Using Multiple Speech Recognition Results to Enhance STD with Suffix Array on the NTCIR-10 SpokenDoc-2 Task

Kouichi Katsurada
Toyohashi Univ. of Tech.
1-1 Hibarigaoka, Tempaku-cho
Toyohashi 441-8580, JAPAN
+81-532-44-6884
katsurada@cs.tut.ac.jp

Koudai Katsuura
Toyohashi Univ. of Tech.
1-1 Hibarigaoka, Tempaku-cho
Toyohashi 441-8580, JAPAN
+81-532-44-6884
katsuura@vox.cs.tut.ac.jp

Kheang Seng
Toyohashi Univ. of Tech.
1-1 Hibarigaoka, Tempaku-cho
Toyohashi 441-8580, JAPAN
+81-532-44-6884
kheang@vox.cs.tut.ac.jp

Yurie Iribe
Toyohashi Univ. of Tech.
1-1 Hibarigaoka, Tempaku-cho
Toyohashi 441-8580, JAPAN
+81-532-44-6638
iribe@imc.tut.ac.jp

Tsuneo Nitta
Waseda University
27-40-305-2 Waseda-cho, Shinjuku-ku
Tokyo 162-0042, JAPAN
+81-3-3203-4450
nitta@cs.tut.ac.jp

ABSTRACT
We have previously proposed a fast spoken term detection method that uses a suffix array as a data structure. By applying dynamic time warping on a suffix array, we achieved very quick keyword detection from a very large-scale speech document. In this study, we modify our method so that it can deal with multiple recognition results. By using these results obtained from various speech recognizers, search performance will improve as a consequence of the complementary effect of using different language and acoustic models. Experimental results show the maximum value of F-measure and the MAP score increased by 6% to 10%.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Search process.

General Terms
Experimentation.

Keywords
Spoken term detection, large scale speech document, suffix array, multiple recognition results.

Team Name
NKI-lab.

Subtasks
Spoken Term Detection.

1. INTRODUCTION
A substantial amount of research has been conducted on spoken term detection (STD) since it was selected as a benchmark test at NIST in 2006 [1]. The main goal of this research has been improvement of search performance; additionally, high-speed search is becoming increasingly relevant as large-scale speech documents are employed as target databases [2][3][4]. We have proposed a fast STD method that uses a suffix array as a data structure [5][6]. By applying dynamic time warping (DTW) to a suffix array, we have achieved very quick keyword detection with a very large-scale speech document. In this study, we modify our method so that it can deal with multiple recognition results.

Various methods that use multiple recognition results have been proposed. The most common approach is to use two recognition results: one obtained using a word N-gram language model, and the other using a sub-word N-gram language model. This type of approach usually employs a two-way search process according to the given keyword [2][7]. If the given keyword is an in-vocabulary word, the word N-gram language model-based result is used for taking advantage of word bias in the recognition process. Otherwise, the sub-word N-gram language model-based result is used for avoiding phonetically erroneous recognition results coming from the word language model.

Other approaches use a unique data structure that does not separate the search process. Nishizaki et al. uses 10 speech recognizers constructed from five language models and two acoustic models [8]. By applying DTW to the confusion network constructed from the speech recognition results obtained from these 10 recognizers, they achieved an improvement in the search results of more than 20%. The advantage of this approach is derived from its use of multiple speech recognizers, which leads to an improvement in search performance, which is a desirable feature for more accurate STD. However, the search speed of this method is slow because DTW is applied from the beginning of the confusion network to its terminal sequentially.

In this paper, we propose a very quick and accurate STD method that uses a suffix array as a data structure and multiple recognition results as the target speech documents. The main idea is to construct a suffix array from combined multiple recognition results. A suffix array guarantees high speed of the search, while multiple recognition results increase the search performance. We conduct the experiments with one to three recognition results obtained by using different language models. We evaluate our approach in terms of both search performance and speed.

This paper is organized as follows. In section 2, we outline our method. In section 3, we conduct some experiments to

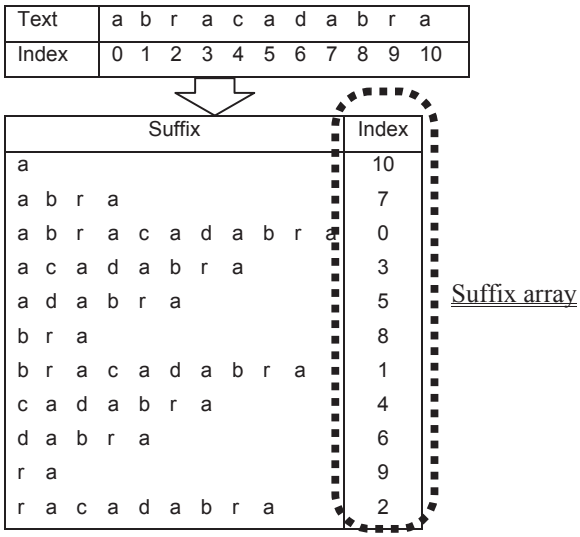

Figure 1: *Example suffix array*.

	a	i	u	e	o	k	s	...
low	-	+	+	-	-	-	-	
high	+	-	-	-	-	+	-	
plosive	-	-	-	-	-	+	-	
affricative	-	-	-	-	-	-	-	
:								

Figure 2: *Table of distinctive phonetic features*.

demonstrate the effectiveness of our method. Lastly, in section 4, we conclude our work and provide an outline of future work.

2. OUTLINE OF OUR METHOD

Our search method uses a suffix array as a data structure to which DTW is applied. In this section, we first outline our previous method, which used a single recognition result, and then explain how to extend it to deal with multiple recognition results. For more details on our previous method, please refer to our former papers [5][6][9].

2.1 Similarity search on a suffix array

A suffix array [10] is a data structure used for quickly searching for keywords in a text database. We employ it for phoneme-based keyword detection. The array holds sorted indexes of all suffixes of the phoneme string in a database, as shown in Figure 1. The index values in the figure represent the position at which the suffixes start in the string. Because the indexes are sorted by the dictionary order of suffixes, we can use a quick-search algorithm on it. However, the original suffix array should be used for exact search. Consequently, we need to introduce a technique for a similarity search to use alongside the suffix array. For this purpose, a search algorithm using DTW on the suffix array is proposed [11]. This algorithm regards a suffix array as a tree, and DTW is applied to all paths from the root of the tree. We employ distinctive phonetic features to define the distance between phonemes used in the DTW process. The distinctive phonetic features represent a phoneme using 15 articulatory features such as plosive and affricative. Figure 2 shows a fragment of the

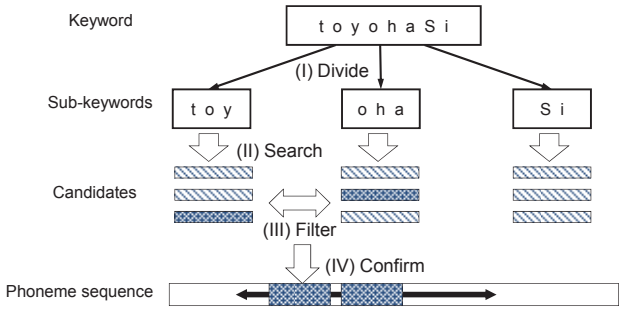

Figure 3: *Outline of our previous keyword search*.

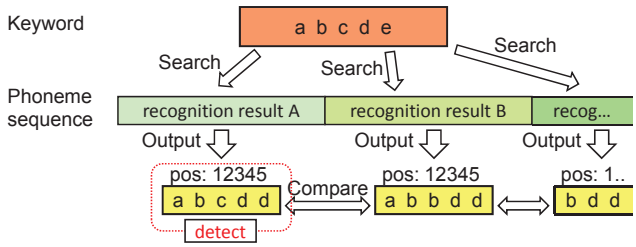

Figure 4: *Keyword search using multiple recognition results*.

relationship between phonemes and articulatory features. We used the Hamming distance of these features to calculate the distance between two phonemes.

2.2 Keyword division

The search method described in the previous section has an issue that, if the keyword is long, the search time increases exponentially because all paths within the threshold are temporarily stored in the memory. To avoid this problem, a long keyword is divided into short sub-keywords, which are then searched for in the array instead of the original keyword.

Of course, the results obtained by using sub-keywords, hereafter referred to as the candidates, may not actually match the results when the original keyword is used. Thus, to guarantee that the same results will be obtained, we have proposed a search algorithm constructed from the following four steps [6][9]. Figure 3 illustrates the outline of a keyword search:

1. Divide the keyword into sub-keywords.
2. Search for the sub-keywords in the suffix array and find candidates.
3. Filter the candidates by detecting adjacent candidates.
4. Confirm the validity of the candidate by DTW.

In step 2, the threshold assigned to each sub-keyword is defined using the following equation:

$$T_s = \frac{T}{n - m + 1} \qquad (1)$$

where T_s is the modified threshold assigned to a sub-keyword, T is the threshold assigned to the original keyword, n is the number of sub-keywords, and at least m of n sub-keywords are detected in the adjacent area of the database. The details of both the search process and derivation of equation (1) are discussed in paper [9].

Table 1: *Results of experiments using SDPWD corpus.*

SystemID	transcription	Max F-measure (micro)	MAP	Index size [MB]	Search time [ms]
NKI13-1	(unmatch) syl+wo+amon	33.81	0.442	15.9	1.250
NKI13-2	(match) syl+wo+amon	40.24	0.456	15.6	0.860
NKI13-3	(unmatch) syl+wo	34.62	0.434	10.7	0.785
NKI13-4	(match) syl+wo	41.15	0.446	10.3	0.700
NKI13-5	(unmatch) wo+amon	28.41	0.348	10.6	0.620
NKI13-6	(match) wo+amon	37.56	0.390	10.5	0.545
NKI13-7	(unmatch) syl+amon	26.24	0.382	10.6	0.705
NKI13-8	(match) syl+amon	27.24	0.350	10.3	0.310

Table 2: *Results of experiments using CSJ corpus.*

SystemID	transcription	Max F-measure (micro)	MAP	Index size [MB]	Search time [ms]
NKI13-1	(unmatch) syl+wo	60.90	0.673	183.3	2.96
NKI13-2	(match) syl+wo	56.09	0.608	168.1	2.49
NKI13-3	(unmatch) wo	52.10	0.574	92.3	1.88
NKI13-4	(match) wo	50.58	0.511	83.0	1.23
NKI13-5	(unmatch) syl	50.56	0.566	91.0	1.87
NKI13-6	(match) syl	45.17	0.525	85.1	1.71

2.3 Construction of a suffix array from multiple recognition results

To improve search performance, we construct a suffix array from combined multiple recognition results. The flow of the search is shown in Figure 4. In the search process, multiple search results may be detected at the same position of the original speech document from multiple recognition results. In this case, we retain only the result closest to the search keyword. The remaining results are deleted.

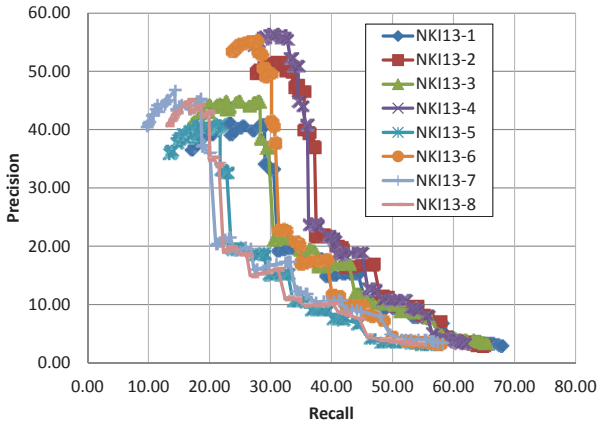

Figure 5: *Recall-precision curve of experimental results using SDPWD corpus.*

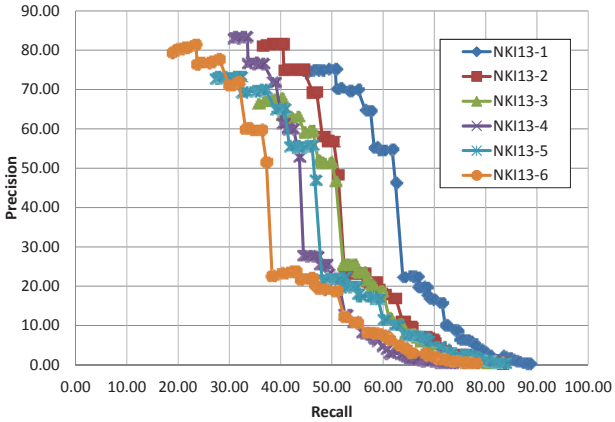

Figure 6: *Recall-precision curve of experimental results using CSJ corpus.*

3. EXPERIMENTS

3.1 Experimental setup

Experiments were conducted on a PC with a 3.4 GHz Intel Core i7-2600 processor and 8 GB main memory. CSJ (Corpus of Spontaneous Japanese) and SDPWS (corpus of Spoken Document Processing WorkShop) corpora are used to evaluate effectiveness of our method. For speech recognition we used word-based and syllable-based transcriptions provided by the NTCIR-10 SpokenDoc organizer, and the recognition results obtained from our own recognizer AMON. AMON is a phoneme recognizer that utilizes articulatory features such as place of articulation and manner of articulation. We submitted 8 results for SDPWD corpus, and 6 for CSJ corpus. We used either unmatched transcription or matched transcription in each experiment.

We set the value of m in equation (1) as 1, and n as the value where the length of sub-keywords is 6. These values were confirmed to be optimal in previous experiments [6]. Moreover, the following evaluation formula is introduced to consider the length l of the keyword:

$$score = \frac{1}{t/\sqrt{l}+1} \quad (2)$$

In the above equation, t is the threshold value par a phoneme (i.e., $t = T/l$). We attached binary decisions "yes" to the results whose score is 0.90 or more. This score is obtained by the preliminary experiment.

3.2 Experimental results

Tables 1 and 2 show the results of the experiments conducted using SDPWS corpus and CSJ corpus, respectively. Figure 5 and 6 illustrate recall-precision curve of these results. In these tables and figures, "syl," "wo" and "amon" represents the syllable-based transcription, the word-based transcription, and the result obtained from our own recognizer AMON, respectively. The sign '+' represents multiple transcriptions are used. For example, in the experiment of the second row of table 1, we used unmatched syllable and word transcriptions, and our recognizer AMON.

Tables 1 and 2 show that our method achieves very quick and accurate search with small size of index. These tables also show that the accuracy of search increases when multiple recognition results are used. Especially in the result of experiments using CSJ corpus, we could confirm that two different types of transcriptions improve the maximum F-measure values and the mean average precision scores by 6 to 10% compared with the results obtained using a single transcription. Although the index size and search time increase according to the number of transcriptions, they are enough smaller and quicker than the other results of SpokenDoc-2 [12].

However, in the experiment of SDPWD corpus, we could not achieve significant improvement by incorporating the result obtained from our recognizer AMON. This is because the phoneme recognition rate of AMON was not enough good by reason that the acoustic model does not match the corpus. If we used appropriate acoustic model in AMON, we could improve the performance of our recognizer and the accuracy of keyword detection.

4. CONCLUSIONS

We proposed a quick and accurate STD using a suffix array and multiple recognition results. The experimental results show that the maximum value of F-measure and the MAP score increased by 6 to 10%. In future work, we will attempt various combinations of recognition results and analyze which combination is the best for accurate search. We will then embed further recognition results obtained from the phoneme recognizer being developed by our research group.

5. ACKNOWLEDGEMENTS

This work has been supported by Grant-in-Aid for Young Scientists (B) 24700167 2012 and for Scientific Research (B) 22300060 2012 by MEXT, Japan, and the Kayamori Foundation of Information Science Advancement

6. REFERENCES

[1] Fiscus, J., Ajot, J., Garofolo, J. and Doddington, G., "Results of the 2006 Spoken Term Detection Evaluation", SIGIR'07 Workshop in Searching Spontaneous Conversational Speech, 2007.

[2] Kanda, N., Sagawa, H., Sumiyoshi, T., and Obuchi, Y., "Open-vocabulary keyword detection from super-large scale speech database", IEEE MMSP 2008, pp.939-944, 2008.

[3] Pinto, J., Szoke, I, Prasanna, S. R. M. and Hermansky, H., "Fast Approximate Spoken Term Detection from Sequence of Phonemes", SIGIR '08 Workshop, pp.28-33, 2008.

[4] Wallace, R., Vogt, R. and Sridharan, S., "Spoken term detection using fast phonetic decoding", ICASSP'09, pp.2135-2138, 2009.

[5] Katsurada, K., Teshima, S. and Nitta, T., "Fast Keyword Detection Using Suffix Array", InterSpeech2009, pp.2147-2150, 2009.

[6] Katsurada, K., Sawada, S., Teshima, S., Iribe, Y. and Nitta, T., "Evaluation of Fast Keyword Detection Using a Suffix Array", InterSpeech2011, pp.909-912, 2011.

[7] 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. NTCIR-9 Workshop Meeting, pp.242-248, 2011.

[8] Nishizaki, H., Furuya, Y., Natori, S. and Sekiguchi, Y., "Spoken Term Detection Using Multiple Speech Recognizers' Outputs at NTCIR-9 SpokenDoc STD subtask", Proc. NTCIR-9 Workshop Meeting, pp. 236-241, 2011.

[9] Katsurada, K., Katsuura, K., Iribe, Y. and Nitta, T., "Utilization of Suffix Array for Quick STD and Its Evaluation on the NTCIR-9 SpokenDoc Task", Proc. NTCIR-9 Workshop Meeting, pp.271-274, 2011.

[10] Manber, U. and Myers, G., "Suffix arrays: A new method for on-line string searches", SIAM J. Computation, vol.22, no.5, pp.935-948, 1993.

[11] Yamasita, T. and Matsumoto, Y., "Full Text Approximate String Search using Suffix Arrays", IPSJ SIG Technical Reports 1997-NL-121, pp.23-30, 1997. (In Japanese)

[12] Akiba, T. et al. "Overview of the NTCIR-10 SpokenDoc-2 Task", The 10th NTCIR Conference, 2013.