, and most of them focused on the out-of-vocabulary (OOV) and speech recognition error problems. For example, STD techniques that employ entities such as subword lattices and confusion networks (CNs) were proposed.

In this study, we propose an STD technique that uses subword-based CN. We use a phoneme transition network (PTN)-formed index derived from multiple speech recognizers' 1-best hypothesis and an edit distance-based dynamic time warping (DTW) framework to detect a query term.

PTN-based indexing originates from the concept of CN being generated from a speech recognizer. CN-based indexing for STD is a powerful indexing method because CN has abundant information when compared with that of the 1-best output from the same speech recognizer. In addition, it is known that many candidates are obtained by one or more speech recognizers that have different language models (LMs) and acoustic models (AMs). For example, multiple speech recognizers' outputs improve the speech recognition effectively. Fiscus [5] proposed the ROVER (recognizer output voting error reduction) method, which adopts a word voting scheme. Utsuro et al. [6] developed a technique for combining multiple recognizers' outputs using a support vector machine to improve the speech recognition. The application of the characteristics of the word (or subword) sequence output by recognizers may enhance STD because these characteristics are different for each speech recognizer. PTNs that are based on multiple speech recognizers' outputs can cover more subword se-

[1] Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi, Kofu, Yamanashi 400–8511, Japan
a) natori@alps-lab.org
b) furuya@alps-lab.org
c) hnishi@yamanashi.ac.jp
d) sekiguti@yamanashi.ac.jp

quences of spoken terms. Therefore, the use of multiple speech recognizers may improve STD relatively to that of a single recognizer's output. This is the principal idea in this study.

This study employs 10 types of speech recognition systems with the same decoder used for all types. Two types of AMs (triphone- and syllable-based Hidden Markov Models (HMMs)) and five types of LMs (word- and subword-based) were prepared. The multiple speech recognizers can generate the PTN-formed index by combining subword (phoneme) sequences from the output of these recognizers into a single CN.

We evaluated the PTN-formed index derived from the 10 recognizers' outputs. The experimental result for the Japanese STD test collection [7] showed that the use of the PTN-formed index effectively improved the STD compared with that of the CN-formed index, which was derived from the phoneme-based CN comprising the 10-best phoneme sequence outputs from a single speech recognizer [8], [9].

However, many false detection errors occurred because the PTN-formed index had redundant phonemes that were incorrectly recognized by a few speech recognizers. The use of more speech recognizers can achieve a better recognition performance, but more errors may occur at the same time.

Therefore, we introduce the concept of majority voting to calculate the edit distance between a query term and the index. In addition, a measure of the ambiguity in the PTN is adopted into the DTW. New parameters based on majority voting and ambiguity are easily derived from the PTN and are considered for distance calculation. We aim to improve the STD by effectively utilizing the advantages realized by using multiple speech recognizers. This is an original concept in the field of STD research.

The majority voting scheme is a powerful technique, and it has applications in a wide range of research areas. For example, "AKARA 2010," a Japanese computer chess game (Shogi), has also incorporated a majority voting scheme [10], and won the mistress of the Shogi player in Japan. As will be shown later, the ROVER method has been successfully used in speech recognition.

To prevent false detections, we applied the majority voting and ambiguity parameters of the PTN to our term search engine on the basis of the DTW. The improved term search engine drastically decreased the number of false detections.

The remainder of this paper is organized as follows. In Section 2, we will introduce a few previous studies on STD. In Section 3, we explain the types of indices that deal with the study and the term search engine using the DTW framework. Moreover, the STD experiment for OOV query terms is discussed in this section. Section 4 describes a false detection control technique in the term search engine. We discuss the STD experimental results for the OOV terms using the improved engine. In Section 5, we confirm the effectiveness of our indexing and parameters using a different query set which is different from the one used in Section 4. In addition, we compare our STD results with those of others. Finally, we summarize this study in Section 6.

2. Related Works

There have been many studies on speech recognition errors, OOV, and term pronunciation problems in STD [11], [12], [13]. This study addresses only the recognition errors and OOV problems.

STD may be improved by refining the speech recognition. Many papers have already reported improvements in speech recognition. In particular, a few studies [5], [6], [14] have proposed methods to improve recognition with the help of combination of multiple speech recognizers' outputs.

Subword-based speech recognition and subword-based term matching are generally used to solve the OOV problem. Large vocabulary continuous speech recognition (LVCSR) is effective in detecting in-vocabulary (INV) words. However, it cannot detect any OOV term. The deal with this problem, a combination of phoneme recognition and LVCSR output was proposed [15], [16]. In addition, Wallace et al. [17] proposed a language modeling method for improving the phoneme recognition.

In addition, a few indexing models/structures for STD were reported, and a few studies on STD dealt with the lattice-formed or CN-formed index [18], [19]. These indices have a high expression ability for subword sequences resulting from a speech recognizer, and they can enhance the STD. Iwami et al. [20] proposed a subword (syllable)-based N-gram indexing of N-best hypotheses from single speech recognizers, and Katsurada et al. [21] proposed the use of a suffix array index with a tree structure. Kaneko et al. [22] proposed metric space indexing for a rapid STD.

Our proposed method for STD, PTN-based indexing, is based on using the output of multiple speech recognizers [8], [9] and a CN-formed index. The use of multiple speech recognition systems results in a variety of speech recognition results, which can improve the speech recognition [6]. In addition, the confidence measure of speech recognized words on the basis of majority voting can effectively detect recognition errors [23]. So, it is able to reduce false detection errors of the STD. We apply this knowledge [6], [23] to study the STD. However, no studies have used a large number of speech recognizers and the confidence measure on the basis of majority voting on STD research. The proposed PTN-formed index has abundant information than that of a CN-formed index from a single speech recognizer, and the reliable majority voting can be used when a query term is searched to control false detection errors. This is an original approach to STD research.

3. STD Framework

3.1 Outline of the STD Framework

Figure 1 illustrates the STD framework in this study.

In the indexing phase, speech data is processed by speech recognition, and the output (word or subword sequences) is converted into the index for STD. In the search phase, the word-formed query is converted into the phoneme sequence, and the phoneme-formed query is then input into the term search engine. For English queries, we have to consider a variety of pronunciations for the queries. A few reports have discussed the pronunciation problem [24], but in this study, we deal with only Japanese STD. Most of the Japanese words can be completely converted into phoneme sequences (pronunciation) using a dictionary in which the relationship between words and their pronunciations is

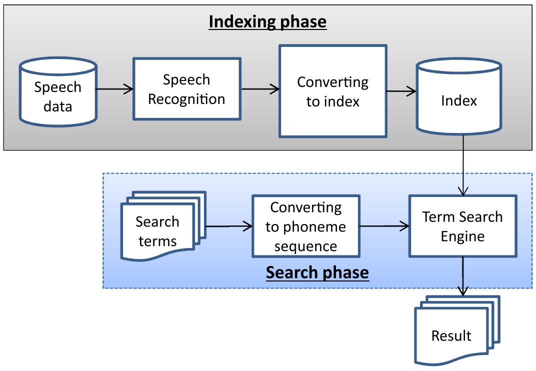


Fig. 1 Overview of our STD framework.

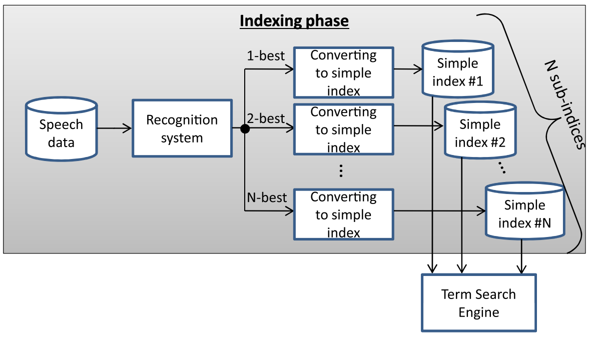


Fig. 2 Phase of creating a simple index.

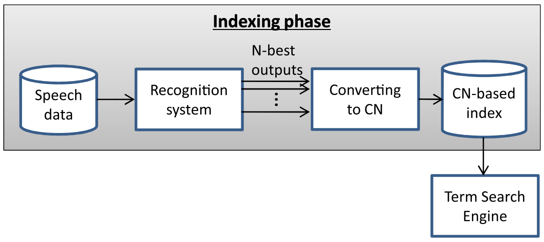


Fig. 3 Phase of creating the CN-formed index.

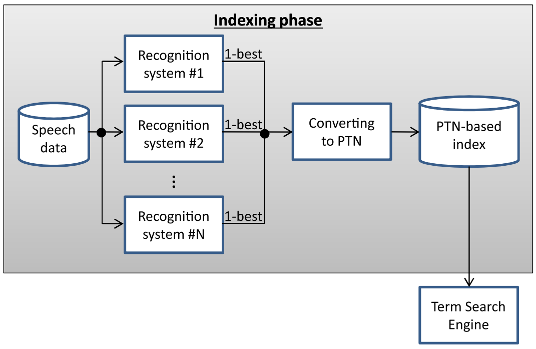


Fig. 4 Phase of creating the PTN-formed index.

listed. In addition, most of the Japanese words have unambiguous pronunciations. Therefore, we do not consider the pronunciation problem in this study.

The term search engine searches the input query term from the index at the phoneme level using the DTW framework.

3.2 Types of Indices for STD

To compare our proposed PTN-formed index with other indices, we prepared three types of indices for the STD task: one is a simple index, and the others are network-formed tasks, as follows:

- subword-based simple index (denoted as "simple"),
- CN-formed index (denoted as "phoneme confusion network: PCN"),
- PTN-formed index.

3.2.1 Simple Index

Figure 2 represents an indexing phase for the simple index.

First, the speech data is transcribed by a speech recognizer, and the recognizer then outputs the N-best hypotheses. Each hypothesis is converted into the simple index. The simple index is a very simple structure that stores only phoneme sequences without any additional information such as scores from the recognizer. For example, the speech "cos θ and sin θ" is automatically transcribed and converted to "k o s a i N sh i: t a t o s a i N sh i: t a" (Japanese phoneme sequence of cos θ and sin θ), and the sequence is stored to a simple index. The term search engine can detect the input query term by performing DTW-based word spotting against the sequence in the simple index.

When the engine searches a query term, it matches the term to each sub-simple index. When the engine hits the query term for at least one sub-simple index, it is extracted even if the term does not match the other sub-indices.

3.2.2 CN-formed Index

Figure 3 shows the indexing phase for creating CN-formed index. The N-best hypotheses from a speech recognizer are combined by aligning all N-best sequences using dynamic programming (DP) and are then converted into the CN-formed index. A CN can effectively represent multiple symbol (phoneme) sequences with the time order of the symbols. We created the CN-formed index from the 10-best hypotheses output from a speech recognizer.

3.2.3 PTN-formed Index

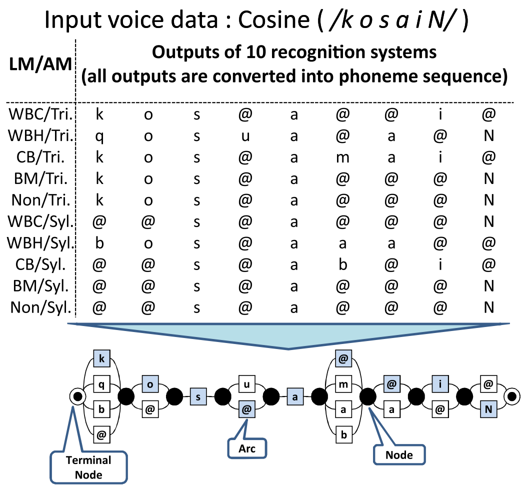
Figure 4 shows the index phase for creating the PTN-formed index. Speech data was recognized by N speech recognizers. We used 10 ($N = 10$) sorts of speech recognizers in this study. Each 1-best hypothesis was translated into the phoneme sequence, and all N sequences were then combined to a PTN-formed index.

Figure 5 shows an example of the development of a PTN-formed index for the speech "*cosine*" (Japanese pronunciation is /k o s a i N/) by aligning N phoneme sequences from the 1-best hypothesis of all recognizer. The speech was recognized by the 10 recognizers to yield 10 hypotheses, which were then converted into phoneme sequences. Next, we obtained "aligned sequences" using the DP scheme which is the same scheme described in Ref. [5]. Finally, PTN was obtained by converting the aligned sequences. The term "@" in Fig. 5 indicates a null transition. Arcs between the nodes in PTN have a few phonemes and null transitions with an occurrence probability. However, in this study, we did not consider any phoneme-occurrence probabilities.

3.3 Term Search Engine

3.3.1 For Simple Index

We adopted the DTW-based word spotting method. In this study, **Fig. 6** shows the permitted paths on the DTW lattice. X



LM/AM	Outputs of 10 recognition systems (all outputs are converted into phoneme sequence)								
WBC/Tri.	k	o	s	@	a	@	@	i	@
WBH/Tri.	q	o	s	u	a	@	a	@	N
CB/Tri.	k	o	s	@	a	m	a	i	@
BM/Tri.	k	o	s	@	a	@	@	@	N
Non/Tri.	k	o	s	@	a	@	@	@	N
WBC/Syl.	@	@	s	@	a	@	@	@	N
WBH/Syl.	b	o	s	@	a	a	a	@	@
CB/Syl.	@	@	s	@	a	b	@	i	@
BM/Syl.	@	@	s	@	a	@	@	@	N
Non/Syl.	@	@	s	@	a	@	@	@	N

Fig. 5 Creating PTN-formed index by alignment using the DP scheme.

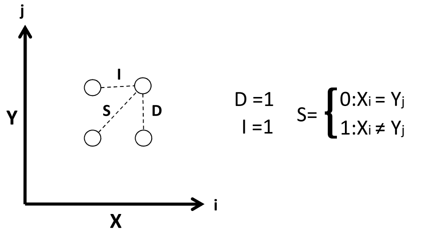


Fig. 6 Definition of the DTW path.

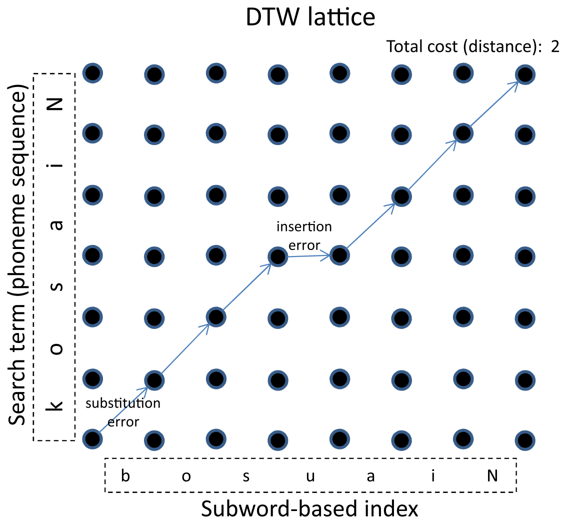


Fig. 7 Example of term search by using the simple index.

and Y indicate an index and a query term, respectively. Moreover, we used the edit distance to calculate the cost on the DTW paths.

Figure 7 shows the DTW framework between the search term "k o s a i N" (cosine) and the subword-based simple index. The costs for the substitution, insertion, and deletion errors were generally set to 1.0. The total cost at the grid point (i, j) ($i = \{0, ..., I\}$, $j = \{0, ..., J\}$, where I and J are the number of phonemes in an utterance for the index and the query term, respectively) on the DTW lattice was calculated by the following equations:

$$D(i, j) = \min \begin{cases} D(i, j-1) + 1.0 \\ D(i-1, j) + 1.0 \\ D(i-1, j-1) + Match(i, j) \end{cases} \tag{1}$$

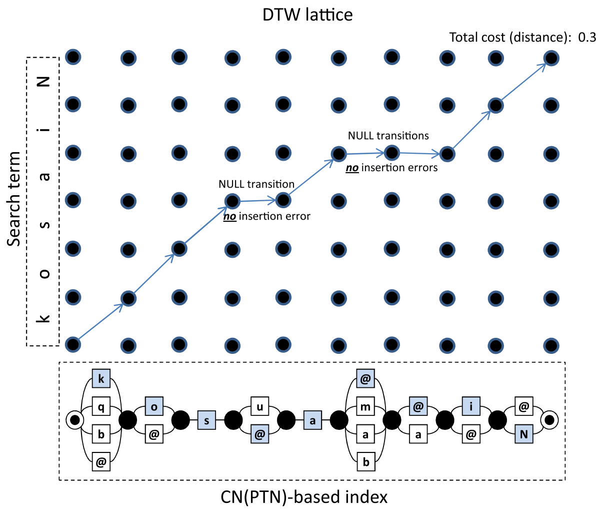


Fig. 8 Example of term search on network-formed index.

$$Match(i, j) = \begin{cases} 0.0 : Query(j) = ph(i) \\ 1.0 : Query(j) \neq ph(i) \end{cases} \tag{2}$$

Here $Query(j)$ indicates the j-th phoneme label in the query term and $ph(i)$ is the i-th phoneme label in the simple index.

To start searching a query term, the term search engine initializes $D(i, 0) = 0$, and then, it calculates $D(i, j)$ using Eq. (1) ($i = \{1, ..., I\}$, $j = \{0, ..., J\}$). Furthermore, $D(i, J)$ is normalized by the length of the DTW path.

After completing the calculation, the engine outputs the detection candidates which have a normalized cost $D(i, J)$ below a threshold θ. By changing the θ value, the recall and precision rates for STD can be controlled.

As an example, in Fig. 7, the total matching distance between the query term and the index is 2.0 (one substitution and one insertion error; it is not normalized).

3.3.2 For Network-formed Index

Figure 8 shows an example of the DTW framework for the PTN- (or CN-) formed index. PTN (or CN) has multiple arcs between adjacent nodes. These arcs are compared with phoneme labels of a query term.

In addition, PTN (or CN) has null transitions. Therefore, the cost equation (1) was extended to the following equations:

$$D(i, j) = \min \begin{cases} D(i, j-1) + 1.0 \\ D(i-1, j) + NULL(i) \\ D(i-1, j-1) + Match(i, j) \end{cases} \tag{3}$$

$$Match(i, j) = \begin{cases} 0.0 : Query(j) \in PTN(i) \\ 1.0 : Query(j) \notin PTN(i) \end{cases} \tag{4}$$

$$Null(i) = \begin{cases} 0.1 : NULL \in PTN(i) \\ 1.0 : NULL \notin PTN(i) \end{cases} \tag{5}$$

Here $PTN(i)$ is the set of phoneme labels of arcs at the i-th node in the PTN.

When the query term matches to null (@) in the PTN (or CN), the transition cost is set to 0.1. This value is empirically determined.

3.4 Speech Recognition

As described in Section 3.2.3, the speech data was processed by 10 speech recognizers. Julius ver. 4.1.3 [25], an open source decoder for LVCSR, was used in all the systems.

We prepared two types of AMs and five types of LMs for constructing the PTN. AMs are triphone-based (Tri.) and syllable-based HMMs (Syl.), both of which are trained on spoken lectures in the Corpus of Spontaneous Japanese (CSJ) [26].

All the LMs were word- and character-based trigrams as follows:

WBC: word-based trigram in which words are represented by a mix of Chinese characters, Japanese Hiragana, and Katakana.

WBH: word-based trigram in which all words are represented only by Japanese Hiragana. The words composed of Chinese characters and Katakana are converted into Hiragana sequences.

CB: character-based trigram in which all characters are represented by Hiragana.

BM: character-sequence-based trigram in which the unit of language modeling is comprised of two Hiragana characters.

Non: LM is not used. Speech recognition without any LM is equivalent to phoneme (or syllable) recognition.

Each model is trained by using transcriptions of the CSJ.

Finally, 10 combinations, which comprise two AMs and five LMs, are formed.

3.5 Japanese STD Test Collection

3.5.1 Speech Data and Speech Recognition Performance

We used a subset of the Japanese test collection for the STD [7] to verify our method. This test collection was created by a working group of the Special Interest Group-Spoken Language Processing (SIG-SLP) of Information Processing Society of Japan (IPSJ).

The CSJ is used as the target spoken documents set of this test collection. It totally contains 2,702 speeches files, including actual academic presentations and simulated public speeches. However, only 177 speeches (44 hours) of them are contained in this collection. These speeches are referred to as the "CORE" part of the CSJ. They are not included in the training data set of AM and LM.

Table 1 shows syllable-based correct and accuracy rates of each speech recognizer which are represented by "Corr." and "Acc.". The combination of 10 recognizers can obtain the high correct rate but a very low accuracy rate. This is because the accuracy rate of the combination of all the recognizers' outputs is based on the method used to calculate the accuracy. First, each recognizer's output was converted into a syllable sequence, and then, 10 types of syllable sequences were obtained. We aligned all the syllable sequences using dynamic programming on a syllable level. The syllable that appeared in the recognizer's output temporally corresponded to other syllables in other outputs simultaneously. Finally, we found that the one syllable sequence in which syllables appeared simultaneously were adjacently arranged by combining the aligned syllable sequences. We calculated the accuracy using the reference and the combined syllable sequences. Therefore, the syllable-based correct rate was drastically improved, but the insertion errors also increased.

Table 1 Syllable recognition rates for the CSJ CORE lectures [%].

	1-best		10-best comb.	
LM / AM	Corr.	Acc.	Corr.	Acc.
WBC / Tri	**86.46**	**83.01**	**89.96**	**44.88**
WBH / Tri	86.27	81.42	89.95	35.06
CB / Tri	81.83	77.42	85.99	41.74
BM / Tri	83.60	78.64	88.35	39.47
Non / Tri	71.00	51.20	74.56	21.06
WBC / Syl	79.11	76.35	84.19	35.73
WBH / Syl	79.32	75.83	84.29	29.90
CB / Syl	73.84	71.18	79.47	42.10
BM / Syl	77.89	74.42	84.60	37.26
Non / Syl	63.68	45.43	67.96	21.57
10 Systems comb.	94.28	−13.78	96.47	−243.51

The best combination of AM and LM for syllable recognition is the word trigram LM (WBC) and triphone-based AM (Tri.).

3.5.2 Query Set

Various search terms, which include Japanese single word and multiword terms, and common and rare terms were prepared in the test set. All terms from the test speech data were spoken.

The Japanese test collection for the STD includes OOV and INV query sets. The OOV (mainly used for the STD experiment) set has 50 terms totally, which were spoken 233 times in the CORE lecture speeches. All of these OOV terms are picked up with respect to the speech recognition dictionary of the WBC LM. Meanwhile, the INV set also has 50 terms, which were spoken 742 times in the CORE. All the terms are included in the dictionary.

3.6 STD Experiment

3.6.1 Evaluation Metric

The evaluation metrics used in this study were the recall, precision, F-measure, and mean average precision (MAP) values. These measurements are frequently used to evaluate the information retrieval performance, and they are defined as follows:

$$Recall = \frac{N_{corr}}{N_{true}} \tag{6}$$

$$Precision = \frac{N_{corr}}{N_{corr} + N_{spurious}} \tag{7}$$

$$F - measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{8}$$

Here N_{corr} and $N_{spurious}$ are the total number of correct and spurious (false) term detections, and N_{true} is the total number of true term occurrences in the speech data. The F-measure values for the optimal balance of *Recall* and *Precision* values are denoted by "maximum F-measure."

The STD performance for the query sets can be displayed by a recall-precision curve, which is plotted by changing the threshold θ value on the DTW-based word spotting.

MAP is the mean of the average precision values for each query. It can be calculated as follows:

$$MAP = \frac{1}{Q} \sum_{i=1}^{Q} AveP(i) \tag{9}$$

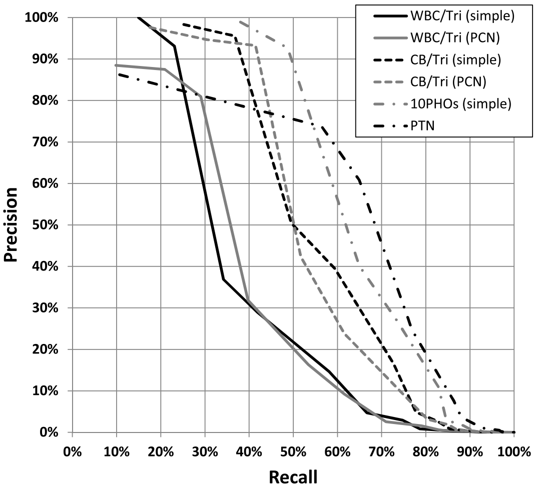


Fig. 9 Recall-precision curves for OOV terms detection. Our proposed PTN is compared with the other types of indices.

Table 2 Maximum F-measure and MAP values for each index.

Index type	Maximum F-measure	MAP
WBC/Tri. (simple)	0.370	0.565
WBC/Tri. (PCN)	0.428	0.585
CB/Tri. (simple)	0.531	0.695
CB/Tri. (PCN)	0.574	0.695
10PHOs (simple)	0.639	0.775
PTN	0.639	0.806

where Q is the number of whole queries and $AveP(i)$ denotes the average precision of the i-th query of the query set. Average precision is calculated by averaging the precision values computed for each relevant term in the list in which retrieved terms are ranked by a relevance measure.

$$AveP(i) = \frac{1}{Rel_i}\sum_{r=1}^{N_i}(\delta_r \cdot Precision_i(r)) \qquad (10)$$

where r is the rank, N_i is the rank number at which all the relevance terms of query i are detected, and Rel_i is the number of the relevance terms of the query i. δ_r is a binary function for a given rank r.

3.6.2 Experimental Results

As mentioned above, we prepared the following four types of indices for the STD experiments:

- **Simple** is phoneme based, and it is derived from the 10-best hypotheses of the WBC/Tri. (LM/AM) recognizer, which shows the best phoneme-based speech recognition rate among all the recognizers. Moreover, we use the index comprising 10-best hypotheses of the CB/Tri. and 10 of the 1-best hypotheses from the 10 types of speech recognizers' outputs.
- **PCN** index is formed of CN, and it comprises 10-best outputs of the WBC/Tri. or CB/Tri. recognizer.
- **PTN** is our proposed PTN-formed index from the 10 types of speech recognizers' outputs.

Figure 9 shows the recall-precision curves for the OOV STD test collection. In addition, **Table 2** represents the maximum F-measure and MAP values for each index. "10PHOs (simple)" indicates the recall-precision curve from 10 sub-indices formed of 10 1-best hypotheses from the 10 speech recognizers' outputs.

First, we compared different speech recognition systems. By comparing the curves of WBC/Tri. with CB/Tri., the index of the CB/Tri. was found to result in a better performance than the WBC/Tri. In addition, the F-measure and MAP values of the CB/Tri. were also higher than the WBC/Tri.

We used the OOV query set, and not all the queries are registered in the dictionary of the WBC/Tri. system. Therefore, the phoneme sequences converted from the word-level transcription were unsuitable for detecting the OOV terms. While performing speech recognition of an OOV term using the recognizer with the word-based LM, the OOV term was confused with other words or word sequences. In many cases, their pronunciations were different from the original term. On the other hand, the phoneme sequences from the recognizer with the subword-based LM was more similar to the pronunciation of the OOV term. This influenced the performance for the OOV term detection.

Using the outputs of 10 speech recognizers leads to a better performance than that obtained with only the CB/Tri. recognizer. The "10PHOs (simple)" results in a 20% improvement in the F-measure when it is compared with that of the simple index which is from the subword-based recognizer (CB/Tri.). This is because the combination of 10 recognizers achieved a good speech recognition performance, as shown in Table 1, although many insertion errors occurred.

Next, we compared two types of index structures: simple and PTN (CN). In either of three cases (WBC/Tri., CB/Tri., and 10 recognizers), PTN- (CN-) based indices exhibited a better performance in both F-measure or MAP than the simple index. CN can effectively represent phoneme sequences using the time order of symbols, and it is apparently suitable for the STD task.

From Fig. 9 and Table 2, our proposed PTN index has the best performance among the six indices. However, the precision rate of the PTN index is not higher than that of the simple index in the low-level (under approximately 60%) range of the recall rate because of the occurrence of many false detections. Therefore, we introduced false detection control parameters to our search engine.

4. False Detection Control

By using the PTN-formed index based on outputs from multiple speech recognizers, we obtained a good STD performance value. However, there were many false detections because we did not use any parameters that were related to CN, such as the posterior probability.

In this section, we introduce a few false detection control parameters that are related to the characteristics while using multiple recognizers and CN.

4.1 Parameters for False Detection Control

We provide the following two parameters to control false detection:

- "$Voting(p)$" is the number of speech recognizers that output the same phoneme p at the same arc. A larger $Voting(p)$ will give a higher reliability on the phoneme p.
- "$ArcWidth(i)$" is the number of arcs (phoneme labels) at $PTN(i)$. Having fewer values of $ArcWidth(i)$ also improves

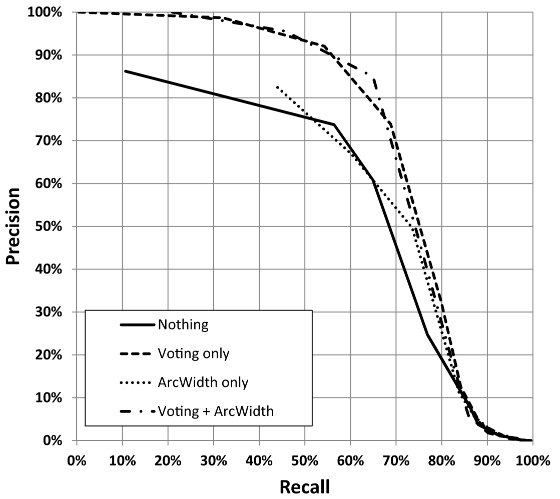


Fig. 10 Recall-precision curves for OOV terms detection with the false detection control technique.

the reliability of phonemes at $PTN(i)$.

The control parameters are applied to Eq. (3) as follows:

$$D(i,j) = \min \begin{cases} D(i,j-1)+1.0 \\ D(i-1,j)+NULL(i) \\ D(i-1,j-1)+Match(i,j) \\ \qquad\qquad +Vot(i,j)+Acw(i) \end{cases} \tag{11}$$

$$Vot(i,j) = \begin{cases} \frac{\alpha}{Voting(p)} : \exists p \in PTN(i), p = Query(j) \\ 1.0 : Query(j) \notin PTN(i) \end{cases} \tag{12}$$

$$Acw(i) = \beta \cdot ArcWidth(i) \tag{13}$$

where α and β are hyper parameters, which are set to 0.5 and 0.01, respectively.

We allow a null transition between two nodes in the PTN-(CN-) formed index with a cost of 0.1. Therefore, the values of $Vot(i,j)$ and $Acw(i)$ must be less than the null transition cost. Therefore, α is set to be less than 0.1. If $\alpha \geq 0.1$, the null transitions have an advantage in the term search process. This may nullify the voting parameter. The parameter β must be set to 0.1 or less for the same reason.

$Voting(p)$ has a range of 1 to 10. Larger values of $Voting(p)$ further strengthens the confidence of phoneme p. By setting α to 0.5 intuitively, the phoneme p output by more than five recognizers may be accurate. In addition, $ArcWidth(i)$ has a range of 1 to 10. Arcs that have fewer values of $ArcWidth(i)$ are more accurate. The i-th node has the most inaccurate arcs when $ArcWidth(i)$ is 10. In this case, we set β to 0.01, then $Acw(i)$ becomes 0.1, it is the same value of the null transition cost.

Eq. (11) considers both parameters, however, sometimes, we can only use one parameter. If the voting parameter is only applied, the $Acw(i)$ is set to 0.

4.2 STD Experiment with False-Detection Control

Figure 10 shows the recall-precision curves of OOV term detection in different search engine environments. **Table 3** shows the F-measure and MAP values for each false detection parameter on the same test set.

Table 3 Maximum F-measure and MAP values for each false detection control parameter on OOV term detection.

Parameter	Maximum F-measure	MAP
Nothing	0.639	0.806
Voting only	0.712	0.860
ArcWidth only	0.634	0.821
Voting + ArcWidth	0.736	0.850

"Nothing" indicates that the term search engine did not use any parameters. "Voting only" and "ArcWidth only" indicate that the term search engine used the Voting or ArcWidth parameters, respectively. "Voting + ArcWidth" indicates that the engine used both the parameters.

As shown in Fig. 10 and Table 3, the voting parameter effectively decreases the false detections in a wide range of recall rates. The F-measure and MAP values (in "Voting only") are appreciably improved from 0.639 (in "Nothing") to 0.712 and from 0.806 to 0.860, respectively. However, the effect of introducing the ArcWidth parameter is less than that of the Voting parameter. With the "ArcWidth," the MAP value achieved an improvement of 0.015 points, but the F-measure value decreased slightly. The combination of the two parameters further improves the F-measure value up to 0.736, but it does not improve the MAP value.

The experiments with false detection control indicate that the majority voting scheme on multiple speech recognizers is a very powerful technique. In addition, the ArcWidth parameter has a positive influence on the STD performance only when it is combined with the voting scheme.

5. Discussion

It was shown that our proposed PTN-formed index with the majority voting scheme could effectively search for OOV terms. In this section, we discuss the superiority of PTN indexing by performing other STD experiments on another query set. In addition, we will compare our indexing and searching techniques with other studies reported at NTCIR-9 [*1] SpokenDoc task [27].

First, we describe the experimental results using a different query set. We used the INV query set of the test collection provided by the working group of SIG-SLP and IPSJ [7]. All the terms in the query set are included in the speech recognition dictionary of WBC/Tri. or WBC/Syl. recognizer. The set is completely different from the OOV query set used in the previous section.

Table 4 shows the experimental results for the F-measure and MAP values for each index using the INV term set. "Grep" shows the STD result by performing a Unix command "grep" against the transcriptions of the CORE speeches generated by the WBC/Tri. recognizer. The value of the F-measure is the same as that shown in Ref. [7]. This is the simplest search method.

In Table 4, the only difference between "Grep" and "WBC/Tri.(simple)" is the search method. The values of "WBC/Tri.(simple)" are derived from the STD engine with the DTW framework. This results in a predictably better perfor-

[*1] The NTCIR Workshop (http://research.nii.ac.jp/ntcir/ntcir-9/index.html) is a series of evaluation workshops designed to enhance research in information access technologies.

Table 4 Maximum F-measure and MAP values for each index on INV term detection.

Index type	Maximum F-measure	MAP
Grep (simple)	0.686	N/A
WBC/Tri.(simple)	0.715	0.679
WBC/Tri.(PCN)	0.726	0.728
PTN (no control param.)	0.770	0.776
PTN (with Voting)	0.774	0.810

Table 5 STD evaluation results of all the participants of the CORE set for the NTCIR-9 SpokenDoc task. The data was a part of Table 4 in Ref [27].

System ID	Maximum F-measure	MAP
AKBL-1	0.393	0.264
AKBL-2	0.385	0.272
ALPS-1	**0.725**	**0.837**
ALPS-2	0.714	0.757
IWAPU-1	0.644	0.772
IWAPU-2	0.510	0.733
NKGW-1	0.645	0.491
NKI11-1	0.570	0.684
NKI11-2	0.569	0.672
RYSDT-1	0.318	0.393
RYSDT-2	0.526	0.468
RYSDT-3	0.521	0.469
YLAB-1	0.425	0.344

mance than that obtained by using a simple search such as "grep" command.

By comparing the simple index with the CN-formed index which is from the output of the single recognizer, the CN-formed index is found superior to the simple index in both the F-measure and MAP values. In addition, the 10 speech recognizers resulted in a good performance. Furthermore, our proposed indexing and searching techniques including the PTN-formed index and false detection control parameters obtained the best STD performance among all indexing and searching techniques on the INV query set.

However, the F-measure with the control parameter (Voting) is slightly improved compared to that without the parameter. The parameters controlled the false detections of query terms effectively, which consist of less than 10 phonemes in the OOV query set. This is because there is a difference between the phoneme sequences produced by the 10 speech recognizers for an utterance including OOV term(s). On the other hand, the 10 recognizers output similar phoneme sequences for an utterance without any OOV term. In this case, the parameter has little impact on the control of false detections.

Next, we discuss the comparison of other studies at the NTCIR-9 SpokenDoc task. Seven teams participated in the task. **Table 5** shows the STD evaluation results for all the participants. All the STD systems were evaluated for the CORE set, in which 31 of the 50 queries were OOV queries. We submitted two STD systems: one was the PTN with false detection control and the other was the PTN without the control, and these are denoted in the table as "ALPS-1" and "ALPS-2," respectively.

Among all the systems, our proposed technique ("ALPS-1") realized the best F-measure and MAP values. The other studies excepting "NKGW-1" did not use many multiple speech recognizers. "NKGW-1" used both the word-based and syllable-based recognition results with N-best hypotheses [28]. Among all the participants, "NKGW-1" realized the second best F-measure because the index formed of N-best hypotheses should have more abundant information than that of a single speech recognizer. Thus, an index having a lot of information improves the STD performance.

Our technique realized the best performance, but our search engine could not rapidly search for terms. "AKBL-1/2," "NKGW-1," and "NKI11-1/2" achieved fast searches, which can find terms for a query in less than 1 second [27]. On the other hand, our engine took approximately 5 seconds for each query.

6. Conclusion

This study describes the STD techniques for OOV queries.

First, we introduced PTN-based indexing, which is essentially a phoneme-based CN. One of the aims of this study was to use multiple outputs of speech recognition systems to construct the PTN-formed index for the STD, which is different from the subword-based approaches proposed earlier.

The experimental results showed that the PTN-formed index with the DTW framework improved the OOV STD performance when it is compared with that of simple and CN-formed indices from the single speech recognizer's output. Finally, the use of multiple recognizers achieved 20% and 11% improvements in the OOV search task compared with the simple and CN-formed indices, respectively.

However, using the 10 speech recognizers resulted in many false detections such as reducing the precision to lower than 60% of the recall-rate range. In order to overcome this problem, we applied the false detection control parameters, majority voting, and the width of the arc in PTN to the DTW framework. The results indicated that false detections were effectively controlled in the OOV query set. In addition, our proposed searching methods were found to achieve better results than the other studies of the NTCIR-9 STD task.

In future, we intend to develop a fast search algorithm under the DTW framework because the processing speed of our engine is still very slow for practical applications.

Acknowledgments This work was supported by JSPS KAKENHI Grant-in-Aid for Young Scientists (B) Grant Number 23700111 and Grant-in-Aid for Scientific Research (C) Grant Number 24500225.

References

[1] Garofolo, J.S., Auzanne, C.G.P. and Voorhees, E.M.: The TREC Spoken Document Retrieval Track: A Success Story, *Proc. Text Retrieval Conference* (*TREC*) *8*, pp.16–19 (2000).

[2] NIST: The Spoken Term Detection (STD) 2006 evaluation plan (2006). available from ⟨http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf⟩

[3] Vergyri, D., Shafran, I., Stolcke, A., Gadde, R.R., Akbacak, M., Roark, B. and Wang, W.: The SRI/OGI 2006 Spoken Term Detection System, *Proc. 8th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2007*), pp.2393–2396 (2007).

[4] Meng, S., Shao, J., Yu, R.P., Liu, J. and Seide, F.: Addressing the Out-of-Vocabulary Problem for Large-scale Chinese Spoken Term Detection, *Proc. 9th Annual Conference of the International Speech*

Communication Association (*INTERSPEECH2008*), pp.2146–2149 (2008).

[5] Fiscus, J.G.: A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER), *Proc. 1997 IEEE Workshop on Automatic Speech Recognition and Understanding* (*ASRU'97*), pp.347–354 (1997).

[6] Utsuro, T., Kodama, Y., Watanabe, T., Nishizaki, H. and Nakagawa, S.: An Empirical Study on Multiple LVCSR Model Combination by Machine Learning, *Proc. Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (*HLT-NAACL 2004*), pp.13–16 (2004).

[7] Itoh, Y., Nishizaki, H., Hu, X., Nanjo, H., Akiba, T., Kawahara, T., Nakagawa, S., Matsui, T., Yamashita, Y. and Aikawa, K.: Constructing Japanese Test Collections for Spoken Term Detection, *Proc. 11th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2010*), pp.677–680 (2010).

[8] Natori, S., Nishizaki, H. and Sekiguchi, Y.: Japanese Spoken Term Detection Using Syllable Transition Network Derived from Multiple Speech Recognizers' Outputs, *Proc. 11th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2010*), pp.681–684 (2010).

[9] Natori, S., Nishizaki, H. and Sekiguchi, Y.: Network-formed Index from Multiple Speech Recognizers' Outputs on Spoken Term Detection, *Proc. 2nd Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* (*APSIPA ASC 2010*) (*student symposium*), p.1 (2010).

[10] Obata, T., Sugiyama, T., Hoki, K. and Ito, T.: Consultation Algorithm for Computer Shogi: Move Decisions by Majority, *Computers and Games, Lecture Notes in Computer Science*, Vol.6515/2011, pp.156–165 (2011).

[11] Motlicek, P., Valente, F. and Garner, P.N.: English Spoken Term Detection in Multilingual Recordings, *Proc. 11th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2010*), pp.206–209 (2010).

[12] Chan, C.-A. and Lee, L.-C.: Unsupervised Spoken-Term Detection with Spoken Queries Using Segment-based Dynamic Time Warping, *Proc. 11th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2010*), pp.693–696 (2010).

[13] Wang, D., King, S., Evans, N. and Troncy, R.: CRF-based Stochastic Pronunciation Modeling for Out-of Vocabulary Spoken Term Detection, *Proc. 11th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2010*), pp.1668–1669 (2010).

[14] Liu, X., Gales, M.J.F. and Woodland, P.C.: Language Model Cross Adaptation For LVCSR System Combination, *Proc. 11th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2010*), pp.342–345 (2010).

[15] Iwata, K., Shinoda, K. and Furui, S.: Robust Spoken Term Detection Using Combination of Phone-based and Word-based Recognition, *Proc. 9th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2008*), pp.2195–2198 (2008).

[16] Mamou, J., Mass, Y., Ramabhadran, B. and Sznajder, B.: Combination of Multiple Speech Transcription Methods for Vocabulary Independent Search, *Proc. 2nd Workshop on Searching Spontaneous Conversational Speech* (*SSCS2008*), pp.20–27 (2008).

[17] Wallace, R., Baker, B., Vogt, R. and Sridharan, S.: The Effect of Language Models on Phonetic Decoding for Spoken Term Detection, *Proc. 3rd Workshop on Searching Spontaneous Conversational Speech* (*SSCS2009*), pp.31–36 (2009).

[18] Gao, J., Zhao, Q., Yan, Y. and Shao, J.: Efficient System Combination for Syllable-confusion-network-based Chinese Spoken Term Detection, *Proc. International Symposium on Chinese Spoken Language Processing* (*ISCSLP2008*), pp.366–369 (2008).

[19] Han, I., Park, C., Cho, J. and Kim, J.: A Hybrid Approach to Robust Word Lattice Generation Via Acoustic-Based Word Detection, *Proc. 11th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2010*), pp.210–213 (2010).

[20] Iwami, K., Fujii, Y., Yamamoto, K. and Nakagawa, S.: Out-of-vocabulary Term Detection by N-gram Array with Distance from Continuous Syllable Recognition Results, *Proc. 2010 IEEE Workshop on Spoken Language Technology* (*SLT2010*), pp.200–205 (2010).

[21] Katsurada, K., Teshima, S. and Nitta, T.: Fast Keyword Detection Using Suffix Array, *Proc. 10th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2009*), pp.2147–2150 (2009).

[22] Kaneko, T. and Akiba, T.: Metric Subspace Indexing for Fast Spoken Term Detection, *Proc. 11th Annual Conference of the International Speech Communication Association* (*INTERSPEECH2010*), pp.689–692 (2010).

[23] Kodama, Y., Utsuro, T., Nishizaki, H. and Nakagawa, S.: Experimental Evaluation on Confidence of Agreement among Multiple Japanese LVCSR Models, *Proc. EUROSPEECH 2001*, pp.2549–2502 (2001).

[24] Wang, D., King, S. and Frankel, J.: Stochastic Pronunciation Modelling for Out-of-Vocabulary Spoken Term Detection, *IEEE Trans. on Audio, Speech, and Language Processing*, Vol.19, No.4, pp.688–698 (2011).

[25] Lee, A. and Kawahara, T.: Recent Development of Open-Source Speech Recognition Engine Julius, *Proc. 1st Asia-Pacific Signal and Information Processing Association Annual Summit and Conference* (*APSIPAASC2009*), pp.131–137 (2009).

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

[27] Akiba, T., Nishizaki, H., Aikawa, K., Kawahara, T. and Matsui, T.: Overview of the IR for Spoken Documents Task in NTCIR-9 Workshop, *Proc. 9th NTCIR Workshop Meeting*, pp.223–235 (2011).

[28] Iwami, K. and Nakagawa, S.: High Speed Spoken Term Detection by Combination of N-gram Array of a Syllable Lattice and LVCSR Result for NTCIR-SpokenDoc, *Proc. 9th NTCIR Workshop Meeting*, pp.242–248 (2011).

Satoshi Natori was born in 1985. He received his B.E. and M.E. degrees in computer and media sciences from University of Yamanashi in 2007 and 2010, respectively. He is now an engineer working for TOKYO ELECTRON TS LIMITED in Yamanashi.

Yuto Furuya was born in 1990. He received his B.E. degrees in computer and media sciences from University of Yamanashi in 2011. He is now a master course student at Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi. His research interests include spoken language processing. He is a student member of the ASJ.

Hiromitsu Nishizaki was born in 1975. He received his B.E., M.E., and D.Eng. degrees in information and computer sciences from Toyohashi University of Technology in 1998, 2000, and 2003. He is now an assistant professor in the Department of Research, Interdisciplinary Graduate School of Medicine and Engineering at University of Yamanashi. His research interests include spoken language processing. He is a member of IEEE, the IEICE, IPSJ, and the ASJ.

Yoshihiro Sekiguchi was born in 1948. He received his B.E., and M.E. degrees in electronics from Yamanashi University in 1971 and 1973, and he received his D.Eng. degree in information and computer sciences from Kyoto University in 1985. He is now a professor in Department of Research, Interdisciplinary Graduate School of Medicine and Engineering at University of Yamanashi. His research interests include spoken language processing. He is a member of the Institute of Electrical Engineers of Japan (IEEJ), the IEICE, IPSJ, and the ASJ.