Introduction of False Detection Control Parameters in Spoken Term Detection

Yuto Furuya*, Satoshi Natori*, Hiromitsu Nishizaki† and Yoshihiro Sekiguchi†
* Department of Education, Interdisciplinary Graduate School of Medicine and Engineering,
† Department of Research, Interdisciplinary Graduate School of Medicine and Engineering,
University of Yamanashi, Kofu-shi, Yamanashi, Japan
E-mail: {furuya,natori,nisizaki,sekiguti}@alps-lab.org Tel/Fax: +81-55-220-8361/8778

***Abstract*—This paper describes spoken term detection (STD) with false detection control. Our STD method uses phoneme transition network (PTN) derived by multiple automatic speech recognizers (ASRs) as an index. An PTN is almost the same to a sub-word based confusion network (CN), which is derived from an output of an ASR. The PTN-based index we proposed is made of the outputs of multiple ASRs, which is known to be robust to certain recognition errors and the out-of-vocabulary problem. Our PTN was very effective at detecting query terms. However, the PTN generates a lot of false detections especially for short query terms. Therefore, we applied two false detection control parameters to the Dynamic Time Warping-based term detection engine. In addition, we changed the search parameters depending on length of a query term. Finally, the STD performance was better (0.785 of F-measure) than without any parameters (0.717).**

I. Introduction

STD is difficult with respect to searching for terms in a vocabulary-free framework because search terms are unknown before using the automatic speech recognizer (ASR). Many studies [1], [2] that address STD tasks have been proposed, but most of them focused on the out-of-vocabulary (OOV) and speech recognition error problems.

The main ideas in our work are to use multiple ASRs and a dynamic time warping (DTW) framework with false control parameters at term searching.

We have already proposed the STD framework for spontaneous spoken lectures using a phoneme transition network (PTN)-formed index derived from multiple ASRs' 1-best hypothesis [3], [4].

PTN-based indexing originates from the concept of confusion network (CN) beging generated from an ASR. 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 ASR. 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). The use of the multiple ASRs and their outputs improves the speech recognition effectively. For example, Fiscus [5] proposed the ROVER method which adopts a word voting scheme. Utsuro et al. [6] developed a technique for combining multiple ASRs' outputs by using a support vector machine (SVM) to improve speech recognition performance. Therefore, multiple ASRs may improve STD relative to that of a single ASR's output.

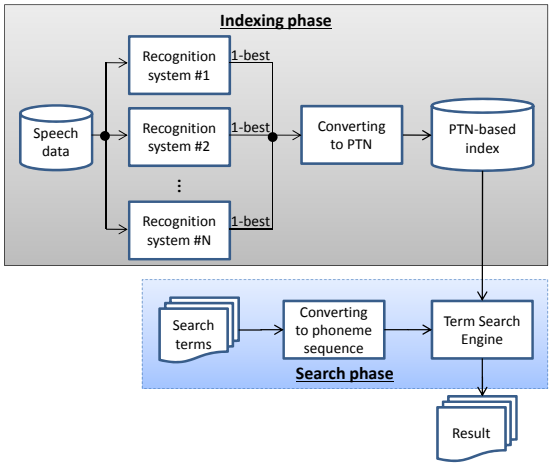

Fig. 1. *Overview of our STD framework.*

The PTN-formed index is made by merging the phoneme sequences of ASRs' outputs to a single CN. We showed that this prevents false detections of terms in our previous works [3], [4]. Although our proposed indexing is robust for the false detections, however, it raises the number of false detections because it has more complicated structure of the CN made by an ASR. Therefore, in this paper, we propose introducing a majority voting parameter and a measure of ambiguity, which are easily derived from the PTN, into the term search engine. We call these two parameters "false detecton control parameters." In addition to this, we investigate the effectiveness of changing the DTW cost paremeters depending on length of a query term in the DTW-based search engine.

To prevent false detections, we installed the voting and ambiguity parameters of PTN in our term detection engine on the basis of DTW. Furthermore, we refined the detection engine, where the DTW cost used to calculate the distance is dynamically changed depending on the number of phonemes consisting of a query term. Our technique was evaluated on the NTCIR-9 test set, and the improved term detection engine decreased some false detections and achieved relative 9.5% improvement in F-measure.

II. STD framework

A. Outline

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

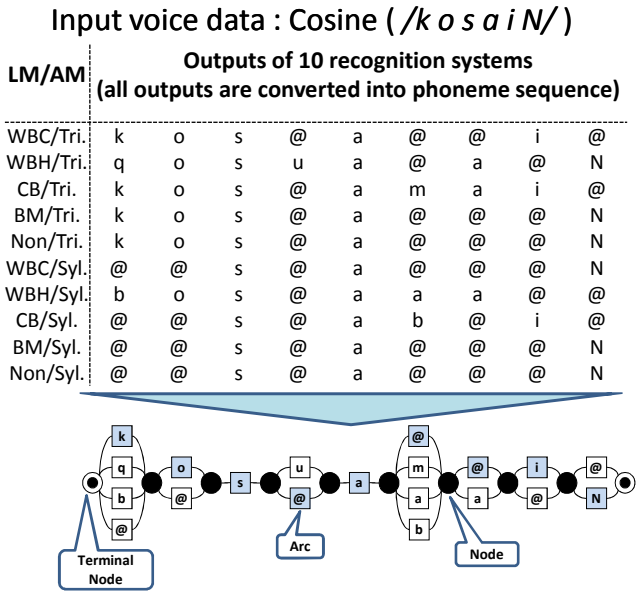

Fig. 2. *Making PTN-based index by performing alignment using on DP and converting to PTN.*

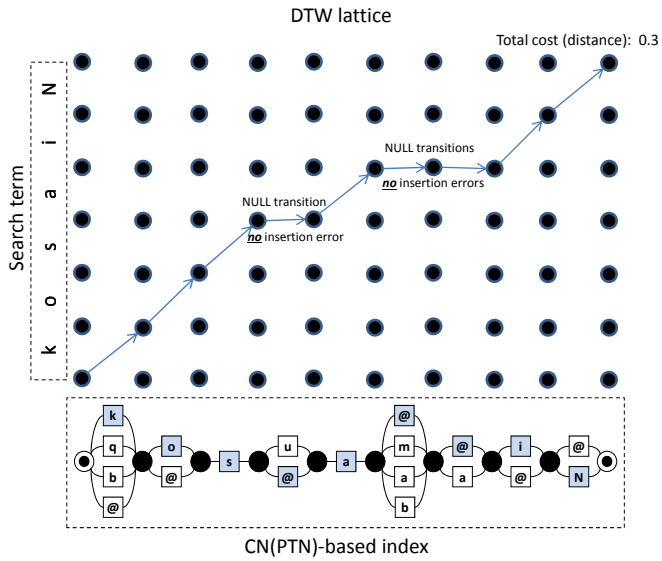

Fig. 3. Example of term search on network based index.

In the indexing phase, speech data is performed by speech recognition and the recognition outputs (word or sub-word sequences) are converted into the PTN index for STD. In the search phase, the word-formed query is converted into the phoneme sequence, then the phoneme-formed query is input to the term detection engine. In the case of treating English queries, we have to consider the variety of pronunciations of the queries. There are some reports fighting the pronunciation problem[7]. In this paper, however, we handle Japanese STD. Most of Japanese words can be completely translated to phoneme sequence (pronunciation). Therefore, we do not consider the pronunciation problem in this paper.

The term detection engine searches the input query term from the index in phoneme level using the DTW framework.

B. PTN-based indexing

Figure 2 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 the ASR. We used 10 types of speech recognizers to create the PTN-formed index. The speech was recognized by the 10 recognizers to yield 10 hypotheses, which were then converted into phoneme sequences (Fig. 2). Next, we obtained "aligned sequences" using the dynamic programming (DP) scheme described previously [5]. Finally, PTN was obtained by converting the aligned sequences. The term "@" in Fig. 2 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.

C. Term detection engine with false detection control

The term detection engine uses the DTW-based word spotting method. Figure 3 represents an example of the DTW framework between the search term "k o s a i N" (cosine) and for the PTN-formed index. The PTN has multiple arcs between adjoining two nodes. These arcs are compared to one of phoneme labels of a query term.

We used edit distance as cost on the DTW paths, and the costs for substitution, insertion and deletion errors were commonly set to 1.0 when the number of phonemes consisting of a query term was N or larger than N. On the other hand, each cost was commonly set to 1.5 when the number of phonemes was less than N to avoid false term detections in query terms, having less number of phonemes. This cost (=1.5) was optimized using a development query set.

The total cost $D(i,j)$ at the grid point (i,j) ($i = \{0,...,I\}$, $j = \{0,...,J\}$, where I and J are the number of the set of arcs in the index and query term, respectively) on the DTW lattice was calculated by the following equations:

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

$$Match(i,j) = \begin{cases} 0.0 : Query(j) \in PTN(i) \\ 1.0 : Query(j) \notin PTN(i), J \geq N \\ 1.5 : Query(j) \notin PTN(i), J < N \end{cases} \quad (2)$$

$$Del = \begin{cases} 1.0 : J \geq N \\ 1.5 : J < N \end{cases} \quad (3)$$

$$Null(i) = \begin{cases} \frac{\alpha}{Voting(@)} : NULL \in PTN(i), J \geq N \\ \frac{\beta}{Voting(@)} : NULL \in PTN(i), J < N \\ 1.0 : NULL \notin PTN(i), J \geq N \\ 1.5 : NULL \notin PTN(i), J < N \end{cases} \quad (4)$$

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. We allowed a null transition between two nodes in the PTN-formed index with the cost defined in Eq.(4). When the query term matches to null (@) in the

PTN, a transition cost was dynamically set as shown in Eq.(4). $Voting(@)$ means the number of ASRs outputting NULL at the same arc. We call it "null voting." α and β are hyper parameters, which were optimized using the development set. The appropriate null cost achieves increasing term detection and decreasing false detections.

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

$$Vot(i,j) = \begin{cases} \frac{\gamma}{Voting(p)} : \\ \quad \exists p \in PTN(i), p = Query(j) \\ 0.0 : Query(j) \notin PTN(i) \end{cases} \quad (5)$$

$$Acw(i) = \delta \cdot ArcWidth(i) \quad (6)$$

We provided two types of parameters to control false detection as follows:

- "$Voting(p)$" is the number of ASRs outputting the same phoneme p at the same arc. More value of $Voting(p)$ makes reliability of phoneme p better.
- "$ArcWidth(i)$" is the number of arcs (phoneme labels) at $PTN(i)$. Less value of $ArcWidth(i)$ makes also reliability of phonemes at $PTN(i)$ better.

γ and δ are also hyper parameters and they were set to 0.5 and 0.01, respectively, optimized by the development query set.

In advance searches for the query term, the term detection engine initializes $D(i,0) = 0$, and then, it calculates $D(i,j)$ using Eq.(1) ($i = \{0,...,I\}$, $j = \{1,...,J\}$). Furthermore, $D(i,J)$ are 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 the recall and precision rates for STD can be controlled.

III. STD EXPERIMENT

A. Speech Recognition

As shown in Figure 1, the speech data was recognized by the 10 ASRs. Julius ver. 4.1.3 [8], which is an open source decoder for LVCSR, was used in all the systems.

We prepared two types of acoustic models (AMs) and five types of language models (LMs) for constructing PTN. The AMs are triphone based (Tri.) and syllable based HMMs (Syl.), where both of which were trained on the spoken lectures in the Corpus of Spontaneous Japanese (CSJ) [9].

All LMs are 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 two of Hiragana characters.
- Non : LM is not used. Speech recognition without any LM is equivalent to phoneme (or syllable) recognition.

Each model is trained from the many transcriptions in the CSJ under the open for the speech data of STD.

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

B. STD test collection

We used the two test collections: one is the CORE-OOV query set of the Japanese test collection for STD (JCT-STD)[10] for developing (tuning) the parameters mentioned in Section II-C, the other is the formal-run CORE query set of the SpokenDoc STD sub-task at the ninth NTCIR Workshop (NTCIR-9) [11] for testing our method.

The CSJ was used as spoken documents set in both the test collections. It includes 2,702 speeches including actual academic presentations and simulated public speech. Both the query sets of the development and test set in this paper were for only 177 speeches (44 hours), specially called "CORE", from the whole CSJ data set. They were not included in the training data set of the AMs and LMs.

The parameters (α, β, γ, and δ) were tuned using the development set. The development set has total 50 types of terms, 233 times uttered in the CORE lecture 177 speeches.

The test set, from the NTCIR-9 SpokenDoc STD sub-task, had 50 query terms which had the 31 OOV terms that were not included in the recognition dictionary for the WBC LM. The total occurrences of the all terms in the CORE query set was 366.

C. Evaluation metric

The evaluation metrics used at the STD task are the recall, precision, F-measure, and mean average precision (MAP) values[11]. These measurements are frequently used to evaluate retrieval performance on information retrieval. An F-measure value with the optimal balance of recall and precision rates is just denoted by "F-measure." The STD performances for the query set can be displayed by recall-precision curves which were drawn by changing the threshold θ value on the DTW-based word spotting.

D. Experimental result

Figure 4 shows the recall-precision curves that are the STD performances of the baseline system and the our STD systems. TABLE I also represents the F-measure and MAP values for the false detection control parameters and the baseline on the same query set.

The baseline (#1) system used the same DTW-based word spotting for the phoneme sequences from the transcription by only the ASR with the CB-based LM and the triphone-based AM. In other words, the baseline system used only the single ASR. The system #2, #3, and #4 show the results when the DTW costs were always set to 1.0. In other wards, the costs do not depend on the length (the number of phonemes) of a query term. #2 (ED_f) did not use any the control parameters with the fixed null transition cost of 0.1. #3 used the false detection

TABLE I
F-MEASURE AND MAP VALUES OF EACH STD SYSTEM.

ID	Parameter	F-measure	MAP
#1	Baseline	0.537	0.642
#2	ED_f	0.717	0.742
#3	+ $Vot(i,j)$ + $Acw(i)$	0.715	0.825
#4	+ $Null(i)$	0.725	0.825
#5	ED_d+$Vot(i,j)$+$Acw(i)$+$Null(i)$	**0.785**	0.825

control parameters and #4 used the null voting in additio
#3. #5 system (ED_d) is similar to #4, however, the differe
is to dynamically change the DTW costs depending on
length of a query term. In this experiment, N was set tc
because it was cleared that the STD performances of the qu
terms that consist of less than 10 of phonemes were worse t
the terms consisting of 10 or more than 10 phonemes in
development set.

By comparing #1 (single ASR) with #2 (multiple AS
first, using multiple ASRs' output makes the F-measure
MAP values drastically improved. #2 maintains the high-l
precision rate in the middle range of racall rate. However
is the less precision rate in the lower recall rate (under 3
because the false detections occured. Therefore, introdu
the control parameters ($Vot(i,j)$ and $Acw(i)$) to #2 impro
the precision rate under 65% of the recall rate. This ra
the MAP value to 0.825 (#3) from 0.742. In addition to
the null voting (#4) slightly improved the F-measure to 0.725
from 0.715 (#3).

The DTW costs used in #1 $\sim$ #4 was fixed for the all query terms. The result of #5 shows that the dynamically changing of the DTW costs depending on the length of a query term improved the precision rate in the higher recall rate (over 60%). Therefore, the best F-measure (0.785) was obtained among the STD systems. In this experiment, we tried two sorts of the DTW costs: 1.0 ($J \geq N$) or 1.5 ($J < N$). Using the fine-graded DTW costs may make the STD performance better. This is an issue in the future.

Finally, the experimental results claimed that the following three techniques in the STD study were very effective: using multiple ASRs, introducing the false detection control parameters, and changing the DTW costs depending on the length of a query term.

IV. CONCLUSION

This paper described the STD technique and its effectiveness on the test set at the NTCIR-9. First, we introduced the PTN-based indexing, essentially a phoneme-based CN, derived from multiple ASRs' outputs. The one of aims in this study was to use multiple outputs of ASRs for constructing the PTN-formed index for STD, which is different from the sub-word based approaches proposed earlier.

The experimental results showed that the PTN-based indexing functioned well in improving the STD performance under the DTW framework compared with the simple index. However, using the 10 ASRs generated a lot of false detections. Therefore, we installed the false detection control parameters, relating to the majority voting and the width of

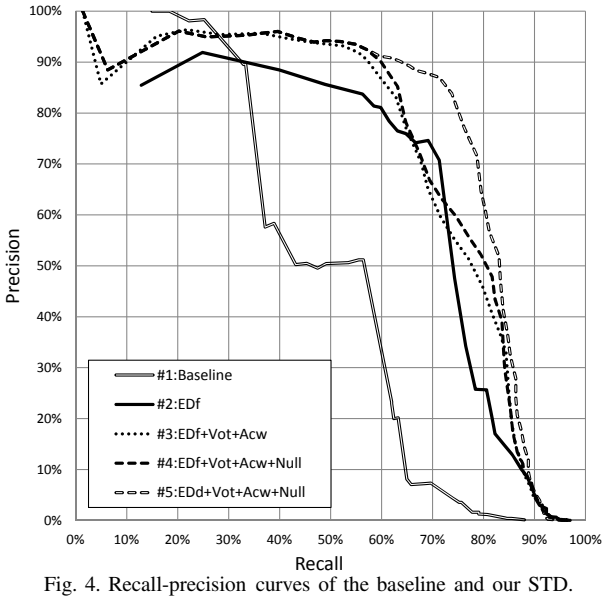

Fig. 4. Recall-precision curves of the baseline and our STD.

arc in the PTN, to the DTW framework. As the result, we succeeded at reducing false detections below 65% of the recall rate. In addition to this, by introducing the null voting and changing the DTW costs depending on the length of query term improved the STD performance.

In future work, we intend to develop a fast search algorithm under the DTW framework. The process speed of our engine is too slow to put into practical use.

REFERENCES

[1] 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 Proc. of INTERSPEECH2007, pp. 2393–2396, 2007

[2] 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 Proc. of INTERSPEECH2008, pp. 2146–2149, 2008.

[3] S. Natori, H. Nishizaki, and Y. Sekiguchi, "Japanese spoken term detection using syllable transition network derived from multiple speech recognizers' outputs," in Proc. of INTERSPEECH2010, pp. 681–684, 2010.

[4] S. Natori, H. Nishizaki, and Y. Sekiguchi, "Network-formed index from multiple speech recognizers' outputs on spoken term detection," in Proc. of APSIPA ASC 2010 (student symposium), p. 1, 2010.

[5] J. G. Fiscus, "A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," in Proc. of ASRU'97, pp. 347–354, 1997.

[6] T. Utsuro, Y. Kodama, T. Watanabe, H. Nishizaki, and S. Nakagawa, "An empirical study on multiple LVCSR model combination by machine learning," in Proc. of HLT-NAACL 2004, pp. 13–16, 2004.

[7] 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), pp. 688–698, 2011.

[8] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine julius," in Proc. of APSIPA ASC2009, 6 pages, 2009.

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

[10] Y. Itoh et al.,"Constructing Japanese Test Collections for Spoken Term Detection," in Proc. of INTERSPEECH2010, pp.677–680, 2010.

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