Spoken Term Detection Using Multiple Speech Recognizers' Outputs at NTCIR-9 SpokenDoc STD subtask

Hiromitsu Nishizaki
University of Yamanashi
4-3-11 Takeda, Kofu
Yamanashi, 400-8511, Japan
hnishi@yamanashi.ac.jp

Hiroto Furuya
University of Yamanashi
4-3-11 Takeda, Kofu
Yamanashi, 400-8511, Japan
furuya@alps-lab.org

Satoshi Natori
University of Yamanashi
4-3-11 Takeda, Kofu
Yamanashi, 400-8511, Japan
natori@alps.cs.yamanashi.ac.jp

Yoshihiro Sekiguchi
University of Yamanashi
4-3-11 Takeda, Kofu
Yamanashi, 400-8511, Japan
sekiguti@yamanashi.ac.jp

ABSTRACT

This paper describes spoken term detection (STD) with false detection control using a phoneme transition network (PTN) derived from multiple speech recognizers' outputs at NTCIR-9 SpokenDoc STD subtask. Using the output of multiple speech recognizers, the PTN method is effective at correctly detecting out-of-vocabulary (OOV) terms and is robust to certain recognition errors. However, it exhibits a high false detection rate. Therefore, we applied two false detection control parameters to the search engine that accepts the PTN-formed index. One of the parameters is based on the consept of the majority voting scheme, and the other is a measure of ambiguity in confusion networks (CN). These parameters improve the STD performance (F-measure value of 0.725) compared to that without any parameters (F-measure value of 0.714).

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Performance

Keywords

Multiple recognizers, NTCIR-9, phoneme transition network, spoken term detection, voting

Term name: [**ALPS**]
Subtask: [**Spoken Term Detection**]
Language: [**Japanese**]

1. INTRODUCTION

Recently, information technology environments have evolved in which numerous audio and multimedia archives, such as video archives, and digital libraries, can be easily used. In particular, a rapidly increasing number of spoken documents, such as broadcast programs, spoken lectures, and recordings of meetings, are archived, with some of them accessible through the Internet. Although the need to retrieve such spoken information is growing, an effective retrieval technique is currently not available; thus, the development of technology for retrieving such information has become increasingly important.

In the Text REtrieval Conference (TREC) Spoken Document Retrieval (SDR) track hosted by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA) in the second half of the 1990s, many studies of SDR were presented using English and Mandarin broadcast news documents [3]. 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 [10]. The aim of STD is to find the position of spoken terms selected for evaluation from an audio archive.

The difficulty in STD lies in the search for terms in a vocabulary-free framework, because search terms are not known a prior to the speech recognizer being used. Many studies dealing STD tasks have been proposed, such as [12, 7]. Most STD studies focus on the out-of-vocabulary (OOV) and speech recognition error problems. For example, STD techniques that use entities such as sub-word lattice and confusion network (CN) have been proposed.

This paper proposes the use of a phoneme trasition network (PTN)-formed index derived from multiple speech recognizers' 1-best hypothesis and a dynamic time warping (DTW) framework with false control parameters applied at the term searching.

PTN-based indexing originates from the idea of the CN being generated from a speech recognizer. CN-based indexing for STD is a powerful indexing method. The multiple speech recognizers cangenerate the PTN-formed index by combining sub-word (phoneme) sequences from the output of these recognizers into a single CN. This study uses 10 types of speech recognition systems with the same decoder used for all. Two types of acoustic models (triphone-based and syllable-based Hidden Markov Models (HMMs)) and five types of language models (word-based or sub-word based) were prepared.

The use of the 10 recognizers and their output is very ef-

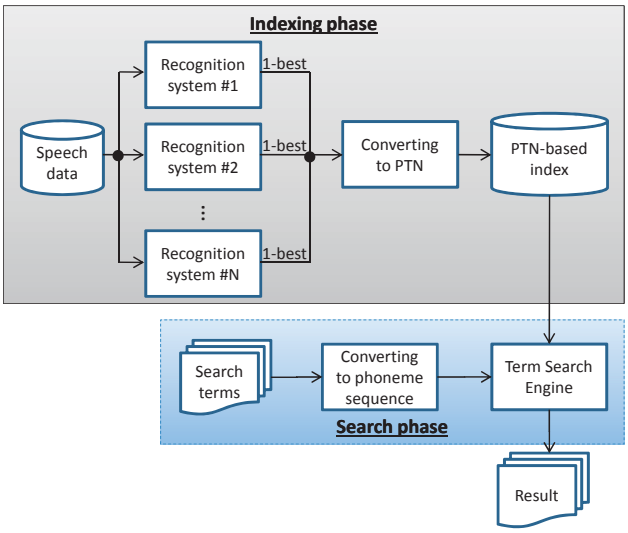

Figure 1: Overview of the STD framework.

fective in improving speech recognition performance. For example, Fiscus [2] proposed the ROVER method that adopts a word voting scheme. Utsuro et al. [11] developed a technique for combining multiple recognizers' output by using a support vector machine (SVM) to improve speech recognition performance. Application of the characteristics of the word (or sub-word) sequence output by recognizers may increase STD performance since these characteristics are different for each speech recognizer. The PTN-based on multiple speech recognizers' output can cover more sub-word sequences of spoken terms. Thus, multiple speech recognizers may improve STD performance compared to that of a single recognizer's output.

We evaluated the PTN-formed index derived from the 10 recognizers' output. The experimental result for the Japanese STD test collection [4] showed that using the PTN formed index was effective in improving STD performance compared to that of the CN-formed index, which was derived from the phoneme-based CN made up to 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 mistakenly recognized by some speech recognizers. Using more speech recognizers can achieve a higher speech recognition performance but more errors may occur. Therefore, we adopted a majority voting parameter and a measure of ambiguity, which are easily derived from the PTN, into the term search engine.

We installed the voting and ambiguity parameters of the PTN to our term search engine to prevent false detections. The improved term search engine decreased the number of false detections for the formalrun query for CORE at the NTCIR-9 SpokenDoc STD subtask. In addition, our STD technique achieved the best performance values among the other teams' results at the same task.

2. STD FRAMEWORK

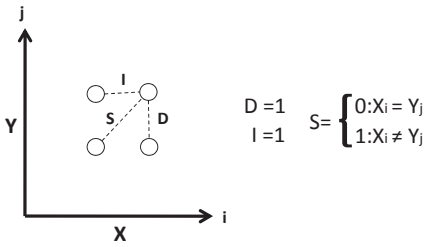

Figure 3: Definition of DTW path.

2.1 Outline

Figure 1 represents an outline of the STD framework in this paper.

In the indexing phase, speech data is processed by speech recognition and the output (word or sub-word sequences) is converted into the PTN index for STD. In the search phase, the word formed query is converted into the phoneme sequence, and then the phoneme-formed query is input to the term search engine. In the case of English queries, we have to consider the variety of pronunciations of the queries. There are some reports that discuss the pronunciation problem[13].

In this paper, however, we only handle Japanese STD. Most Japanese words can be fully translated into phoneme sequence (pronunciation) using a dictionary in which relationship between words and their pronounciations is listed. In addition to this, most Japanese words have no ambiguous pronounciations. Therefore, we do not consider the pronunciation problem in this paper.

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

2.2 PTN-based indexing

Figure 2 shows an example of the construction of a PTN-formed index of an utterance "*cosine*" (Japanese pronunciation is /k o s a i N/) by performing the alignment process of N of phoneme sequences from 1-best hypothesis of the recognizer. We use 10 types of speech recognizers to create the PTN formed index. The utterance is recognized by the 10 recognizers to yield the 10 hypotheses, which are then converted into the phoneme sequences (Figure 2). Next, we obtain "aligned sequences" by using a dynamic programming (DP) scheme described in [2]. Finally, the PTN is obtained by converting the aligned sequences. The "@" in Figure 2 indicates a null transition. Arcs between nodes in the PTN have some phonemes and null transitions with an occurrence probability. However, in this paper, we do not use any phoneme occurrence probabilities.

2.3 Term search engine with false detection control

We adopt the DTW-based word-spotting method. The paths of the DTW lattice are shown in Figure 3. X and Y indicate index and query terms, respectively.

Figure 4 shows an example of the DTW framework. In the example, the search term "k o s a i N" (cosine) and the PTN-formed index are used. The PTN has multiple arcs between two adjoining nodes. These arcs are compared to one of the phoneme labels of a query term.

We use edit distance as cost on the DTW paths, and the

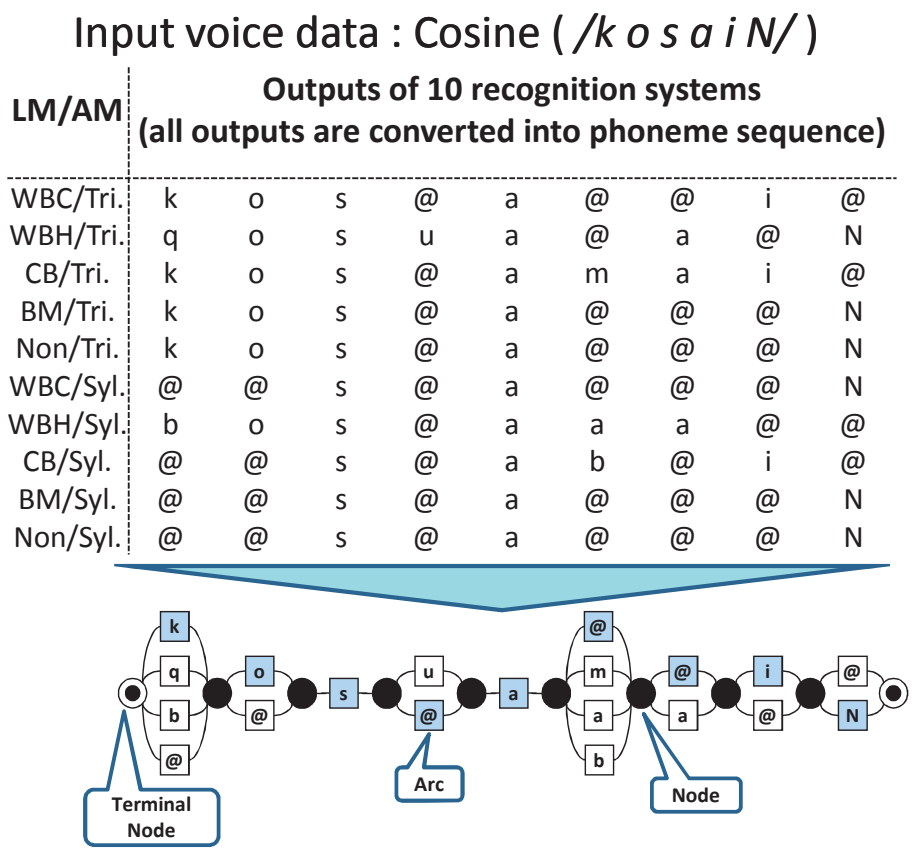

Figure 2: Making PTN-formed index by performing alignment using a DP and converting to a PTN.

cost value for substitution, insertion and deletion errors is set as 1.0. The total cost $D(i,j)$ at the grid point (i,j) ($i = \{0,...,I\}$, $j = \{0,...,J\}$, where I and J are the numberss of the set of arcs in an index and a query term, respectively) on the DTW lattice is calculated using the following equations:

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

$$Match(i,j) = \begin{cases} 0.0 : Query(j) \in PTN(i) \\ 1.0 : Query(j) \notin PTN(i) \end{cases} \quad (2)$$

$$NULL(i) = \begin{cases} 0.1 : NULL \in PTN(i) \\ 1.0 : NULL \notin PTN(i) \end{cases} \quad (3)$$

where $PTN(i)$ is the set of phoneme labels of the arcs at the i-th node in the PTN, and $Query(j)$ indicates the j-th phoneme label in the query term. When the query term equals null (@) in the PTN, the transition cost is set as 0.1. This value is empirically determined in this paper.

"$Vot(i,j)$" and "$Acw(i)$" in Equation (1) are related to the false detection control parameters and calculated as follows:

$$Vot(i,j) = \begin{cases} \frac{\alpha}{Voting(p)} : \\ \quad \exists p \in PTN(i), p = Query(j) \\ 1.0 : Query(j) \notin PTN(i) \end{cases} \quad (4)$$

$$Acw(i) = \beta \cdot ArcWidth(i) \quad (5)$$

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

We provided two types of parameters to control false detection:

- "$Voting(p)$" is the number of speech recognizers outputting the same phoneme p at the same arc. The higher the value of $Voting(p)$, the higher the reliability of phoneme p.
- "$ArcWidth(i)$" is the number of arcs (phoneme labels) at $PTN(i)$. The lower the value of $ArcWidth(i)$, the higher the reliability of phonemes at $PTN(i)$.

We allow a null transition between two nodes in the PTN-formed index with cost= 0.1. Thus, the values of $Vot(i,j)$ and $Acw(i)$ must be less than the null transition cost. Therefore, the value of α is set below 1.0. When $Voting(p) = 1$, if α is 0.1 or more than 0.1, the null are given priority in the term search process. However, this may confuse the voting parameter. In this paper, we empirically set α as 0.5, which means that $Voting(p)$ value of more than 5 can be considered reliable.

β must be set in the range from 0.01 to 0.1 because of the same reason. We empirically set β as 0.01. This means that less than 10 of $ArcWidth(i)$ can be considered reliable.

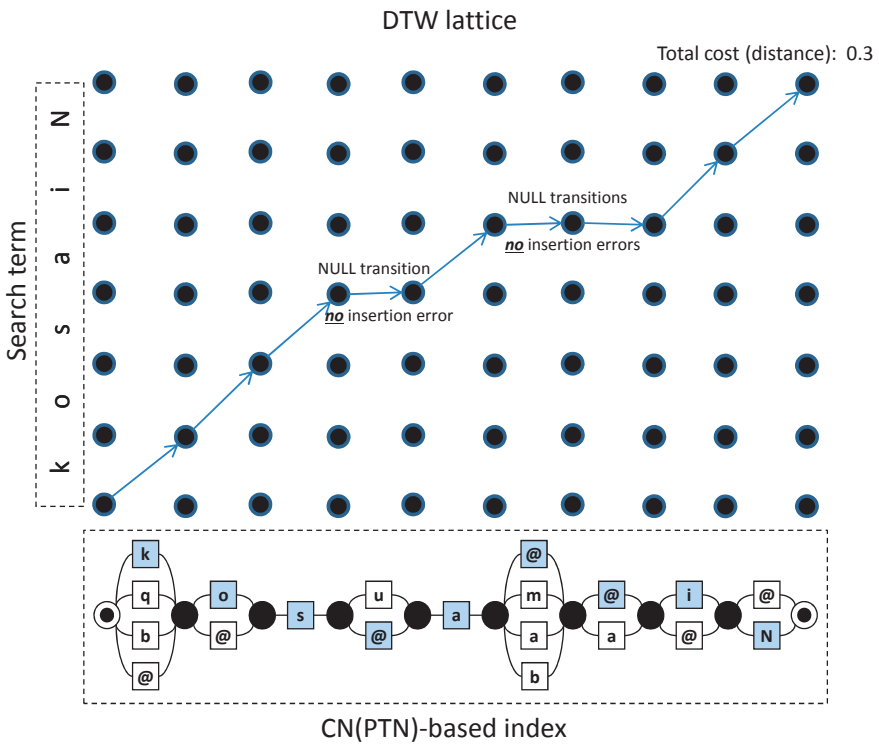

Figure 4: Example of term search using the network formed index.

In advance searching of the query term, the term search engine initializes $D(i,0) = 0$ and then calculates $D(i,j)$ using Equation(1) ($i = \{0, ..., I\}$, $j = \{1, ..., J\}$). Furthermore, $D(i,J)$ are normalized by the length of the DTW path.

After the calculation is complete, the engine outputs the detection candidates that have a normalized cost $D(i,J)$ below a threshold θ. Changing the θ value enables us to control the recall and precision rates on STD performance.

3. STD EXPERIMENT

3.1 Speech Recognition

As shown in Figure 1, the speech data is processed by 10 speech recognizers. Julius ver. 4.1.3 [5], an open source decoder for LVCSR, is used in all the systems.

We prepared two types of acoustic models (AMs) and five types of language models (LMs) for constructing the PTN. The AMs are triphone-based (Tri.) and syllable-based HMMs (Syl.), and both types of HMMs are trained on spoken lectures in the Corpus of Spontaneous Japanese (CSJ) [6].

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

WBC : word-based trigram with 27k vocabulary 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 two of Hiragana characters.

Non : No LM is used. Speech recognition without any LM is equivalent to phoneme (or syllable) recognition.

Each model is trained using the many transcriptions in the CSJ under the open condition for the speech data of STD. The condition is completely the same as the description in the overview paper [1].

Finally, the 10 combinations, comprising two AMs and five LMs, are formed.

3.2 Query set of the STD subtask

The SpokenDoc organizers provided two types of query sets; one is for the all lectures and the other is for only the core lectures in CSJ[1].

In this paper, however, we use only 177 speeches (44 hours), called the "CORE," from the all lecture speeches in CSJ. Therefore, we use the CORE query set among the STD formal-run query sets.

The CORE query set has 50 query terms, and 31 of the all 50 queries are out-of-vocabulary queries which do not included in the ASR dictionary and the others are in-vocabulary queries.

3.3 Evaluation metric

The official evaluation measure for effectiveness is F-measure at the decision point, based on recall and precision averaged over queries (described as "F-measure(spec.)"). F-measure

at the maximum decision point (described as "F-measure(max)"), Recall-Precision curves and mean average precision (MAP) will also be used for analysis purpose.

They are defined as follows:

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

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

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

where N_{corr} and $N_{spurious}$ are the total number of correct and spurious (false) term (IPU) detections whose scores are greater than or equal to the threshold, and N_{true} is the total number of true term occurrences in the speech data. Recall-precision curves can be plotted by changing the threshold value. In the evaluation, the threshold value is varied in 100 steps. F-measure at the maximum decision point is calculated at the optimal balance of *Recall* and *Precision* values from the recall-precision curve.

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

MAP for the set of queries is the mean value of the average precision values for each query. It can be calculate as follows:

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

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

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

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

3.4 Experimental result

Figure 5 shows the recall-precision curves that show the STD performance for the formal-run CORE query set using the term search engine with the false detection control parameters or without any parameters. Table 1 also indicates the F-measure and MAP values for each false detection parameter on the same test set.

The decision point for culculating "F-measure(spec.)" was decided by the result of the NTCIR-9 SpokenDoc STD dry-run query set. We adjusted the threshold to be the best F-measure value on the dry-run set which was used as a development set.

"Only edit distance (ALPS-2)" indicates the term search engine did not use any parameters. This results is the same as the one labeled as "ALPS-2" on the overview paper of the SpokenDoc[1]. "Voting + ArcWidth (ALPS-1)" indicates that the engine used both the parameters. "Baseline" shows the result provided by the SpokenDoc organizers. The baseline system has a dynamic programming (DP) based word spotting. The score between a query term and an IPU is calculated based on phoneme-based edit distance.

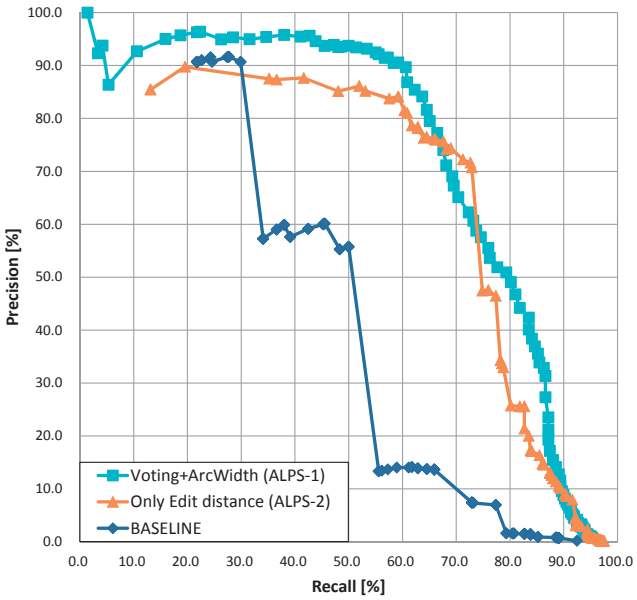

Figure 5: Recall-precision curves for the CORE query set with the false detection control technique.

The effectiveness of our STD technique has been showed in the previous papers [8, 9]. In addition to this, as shown in Figure 5 and Table 1, using multiple recognizers' outputs both with the false detection control parapeters and without them is very effective to improve the STD performance comparing with the Baseline for the NTCIR-9 SpokenDoc STD task.

The false detection control parameters slightly decreases false detections on a wide range of recall rates on the recall-precision curve. The F-measure(max) and F-measure(spec.) are appreciably improved to 0.725 and 0.708, respectively. However, the control parameters significantly improved the MAP value from 0.757 to 0.837.

The experiment with false detection control indicates that the majority voting scheme using multiple speech recognizers is a very powerful technique.

3.5 Time consumption of the STD processing

Our technique has high performance on a precision-recall curve and a MAP value but needs too much time to search a term from the large scale speech data.

It takes 13.5 seconds to search a term from the CSJ CORE speeches (44 hours data) on a computer which has "Intel Core i7 975 3.33GHz" CPU.

4. CONCLUSION

This paper describes the STD techniques for the NTCIR-9 SpokenDoc STD subtask.

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 for constructing the PTN-formed index for STD, which is different from the sub-word based approaches proposed earlier.

Table 1: F-measure and MAP values for each false detection parameter for the CORE query set.

Parameter	F-measure(max)	F-measure(spec.)	MAP
Baseline	0.527	0.398	0.596
Only Edit distance (ALPS-2)	0.714	0.697	0.757
Voting + ArcWidth (ALPS-1)	0.725	0.708	0.837

In our previous works [8, 9], the experimental results showed that PTN-based indexing improving the STD performance in searching the OOV terms under the DTW framework, compared with the simple index and CN-formed index from the single speech recognizer's output.

However, using the 10 speech recognizers resulted in a lot of false detections. Therefore, we installed the false detection control parameters, majority voting and the width of the arc in the PTN, to the DTW framework. The results indicate that this was very effective in controlling false detection in the query term set.

In future work, we intend to develop a fast search algorithm under the DTW framework. The processing speed of our engine is too slow for practical use. In addition, we want to use our methods on the English test collection of STD by NIST.

5. ACKNOWLEDGMENTS

This work was supported by KAKENHI (Grant-in-Aid for Young Scientists(B), 23700111).

6. REFERENCES

[1] T. Akiba, H. Nishizaki, K. Aikawa, T. Kawahara, and T. Matsui. Overview of the ir for spoken documents task in NTCIR-9 workshop. In *Proceedings of the NTCIR-9 Workshop*, page 8 pages, 2011.

[2] J. G. Fiscus. A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER). In *Proceedings of the 1997 IEEE Workshop on Automatic Speech Recognition adn Understanding (ASRU'97)*, pages 347–354, 1997.

[3] J. S. Garofolo, C. G. P. Auzanne, and E. M. Voorhees. The trec spoken document retrieval track: A success story. In *Proceedings of the Text Retrieval Conference (TREC) 8*, pages 16–19, 2000.

[4] Y. Itoh, H. Nishizaki, X. Hu, H. Nanjo, T. Akiba, T. Kawahara, S. Nakagawa, T. Matsui, Y. Yamashita, and K. Aikawa. Constructing japanese test collections for spoken term detection. In *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH2010)*, pages 677–680. ISCA, 2010.

[5] 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)*, page 6 pages, 2009.

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

[7] S. Meng, J. Shao, R. P. Yu, J. Liu, and F. Seide. Addressing the out-of-vocabulary problem for large-scale chinese spoken term detection. In *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH2008)*, pages 2146–2149. ISCA, 2008.

[8] 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 (INTERSPEECH2010)*, pages 681–684. ISCA, 2010.

[9] S. Natori, H. Nishizaki, and Y. Sekiguchi. Network-formed index from multiple speech recognizers' outputs on spoken term detection. In *the proceedings of the 2nd Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2010) (student symposium)*, page 1, 2010.

[10] The spoken term detection (STD) 2006 evaluation plan, 2006. http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf.

[11] T. Utsuro, Y. Kodama, T. Watanabe, H. Nishizaki, and S. Nakagawa. An empirical study on multiple LVCSR model combination by machine learning. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2004)*, pages 13–16, 2004.

[12] D. Vergyri, I. Shafran, A. Stolcke, R. R. Gadde, M. Akbacak, B. Roark, and W. Wang. The SRI/OGI 2006 spoken term detection system. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH2007)*, pages 2393–2396. ISCA, 2007.

[13] D. Wang, S. King, and J. Frankel. Stochastic pronunciation modelling for out-of-vocabulary spoken term detection. *IEEE Trans. on Audio, Speech, and Language Processing*, 19(4):688–698, 2011.