Utilizing Confusion Network in the STD with Suffix Array and Its Evaluation on the NTCIR-11 SpokenQuery & Doc SQ-STD 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

Genki Ishihara
Toyohashi Univ. of Tech.
1-1 Hibarigaoka, Tempaku-cho
Toyohashi 441-8580, JAPAN
+81-532-44-6884
ishihara@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
Aichi Prefectual University
1522-3 Ibaragabasama, Nagakute-shi,
Aichi 480-1198, JAPAN
+81-561-76-8793
iribe@ist.aichi-pu.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
The authors have proposed a fast spoken term detection that uses a suffix array as a data structure. This method enables very quick and memory saving search by using such techniques as keyword division, dynamic time warping, and employment of articulatory-feature-based local distance definition. In this paper, we investigate a new approach that utilizes a confusion network in the suffix array. The experimental results show that this approach has both good and bad effect on the search. Although it increases the search time, it can reduce the size of the search index. The search accuracy is almost same as the original one.

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

General Terms
Experimentation.

Keywords
Spoken term detection, suffix array, confusion network.

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][7]. By applying dynamic time warping (DTW) to a suffix array, we have achieved very quick keyword detection with a very large-scale speech document.

From the viewpoint of search accuracy, some methods have utilized multiple speech recognition results obtained from different ASR systems. The most common approach is the one using two recognition results: a word N-gram language model-based one, a sub-word N-gram language model-based one [2][8]. This type of approach usually employs a two-way search process according to the given keyword. 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 [9]. 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 by more than 20%.

We have also provided a very quick and accurate STD method that constructs a suffix array from multiple recognition results, and showed that our approach indicates good performance both in search accuracy and speed [10]. However, it has a problem that the index size proportionately increases according to the number of recognition results. To suppress the enlargement of the index size, this paper employs confusion network as a data structure and transforms it into linear structure so that it can be held as a suffix array.

This paper is organized as follows. In section 2, we outline our method. In section 3, we conduct some experiments to demonstrate the effectiveness of our method. Lastly, in section 4, we conclude our work and provide an outline of future work.

Text	a	b	r	a	c	a	d	a	b	r	a
Index	0	1	2	3	4	5	6	7	8	9	10

Suffix	Index
a	10
a b r a	7
a b r a c a d a b r a	0
a c a d a b r a	3
a d a b r a	5
b r a	8
b r a c a d a b r a	1
c a d a b r a	4
d a b r a	6
r a	9
r a c a d a b r a	2

Suffix array

Figure 1: *Example suffix array*.

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

Figure 2: *Table of distinctive phonetic features*.

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][7].

2.1 Similarity search on a suffix array

A suffix array [11] 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 [12]. 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 relationship between phonemes and articulatory features. We used the Hamming distance of these features to calculate the distance between two phonemes.

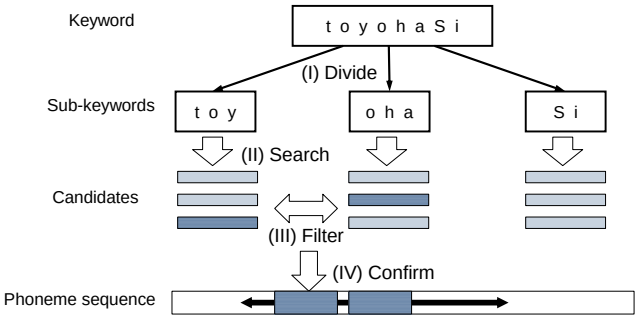

Figure 3: *Outline of our previous keyword search*.

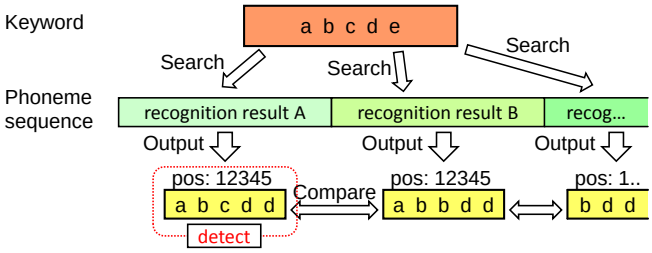

Figure 4: *Keyword search using multiple recognition results*.

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][7]. 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:

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

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 [7].

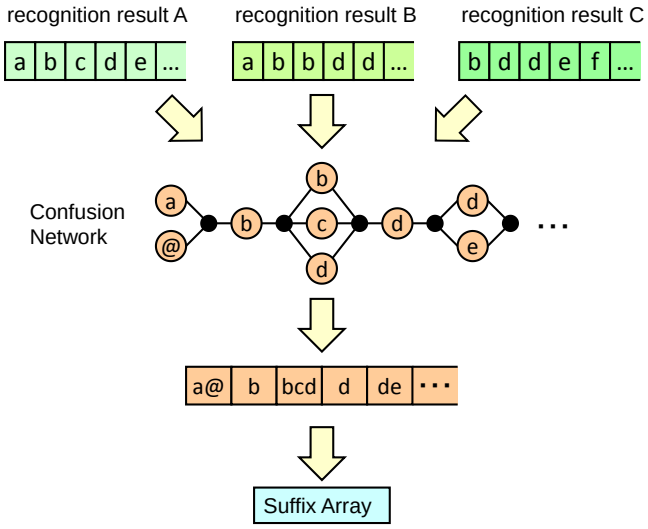

Figure 5: *Construction of suffix array from multiple results*.

2.3 Construction of a suffix array from multiple recognition results

To improve search performance, we have constructed a suffix array from the sequence of multiple recognition results [10]. The flow of the search is shown in Figure 4. Although this method increases search accuracy by around 10%, it has a problem that the size of index increases according to the number of recognition results. To suppress the enlargement of the index size, we employs confusion network as a data structure. However, confusion network cannot be stored in a suffix array because a suffix array is a linear structure while a confusion network is a configuration of restricted lattice.

In order to store a confusion network in a suffix array, we convert the confusion network by collecting the branches appeared in the confusion network into a line, and construct a linear structure that can be hold as a suffix array. Figure 5 shows the outline of our proposed approach.

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. SDPWS (corpus of Spoken Document Processing WorkShop) corpus is used to evaluate effectiveness of our method. For speech recognition we used word-based and syllable-based transcriptions provided by the NTCIR-11 SpokenDoc organizer. We used either unmatched transcription or matched transcription in each experiment. While NKI14-1 and NKI14-2 use our previous method [10] for constructing the suffix array, NKI14-3 and NKI14-4 use confusion network and its conversion method.

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:

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

Table 1: *Results of experiments*.

SystemID	transcription	Max F-measure (micro)	MAP	Index size [MB]	Search time [ms]
NKI14-1	(match) syl+wo	58.81	0.510	10.23	0.65
NKI14-2	(unmatch) syl+wo(LM) +wo(LMAM)	49.70	0.442	15.27	0.88
NKI14-3	(match,CN) syl+wo	57.06	0.506	5.47	11.70
NKI14-4	(unmatch,CN) syl+wo(LM) +wo(LMAM)	45.12	0.430	5.83	53.43

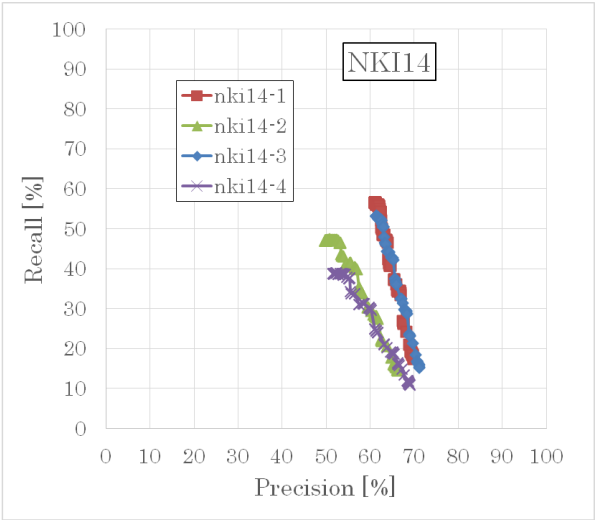

Figure 5: *Recall-precision curve of experimental results.*

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

Table 1 shows the results of the experiments, and Figure 6 illustrates their recall-precision curves. In the table, "syl," "wo" represents the syllable-based transcription and the word-based transcription, respectively. "CN" means the run uses a confusion network-based suffix array. The sign '+' represents multiple transcriptions are used. LM and AM in the table represent a language model and an acoustic model. Therefore, in the experiment of the last row of Table 1, we used three transcriptions obtained using unmatched syllable language model, unmatched word language model, and unmatched word language and acoustic models, and constructed a suffix array from a confusion network.

Table 1 shows that our method achieves very quick and accurate search with small size of index. Especially in the search time, our results are shown to be very quicker than the other participants of the NTCIR-11 SpokenDoc task [13]. Comparing NKI14-3 and NKI14-4, that construct the suffix arrays from the confusion

network, with NKI14-3 and NKI14-4, they require more time to get the search results. However, the index sizes are definitely smaller than NKI14-1 and NKI14-2 whose suffix arrays are constructed from the sequence of multiple recognition results. This is a desirable feature when we employ a method using many recognition results such as the method proposed in [9]. The search accuracy of NKI14-3 and NKI14-4 are almost same as that of NKI14-1 and NKI-2, or slightly worse than them.

4. CONCLUSIONS

We proposed a method to construct a suffix array from a confusion network that contains multiple recognition results. The experimental results show that our new approach reduces the size of index to one third or half compared with our previous approach. However, it is shown that the search time drastically increases because of increase of the search space. In future work, we will attempt to reduce the search time by constructing a suffix array that does not need a large search space.

5. ACKNOWLEDGEMENTS

This work has been supported by Grant-in-Aid for Young Scientists (B) 24700167 2012.

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] 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.

[8] 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.

[9] 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.

[10] Katsurada, K., Katsuura, K., Seng, K., Iribe, Y. and Nitta, T., "Using Multiple Speech Recognition Results to Enhance STD with Suffix Array on the NTCIR-10 SpokenDoc-2 Task", Proc. the NTCIR-10 Conference, pp.588-591, 2013.

[11] 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.

[12] 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)

[13] Akiba, T. et al. "Overview of the NTCIR-11 SpokenQuery & Doc task", Proc. the NTCIR-11 Conference, 2014.