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

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

Spoken Term Detection Using SVM-Based Classifier Trained with Pre-Indexed Keywords

Kentaro DOMOTO[†*], *Nonmember*, **Takehito UTSURO**[†a)], *Senior Member*, **Naoki SAWADA**[††], *Nonmember*, *and* **Hiromitsu NISHIZAKI**[††], *Senior Member*

SUMMARY This study presents a two-stage spoken term detection (STD) method that uses the same STD engine twice and a support vector machine (SVM)-based classifier to verify detected terms from the STD engine's output. In a front-end process, the STD engine is used to pre-index target spoken documents from a keyword list built from an automatic speech recognition result. The STD result includes a set of keywords and their detection intervals (positions) in the spoken documents. For keywords having competitive intervals, we rank them based on the STD matching cost and select the one having the longest duration among competitive detections. The selected keywords are registered in the pre-index. They are then used to train an SVM-based classifier. In a query term search process, a query term is searched by the same STD engine, and the output candidates are verified by the SVM-based classifier. Our proposed two-stage STD method with pre-indexing was evaluated using the NTCIR-10 SpokenDoc-2 STD task and it drastically outperformed the traditional STD method based on dynamic time warping and a confusion network-based index.

key words: *decision process, pre-indexing, spoken term detection, support vector machine, verification*

1. Introduction

Spoken term detection (STD) is one of the core technologies in spoken language processing. STD enables us to search for a specified word from recorded speech, and its effectiveness has been demonstrated through an electronic note-taking system [1]. However, STD is difficult to use when searching for terms within a vocabulary-free framework because search terms are not known by the STD process prior to implementing a large vocabulary continuous speech recognition (LVCSR) system. Several studies have proposed methods to address this issue with STD [2], [3].

For example, STD techniques that use lattice- [4], [5] or confusion network (CN)-formed [6], [7] transcriptions to handle automatic speech recognition (ASR) errors and subword-based transcriptions for a vocabulary-free framework [8], [9] have been studied, and they improved STD performance. In contrast to these ASR-based approaches, Prabhavalkar et al. [10] proposed articulatory models that included discriminative training for STD under low-resource settings. They challenged an STD framework without any ASR system, and their models could directly detect a query term from acoustic feature vectors.

In ASR-based approaches, STD using lattice- or CN-formed phoneme sequences that have more rich information than probable (1-best) phoneme sequences from an ASR system's output were very robust against ASR errors [11], [12]. However, diverse information hinders STD performance because it contains significant amounts of redundant information. For example, for phoneme-level matching between a query term and CN-formed transcriptions in a dynamic time warping (DTW)-based framework, a query term falsely matches the incorrect phoneme paths on the CNs [12]. As a result, several false term detections (false alarms) are output. In particular, a short query term composed of fewer syllables is likely to be falsely detected. Therefore, preventing such false detections by an STD engine is important.

As a result, an increasing number of machine learning approaches for STD have been proposed. For example, deep learning, multiple linear regression, support vector machines (SVMs), and multilayer perceptrons have been used to estimate the confidence level of detected candidates in decision [13]–[15] and re-ranking processes [16], [17].

In this study, we also employ a decision process for detected candidates using a machine learning approach. Our approach, which is two-stage STD with a pre-indexing framework, uses an STD engine twice and an SVM-based classifier. Figure 1 and Fig. 2 show an example of the process of the proposed two-stage STD with a pre-indexing framework. As shown in Fig. 1, a keyword list is built from the result of an ASR process on spoken documents. Then, each keyword in the list is searched for using a DTW-based STD engine [18]. Note that a detected keyword candidate has a matching cost and an occurrence position. Therefore, different keywords are detected at the same position (competitive position). In that case, we select one keyword from a competitive position. The selected keyword is registered as the pre-indexed keyword. This is a front-end process of the whole framework.

The STD process is illustrated in Fig. 2. In the STD process, which is divided into two stages, a query term is first input to the same STD engine used for pre-indexing and is then searched for. Then, detection candidates are ob-

Manuscript received February 5, 2016.
Manuscript revised May 17, 2016.
Manuscript publicized July 19, 2016.

[†]The authors are with the Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba-shi, 305–8573 Japan.
[††]The authors are with the Integrated Graduate School of Medicine, Engineering, and Agricultural Sciences, University of Yamanashi, Kofu-shi, 400–8511 Japan.
[*]Presently, with NTT DATA CORPORATION.
a) E-mail: utsuro@iit.tsukuba.ac.jp (Corresponding author)

DOI: 10.1587/transinf.2016SLP0017

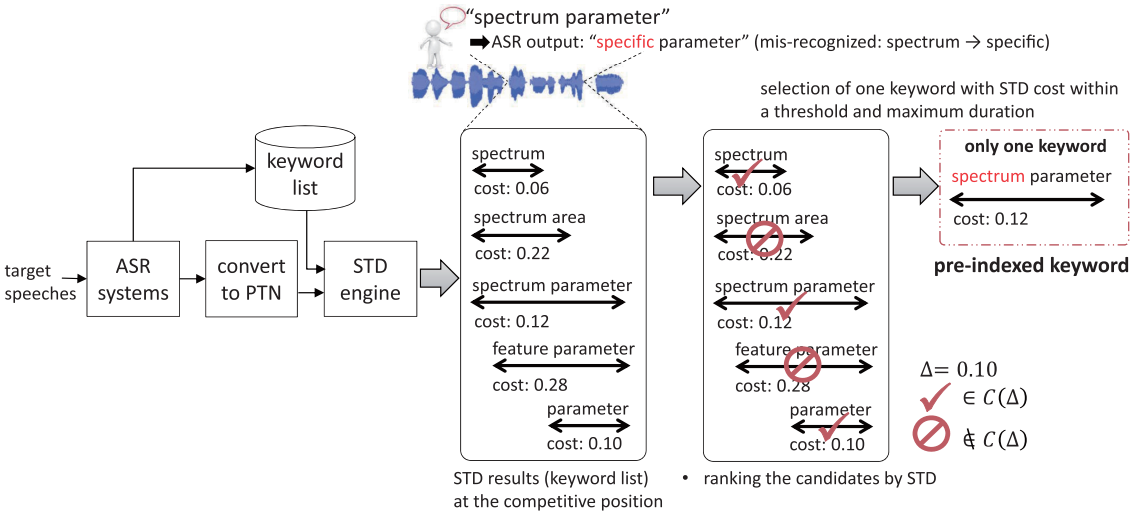

Fig. 1 Pre-indexing for SVM classifier to verify candidates in Fig. 3; selection of the best matched keyword from a competitive interval position.

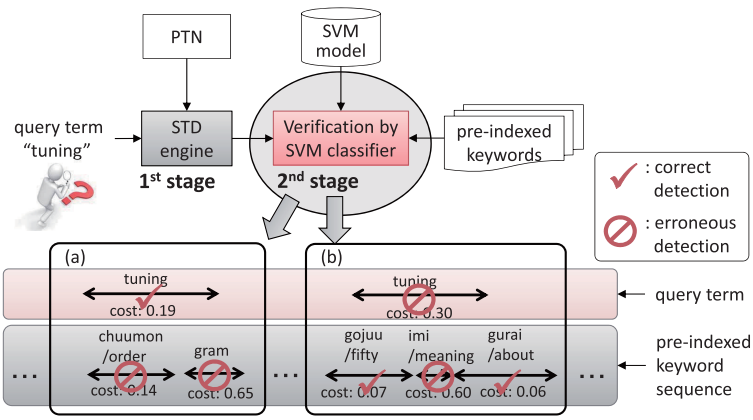

Fig. 2 STD using SVM verification in Fig. 3.

tained. In the second stage of STD, the detection candidates are verified to determine whether they are confident using an SVM-based classifier. Figure 2 shows two verification cases using the SVM-based classifier. For case (a) in Fig. 2, the query term is preferred to competitive keywords, and for case (b) in Fig. 2, some competitive keywords are preferred to the query term. Case (b) in Fig. 2 can be considered a typical example of avoiding over-detection with the proposed framework.

Our two-stage approach is similar to previous methods that have focused on a decision process. However, the proposed framework differs from previous studies in that features derived from a pre-index of spoken documents are used to train a classifier. Most previous studies used acoustic features [19], [20], ASR-related information [21], lattice-based information [22], [23], and similar approaches. In contrast, we demonstrate the effectiveness of the proposed decision process, wherein STD outputs are verified using an SVM-based classifier trained with pre-indexed best match keyword-based features. This is quite novel in that an SVM-based classifier trained with features that are completely different from those proposed previously is used in a decision process. Incidentally, our approach does not depend on a specific STD engine as studied in [18]. We have already reported an SVM-based classification method [24]. However, this paper provides more detailed discussion about the types of features used for SVM training and the length of query terms.

We have evaluated the proposed framework using academic lecture speeches as spoken documents for the STD task. The proposed framework verified the detected candidates effectively because the precision rate drastically improved in the lower 60% of the recall rate compared to traditional baseline STD. In particular, the proposed framework elicited a beneficial effect on verification for the detection candidates of short query terms.

The remainder of this paper is organized as follows. DTW-based STD using multiple ASR systems is described in Sect. 2. Section 3 discusses the two-stage STD method, which includes a pre-indexing method prior to an STD pro-

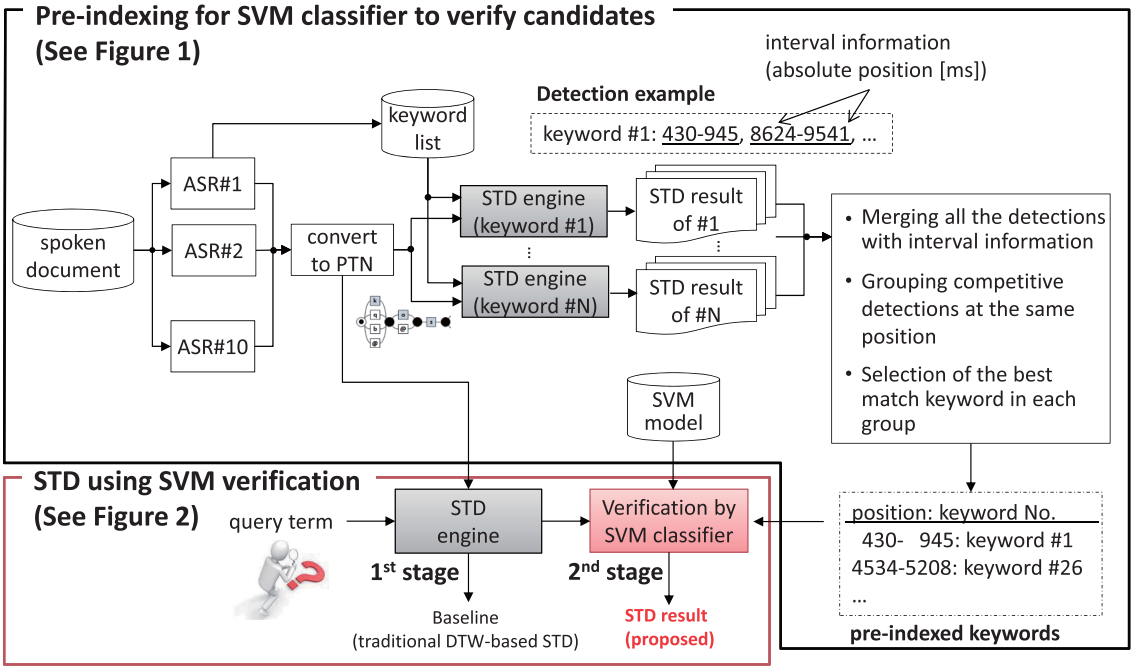

Fig. 3 Overview of the proposed two-stage STD framework with SVM verification.

cess and the decision process using an SVM-based classifier. Evaluation results are given in Sect. 4, and conclusions are presented in Sect. 5.

2. DTW-Based STD Engine and ASR

We have employed an STD engine [18] that uses subword-based CNs. We used a phoneme transition network (PTN)-formed index derived from one-best hypotheses of multiple ASR systems and an edit distance-based DTW framework to detect a query term. We employed 10 types of ASR systems, and the same decoder was used for all ASR systems. Two types of acoustic models and five types of language models were prepared. The multiple ASR systems can generate a PTN-formed index by combining subword (phoneme) sequences from the output of the ASR systems into a single CN. The details of the STD engine are explained in the literature [18]. The STD engine includes some parameters for DTW. In addition, we used an STD engine with false-detection parameters of "Voting" and "AcwWith," which achieved the best STD performance on the evaluation set of the 9th NII Testbeds and Community for Information Access Research (NTCIR-9) SpokenDoc-1 STD task [25].

Julius ver. 4.1.3 [26], which is an open-source decoder for ASR, was used in all systems. The acoustic models were triphone- and syllable-based hidden Markov models (HMMs), wherein each state uses a Gaussian mixture model (GMM). The acoustic and language models were trained with spoken lectures from the Corpus of Spontaneous Japanese (CSJ) [27]. All language models were word- and character-based trigrams as follows:

WBC: Word-based trigram where words are represented by a mix of Chinese characters and Japanese Hiragana and Katakana.

WBH: Word-based trigram where all words are represented by only Japanese Hiragana. Words comprising Chinese characters and Japanese Katakana are converted to Hiragana sequences.

CB: Character-based trigram where all characters are represented by Japanese Hiragana.

BM: Character-sequence-based trigram where the unit of language modeling comprises two Japanese Hiragana characters.

None: No language model is used. Speech recognition without any language model is equivalent to phoneme (or syllable) recognition.

Each model except for "None" was trained using the CSJ transcriptions. The training conditions for all acoustic and language models and the ASR dictionary are the same as the STD/SDR test collections used in the NTCIR-10 SpokenDoc-2 moderate size task [28]. The DTW-based STD engine outputs detections with matching costs that are normalized with the phoneme length of the query term and are scaled from 0 to 1.

3. Two-Stage STD with Pre-Indexing

3.1 Overview

Figure 3 shows an outline of the proposed two-stage STD framework, which uses the same STD engine in both the pre-indexing and STD processes, and an SVM-based classifier. Our STD framework is primarily divided into three processes: (1) a pre-indexing (front-end) process using an

STD engine, (2) the first term search process for an input query term using the same STD engine, and (3) a verification process in which the candidates detected in the STD phase are either accepted or rejected as final output using an SVM-based classifier.

The pre-indexing process includes building PTNs from search target spoken documents and creating pre-indexed keywords. First, the 10 types of ASR systems were applied to the spoken documents to create PTNs. Note that we used the DTW-based STD engine with the PTN described in Sect. 2; however, our framework is independent of specific STD methods[†]. Next, all keywords other than those with one morae are extracted from the transcription of the spoken documents by the ASR system, wherein the word-based trigram and triphone-based GMM-HMM models are used, and are collected into the keyword list[††]. All keywords in the keyword list are searched for by the STD engine, and all detected candidates are merged using the interval information of each candidate. Finally, one keyword is selected from a competitive detection group and registered in the pre-index of the spoken documents.

In the term search process, when a query is input to the same STD engine as the pre-indexing process, it outputs detection candidates. Finally, the candidates are verified using an SVM-based classifier trained with features related to the competitive keywords in the verification process. Finally, only confident candidates are the final STD results.

3.2 Pre-Indexing Process

Figure 1 shows the pre-indexing method used for the target spoken documents. The method indexes the keywords with a matching cost that is less than the threshold at the competitive position. We explain the details of the pre-indexing method in the following paragraphs.

First, we define a quadruplet, which comprises a keyword w, the start time t of its detection interval, its end time t', and the STD matching cost of the keyword $cost$. The competitive detection set C comprising N quadruplets $\langle w, t, t', cost\rangle$ is defined as follows:

$$C = \left\{\langle w_1, t_1, t'_1, cost_1\rangle, \ldots, \langle w_N, t_N, t'_N, cost_N\rangle\right\}$$

where a time interval $[t_i, t'_i]$ must overlap at least one other time interval $[t_j, t'_j]$ $(i \neq j)$ in the competitive detection set. C in the case shown in Fig. 1 is represented as follows:

$$\begin{aligned} C = \Big\{ & \langle\text{“spectrum”}, t_1, t_3, 0.06\rangle, \\ & \langle\text{“spectrum area”}, t_1, t_4, 0.22\rangle, \\ & \langle\text{“spectrum parameter”}, t_1, t_5, 0.12\rangle, \\ & \langle\text{“feature parameter”}, t_2, t_5, 0.28\rangle, \\ & \langle\text{“parameter”}, t_3, t_5, 0.10\rangle \Big\}. \end{aligned}$$

Here, $t_1 < t_2 < t_3 < t_4 < t_5$. In this method, we first find the quadruplet with the smallest matching cost from C:

$$\langle w_{min}, t, t', cost_{min}\rangle.$$

In the case shown in Fig. 1, the following quadruplet is selected:

$$\langle\text{“spectrum”}, t_1, t_3, 0.06\rangle.$$

Next, the candidate set $C(\Delta)$ for pre-indexing is created by filtering quadruplets in C based on cost-range Δ because quadruplets whose matching cost are out of range for the quadruplet with smallest cost are unlikely to be incorrect candidates. Note that the quadruplets in $C(\Delta)$ have an STD cost less than $(cost_{min} + \Delta)$:

$$C(\Delta) = \left\{\langle w, t, t', cost\rangle \in C \mid cost \leq (cost_{min} + \Delta)\right\}.$$

For example, in Fig. 1, if $\Delta = 0.10$, then $C(\Delta = 0.10)$ is represented as follows:

$$\begin{aligned} C(\Delta = 0.10) = \Big\{ & \langle\text{“spectrum”}, t_1, t_3, 0.06\rangle, \\ & \langle\text{“spectrum parameter”}, t_1, t_5, 0.12\rangle, \\ & \langle\text{“parameter”}, t_3, t_5, 0.10\rangle \Big\}. \end{aligned}$$

Finally, we select the quadruple $\langle w_{ld}, t, t', cost_{ld}\rangle$ from $C(\Delta)$ that has the longest duration (ld)[†††], where the duration of a detected keyword is defined as $t' - t$. The keyword w_{ld} is output as the STD result and used as a pre-indexing word for spoken documents. In the example shown in Fig. 1, the following quadruple is the final output:

$$\langle\text{“spectrum parameter”}, t_1, t_5, 0.12\rangle.$$

This is a pre-indexed keyword in the interval $[t_1, t_5]$. Note that the same pre-indexing process is performed for the other intervals of the spoken documents. As a result, each interval

[†]Although the proposed framework is independent of specific STD methods, the employed pre-indexing based on the PTN approach is advantageous compared to other approaches, such as the word-graph based approach. This is primarily because a subword based STD [18], where query terms are matched against lattices of subwords (phonemes), outperforms [18] word-graph based STD, where query terms are matched against lattices of words.

[††]As discussed in Sect. 4.1, the keywords to be used in the pre-indexing process are collected from the spoken documents transcribed by the ASR system with a vocabulary of 27,000 words. For example, for spoken documents with a total duration of 28.6 hours used in our evaluation, the number of keywords collected and used in the pre-indexing process was 6,575. Thus, the number of the keywords used in the pre-indexing process is constant in proportion to the size of the evaluation speech documents, which implies that the computational complexity of generating the keyword list to be used in the pre-indexing process is linear in proportion to the size of the evaluation speech documents. In addition, since the number of the keywords used in the pre-indexing process is constant in proportion to the size of the evaluation speech documents, the computational complexity of the pre-indexing process is linear in proportion to the size of the evaluation speech documents.

[†††]The preliminary evaluation includes an examination of keyword selection based on the shortest duration criterion as well as random selection, where the longest duration criterion performed the best.

Table 1 Features used for training SVM-based classifier (the asterisk (∗) next to each feature ID indicates that the feature is optimal)

Type	ID	Feature	Definition
Competitive set	f_1^c*	The detected query does or does not have maximum duration	Whether the interval $t'_q - t_q$ of the detected query $q = \langle w_q, t_q, t'_q, cost_q \rangle$ is the maximum duration in the competitive detection set C_q.
	f_2^c*	The number of competitive keywords	$\lvert C_q \rvert$
	f_3^c*	The minimum cost value at the competitive position	The minimum cost value $cost$ of $\langle w, t, t', cost \rangle$ within the competitive detection set C_q.
Query	f_1^q*	The number of morae of the query	The number of morae of the query
	f_2^q	The query is out-of-vocabulary (OOV) or in-vocabulary (INV)	Set to 1 if the query is INV; otherwise, set to 0
	f_3^q*	The matching cost of the query	$cost_q$
	f_4^q	The duration time of the query	$t'_q - t_q$
Competitive keyword w_k	f_1^{wk}	The number of morae of a competitive keyword	The number of morae of a competitive keyword
	f_2^{wk}	Character type of a competitive keyword	For each of the five character types of Kanji, Hiragana, Katakana, mixture of Kanji and Hiragana, and alphanumeric characters, add a binary feature indicating the character type of the competitive keyword.
	f_3^{wk}	Part-of-speech (POS) of a competitive keyword	For each of 14 POS types, add a binary feature indicating the POS of the competitive keyword.
	f_4^{wk}	The DTW-based matching cost of a competitive keyword	$cost$
	$f_5^{wk}*$	The duration time of a competitive keyword	$t' - t$
	$f_6^{wk}*$	The overlapped duration between a competitive keyword and the query	$min(t'_q, t') - max(t_q, t)$
	$f_7^{wk}*$	The overlapped rate between a competitive keyword and the query	$\frac{(min(t'_q, t') - max(t_q, t))}{(t'_q - t_q)}$
	f_8^{wk}	The difference between the matching costs of a competitive keyword and the query	$cost_q - cost$
	f_9^{wk}	The distance between a competitive keyword and the query	An edit distance at the phoneme level

has one pre-indexed keyword[†]. The pre-indexed keyword-based features are used to train an SVM-based classifier that verifies the term detection candidates of a search query term.

3.3 Term Search Process

In the term search process, the same STD engine is used to detect query terms. As a query term is input to the STD engine, the query is searched for, and the detection candidates are output. This is the same as the traditional STD process [18], which is used as a baseline system in this study. In addition, we create competitive detection sets using the detection candidates for the verification process. Each detection of a query term denoted as $\langle w_q, t_q, t'_q, cost_q \rangle$ is included in one competitive detection set, where all keywords in the competitive detection set overlap with the query. Details are given in the following.

The pre-indexed keywords can be represented as a sequence of quadruples $\langle w, t, t', cost \rangle$ as follows:

$$\langle w, t, t', cost \rangle_l, \ldots, \langle w, t, t', cost \rangle_m. \quad (1)$$

Note that the intervals $([t, t']_l, \ldots, [t, t']_m)$ do not overlap[††]. When a query term w_q is input to the same STD engine as that used to create the pre-index, the STD engine outputs detected candidates. We define a detection candidate denoted q as a quadruple as follows.

$$q = \langle w_q, t_q, t'_q, cost_q \rangle$$

Here, the interval $[t_q, t'_q]$ of q overlaps one or more pre-indexed keywords out of the sequence of competitive keywords of Eq. (1) as follows:

$$\langle w, t, t', cost \rangle_u, \ldots, \langle w, t, t', cost \rangle_v.$$

In this case, we can represent the competitive detection set C as C_q as follows:

$$C_q = \Big\{ \langle w, t, t', cost \rangle_u, \ldots, \langle w, t, t', cost \rangle_v, \langle w_q, t_q, t'_q, cost_q \rangle \Big\}. \quad (2)$$

The final STD decision process verifies whether q is confident using an SVM-based classifier. Note that this process is performed for all detected candidates q.

[†]Here, rather than selecting a single keyword from a competitive keyword set of an interval, we compare selecting two or three keywords from a competitive keyword set of an interval. The result shows that selecting only a single keyword performs the best.

[††]In our evaluation, for each interval of a query term, at least one pre-indexed keyword overlaps with the query term interval.

3.4 Verification Process Using SVM-Based Classifier

For each competitive detection set C_q from Eq. (2), we prepared 16 types of features to train the SVM-based classifier. Table 1 shows a list of these features. All of the features listed in Table 1 are represented as the set F_{all}:

$$
\begin{aligned}
&F_{all} = \\
&\{\ f_1^c,\ f_2^c,\ f_3^c, && \text{(competitive set feature)}\\
&\quad f_1^q,\ f_2^q,\ f_3^q,\ f_4^q, && \text{(query feature)}\\
&\quad f_1^{w1},\ \ldots,\ f_9^{w1}, && \text{(feature for the competitive keyword } w_1)\\
&\qquad\vdots && \qquad\vdots\\
&\quad f_1^{wk},\ \ldots,\ f_9^{wk} && \text{(feature for the competitive keyword } w_k)\\
&\qquad\vdots && \qquad\vdots\\
&\quad f_1^{wM_{max}},\ \ldots,\ f_9^{wM_{max}} && \text{(feature for the competitive keyword } w_{M_{max}})\\
&\}
\end{aligned}
$$

Note that all features are related to a competitive set, competitive keywords registered in a pre-index, and a query. "Competitive set" and "Characteristics of the competitive keyword set" in Table 1 are obtained by the pre-indexing process. In addition, for each competitive keyword w_k, we use nine features $f_1^{wk},\ \ldots,\ f_9^{wk}$ as shown in Table 1. Here, among all competitive detection sets C_q in the training data, let M_{max} be the maximum number of competitive keywords as follows:

$$M_{max} = \max_{C_q} |C_q| - 1$$

Then, we define the set F_{all} of features for a competitive keyword set C_q as having features for the maximum number of competitive keywords $w_1, \ldots, w_{M_{max}}$[†] [††] [†††].

[†]In the evaluation in Sect. 4, more than 96% of the competitive detection set C_q had less than or equal to three competitive keywords, and the maximum number M_{max} of competitive keywords was seven. When assigning feature numbers to those competitive keywords within a competitive detection set C_q, we ordered them in ascending order of cost.

[††]For nearly all of the competitive detection sets C_q, for competitive keywords $w_1, \ldots, w_M$, their number M was less than M_{max}, and we assigned the values of the features of competitive keywords $w_{M+1}, \ldots, w_{M_{max}}$, which are actually not included in the competitive detection set C_q, to 0.

[†††]For the number of competitive keywords included in the feature set F_{all}, we compared the following four approaches in a preliminary evaluation:

(1) only the competitive keyword with minimum STD matching cost is included;
(2) only competitive keywords with the minimum and second minimum STD matching costs are included;
(3) only competitive keywords with the minimum, second minimum, and third minimum STD matching costs are included;
(4) all competitive keywords $w_1, \ldots, w_{M_{max}}$ are included

In our preliminary evaluation, the approach (4) achieved the best performance.

As shown in Table 1, the eight types of features with IDs with an asterisk (∗) are the optimal features selected by a backward elimination method [29]. First, we define F as the feature set for SVM training and F_f^{-1} indicates a feature set where only one feature f is removed from F. For all L features, L of the SVM-based classifiers are trained using $F_1^{-1}, \ldots, F_L^{-1}$ and are evaluated through 100-fold cross-validation (Sect. 4.1). Then, F_f^{-1}, which yields the best STD performance among $F_1^{-1}, \ldots, F_L^{-1}$, is identified and the feature f is then removed from F. Repeating this procedure to remove a feature one-by-one, we obtained eight features that achieved the best performance compared to other feature sets.

4. Evaluation

4.1 Experimental Setup

We used the moderate-size STD NTCIR-10 SpokenDoc-2 task [28] as the STD task for our evaluation. The evaluation speech data were obtained from the Corpus of Spoken Document Processing Workshop. It consisted of recordings of the first to sixth annual Spoken Document Processing Workshop, comprising 104 real oral presentations (total duration is 28.6 hours). The ASR system, wherein the word-based trigram with a vocabulary of 27,000 words and triphone-based GMM-HMM models are used, was used to create word-based transcriptions of the evaluation speech data. Next, all keywords other than those with one morae were extracted from the transcription. The number of keywords collected from the transcription of the spoken documents that were used in the pre-indexing process was 6,575. The number of query terms was 100, where 47 of all query terms were INV (in vocabulary) queries included in the ASR vocabulary of the word-based trigram model. The other 53 queries were OOV (out of vocabulary). The number of INV and OOV query occurrences in all speech data was 444 and 456, respectively. The word error rate was 36.9% when the WBC language model and triphone-based GMM-HMM were used. The STD cost range Δ was determined as 0.20 by examining Δ at 0.05, 0.10, 0.15, 0.20, 0.25, and 0.30 through 100-fold cross-validation.

We used the LIBSVM tool [30] as the SVM-based classifier with the radial basis function (RBF) kernel. A cost parameter and a gamma parameter of the RBF kernel were set from the results of a grid search using a five-fold cross-validation framework using the target speech data. All feature values were scaled from 0 to 1 using LIBSVM's svm-scale tool. Note that the SVM-based classifier can output a result with a confidence score. The detected candidates, which are determined as "confident," were sorted in descending order of confidence scores. We can draw a recall-precision curve by changing the threshold for the confidence score.

An SVM-based classifier is generally influenced by the balance of the numbers of positive (query is uttered) and negative (query is not uttered) examples in the training data.

Of the total number of positions where a query term was detected by the DTW-based STD engine, the number of negative (query is not uttered) examples was approximately 3,800 times greater than that of positive (query is uttered) examples. Therefore, we removed negative examples whose matching costs were over an upper bound of 0.35, where the number of negative examples is approximately 10 times greater than that of positive examples[†].

In the training and testing of the SVM classifier, we trained the SVM classifier using 90% of the speech data and 90% of the query terms, and we tested the SVM classifier using the remaining speech data and query terms. We repeated this procedure 100 times by varying the combination of 90% and 10% decomposition of the speech data and query terms. Finally, we concatenated the results of the 100 procedures and obtained an evaluation result against 100% of the speech data and 100% of the query terms.

We compared the results of the baseline and **three** types of STD systems, depending on the type of feature sets, to demonstrate the effectiveness of the proposed STD system. Here, "**baseline**" was used as the DTW-based STD engine (Sect. 2). We used the following three types of feature sets:

"all features:" All 16 types of features in Table 1 were used for training;

"selected features:" Only the eight types of effective features were used;

"query features only:" Only the four types of features (f_1^q, f_2^q, f_3^q, and f_4^q) related to a query term were used. Note that these features not arise from the pre-indexing process. The goal of using this feature set was to investigate the effectiveness of the pre-indexed features. The preparation of these features does not require the pre-indexing process; therefore, we can compare the verification abilities of a classifier trained with and without pre-indexed keywords-related features.

4.2 Evaluation Metrics

The evaluation metrics used in this study were recall, precision, and F-measure. These measurements are frequently used to evaluate information retrieval performance and are defined as follows.

$$Recall = \frac{N_{corr}}{N_{true}} \tag{3}$$

$$Precision = \frac{N_{corr}}{N_{corr} + N_{spurious}} \tag{4}$$

[†]When we set the upper bound of matching costs to greater than 0.35, the SVM training procedure did not converge. In addition, when we set the upper bound to less than 0.35, more negative and more positive examples were removed from both the training and test data, and we obtained significantly reduced recall compared to when the upper bound was set to 0.35.

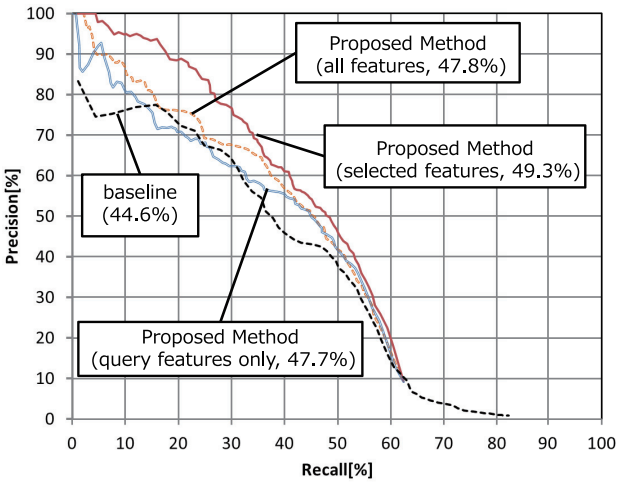

Fig. 4 Recall-precision curves and Max. F-measure values for each STD system (numbers in parentheses show Max. F-measure values)

$$F - measure = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{5}$$

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 F-measure values for the optimal balance of *Recall* and *Precision* values are denoted by "Max. F-measure."

4.3 Evaluation Results

Figure 4 shows the results of the baseline and three types of feature sets used to train the SVM-based classifiers.

As shown in Fig. 4, the proposed STD system (the selected feature set) achieved better STD performance compared to the baseline STD system. The proposed method drastically improved the precision rates in the range below 60% recall rate (with a 1% level of significance in the range 10% to 60% recall). Max. F-measure value of the curve generated by the proposed STD system also improved. This shows that the verification process of the SVM-based classifier is effective to filter out false detections output by the baseline STD engine.

Next, we discuss the three types of feature sets for SVM-based classifier training, i.e., all features, selected features, and only query features. As shown in Fig. 4, the selected feature set prevailed against the all feature set. The optimal feature set made the classifier better.

On the other hand, the SVM-based classifier trained with query features only did not outperform the others although it won the baseline. Note that this SVM-based decision process does not necessarily require pre-indexed keywords. However, further analysis showed that the pre-indexed keyword-based features worked well with this SVM-based decision process. This result is shown in Fig. 4 where "Proposed Method (selected features)" outperformed "Proposed Method (query features only)" for precision at a 1% level of significance in the range 10% to 30% recall and

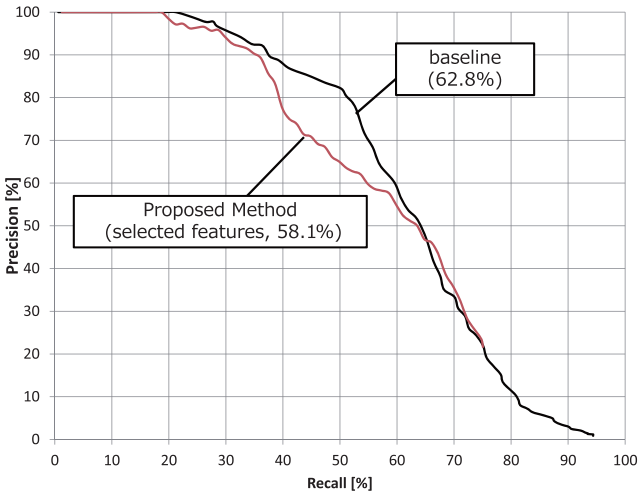

Fig. 5 Recall-precision curves and Max. F-measure values for long query terms with more than six syllables (50 of the total 100 query terms; numbers in parentheses show Max. F-measure values)

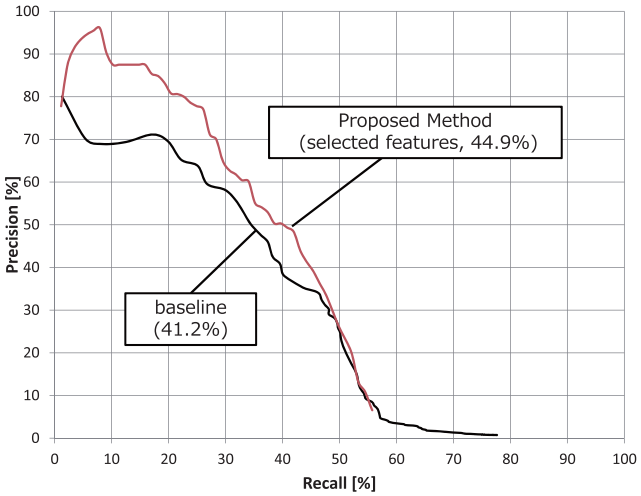

Fig. 6 Recall-precision curves and Max. F-measure values for short query terms with less than seven syllables (50 of the total 100 query terms; numbers in parentheses show Max. F-measure values)

at a 5% level of significance in the range 10% to 60% recall. Therefore, it can be said that it effectively prepared the pre-index with richer information when using the STD engine[†].

Figure 5 and Fig. 6 show the effectiveness of the proposed framework for two types of query sets. The average number of syllables consisting of a query term in the evaluation set was 6.6. We divided the query set into two sets, i.e., a long query term set, where the query terms had more than six syllables (50 query terms of the total 100), and a short query term set, where the query terms had less than seven syllables (50 query terms of the total 100). Figure 5 and Fig. 6 show the recall-precision curves of the baseline system and the SVM-based classifier with the two query sets. As can be seen in these figures, the SVM-based classifier had significant influence on the short query terms. To examine the reason for this result, we observed the relation of the length of the query terms and the confidence assigned by the SVM-based classifier. Here, we discovered that most long query terms were assigned higher confidence even if they were actually not uttered in the speech data. On the other hand, for short query terms, most were assigned much lower confidence and thus contributed to avoiding over-detection. This is primarily because we trained a single SVM-based classifier for both the long and short query terms. If we train two SVM-based classifiers separately, i.e., one for long query terms and another for short query terms, then it is expected that we could achieve much higher performance for the long query terms.

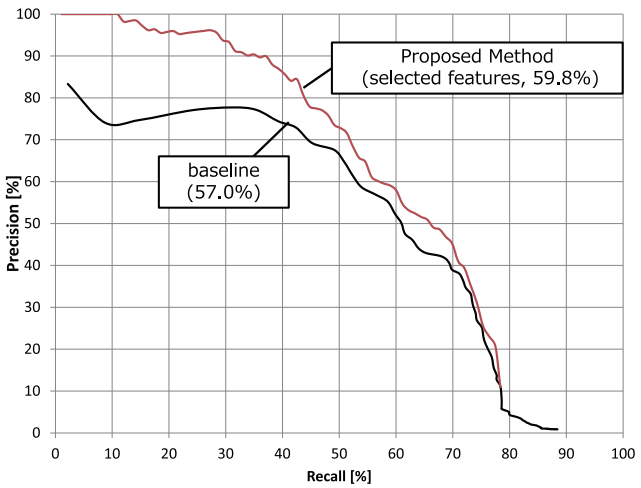

Fig. 7 Recall-precision curves and Max. F-measure values for INV query terms (numbers in parentheses show Max. F-measure values)

Furthermore, we divided the total 100 query terms into 47 INV query terms and 53 OOV query terms. We compare their performance in Fig. 7 and Fig. 8, respectively. As shown by the results, the proposed framework outperformed the baseline for both INV and OOV query terms. This result is quite remarkable considering the fact that the pre-indexing process of the proposed framework utilizes the keyword list collected from only an ASR transcription. Similarly, from the 47 INV query terms, we extracted 26 query

[†]We also compared the MAP (mean average precision) values [25] of the four plots in Fig. 4, i.e., the baseline and three types of feature sets, where their MAP values are 0.594 for the baseline, 0.547 for all the feature set, 0.572 for the selected feature set, and 0.548 for the query features only. Among these results, it is most important to note that the proposed method with the selected feature set significantly outperforms the proposed methods with the other two feature sets, which agrees with the results of the recall-precision curves shown in Fig. 4, where the proposed method with the selected feature set clearly outperforms the other methods. In addition, the baseline outperforms the proposed methods simply because it outputs more candidates for detecting query terms than the proposed methods, thereby resulting in a recall-precision curve in the range greater than 60% recall but less than 10% precision. On the other hand, for the three plots of the proposed methods, the proposed method carefully selects candidates to detect query terms, thereby resulting in recall-precision curves that end within the range greater than 10% precision but less than 60% recall.

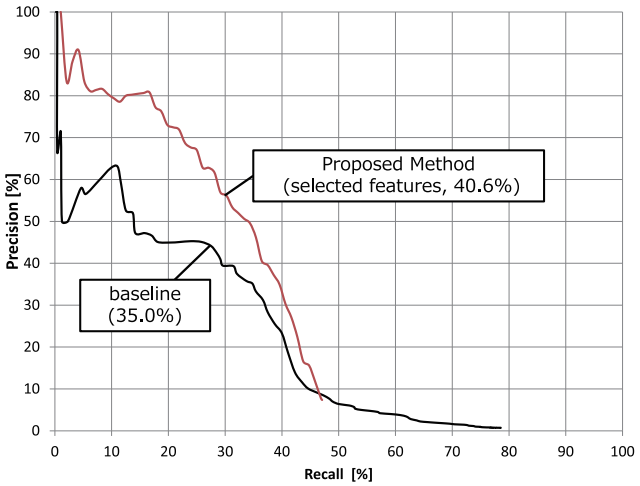

Fig. 8 Recall-precision curves and Max. F-measure values for OOV query terms (numbers in parentheses show Max. F-measure values)

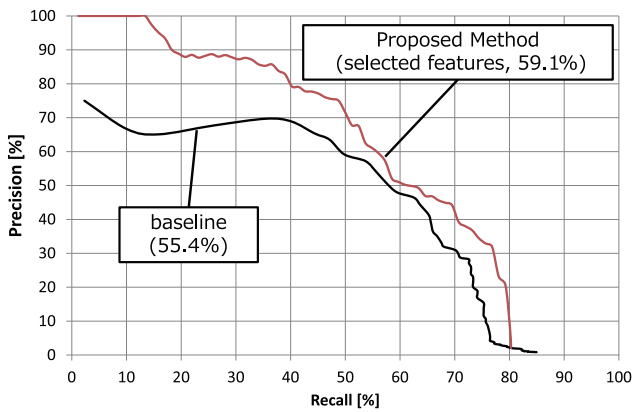

Fig. 9 Recall-precision curves and Max. F-measure values for the 26 query terms included in the 6,575 pre-indexing keywords (numbers in parentheses show Max. F-measure values)

terms that were among the 6,575 keywords used in the pre-indexing process. We then compared the performance of those 26 query terms among the 6,575 pre-indexing keywords and the remaining 74 query terms not in the 6,575 pre-indexing keywords to those of the baseline, as shown in Fig. 9 and Fig. 10, respectively. As can be seen, the proposed framework outperformed the baseline for the 26 query terms among the 6,575 pre-indexing keywords. However, it only slightly outperformed the baseline for the remaining 74 query terms. This result clearly indicates that the proposed framework is definitely applicable to query terms that are among the pre-indexing keywords. On the other hand, the remaining 74 query terms consisted of 53 OOV query terms (Fig. 8 shows a comparison to the baseline) and 21 INV query terms that were not correctly recognized by the ASR engine. A comparison of the evaluation results in Fig. 8 and Fig. 10 indicates that, for the 21 INV query terms not correctly recognized by the ASR engine, the proposed method probably underperforms compared to the baseline.

In summary, our results indicate that the two-stage

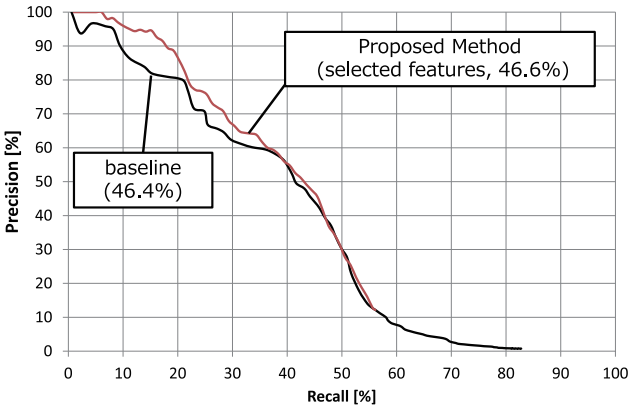

Fig. 10 Recall-precision curves and Max. F-measure values for the 74 query terms not included in the 6,575 pre-indexing keywords (numbers in parentheses show Max. F-measure values)

STD method with pre-indexing works well. The method involves pre-indexing of the target spoken documents from the keyword set collected from the ASR result using the STD engine and the SVM-based classifier trained by the features related to the pre-index.

5. Conclusion

This paper has described a two-stage STD framework that uses the same STD engine twice. First, an STD process with a keyword list from an ASR result generates a pre-indexed keyword list of spoken documents in a front-end process. In the query term search process, the STD engine searches for an input query from the target documents and outputs detection candidates. These detected candidates are then verified by an SVM-based classifier trained with features related to pre-index and query information.

The experimental results show that the proposed STD framework is very effective in drastically reducing the number of falsely detected candidates in the range less than 60% of recall in the verification process. The best SVM-based classifier, which was trained with the selected feature set including the pre-index-related features, improved the Max. F-measure value on the recall-precision curve by up to 49.3% compared to the baseline (44.6%). In particular, it worked better with short query terms (less than seven syllables) compared to long query terms.

In future, we plan to evaluate the proposed STD system with the test collection-free framework, wherein the classifier is trained with features from another dataset. We also plan to integrate SVM features from all of the keywords in a competitive interval.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP26282049, JP24500225, JP15K00254.

References

[1] C. Yonekura, Y. Furuya, S. Natori, H. Nishizak, and Y. Sekiguchi, "Evaluation of the Usefulness of Spoken Term Detection in an Electronic Note-Taking Support System," Proc. 5th APSIPA ASC, pp.1–4, 2013.
[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. 8th INTERSPEECH, 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. 9th INTERSPEECH, pp.2146–2149, 2008.
[4] S. Meng, P. Yu, F. Seide, and J. Liu, "A study of lattice-based spoken term detection for chinese spontaneous speech," Proc. ASRU, pp.635–640, 2007.
[5] M. Akbacak, D. Vergyri, and A. Stolcke, "Open-vocabulary spoken term detection using graphone-based hybrid recognition systems," Proc. ICASSP, pp.5240–5243, 2014.
[6] L. Mangu, B. Kingsbury, H. Soltau, H.-K. Kuo, and M. Picheny, "Efficient spoken term detection using confusion networks," Proc. ICASSP, pp.7844–7848, 2014.
[7] V. Soto, E. Cooper, L. Mangu, A. Rosenberg, and J. Hirschberg, "Rescoring confusion networks for keyword search," Proc. ICASSP, pp.7138–7142, 2014.
[8] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," Proc. 30th SIGIR, pp.615–622, 2007.
[9] I. Szoke, L. Burget, J. Cernocky, and M. Fapso, "Sub-word modeling of out of vocabulary words in spoken term detection," Proc. SLT, pp.273–276, 2008.
[10] R. Prabhavalkar, K. Livescu, E. Fosler-Lussier, and J. Keshet, "Discriminative articulatory models for spoken term detection in low-resource conversational settings," Proc. ICASSP, pp.8287–8291, 2013.
[11] T. Mertens, D. Schneider, and J. Köhler, "Merging search space for subword spoken term detection," Proc. 10th INTERSPEECH, pp.2127–2130, 2009.
[12] S. Natori, H. Nishizaki, and Y. Sekiguchi, "Japanese spoken term detection using syllable transition network derivedfrom multiple speech recognizers' outputs," Proc. 11th INTERSPEECH, pp.681–684, 2010.
[13] X. Wang, T. Li, Y. Xiao, J. Pan, and Y. Yan, "Improved mandarin spoken term detection by using deep neural network for keyword verification," Proc. 10th ICNC, pp.144–148, 2014.
[14] D. Wang, S. King, J. Frankel, and P. Bell, "Term-dependent confidence for out-of-vocabulary term detection," Proc. 10th INTERSPEECH, pp.2139–2142, 2009.
[15] J. Tejedor, A. Echeverria, and D. Wang, "An evolutionary confidence measurement for spoken term detection," Proc. 9th CBMI, pp.151–156, 2011.
[16] T.-W. Tu, H.-Y. Lee, and L.-S. Lee, "Improved spoken term detection using support vector machines with acoustic and context features from pseudo-relevance feedback," Proc. ASRU, pp.383–388, 2011.
[17] N. Sawada, S. Natori, and H. Nishizak, "Re-ranking of spoken term detections using crf-based triphone detection models," Proc. 6th APSIPA ASC, pp.1–4, 2014.
[18] S. Natori, Y. Furuya, H. Nishizaki, and Y. Sekiguchi, "Spoken Term Detection Using Phoneme Transition Network from Multiple Speech Recognizers' Outputs," Journal of Information Processing, vol.21, no.2, pp.176–185, 2013.
[19] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An Acoustic Segment Modeling Approach to Query-by-Examble Spoken Term Detection," Proc. 37th ICASSP, pp.5157–5160, 2012.
[20] S.R. Shiang, P.W. Chou, and L.C. Yu, "Spoken Term Detection and Spoken Content Retrieval: Evaluations on NTCIR-11 Spoken-Query&Doc Task," Proc. 11th NTCIR, pp.371–375, 2014.
[21] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, "Exploiting Diversity for Spoken Term Detection," Proc. 38th ICASSP, pp.8282–8286, 2013.
[22] Y.-N. Chen, C.-P. Chen, H.-Y. Lee, C.-A. Chan, and L.-S. Lee, "Improved Spoken Term Detection with Graph-based Re-Ranking in Feature Space," Proc. ICASSP, pp.5644–5647, 2011.
[23] D. Wang, S. King, J. Frankel, R. Vipperla, N. Evans, and R. Troncy, "Direct posterior confidence for out-of-vocabulary spoken term detection," ACM Transactions on Information Systems, vol.30, no.3, pp.16:1–16:34, 2012.
[24] K. Domoto, T. Utsuro, N. Sawada, and H. Nishizaki, "Two-step spoken term detection using SVM classifier trained with pre-indexed keywords based on ASR result," Proc. 16th INTERSPEECH, pp.834–838, 2015.
[25] T. Akiba, H. Nishizaki, K. Aikawa, T. Kawahara, and T. Matsui, "Overview of the IR for Spoken Documents Task in NTCIR-9 workshop," Proc. 9th NTCIR, pp.223–235, 2011.
[26] A. Lee and T. Kawahara, "Recent Development of Open-Source Speech Recognition Engine Julius," Proc. 1st APSIPA ASC, pp.1–6, 2009.
[27] K. Maekawa, "Corpus of Spontaneous Japanese: Its Design and Evaluation," Proc. SSPR, pp.1–8, 2003.
[28] 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. 10th NTCIR, pp.573–587, 2013.
[29] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," Journal of Machine Learning Research, vol.3, pp.1157–1182, 2003.
[30] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," ACM Transactions on Intelligent Systems and Technology, vol.2, no.3, pp.27:1–27:27, 2011.

Kentaro Domoto was born in 1991. He received his B.E. and M.E. degrees in engineering from University of Tsukuba in 2014 and 2016. After graduating from the master course in Department of Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba in 2016, he has been a member of NTT DATA CORPORATION. His research interests include spoken language processing, in particular speech term detection.

Takehito Utsuro was born in 1966. received his B.E., M.E., and PhD. Eng. degrees in electrical engineering from Kyoto University in 1989, 1991, and 1994. He has been a professor at the Division of Intelligent Interaction Technologies, Faculty of Engineering, Information and Systems, University of Tsukuba, since 2012. His professional interests in natural language processing, Web intelligence, information retrieval, machine learning, spoken language processing, and artificial intelligence. He is a member of the Association for Computational Linguistics (ACL) the Institute of Electronics, Information and Communication Engineers (IEICE), Information Processing Society of Japan (IPSJ), the Japan Society for Artificial Intelligence (JSAI), and the ASJ.

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 doctor course student at 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 ASJ and the 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 Department of Research, Interdisciplinary Graduate School of Engineering at University of Yamanashi. His research interests include spoken language processing, speech interface, and human-computer interaction. He is a member of IEEE, IEICE, IPSJ, JSAI, and the ASJ.