Entropy-based False Detection Filtering in Spoken Term Detection Tasks

Satoshi Natori*, Yuto Furuya*, 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: {natori,furuya,nisizaki,sekiguti}@alps-lab.org Tel/Fax: +81-55-220-8361

***Abstract*—This paper describes spoken term detection (STD) and inexistent STD (iSTD) methods using term detection entropy based on a phoneme transition network (PTN)-formed index. Our previously reported STD method uses a PTN derived from multiple automatic speech recognizers (ASRs) as an index. A PTN is almost the same as a sub-word-based confusion network, which is derived from the output of an ASR. In the previous study, our PTN was very effective in detecting query terms. However, the PTN generated many false detection errors. In this study, we focus on entropy of the PTN-formed index. Entropy is used to filter out false detection candidates in the second pass of the STD process. Our proposed method was evaluated using the Japanese standard test-set for the STD and iSTD tasks. The experimental results of the STD task showed that entropy-based filtering is effective for improving STD at a high-recall range. In addition, entropy-based filtering was also demonstrated to work well for the iSTD task.**

I. Introduction

The difficulty in spoken term detection (STD) lies in the search for terms under a vocabulary-free framework because search terms are not known prior to an automatic speech recognizer (ASR). Many studies tackling STD have already been proposed [1], [2]. Most STD studies have focused on out-of-vocabulary (OOV) and speech recognition error problems. For example, STD techniques that use sub-word lattices and confusion networks (CNs) have been proposed.

We have developed an indexing method that is robust for such problems. The main idea of our work is to use both multiple ASRs for indexing and a dynamic time warping (DTW) framework with false detection control parameters during term searching. We have previously evaluated our STD framework for spontaneous spoken lectures using a phoneme transition network (PTN)-formed index derived from multiple ASRs' 1-best hypotheses [3], [4]. Our STD method showed the best STD performance for the 9th NII Testbeds and Community for Information access Research (NTCIR-9) SpokenDoc STD sub-task [5].

The use of multiple ASRs and their outputs is very effective for improving speech recognition performance. For example, Fiscus [6] proposed the ROVER method, which adopts a word-voting scheme. Utsuro et al. [7] developed a technique for combining the output of multiple ASRs using a support vector machine in order to improve speech recognition performance. Therefore, multiple ASRs realize good STD performance compared to a single ASR's N-best outputs [3]. Our PTN-based indexing is based on the idea of a CN generated from an ASR. CN-based indexing for STD is a powerful indexing method. The PTN-formed index is generated by merging the phoneme sequences of ASRs' output to a single CN.

Our previously proposed PTN-based indexing was robust for miss detections; however, it produced a number of false detection errors because it had a more complicated CN structure. Therefore, in this study, we propose a two-pass STD system using PTN-formed index entropy. The first pass detects terms from target speech data using a STD engine with false detection control parameters [8]. These parameters are majority voting parameters and a measure of ambiguity, which are easily derived from the PTN. Then, the second pass filters out false detection candidates on the basis of their entropy values for the queried term.

For OOV term detection, an ASR cannot correctly transcribe target OOV terms. Therefore, phoneme sequences derived from the output of multiple ASRs vary. The entropy value of a PTN that contains OOV terms becomes greater because the structure of the PTN becomes more complex. If a detection candidate for an OOV queried term has lower entropy, the confidence of the candidate is likely to have low detection confidence. This is performed in the second pass of our DTW-based STD engine to prevent the engine from falsely detecting a queried term. To the best of our knowledge, there is no current STD method that focuses on entropy of an index. This paper presents a possible beneficial effect of entropy-based filtering.

Our two-pass STD system was evaluated on the OOV test-set of the Japanese STD test collection and the inexistent STD (iSTD) test-set of the 10th NTCIR (NTCIR-10) SpokenDoc-2 task [9]. In the iSTD task, an STD engine inspects whether a queried term is existent or inexistent in a speech data collection. This is a new STD-related task proposed at the NTCIR-10 conference. The same STD engine was used for the iSTD task in this study.

The evaluation of the STD task showed that entropy-based filtering was effective for improving STD performance at a high-recall range. In addition, it was demonstrated to work well for the iSTD task.

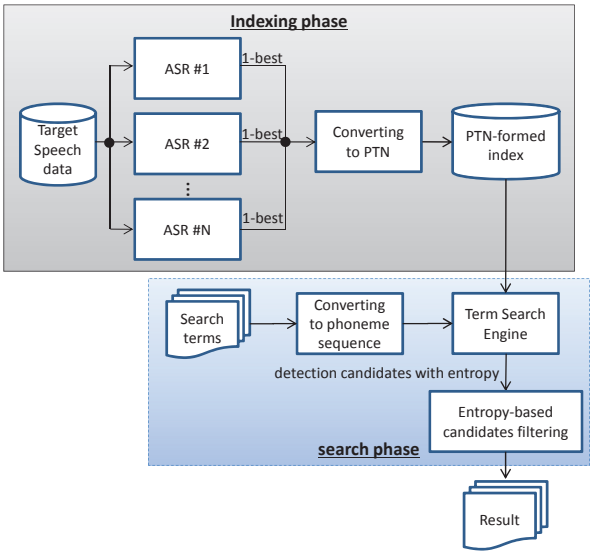

Fig. 1. Overview of our two-pass STD framework.

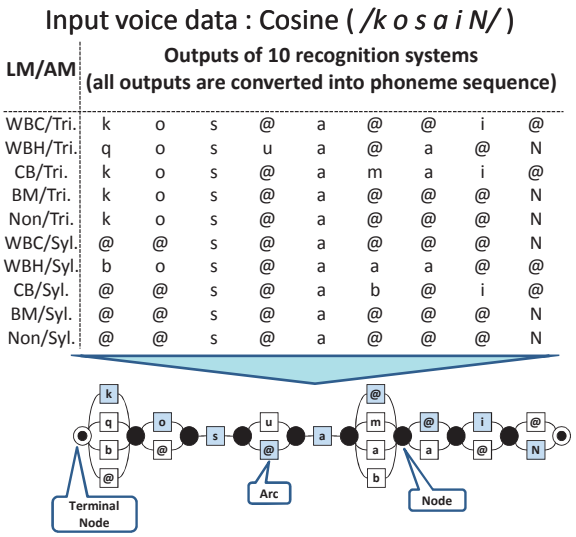

Fig. 2. Generating a PTN-formed index by performing alignment using DP and converting to a PTN.

II. STD FRAMEWORK

A. Outline

Figure 1 shows an outline of the proposed two-pass STD framework.

In the indexing phase, speech data is performed by speech recognition, and the recognition outputs (words or sub-word sequences) are converted into the PTN index for STD. In the search phase, the word-formed query is converted into a phoneme sequence. Then, the phoneme-formed query is input to the term search engine. The term search engine searches the queried term from the index at the phoneme level using the DTW framework. Next, the detected candidates are filtered out based on their entropy values.

B. PTN-based indexing

Speech data is recognized by 10 different ASRs. Each 1-best hypothesis is translated into a phoneme sequence, and then all 10 sequences are combined into the PTN-formed index.

Figure 2 shows an example of generating a part of a PTN-formed index of an utterance of "*cosine*" (Japanese pronunciation is /k o s a i N/) by performing the alignment process of N phoneme sequences from the 1-best hypotheses of the recognizers. We used 10 types of ASRs to generate a PTN-formed index. In Figure 2, the utterance is recognized by the 10 recognizers, and then the 10 hypotheses are obtained. The hypotheses are converted into phoneme sequences. Next, we can obtain "aligned sequences" by performing a dynamic programming (DP) scheme and generating the CN-formed index.

Finally, the PTN is obtained by converting the aligned sequences. "@" 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 study, we did not use phoneme occurrence probabilities.

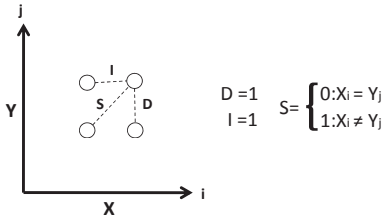

Fig. 3. Definition of DTW path.

C. Term search engine with false detection control

We adopted a DTW-based word spotting method. In this study, the paths on the DTW lattice were allowed, as shown in Figure 3. X and Y denote an index and a query term, respectively.

Figure 4 represents an example of the DTW framework between the search term "k o s a i N" (cosine) and the PTN-formed index. 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 cost value for substitution, insertion, and deletion errors is commonly set to 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 number of the set of arcs in an index and a query term, respectively) on the DTW lattice is calculated by the following equations:

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

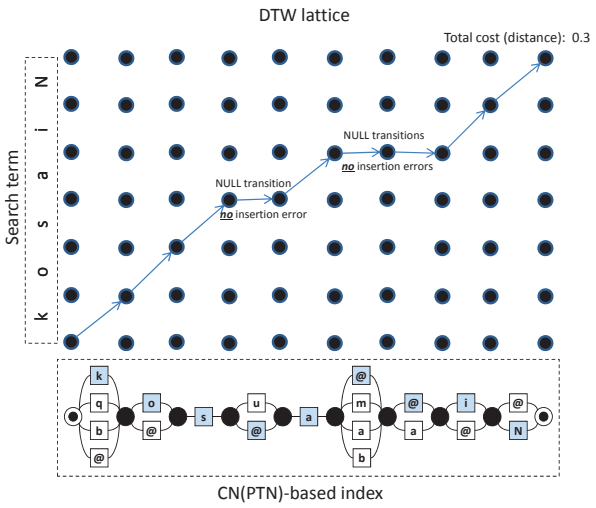

Fig. 4. Example of term search on network-formed index.

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

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

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

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

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

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

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

We provide two types of parameters to control false detection:

- "Voting(p)" is the number of ASRs outputting the same phoneme p at the same arc. Greater $Voting(p)$ values result in greater reliability of phoneme p.
- "ArcWidth(i)" is the number of arcs (phoneme labels) at $PTN(i)$. Lesser $ArcWidth(i)$ values also result in greater reliability of phonemes at $PTN(i)$.

We allow a null transition between two nodes in the PTN-based index with 0.1 cost. Thus, the values of $Vot(i,j)$ and $Acw(i)$ must be less than the null transition cost. Therefore, a is set to less than 1.0. When $Voting(p) = 1$ and a is greater than or equal to 0.1, the null transitions get an advantage during term search. This may make the voting parameter negligible. In this study, we set α to 0.5. This means that $Voting(p)$, which becomes greater than 5, is reliable. β must be set in the range of 0.01-0.1 for the same reason. We set β to 0.01. This means that any ArcWidth(i) value less than 10 is also reliable.

In advance query term searching, the term search engine initializes $D(i,0) = 0$, and then calculates $D(i,j)$ using Equ. (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 that have normalized cost D(i, J) below threshold θ. Changing the θ value enables us to control the recall and precision rates of STD performance.

D. Entropy of detected candidate

The entropy value of an arbitrary interval in a PTN is calculated using the number of phonemes and a posterior probability of a phoneme. A posterior probability at any position in a PTN is calculated based on the number of ASRs that output the phoneme.

Detection entropy (DE_t) of a detected candidate t for a queried term is calculated using the following equations:

$$VE_i = -\sum_{j=1}^{J_i} \frac{Voting(p_{ij})}{R} \log_2 \frac{Voting(p_{ij})}{R} \quad (6)$$

$$DE_t = \frac{1}{T} \sum_{i=t_s}^{t_e-1} VE_i \quad (7)$$

Here VE_i is voting entropy between the i-th and $(i+1)$-th nodes in a PTN, p_{ij} represents the j-th phoneme at the arcs between the i-th and $(i+1)$-th nodes in the PTN, and J_i is the number of phonemes (arcs) between the i-th and $(i+i)$-th nodes. $Voting(p_{ij})$ indicates the number of ASRs that output the phoneme p_{ij}. R is the number of all ASRs that constructed the PTN. In this study, R is 10.

DE_t is calculated by Eq. (7). t_s and t_e are the first and last nodes of a candidate t in a PTN, respectively. T is the number of nodes between t_s and t_e.

The DE of detection candidates is used to filter out false detection candidates in the second-pass of the STD process.

E. Japanese STD test collection

1) Target speech data: We used a subset of the Japanese test collection for the STD [12] to verify our method. This test collection was created by a working group of the Special Interest Group-Spoken Language Processing (SIG-SLP) of the Information Processing Society of Japan (IPSJ).

The Corpus of Spontaneous Japanese (CSJ) was used as the target spoken document set of this test collection. In total, it contains 2,702 speech files, including actual academic presentations and simulated public speeches. However, only 177 speeches (39 hours) are included in this collection. These speeches are referred to as the "CORE" part of the CSJ. They are not included in the acoustic model (AM) and language model (LM) training dataset.

As shown in Figure 1, the speech data was recognized by the 10 ASRs. Julius ver. 4.1.3 [10], an open source decoder for large vocabulary continuous speech recognition (LVCSR), was used in all the systems.

We prepared two types of AMs and five types of LMs to construct the PTN. The AMs were triphone- (Tri.) and syllable-based (Syl.) hidden Markov models (HMMs); both types of HMMs were trained from the spoken lectures in the CSJ [11].

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, and 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 Japanese Hiragana.
- BM : character sequence based trigram in which the unit of language modeling is two Hiragana characters.
- Non : No LM is used. Speech recognition without a LM is equivalent to phoneme (or syllable) recognition.

Each LM was trained from the many transcriptions in the CSJ under the open condition for the target speech data of STD.

Finally, 10 combinations, comprised of two AMs and five LMs, were formed.

2) Query set: We used the OOV test-set from the Japanese test collection [12] for STD to evaluate performance. The OOV test-set has a total of 50 terms, which were spoken 233 times in the CORE lectures. All the OOV terms are not included in the WBC LM speech recognition dictionary.

In addition, the NTCIR-9 SpokenDoc STD test-set [5] was used to analyze entropy.

3) Ealuation metrics for STD task: The evaluation metrics used in this study were recall and precision values. These measurements are frequently used to evaluate information retrieval performance and are defined as follows:

$$Recall = \frac{N_{corr}}{N_{true}} \tag{8}$$

$$Precision = \frac{N_{corr}}{N_{corr} + N_{spurious}} \tag{9}$$

Here N_{corr} and $N_{spurious}$ are the total number of correct and spurious (false) term detections, respectively, and N_{true} is the total number of true term occurrences in the speech data.

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

III. iSTD FRAMEWORK

A. iSTD task

The iSTD task is a new task in the NTCIR-10 SpokenDoc-2[9]. In the iSTD task, a STD engine inspects whether a queried term is existent or inexistent in a spoken document collection. Unlike conventional STD tasks, the iSTD task has two main characteristics: existent and inexistent. Terms in a query set are evaluated together, and each queried term is evaluated to determine whether it exists at least once in a spoken document collection.

The output of the iSTD task is a query list in which the queried terms are sorted in descending order based on their "iSTD scores." The iSTD score is a confidence value that indicates the likelihood of a term being inexistent in the target speech collection. In the NTCIR-10 iSTD task, the preferred score is in the range from 0.0 to 1.0. For example, if a term is considered inexistent, the iSTD score will be close to 1.0.

B. iSTD method

Our iSTD engine is almost the same as the two-pass engine of the STD task. In this study, the iSTD score for a queried term is regarded as the lowest score calculated by the STD engine for a detection candidate.

The first pass of the STD engine outputs an iSTD score based on a DTW-based calculation. The second pass outputs the final iSTD score that is calculated by lineally combining the iSTD score from the first pass and the entropy value of the detection candidate. The first-pass iSTD score is calculated by the following equation.

$$\text{iSTD score (at first pass)} = 1 - \text{"DTW cost"} \tag{10}$$

"DTW cost" is calculated by Eq. (1).

Finally, the 200 queryed terms are ranked (ordered) according to combined iSTD score.

C. Test collection for iSTD

1) Target speech data: We used the Corpus of Spoken Document Processing Workshop (SDPWS) speech collection released by the NTCIR-10 SpokenDoc-2 task organizers [9]. It consists of recordings from the first to sixth annual SDPWS and is comprised of 104 oral presentations (28.6 hours).

These speeches are also speech-recognized by the 10 ASRs used to transcribe the CSJ speeches in the STD task.

2) Query set: We define two classes as follows:

- Class $\in$ is a set of queried terms that exist at least once in the target speech.
- Class $\notin$ is a set of queried terms that are inexistent in any target speech.

Figure 5 shows an example of a query set. The query consists of 200 terms and their ID numbers. The query set contains 100 Class $\notin$ terms and 100 Class $\in$ terms.

3) Evaluation metric on iSTD task: The evaluation metrics used in this task are as follows:

- Maximum F-measure (the balanced point on recall−precision curve),
- F-measure calculated by the top-100-ranked term,

Figure 6 shows an example of iSTD result of the query shown in Figure 5. Recall and precision rates for terms

term ID,	term,	Class
001,	A,	$\notin$
002,	B,	$\in$
003,	C,	$\in$
004,	D,	$\notin$
005,	E,	$\in$
006,	F,	$\notin$
007,	G,	$\in$
008,	H,	$\notin$
...		
200,	J,	$\in$

Fig. 5. An example of a query set for the iSTD task.

rank,	term ID,	score
1,	004,	1.00
2,	002,	0.98
3,	001,	0.90
4,	008,	0.89
5,	005,	0.85
6,	009,	0.80
7,	003,	0.50
8,	007,	0.45
9,	006,	0.40
10,	010,	0.10

Fig. 6. An example of an iSTD result.

positioned at rank r and greater than r are calculated by the following functions:

$$Recall_r = \frac{T_{\notin,r}}{N_{\notin}} \times 100(\%)$$

$$Precision_r = \frac{T_{\notin,r}}{r} \times 100(\%)$$

where $T_{\notin,r}$ denotes the number of $\notin$ terms positioned at rank r and greater than r, and $N_{\notin}$ is the total number of terms that belong to class $\notin$. By changing r from 1 to 200, a recall$-$precision curve can be drawn. The maximum F-measure from the best balanced point in the curve is also used for evaluation.

IV. STD and iSTD Experiments

A. Entropy analysis

The average entropies for the total true occurrences of the terms in each test-set are shown in Table I. "CORE_OOV" is the OOV set of the test collection, and "NTCIR9_OOV" and "NTCIR9_IV" are the OOV terms and the in-vocabulary (IV) terms in the NTCIR-9 test-set, respectively.

As shown in Table I, the average entropy of NTCIR9_IV is lower than the other test-sets. This is because an utterance that includes an OOV term is transcribed into various phoneme sequence patterns by the ASRs. In other words, speech recognition of an OOV term using multiple ASRs is likely to produce a PTN that has more arcs (maximum of 10) than an IV term. On the other hand, speech recognition of an IV term may produce a PTN with the less number of arcs compared with an OOV term.

In this study, we assume that the DE of a queried OOV term increases. Figure 7 shows a scatter chart for DE vs. DTW

TABLE I
AVERAGE ENTROPIES FOR THE TOTAL TRUE OCCURRENCES IN EACH TEST-SET

test-set	true occurences	entropy
CORE_OOV	233	0.63
NTCIR9_OOV	195	0.60
NTCIR9_IV	167	0.48

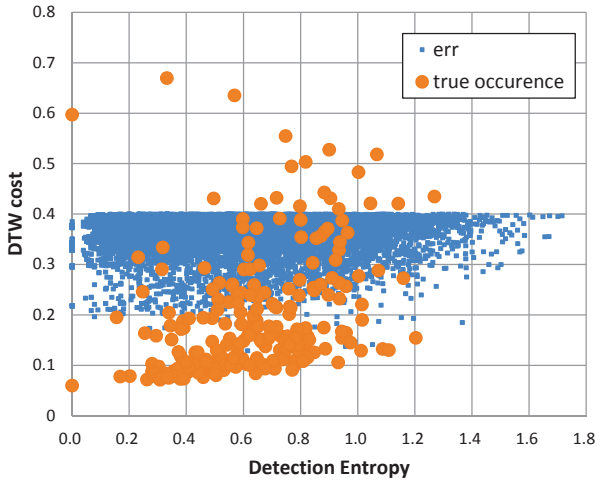

Fig. 7. A scatter chart on DE vs. DTW cost.

cost for all true occurrences and false detection candidates (illustrated as "err") with less than 0.4 DTW cost for the CORE_OOV test-set. It is difficult to completely separate all the detections into true occurrences and false detections. However, many candidates with lower entropy are false detections. In this study, we attempted to separate detections using a linear function $y = ax + b$ (x and y are entropy and DTW cost, respectively) during the second pass of the STD process.

The parameters a and b were set to contribute to the maximization of the STD performance. In this experiment, a and b were set to 0.014 and 0, respectively.

B. STD experimental result

Figure 8 shows the recall-precision curves from our STD system for the CORE_OOV test-set; "w/o entropy" is the curve from the first stage of the framework and "w/ entropy" is the curve from the detection candidates that were filtered out using the DE criterion.

As shown in Figure 8, entropy-based filtering was ineffective for improving precision rate at the lower-recall range (less than 65%); however, it could improve precision rate at the high-recall rate range (from 65$-$80%).

Figure 7 shows that most of the true occurrences with greater than 0.2 DTW cost show high DE (greater than 0.2). In addition, there were many false detection candidates with lower DE (less than 0.2). In this experiment, we clarified that entropy filtering detection candidates with lower DE is effective to a certain extent.

TABLE II
iSTD PERFORMANCES. (*1) RECALL, PRECISION AND F-MEASURE RATES CALCULATED BY TOP-100-RANKED OUTPUTS. (*2) RECALL, PRECISION AND F-MEASURE RATES CALCULATED BY TOP-N-RANKED OUTPUTS. N IS SET TO OBTAIN THE MUXIMUM F-MEASURE

	Rank 100*1			Maximum*2			
	Rec. [%]	Prec. [%]	F. [%]	Rec. [%]	Prec. [%]	F. [%]	Rank
w/o entropy	79.00	79.00	79.00	84.00	78.50	81.16	107
w/ entropy	82.00	82.00	82.00	85.00	80.19	82.52	106

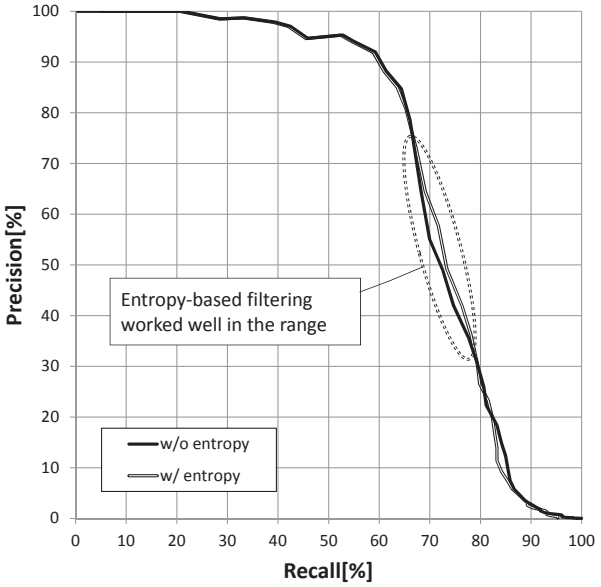

Fig. 8. Recall-precision curves on the CORE_OOV test-set.

C. iSTD experimental result

Table II shows iSTD performance; "w/ entropy" used entropy to rank the query terms and "w/o entropy" only used DTW cost.

The F-measure values calculated by the top-100-ranked and the maximum F-measure value for "w/ entropy" outperformed those for "w/o entropy." Using the entropy value of a detection candidate achieved 3.0% improvement in the F-measure values calculated by the top-100-ranked value in the query list.

V. CONCLUSION

This study described a two-pass STD technique and evaluated the technique's effectiveness on STD and iSTD test collections. First, we introduced PTN-based indexing derived from multiple ASR outputs, which is essentially a phoneme-based CN. PTN-based indexing differs from the sub-word-based approaches proposed earlier.

The first pass of our STD engine outputs detection candidates for a queried term using the DTW framework with false detection control parameters. Although this can control many false detection candidates, there are many surviving false detection candidates included in the first-pass output. Therefore, we tried to filter out false detection candidates using DE in the second pass of the STD process. In the second pass, a candidate with a DE value lower than a threshold is filtered out.

The experimental results of the STD task showed that entropy-based filtering was partially effective for improving STD performance at a high-recall range (65−80%). However, the entropy filtering we proposed cannot remove all false detections.

The entropy-based framework was also performed with the iSTD task. The experimental results indicated that using entropy of a detection candidate worked well and effectively reduced false detections in the iSTD task.

VI. ACKNOWLEDGEMENTS

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] 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)*. ISCA, 2007, pp. 2393–2396.

[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 *Proceedings of the 9th Annual Conference of the International Speech Communication Association (INTERSPEECH2008)*. ISCA, 2008, pp. 2146–2149.

[3] 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)*. ISCA, 2010, pp. 681–684.

[4] 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)*, 2010, p. 1.

[5] 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 9th NTCIR Workshop Meeting*, 2011, pp. 223–235.

[6] 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)*, 1997, pp. 347–354.

[7] 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)*, 2004, pp. 13–16.

[8] H. Nishizaki, Y. Furuya, S. Natori, and Y. Sekiguchi, "Spoken term detection using multiple speech recognizers' outputs at ntcir-9 spokendoc std subtask," in *Proceedings of the 9th NTCIR Workshop Meeting*, 2011, pp. 236–241.

[9] T. Akiba, et al., "Overview of the NTCIR-10 SpokenDoc-2 Task," in *Proceedings of the 10th NTCIR Conference*, 2013.

[10] 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)*, 2009, pp. 131–137.

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

[12] 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)*. ISCA, 2010, pp. 677–680.