IEICE TRANS. INF. & SYST., VOL.E99–D, NO.10 OCTOBER 2016

PAPER Special Section on Recent Advances in Machine Learning for Spoken Language Processing

Re-Ranking Approach of Spoken Term Detection Using Conditional Random Fields-Based Triphone Detection

Naoki SAWADA[†a)], *Nonmember and* **Hiromitsu NISHIZAKI**[†b)], *Senior Member*

SUMMARY This study proposes a two-pass spoken term detection (STD) method. The first pass uses a phoneme-based dynamic time warping (DTW)-based STD, and the second pass recomputes detection scores produced by the first pass using conditional random fields (CRF)-based triphone detectors. In the second-pass, we treat STD as a sequence labeling problem. We use CRF-based triphone detection models based on features generated from multiple types of phoneme-based transcriptions. The models train recognition error patterns such as phoneme-to-phoneme confusions in the CRF framework. Consequently, the models can detect a triphone comprising a query term with a detection probability. In the experimental evaluation of two types of test collections, the CRF-based approach worked well in the re-ranking process for the DTW-based detections. CRF-based re-ranking showed 2.1% and 2.0% absolute improvements in F-measure for each of the two test collections.

key words: *conditional random fields, phoneme-to-phoneme confusion learning, re-ranking, spoken term detection, triphone detection*

1. Introduction

Spoken term detection (STD), a speech data retrieval technology, is designed to determine whether a given utterance includes a query term comprising a word or phrase. STD research has become a popular topic in spoken document processing research, and the number of STD research reports has increased following the 2006 STD evaluation organized by the National Institute of Standards and Technology [1].

The difficulty in STD lies in the search for terms under a vocabulary-free framework because search terms are not known prior to a large vocabulary continuous speech recognition (LVCSR) system. Many studies focusing on STD have been conducted [2], [3]. In the past, most STD studies focused on out-of-vocabulary (OOV) and speech recognition error problems. For example, STD techniques using subword (syllable or phoneme)-based lattices or confusion networks (CN) have been proposed [3].

Another problem in STD is ineffectiveness to detect speech recognition errors. For example, the speech recognition performance of target speeches affects STD performance of a dynamic time warping (DTW)-based matching approach between a subword (such as a phoneme) sequence of a query term and a transcription of speech. Therefore, the STD performance of a DTW-based technique depends on the accuracy of subword-based transcriptions. Therefore, improvement of automatic speech recognition (ASR) performance for target speeches is also important to obtain good STD results. However, it is nearly impossible to remove ASR errors completely, although we use state-of-the-art ASR technologies. Therefore, it is necessary to develop an STD technique that is robust against ASR errors. For example, a lattice-based STD approach [4] has been proposed.

In a recent study, we proposed CN-based indexing and a DTW-based search engine [5]. The CN-based index, a phoneme transition network (PTN)-formed index [5], comprised of ten types of transcriptions generated by ten different ASR systems, including LVCSR and phoneme recognition systems. We have shown that the PTN-formed index comprising the multiple ASR systems' outputs obtained better STD results than that of the n-best output from a single ASR system. This STD system could outperform other STD technologies that participated in the ninth National Institute of Informatics Testbeds and Community for Information Access Research (NTCIR-9) project STD evaluation framework [6].

Although our DTW-based approach using a PTN-formed index for STD was very robust against ASR errors, the approach outputted many false detections from a PTN with a complex structure [7]. It is difficult to speech-recognize correctly an utterance including OOV words or unclearly pronounced words. Therefore, each ASR system can output different phoneme sequences from those with errors. This increases the number of PTN arcs, increasing the complexity of the PTN structure. Consequently, a query term might falsely match such complexly formed PTNs. In particular, a short query term, with a low number of phonemes, is likely to be detected incorrectly in wrong positions.

In this study, we focus on controlling false detections by the DTW-based STD engine [5] in a second-pass stage using a machine learning framework. The principal idea is to verify detections from the first stage, which uses the DTW-based approach, by estimating a correct phoneme sequence, represented as a combination of triphones, of a query term. We achieve the estimation by using conditional random fields (CRF)-based triphone detectors, which can estimate the detection probability of a query term detected by the DTW-based STD engine. This study examines the speech recognition error diversity in STD. Speech recognition error patterns are different in each ASR system. We

Manuscript received February 3, 2016.
Manuscript revised May 25, 2016.
Manuscript publicized July 19, 2016.

[†]The authors are with the Integrated Graduate School of Medicine, Engineering, and Agricultural Sciences, University of Yamanashi, Kofu-shi, 400–8580 Japan.
a) E-mail: sawada@alps-lab.org
b) E-mail: hnishi@yamanashi.ac.jp (Corresponding author)

DOI: 10.1587/transinf.2016SLP0012

show that phoneme error pattern training with different ASR systems can yield better CRF models compared to the one with N-best outputs from an ASR system. The ASR error patterns are trained using a CRF-based framework on each triphone from phoneme-level transcriptions by multiple ASR systems. Then, a correct phoneme (triphone) sequence in a PTN can be re-estimated using the CRF models.

Figure 1 shows our STD framework. In the preprocessing stage, search target speeches are automatically transcribed by NA (the number of ASR systems) types of phoneme-based transcriptions. Then, they are converted to a PTN-formed index. A triphone detection model for each possible triphone is also trained using features generated from NA types of phoneme-based transcriptions by training speeches. In the first-pass of the STD process, the DTW-based STD engine outputs the detections for a query term [5]. In the second-pass, we use a triphone detection model with a CRF-based framework for calculating the detection probability of query terms in an utterance. In this study, we used $NA = 10$ types of ASR systems.

In the process of filtering detected candidates at the second pass, first, a query term is decomposed into triphones, and for each triphone, whether a given utterance includes that triphone is determined using the corresponding CRF-based triphone detection model. A CRF-based model is trained for each triphone. Then, we calculate the term detection probability by calculating the sum of the products of the output probabilities from all the models. This is the detection probability of the query term of the given utterance. Finally, the probability is used to recompute the final detection score using the DTW-based approach. Although the CRF-based approach is used to filter false detections, it can work independently. In the experiment, we show the STD performance of the CRF-based approach only.

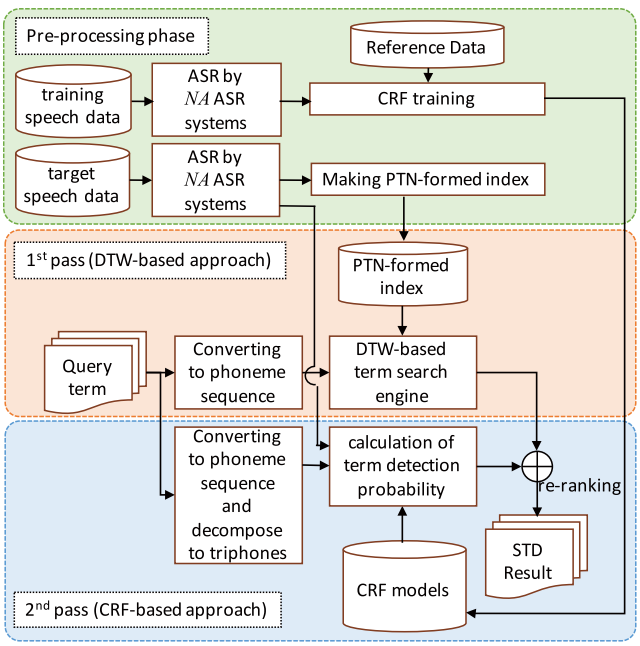

Fig. 1 Overview of the two-pass STD framework using CRF-based triphone detection modeling.

In experimental evaluation with the OOV subset of the Japanese test collection for STD [8] and the NTCIR-10 SpokenDoc-2 moderate-size task [9], the CRF-based approach alone could not outperform the DTW-based single approach on both tasks, but we found that a combination of CRF-based triphone detection and the DTW-based method in the STD process shows better performance than the DTW-based baseline approach on all evaluation metrics. In addition, the CRF-based models trained from the triphone sequences outputted by multiple ASR systems obtained better performance compared to the n-best triphone sequences of the single ASR system.

2. Related Work

Our CRF-based approach is similar to those of previous research [10], [11]. In those approaches, a phoneme sequence of some target speech is estimated using CRF models trained using ASR hypothesis-based features. This idea is similar to an acoustic modeling framework using CRF [12]. Chaudhari's technique [10], [11] was effective for the OOV detection task, because the CRF models learned phoneme confusions well.

Our study is intended as an extension of [10], [11], treating STD as a triphone sequence labeling problem for speech data. Chaudhari's technique [11] trains CRF models using training features related to phoneme-based n-grams extracted from a single phoneme sequence by an ASR system. Although we also use phoneme-based n-grams as a training feature for CRF models, our n-grams are generated from multiple phoneme sequences from multiple ASR systems' outputs. Our STD engine used in the first-step is based on a DTW-based approach using multiple ASR systems' outputs. The features for CRF training are extracted from the outputs of the same ASR systems. Therefore, using these n-gram features makes sense. In addition, The Chaudhari's CRF estimates the posterior probability of each monophone at any position and the posterior-grams (sequences of the posterior probabilities of monophones) are used to search a query term. Conversely, our CRF models can detect a suitable triphone with a probability considering phonetic context. This is a more robust detection of a phoneme from a phoneme label sequence than monophone detection. The detection probability of a triphone is used to calculate the confidence of a detected term by the DTW-based STD engine.

Further machine learning approaches for STD have been proposed recently. For example, Prabhavalkar et al. [13] proposed articulatory models using discriminative training for STD with low resource settings. They proposed an STD framework without any LVCSR system, and their models could detect a query term directly from acoustic feature vectors. Conversely, multiple linear regression, support vector machines, and multilayer perceptrons were also used to estimate confidence in the detected candidates

in a decision [14], [15] or re-ranking process [16]. Our CRF-based models train phoneme-to-phoneme confusion patterns on the basis of multiple types of transcriptions in contrast to these previous studies. In addition, our study investigates the effectiveness of a combination of outputs of multiple ASR systems. This is a new "cherry-picking" approach based on machine learning that trains phoneme-to-phoneme confusions.

The novelty of this study is that CRF is extended to provide the detection probability of a query term based on the estimation of a correct phoneme sequence and also in the decision process on the second pass of our STD framework by combining the DTW-based STD score with the CRF-based probability. In addition, we show that the CRF-based triphone detection models trained with multiple ASR systems' outputs are useful in filtering out false detection candidates. An advantage of this approach is that the DTW-based STD engine and CRF-based triphone detector are independent of ASR systems because they need only phoneme-based label sequences, that is, the proposed method can work on an ASR system-free framework. Therefore, we do not use any parameters related to ASR systems such as acoustic likelihood.

Our approach has been evaluated on the same OOV subset as reported already in the previous paper [17]. In addition, it has also been evaluated on the spoken query (SQ)-STD subtask of the NTCIR-11 SpokenQuery&Doc-1 task [18], [19], which is different from the task we use in this study. In the subtask, our CRF-based re-ranking approach did not outperform the baseline DTW-based STD approach because the CRF models for each triphone were trained using a different speech corpus from the first to sixth Spoken Document Processing Workshops (SDPWSs). However, this paper provides more detailed discussion on the types of features in CRF model training and the length of a query term.

3. DTW-Based Approach Using Multiple ASR Systems' Outputs

DTW-based STD using a PTN-formed index is performed in the first-pass stage on the entire STD framework, as in the baseline approach. Figure 2 shows an overview of the baseline method. In the indexing phase, speech data are processed using ASR, and the recognition outputs (words or sub-word sequences) are converted into the PTN-formed index for STD. Figure 3 shows an example of the development of a PTN-formed index for the speech "*Nepale*" (Japanese pronunciation is /n e p a a r u/ [†]) by aligning *NA* phoneme sequences from the best hypothesis of all the ASR systems. The speech was recognized by the *NA* ASR systems to yield *NA* hypotheses, which were then converted into phoneme sequences. Next, we obtained "aligned sequences" using the same dynamic programming (DP) scheme as described in [20]. Finally, a PTN was obtained by converting the aligned sequences. The symbol "@" in Fig. 3 indicates a null transition.

In the search phase, the word-formed query is converted into a phoneme sequence. Then, the phoneme-formed query is inputted to the term search engine. The term search engine searches for the query term from the index at the phoneme level using the DTW framework. Unlike the combination techniques of multiple STD systems described in [21], the baseline system combines the transcriptions produced by multiple ASR systems.

We calculate the distance between a query and a PTN using Bhattacharyya distance (BD) [22]. The BD between phoneme p and q is calculated by the monophone-based acoustic models of p and q.

The total DTW cost $D(i, j)$ at the grid point (i, j) (i =

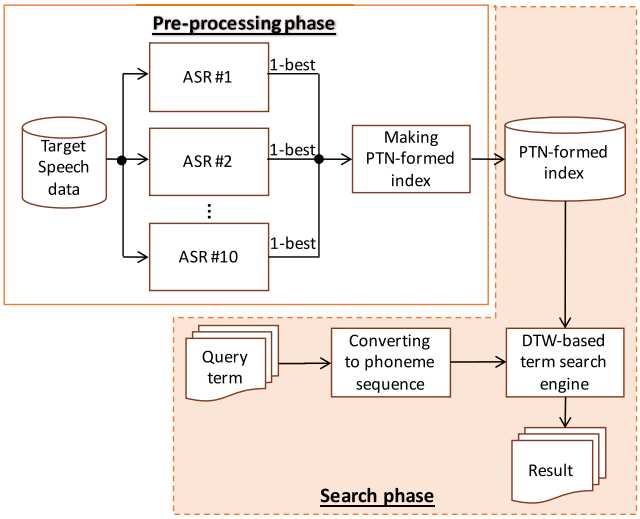

Fig. 2 Overview of the first-pass stage using DTW-based matching.

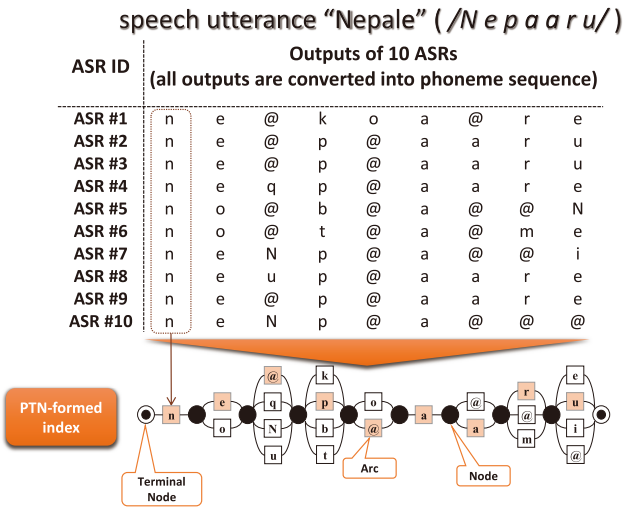

Fig. 3 Generation of a PTN-formed index by performing alignment using DP and converting to a PTN.

[†]In this paper, a vowel and a long vowel are modeled separately in triphone-based acoustic models, such as /a/ and /a:/. Conversely, syllable-based acoustic model does not have long vowels. Therefore, a long vowel is represented by repeating single vowel, such as /a a/.

$\{0, \ldots, I\}$, $j = \{0, \ldots, J\}$, where I and J are the number of the set of arcs in the index and query terms, respectively) on the DTW lattice was calculated using the following equations:

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

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

$$Vot(i, j) = \begin{cases} \alpha \div (Voting(p) + \Sigma_q(BD(q) \\ \qquad \times BDVoting(q))) \\ \qquad : \exists p \in PTN(i), p = Query(j), \\ \qquad q \neq Query(j) \\ \alpha : Query(j) \notin PTN(i) \end{cases} \quad (3)$$

$$Del(i) = min \begin{cases} minBD(p, q) : \exists p \in PTN(i-1), \\ \qquad q = Query(j) \\ minBD(p, q) : \exists p \in PTN(i), \\ \qquad q = Query(j) \end{cases} \quad (4)$$

$$Null(i) = min \begin{cases} NullVot(i) : Null \in PTN(i) \\ minBD(p, q) : \exists p \in PTN(i-1), \\ \qquad q \in PTN(i) \\ minBD(p, q) : \exists p \in PTN(i), \\ \qquad q \in PTN(i+1) \end{cases} \quad (5)$$

$$NullVot(i) = \beta \div Voting(Null) \quad (6)$$

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.

Equations (2) and (3) are related to the cost calculation for a substitution error. $minBD$ in Eq. (2) is the smallest BD between j and any phoneme in $PTN(i)$. $Vot(i, j)$ is a confidence parameter for the matching between $PTN(i)$ and $Query(j)$. $Voting(p)$ is the number of ASR systems outputting the same phoneme p at the same arc. A greater value of $Voting(p)$ improves the reliability of phoneme p. $BD(q)$ is the BD between $Query(j)$ and phoneme q, which does not correspond to $Query(j)$. $BDVoting(q)$ also indicates the number of ASR systems outputting the phoneme q. The cost for a deletion error is calculated based on Eq. (4). $minBD(p, q)$ is a BD between phoneme p and q. Equations (5) and (6) are used for the cost calculation of an insertion error. We allow a null transition between two nodes in the PTN-formed index with the cost $NullVot(i)$ defined in Eq. (6). α and β are hyperparameters set to 0.5 and 0.45, respectively. They were optimized using the development set. The appropriate null cost achieves increasing term detection and decreasing numbers of false detections.

In advanced searches for the query term, the term detection engine initializes $D(i, 0) = 0$ (both endpoints are free), and then it calculates $D(i, j)$ using Eq. (1) ($i = \{0, \ldots, I\}$, $j = \{1, \ldots, J\}$). 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 of $D(i, J)$ below a threshold θ.

4. CRF-Based Triphone Detection

CRFs [23] have been used successfully in numerous text processing tasks, such as named entity extraction [24] and phrase chunking [25]. In speech processing, CRFs are used for sentence boundary detection [26] and OOV detection in speech [27]. In this section, we describe how to calculate a detection probability of a query term in an utterance using CRF-based triphone detectors.

4.1 Overview

Figure 4 shows an overview of the STD process using CRF-based triphone detection modeling in the second-pass stage on the entire STD framework. In this study, we use ten ($NA = 10$) types of phoneme-based transcriptions generated by ten different ASR systems for training CRF-based models. A query term is translated into a phoneme sequence and decomposed into triphones. Then, a CRF-based triphone model calculates the detection probability of the triphone corresponding to that model in an utterance. The final term detection probability (score) is based on the sum of the products of the output probabilities from all of the triphone models. In this research, we prepared two types of acoustic models (AMs), five types of language models (LMs), and a decoder. The AM and LM combinations resulted in ten ASR systems. The details of these models are explained in Sect. 6.1.

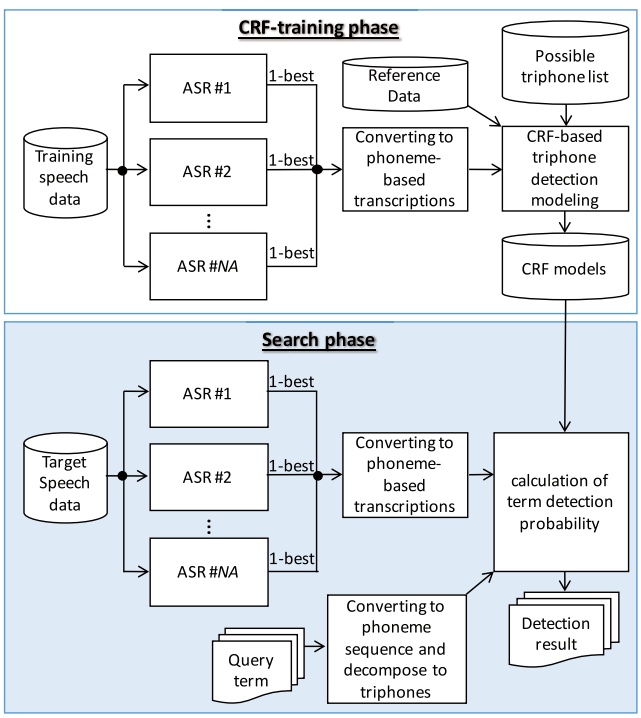

Fig. 4 Overview of the STD framework using CRF-based triphone detection modeling.

4.2 Training Label Definition and Features for CRF Training

In this study, we use CRFs to detect a triphone in an utterance. Therefore, we prepare a CRF-based triphone detector for each triphone. Idealy, although it is desirable that CRFs estimate a term detecion probabity directly, this is impossible, because CRF models detect words that cannot be trained. Conversely, all words can be decomposed into phoneme sequences. Therefore, we use a phoneme detector for word detection. In addition, context information is very useful for speech processing. We create triphone detectors considering phoneme-to-phoneme confusion error patterns in the CRF framework.

Figure 5 shows an example of training features for the triphone "*n-e-p*"and an output label definition on an utterance for CRF training. First, training data for CRFs are generated using the ten sorts of ASR systems, which are the same as those for creating the PTN-formed index described in Sect. 3. A DP-based alignment procedure [20] was also performed on the ten sorts of phoneme-based transcriptions and the reference (correct) phoneme sequence for creating an alignment between the phonemes at a position. Finally, we can obtain the phoneme-based alignment sequence on the transcriptions.

We used BIO (beginning/inside/outside)[†] encoding [27], [28] in the CRF-based triphone detection modeling. CRF with the BIO encoding framework was used to solve a text-chunking problem. As mentioned in Sect. 2, the BIO encoding framework is completely different from that of previous work [11], which directly estimates a phoneme from a phoneme-based transcription using CRF. We attempt to achieve triphone detection by solving a sequence-labeling problem. In the BIO encoding method, the B tag corresponds to the head phoneme of a target triphone, the I tag indicates the inside phones of the target triphone except for the head, and the O tag shows the target triphone. To achieve that, we must prepare BIO tag labeling for each alignment. As each alignment corresponds to the correct phoneme label in the reference, we can put one of the BIO tags on each alignment depending on the triphone. In Fig. 5, the "current token" alignment corresponds to the correct phoneme "e" and the I tag because "e" is the middle phoneme of the triphone "*n-e-p*". The BIO-tag labeling procedure is performed for all possible triphones. A CRF model for a triphone estimates the occurrence probabilities of the B, I, and O tags in an alignment sequence. By considering these probabilities for triphones composing a query term, we can calculate the detection probability of the query term.

Next, we describe features that are used to train a CRF-based model. As shown in Fig. 5, we prepare four types of features: unigrams, in-ASR bigrams, cross-ASR bigrams, and in-ASR trigrams. Table 1 shows a list of training features and the number of features for each type of feature.

[†]IOB representation is alternatively used.

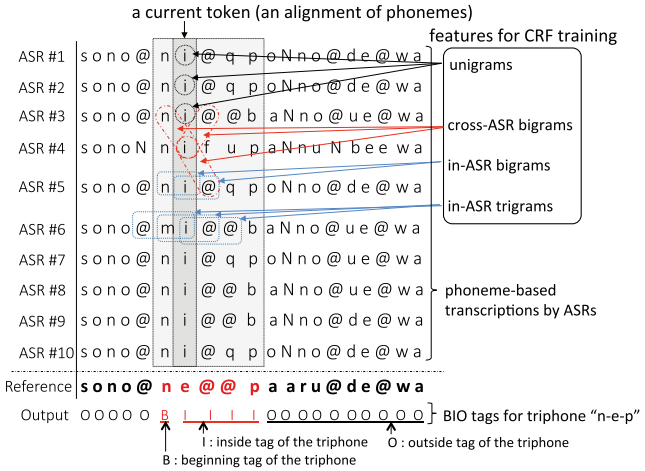

Fig. 5 Example of features for CRF model training and BIO encoding.

Table 1 List of training features for a CRF-based triphone detector. # indicates the number of feature types.

Feature type	Definition	#
Unigram	$(r_p,\ h^i_t)$	10
in-ASR bigram	$(r_p,\ h^i_{p-1},\ h^i_p),\ (r_p,\ h^i_p,\ h^i_{p+1})$	20
cross-ASR bigram	$(r_p,\ h^i_{p-1},\ h^j_p \mid i \neq j),$ $(r_p,\ h^i_p,\ h^j_{p+1} \mid i \neq j)$	180
in-ASR trigram	$(r_p,\ h^i_{p-2},\ h^i_{p-1},\ h^i_p),$ $(r_p,\ h^i_{p-1},\ h^i_p,\ h^i_{p+1}),$ $(r_p,\ h^i_p,\ h^i_{p+1},\ h^i_{p+2})$	30

h^i_p is the phoneme at current position p of the phoneme sequence from the i^{th} ASR system, and r_p is the BIO tag. The aim of this CRF modeling is to learn phonetic confusion error patterns with contexts in each ASR system.

Output label sequences and alignment sequences generated by phoneme-level transcriptions are used to train CRF-based triphone detectors. The conditional probability of an input (alignment) sequence $\mathbf{x}$ for a triphone t, given an output (BIO) label sequence $\mathbf{y}_t$, is calculated as follows:

$$P_t(\mathbf{y}_t|\mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\left\{\sum_k \lambda_k F_k(\mathbf{x}, \mathbf{y}_t)\right\} \tag{7}$$

where $F_k(\mathbf{x}, \mathbf{y}_t)$ is the k^{th} feature representation for the input alignment sequence $\mathbf{x}$, and λ_k is the weight parameter for $F_k(\mathbf{x}, \mathbf{y}_t)$. $Z(\mathbf{x})$ is a normalization factor given by:

$$Z(\mathbf{x}) = \sum_y \exp\left\{\sum_k \lambda_k F_k(\mathbf{x}, \mathbf{y}_t)\right\} \tag{8}$$

We train the CRF models for all possible triphones t. A trained CRF model for a triphone can calculate the posterior probabilities of B, I, and O tags at any alignment position of a testing utterance. By applying all the trained models to testing utterances, we can obtain the sequences of the posterior probabilities of BIO tags for all possible triphones. The number of sequences is equal to the number of possible triphones. We show term detection using these probabilities.

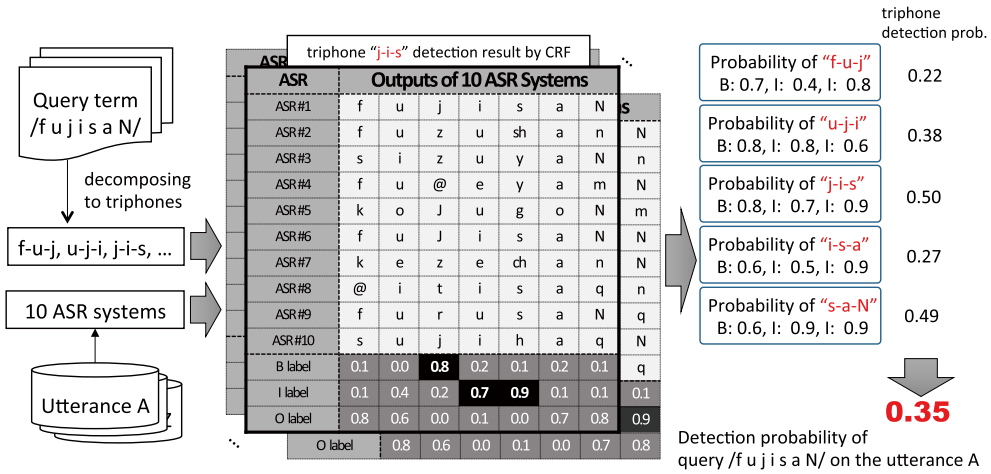

Fig. 6 Example of a term detection based on triphone detection models.

4.3 Term Detection

The term detection probability $P(T|\mathbf{x}_i)$ of a query term T consisting of N triphones in the input alignment sequence $\mathbf{x}$ of utterance i is calculated using the following equation:

$$P(T|\mathbf{x}_i) = \left\{ \prod_{j=1}^{N} P_{t_j}(\mathbf{y}_{t_j}|\mathbf{x}_i) \right\}^{\frac{1}{N}}, \quad (l_{t_1} < l_{t_j} < l_{t_N}) \tag{9}$$

where t_j is the j^{th} triphone of T, $\mathbf{x}_i$ is the input sequence of utterance i, and $\mathbf{y}_{t_j}$ is a part of the BIO label sequence for triphone t_j. l_{t_j} indicates the location (position) of the beginning phoneme of triphone t_j. A triphone is extracted many times in the same utterance. Using the constraint of the occurrence positions of triphones composing T reduces wasteful computation and false alarm detections of T. Note that $P_{t_j}(\mathbf{y}_{t_j}|\mathbf{x}_i)$ is not calculated using the conditional probability of the entire label sequence for utterance i, but using the sum of the products of probabilities of each B and I tag. The probability of O tag output is not considered because our aim is to detect possible triphones. This idea is similar to maximum entropy modeling. However, CRFs can find an optimal labeling for an entire sequence. Therefore, CRF-based models can detect triphones with high accuracy. The detection probability of triphone t_j is calculated using:

$$P_{t_j}(\mathbf{y}_{t_j}|\mathbf{x}_i) = \prod_{L=\mathrm{B}^{\mathrm{t_j}}}^{\mathrm{I}^{\mathrm{t_j}}_{\mathrm{tail}}} P_{t_j}(L|\mathbf{x}_i) \tag{10}$$

where $\mathrm{B}^{\mathrm{t_j}}$ and $\mathrm{I}^{\mathrm{t_j}}_{\mathrm{tail}}$ represent the beginning and tailing tags of triphone t_j, respectively. In other words, the detection probability of t_j is calculated by taking the product of the conditional probability of each tag between the head $\mathrm{B}^{\mathrm{t_j}}$ and tailing $\mathrm{I}^{\mathrm{t_j}}_{\mathrm{tail}}$ tags. The start position for detecting triphone t_j is based on the probability of B^{t_j}. If B^{t_j} has a probability value greater than probability ϕ, the detection probability of t_j is calculated using Eq. (10). If the head triphone of t_j is detected at the different position from B^{t_j} (this is denoted as $B^{t_{j+1}}$), the position $B^{t_{j+1}}$ will show the head triphone of t_j. Therefore, the same triphones are sometimes detected at the close positions. If $P_{t_j}(\mathbf{y}_{t_j}|\mathbf{x}_i)$ is less than probability ϕ, then $P_{t_j}(\mathbf{y}_{t_j}|\mathbf{x}_i)$ is set to ϕ. ϕ is a smoothing parameter that prevents a very low or zero detection probability for T when any triphone consisting of T is detected with very low probability or cannot be detected. In this study, ϕ is heuristically set to 0.01. If $P(T|\mathbf{x}_i)$ is greater than a threshold θ_C, then term T appears to be in utterance i. Changing the threshold θ_C enables us to draw the recall-precision curve of the evaluation.

Figure 6 shows an example of a term "*Mt. Fuji*" (`/f u j i s a N/`) detected using CRF-based triphone detection models. For example, the detection probability of a triphone "*j-i-s*" in the utterance A is 0.50 ($0.8 \times 0.7 \times 0.9$), which is calculated based on the products of the posterior probabilities of the B and I tags. The final detection probability of the query term for the utterance is 0.35, which is the n^{th} root of the sum of the products of all of the triphones comprising the query term.

5. Re-Ranking of First-Pass Detections

We used the simple combination of DTW- and CRF-based scores (detection probabilities) given by Eq. (11), which is well-known as a weighted harmonic mean. The recomputed detection score $\mathrm{RS}(T, i)$ is calculated as follows:

$$\mathrm{RS}(T, i) = \frac{(\gamma^2 + 1) \cdot \mathrm{DTW}(T, i) \cdot \mathrm{CRF}(T, i)}{\gamma^2 \cdot \mathrm{DTW}(T, i) + \mathrm{CRF}(T, i)} \tag{11}$$

where γ is a weight parameter that controls the balance between $\mathrm{CRF}(T, i)$ and $\mathrm{DTW}(T, i)$, and $\mathrm{CRF}(T, i)$ and $\mathrm{DTW}(T, i)$ are the scores of term T in utterance i derived using the CRF- and DTW-based STD methods, respectively. Both scores from the two approaches range from zero to one. γ is set to 0.05, which is determined by the in-vocabulary (INV) subset of the Japanese test collection [8] and is common for all of the query terms in the test collection.

6. STD Experiment

6.1 Test and Development Set

We used two types of STD test collections to verify our proposed method. One is the CORE subtask of the NTCIR-9 SpokenDoc-1 STD task [6] (CSJ-CORE set). The test collection targets 177 lectures (39 hours) in the Corpus of Spontaneous Japanese (CSJ) [29]. The number of utterances is 53,892. This test collection contains a total of 50 query terms. The average number of occurrences per term is 7.1. The other test collection is the moderate-size task of NTCIR-10 SpokenDoc-2, which contains 104 oral presentation speeches (28.6 hours) from the first to sixth annual SDPWS (SDPWS set). The number of query terms is 100. The average number of occurrences per term is 9.0.

In addition, we prepared a development set for γ parameter tuning of Eq. (11). We used the IVN subset of the Japanese test collection for STD [8], which includes 50 IVN query terms.

As shown in Figs. 2 and 4, the speech data were recognized by the ten ASRs. Julius ver. 4.1.3 [30], an open source decoder for LVCSR, was used in all the systems. We prepared two types of AM and five types of LM for constructing the PTN. The AMs are triphone-based (Tri.) and syllable-based hidden Markov models (HMMs) (Syl.) with both types of HMM trained from the spoken lectures except for the 177 lectures in the CSJ. All the 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. Words consisting 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.

None: No LM is 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 speech data of STD. The AMs and LMs are trained under the same condition as in the previous work [5].

Finally, ten combinations, consisting of two AMs and five LMs, are formed.

Table 2 shows the phoneme recognition accuracy rate of each ASR system on the evaluation corpora.

Table 2 Phoneme recognition accuracy rate (%) of each ASR system.

LM / AM	CSJ	SDPWS
WBC / Tri.	93.2	88.4
WBH / Tri.	93.1	88.3
CB / Tri.	90.8	85.4
BM / Tri.	91.8	86.3
None / Tri.	88.2	66.7
WBC / Syl.	88.9	81.1
WBH / Syl.	88.8	81.6
CB / Syl.	88.1	80.9
BM / Syl.	85.7	75.6
None / Syl.	86.0	54.6

6.2 CRF-Based Model Training and Feature Types

Our CRF-based triphone detection models were trained from the portion of the CSJ other than the 177 lectures that used a CRF++ toolkit[†]. A total of 2,525 speeches were used to train models. The number of trained triphone models was 908, derived from 43 types of Japanese monophones, and we did not use any clustering algorithm for grouping similar triphones together as AM training before training the CRF-based models.

We introduced four kinds of feature types in Table 1. In this paper, we use two types of feature sets from Table 1 as follows:

1. all of the types of features,
2. unigrams, in-ASR bigrams, and trigrams.

We verify the effectiveness of learning the relationships of error patterns between and across the ASR systems using these two feature sets.

In addition, we trained CRF models from the n-best phoneme-based transcriptions outputted by the single ASR system (Tri./WBC) that achieved the best ASR performance. They were compared to the CRF models trained from the multiple ASR systems. We use the best and ten best transcriptions of the ASR system in this paper.

6.3 Evaluation Metrics

The evaluation metrics used included recall, precision, F-measure of optimal point on a recall-precision curve, and mean average precision (MAP) values [6]. These measures are used officially in the test collections.

$$Recall = \frac{N_{corr}}{N_{true}} \tag{12}$$

$$Precision = \frac{N_{corr}}{N_{corr} + N_{spurious}} \tag{13}$$

$$F - measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{14}$$

Here, N_{corr} and $N_{spurious}$ are the total numbers of correct and spurious (false) term detections, respectively, and N_{true} is the total number of true term occurrences in the speech data. The F-measure values for the optimal balance of *Recall* and *Precision* values are denoted by "maximum F-measure."

[†]CRF++: Yet Another CRF toolkit, https://code.google.com/p/crfpp/

The STD performance for the query sets can be illustrated using a recall−precision curve, which is plotted by changing the threshold θ_C in the CRF-based STD method or θ_D in the DTW-based baseline.

MAP is the mean of the average precision values for each query term, calculated as follows:

$$MAP = \frac{1}{Q}\sum_{q=1}^{Q} AveP(q) \tag{15}$$

where Q is the number of full queries and $AveP(q)$ is the average precision of the q^{th} query term of the query set. Average precision is calculated by averaging the precision values computed for each relevant term in the list in which retrieved terms are ranked by a relevance measure.

$$AveP(q) = \frac{1}{Rel_q}\sum_{r=1}^{N_q}(\delta_r \cdot Precision_q(r)) \tag{16}$$

where r is the rank, N_q is the rank number at which all the relevance terms of query term q are detected, and Rel_q is the number of the relevance terms of the query term q. δ_r is a binary function for a given rank r.

6.4 Experimental Results

Figures 7 and 8 show the recall-precision curves of each STD approaching the CSJ-OOV and SDPWS sets, respectively. Tables 3 and 4 also represent maximum F-measure and MAP values of the STD methods. We then compared the STD performances of three STD methods. The STD system (1) explained in Sect. 3 is the baseline in this study, and (2)−(5) are the CRF-based approaches only. "CRF-1" refers to the CRF models trained with all of the feature types for CRF training, and "CRF-2" also refers to the models trained with all of the feature types except for cross-ASR bigram features, while "multi" indicates that the training data of the CRF models consist of multiple ASR systems. Conversely, "single" systems use only the best output or ten best outputs from a single ASR system (Tri./WBC). Systems (6), (7), (8), and (9) are the proposed approaches, which recompute the scores of the detections obtained by the baseline, and (7) and (9) are revised versions of (6) and (8), respectively. In (7) and (9), the CRF-based approach is applied only to detections by the baseline system for a query term with fewer than 11 phonemes ($N \leqq 10$) [†]. The CRF-based re-ranking is not applied in (7) and (9) to query terms with more than ten phonemes.

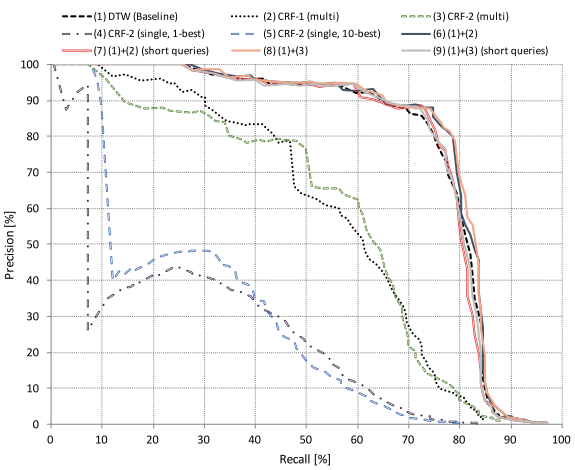

Fig. 7 Recall-precision curves of the STD methods on the CSJ-OOV set.

First, we discuss the feature types of the CRF-based triphone detection models. Comparing CRF-1 and CRF-2 in Table 3, we see that CRF-2 obtained better STD performance than CRF-1 on the CSJ-OOV set. Conversely, for the

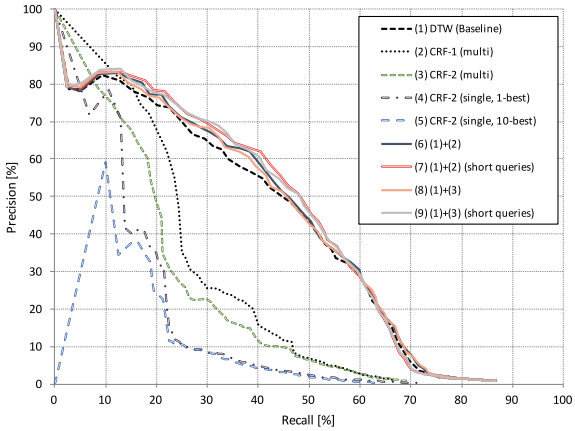

Fig. 8 Recall-precision curves of the STD methods on the SDPWS set.

Table 3 Maximum F-measure and MAP values of the STD methods on the CSJ-OOV set.

	Systems	Max. F-measure (%)	MAP
(1)	DTW	78.61	0.863
(2)	CRF-1 (multi)	58.60	0.771
(3)	CRF-2 (multi)	61.14	0.795
(4)	CRF-2 (single, 1-best)	37.22	0.582
(5)	CRF-2 (single, 10-best)	39.16	0.574
(6)	(1)+(2)	**80.74**	0.882
(7)	(1)+(2) (short queries)	79.72	0.869
(8)	(1)+(3)	80.47	**0.893**
(9)	(1)+(3) (short queries)	79.16	0.869

Table 4 Maximum F-measure and MAP values of the STD methods on the SDPWS set.

	Systems	Max. F-measure (%)	MAP
(1)	DTW	47.35	0.621
(2)	CRF-1 (multi)	32.98	0.477
(3)	CRF-2 (multi)	28.57	0.460
(4)	CRF-2 (single, 1-best)	25.48	0.367
(5)	CRF-2 (single, 10-best)	23.90	0.358
(6)	(1)+(2)	47.92	0.624
(7)	(1)+(2) (short queries)	**49.34**	**0.626**
(8)	(1)+(3)	47.79	0.625
(9)	(1)+(3) (short queries)	49.26	**0.626**

[†]The average number of phonemes consisting of all the query terms is approximately ten. Therefore, we divided the query set equally into two groups based on the number of phonemes more than ten and less than 11.

SDPWS set, CRF-1 performed better than CRF-2 (Table 4). The cross-ASR bigram features can train the relationship of phoneme confusion patterns across ASR systems. On the CSJ-OOV set, the training and testing data are from the same corpus. Therefore, the CRF models were well trained, using only unigram-, in-ASR bigram-, and trigram-based features. For the SDPWS set, the environment between the training and testing of the CRF models was unmatched. In addition, the ASR performance of the SDPWS speeches was lower than that of the CSJ speeches as shown in Table 2. In that case, the CRF model cannot be trained adequately with only in-ASR-based features. The cross-ASR feature was useful for error pattern training.

Next we discuss the training data of the CRF models. As shown in Tables 3 and 4, we were able to improve the CRF models when we used transcriptions outputted from the multiple ASR systems, even though the training volume of multi and single, ten-best are the same.

Although the CRF-based STD-only studies (2)−(5) did not perform well compared to the baseline approach (1), the re-ranking approach (6)−(9) of the DTW- and CRF-based STD outputs improved STD performance (both maximum F-measure and MAP) compared to the baseline approach (1). The recall-precision curves of (6)−(9) in Figs. 7 and 8 from the re-ranked detections also improved substantially. Comparing (6) with (8) on both the sets, we see that there are no significant differences between CRF-1 and CRF-2, both of which are used in the re-ranking approach, on the two evaluation metrics.

The CRF labeling performances[†] were approximately 85% and 70% on the CSJ-OOV and SDPWS sets, respectively. The reason the labeling performance for the SDPWS set was so much worse was the lower ASR performances of each ASR system (Table 2) and the unmatched environment between the training and testing of the CRF-based models. This resulted in the estimation of the correct triphone on the SDPWS being more difficult than for the CSJ. Consequently, for the SDPWS set, a long query term that consists of more triphones is difficult to detect from transcriptions using the CRF models because the detection probability is likely to be low. However, the difficulty of triphone detection on the SDPWS set did not have a negative impact on the STD performance for the short query terms compared to the long query terms because the short queries consisted of a limited number of triphones. As shown in Table 5, the STD performance for the short query terms improved more than any of the query terms in the SDPWS set. As shown in (7) and (9) from Fig. 8 and Table 4, we obtained the best STD performance when the CRF-based re-ranking approach was applied to only the short query terms. Conversely, the CRF-based models trained on the matched environment worked well on the evaluation for the CSJ-OOV set because phoneme-to-phoneme confusion patterns trained using CRF were fit well to the target transcriptions (matched condition). Therefore, in the case of the CSJ-OOV set, applying the CRF-based re-ranking to all of the query terms improved the entire STD performance compared to applying it to only the short query terms.

[†]The correct label (B and I) detection rates were 85.4% (CRF-1) and 86.0% (CRF-2) for the CSJ. The rates for the SDPWS were 70.3% (CRF-1) and 70.5% (CRF-2).

Table 5 Maximum F-measure and MAP values for only the short query terms.

Systems	CSJ-OOV set		SDPWS set	
	F-measure	MAP	F-measure	MAP
(1)	78.18	0.838	39.85	0.537
(6)	80.00	0.855	42.18	0.549
(8)	79.45	0.853	41.46	0.548

Finally, the experimental results showed that CRF-based triphone detection modeling is useful in providing confidence regarding terms detected using the different types of STD technique despite the unmatched condition between training and testing of the CRF models.

7. Conclusion

In this study, we proposed a CRF-based re-ranking approach that recomputes the detection scores provided by the DTW-based STD engine. The CRF model finds triphones comprising a query term from an utterance. We used CRF-based triphone detection models based on features generated from multiple types of phoneme-based transcriptions that are used for creating a PTN-formed index used in the DTW-based approach. The aim of this approach is to train recognition error patterns such as phoneme-to-phoneme confusions on the CRF framework and to control false detections from the DTW approach. In the STD experiment on the OOV subset for Japanese test collection from the CSJ and the NTCIR-10 moderate-size task from the SDPWS speeches, the CRF-based approach alone could not outperform the DTW-based STD approach by using the outputs of multiple ASR systems. However, we found that the CRF-based method could perform accurate detections. In the end, the combination of CRF- and DTW-based methods yielded the best STD performance. In particular, we also showed that STD performance for short query term detection in the unmatched environment between training and testing of the CRF-based models was improved by using the CRF-based approach.

As future work, we intend to study a triphone clustering approach to training CRF-based models. This approach might solve the training data shortage problem and improve the detection accuracy of each triphone. In addition, we will attempt to investigate how to set the weight parameter γ automatically and dynamically for each query term. We expect this to achieve more improvement than a uniform value of γ for all query terms.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP26282049, JP15K00254.

References

[1] "The spoken term detection (STD) 2006 evaluation plan," 2006. http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf.
[2] D. Vergyri, I. Shafran, A. Stolcke, R.R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006 spoken term detection system," Proc. of INTERSPEECH 2007, pp.2393–2396, 2007.
[3] 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," Proc. of INTERSPEECH 2008, pp.2146–2149, 2008.
[4] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," IEEE Trans. on Audio, Speech and Language Processing, vol.19, no.8, pp.2338–2347, 2011.
[5] S. Natori, Y. Furuya, H. Nishizaki, and Y. Sekiguchi, "Spoken Term Detection Using Phoneme Transition Network from Multiple Speech Recognizers' Outputs," J. Information Processing, vol.21, no.2, pp.176–185, 2013.
[6] T. Akiba, H. Nishizaki, K. Aikawa, T. Kawahara, and T. Matsui, "Overview of the IR for Spoken Documents Task in NTCIR-9 workshop," Proc. of NTCIR-9, pp.223–235, 2011.
[7] S. Natori, Y. Furuya, H. Nishizak, and Y. Sekiguchi, "Entropy-based False Detection Filtering in Spoken Term Detection Tasks," Proc. of APSIPA ASC 2013, pp.1–7, 2013.
[8] 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," Proc. of INTERSPEECH 2010, pp.677–680, 2010.
[9] T. Akiba, H. Nishizaki, K. Aikawa, X. Hu, Y. Itoh, T. Kawahara, S. Nakagawa, H. Nanjo, and Y. Yamanashita, "Overview of the NTCIR-10 SpokenDoc-2 Task," Proc of NTCIR-10, pp.573–587, 2013.
[10] U.V. Chaudhari and M. Picheny, "Improved vocabulary independent search with approximate match based on conditional random fields," Proc. of ASRU 2009, pp.416–420, 2009.
[11] U.V. Chaudhari and M. Picheny, "Matching criteria for vocabulary-independent search," IEEE Trans. on Audio, Speech and Language Processing, vol.20, no.5, pp.1633–1643, 2012.
[12] A. Gunawardana, M. Mahajan, A. Acero, and J.C. Platt, "Hidden conditional random fields for phone classification," Proc. of INTERSPEECH 2008, pp.1117–1120, 2005.
[13] R. Prabhavalkar, K. Livescu, E. Fosler-Lussier, and J. Keshet, "Discriminative articulatory models for spoken term detection in low-resource conversational settings," Proc. of ICASSP 2013, pp.8287–8291, 2013.
[14] D. Wang, S. King, J. Frankel, and P. Bell, "Term-dependent confidence for out-of-vocabulary term detection," Proc. of INTERSPEECH 2009, pp.2139–2142, 2009.
[15] J. Tejedor, A. Echeverria, and D. Wang, "An evolutionary confidence measurement for spoken term detection," Proc. of International Workshop on Content-Based Multimedia Indexing (CBMI 2011), pp.151–156, 2011.
[16] T. wei Tu, H. yi Lee, and L. shan Lee, "Improved spoken term detection using support vector machines with acoustic and context features from pseudo-relevance feedback," Proc. of ASRU 2011, pp.383–388, 2011.
[17] N. Sawada, S. Natori, and H. Nishizaki, "Re-Ranking of Spoken Term Detections Using CRF-based Triphone Detection Models," Proc. of APSIPA ASC 2014, pp.1–4, 2014.
[18] T. Akiba, H. Nishizaki, H. Nanjo, and G.J.F. Jones, "Overview of the NTCIR-11 SpokenQuery&Doc Task," Proc. of NTCIR-11, pp.350–364, 2014.
[19] H. Nishizaki, N. Sawada, S. Natori, K. Domoto, and T. Utsuro, "Combination of DTW-based and CRF-based Spoken Term Detection on the NTCIR-11 SpokenQuery&Doc SQ-STD Subtask," Proc. of NTCIR-11, pp.402–408, 2014.
[20] J.G. Fiscus, "A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," Proc. of ASRU'97, pp.347–354, 1997.
[21] M. Akbacak, L. Burget, W. Wang, and J. van Hout, "Rich system combination for keyword spotting in noisy and acoustically heterogeneous audio streams," Proc. of ICASSP 2013, pp.8267–8271, 2013.
[22] B. Mak and E. Barnard, "Phone clustering using the Bhattacharyya distance," Proc. of ICSLP'96, pp.2005–2008, 1996.
[23] J.D. Lafferty, A. McCallum, and F.C.N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," Proc. of ICML '01, pp.282–289, 2001.
[24] K. Nongmeikapam, T. Shangkhunem, N.M. Chanu, L.N. Singh, B. Salam, and S. Bandyopadhyay, "Crf based name entity recognition (ner) in manipuri: A highly agglutinative indian language," Proc. of the 2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS), pp.1–6, 2011.
[25] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," Proc. of NAACL'03, pp.134–141, 2003.
[26] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using conditional random fields for sentence boundary detection in speech," Proc. of ACL'05, pp.451–458, 2005.
[27] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek, "Contextual information improves oov detection in speech," Proc. of HLT-NAACL 2010, pp.216–224, 2010.
[28] W. Chen, Y. Zhang, and H. Isahara, "An empirical study of chinese chunking," Proc. of COLING/ACL 2006, pp.97–104, 2006.
[29] K. Maekawa, "Corpus of Spontaneous Japanese: Its design and evaluation," Proc. of SSPR 2003, pp.7–12, ISCA, 2003.
[30] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine julius," Proc. of APSIPA ASC 2009, pp.131–137, 2009.

Naoki Sawada was born in 1991. He received his B.E. and M.E. degrees in engineering from University of Yamanashi in 2014 and 2016. He is now a doctoral course student at the Integrated Graduate School of Medicine, Engineering, and Agricultural Sciences, University of Yamanashi. His research interests include spoken language processing, in particular speech term detection. He is a student member of the Acoustical Society of Japan (ASJ) and the Information Processing Society of Japan (IPSJ).

Hiromitsu Nishizaki was born in 1975. He received his B.E., M.E., and PhD. Eng. degrees in information and computer sciences from Toyohashi University of Technology in 1998, 2000, and 2003. He is now an associate professor in the graduate school of interdisciplinary research, faculty of engineering, University of Yamanashi. His research interests include spoken language processing. He is a member of IEEE, the ASJ and the IPSJ.