# Two-Step Spoken Term Detection using SVM Classifier Trained with Pre-Indexed Keywords based on ASR Result

*Kentaro Domoto*[1], *Takehito Utsuro*[1], *Naoki Sawada*[2], *Hiromitsu Nishizaki*[2]

[1]Graduate School of Systems and Information Engineering, University of Tsukuba, Japan
[2]Interdisplinary Graduate School of Medicine and Engineering, University of Yamanashi, Japan
utsuro@iis.tsukuba.ac.jp, hnishi@yamanashi.ac.jp

10.21437/Interspeech.2015-261

## Abstract

This paper presents a novel two-step 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 output of the second STD engine. In the first STD process, pre-indexing of the target spoken documents from a keyword list built from the results of automatic speech recognition of the speeches is performed. The first STD process result includes a set of keywords and their detection intervals (positions) in the spoken documents. For the keywords that have competitive intervals, we rank them on the basis of the matching cost of STD and select the best one with the longest duration among competitive detections. The selected keywords are registered in the pre-index. In the second STD process, a query is searched by the same STD engine, and then, the outputted candidates are verified by an SVM classifier. Our proposed two-step STD method was evaluated using the NTCIR-10 SpokenDoc-2 STD task and it drastically outperformed the traditional STD method based on dynamic time warping and the confusion network-based index.

**Index Terms**: spoken document indexing, spoken term detection, verification by support vector machine

## 1. Introduction

Spoken Term Detection (STD) is one of the core technologies in spoken language processing. This technology enables us to search for a specified word from recorded speeches, and its effectiveness has been demonstrated on an electronic note-taking system [1]. 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 the implementation of a large vocabulary continuous speech recognition (LVCSR) system. Many studies tackling this difficulty with STD have already been proposed [2, 3]. In particular, machine learning approaches for STD have been recently increasing. For example, Prabhavalkar et al. [4] proposed articulatory models that included discriminative training for STD under low-resource settings. They challenged an STD framework without any automatic speech recognition (ASR) system, and their models could directly detect a query term from acoustic feature vectors. On the other hand, deep learning, multiple linear regression, support vector machines, and multilayer perceptrons were also used to estimate the confidence level of the detected candidates in a decision process [5, 6, 7] or in a re-ranking process [8, 9].

Our approach, a two-step STD framework, uses an STD engine twice and an SVM classifier. First, a keyword list is built from the result of an ASR process of spoken documents, and then, each keyword in the list is searched for by an STD engine. A detected candidate for a keyword has a matching cost and an occurence position. Therefore, different keywords are detected on 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. In the second STD process, an inputted query term is searched for and detection candidates are outputted. The detection candidates are verified to determine whether they are confident using an SVM classifier.

Our approach is similar to previous studies that focused on a decision process. However, our proposed framework is different from the other previous studies in that features derived from a pre-index of spoken documents are used for training a classifier. Most of the previous studies used acoustic features [10, 11], ASR-related information [12], lattice-based information [13, 14], and similar approaches. Therefore, this paper demonstrates the effectiveness of our proposed decision process, wherein STD outputs are used by an SVM classifier trained with these kinds of features.

We evaluated the proposed framework against academic lecture speeches as the spoken documents for the STD task. The proposed framework effectively verified the detected candidates because the precision rate drastically improved in the lower half of the recall rate compared with the traditional baseline STD.

## 2. Basline STD Engine and ASR

We employed an STD engine [15] that uses subword-based confusion networks. We used a phoneme transition network (PTN)-formed index derived from 1-best hypotheses of multiple ASR systems, and an edit distance-based dynamic time warping (DTW) framework to detect a query term. Our study employed 10 types of ASR systems; the same decoder was used for all types of ASR systems. Two types of acoustic models and five types of language models were prepared. The multiple ASR systems can generate the PTN-formed index by combining subword (phoneme) sequences from the output of these ASR systems into a single CN. The details of the STD engine are explained in [15]. The STD engine includes some parameters for DTW. Our study used the STD engine with false-detection parameters of "Voting" and "AcwWith," which received the best STD performance on the evaluation sets [15].

Julius ver. 4.1.3 [16], an open-source decoder for ASR, was used in all systems. Acoustic models are triphone- and syllable-based hidden Markov models (HMMs), in which each state uses a Gaussian mixture model (GMM). The acoustic and language models are trained with spoken lectures from the Corpus of Spontaneous Japanese (CSJ) [17]. All language models are word- and character-based trigrams. The details of the acoustic and the language models are described in [15]. The training conditions of all acoustic and language models and the ASR vo-

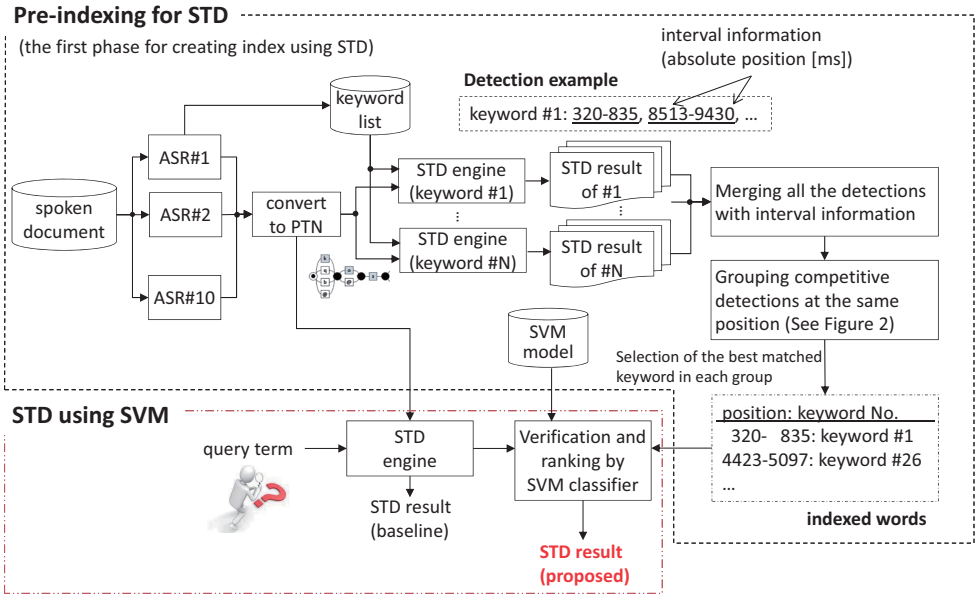


Figure 1: An overview of our proposedp two-step STD framework using SVM verification.

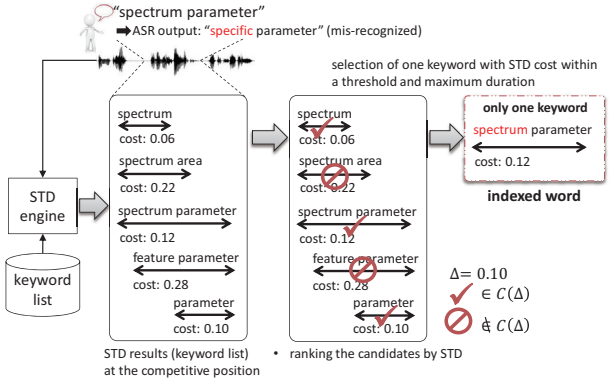


Figure 2: Selection of the best matched keyword from a competitive interval position.

cabulary are the same as in the STD/SDR test collections used in the NTCIR-9 [18] and the NTCIR-10 Workshop [19].

# 3. Two-Step STD

## 3.1. Overview

Figure 1 shows an outline of the proposed two-step STD framework which uses the same STD engine twice and an SVM classifier. Our STD is divided into two parts: (1) a pre-indexing phase using an STD engine and (2) a verification phase that detects candidates with the same STD engine using an SVM classifier.

First, the 10 types of ASR systems were applied to the spoken documents in order to make a PTN. We used the DTW-based STD engine with the PTN described in Section 2; however, our framework is independent of specific STD methods. Next, all the keywords in a keyword list made from the transcription of the spoken documents by the ASR system, wherein the word-based trigram and triphone-based GMM-HMM models are used, are searched for by the STD engine.

All the 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.

When a query is inputted to the same STD engine on the second step, it outputs detection candidates, which are verified using an SVM classifier trained with features related to the competitive keywords. Finally, only the confident candidates are outputted as the final STD result.

## 3.2. Pre-indexing performed by the first STD

Figure 2 shows the pre-indexing method used for target spoken documents. The method indexes the keywords that have a matching cost less than the threshold at the competitive position. We explain the details of the pre-indexing method below.

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:

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

The $C$ in the case shown in Figure 2 is represented as follows:

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

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

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

In the case shown in Figure 2, 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 $\Delta$. The quadruplets in $C(\Delta)$ have an STD cost less than $(cost_{min} + \Delta)$ as follows:

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

Table 1: List of features used for training an SVM classifier. The asterisk ($*$) next to each feature ID indicates that the feature is effective.

| Type | ID | Feature | Definition |
|---|---|---|---|
| Competitive set | F01 | The detected query has maximum duration or not | Set to 1 if $(t'_q - t_q) \geq \max\{t' - t \mid \langle w, t, t', cost\rangle \in C_q\}$ |
| | F02$*$ | The number of competitive keywords | $\lvert C_q \rvert$ |
| | F03$*$ | The minimum cost value at the competitive position | $\min\{cost \mid \langle w, t, t', cost\rangle \in C_q\}$ |
| Characteristics of the query | F04$*$ | The number of morae of the query | |
| | F05$*$ | The query is out-of-vocabulary (OOV) or in-vocabulary (INV) | Set to 1 if the query is INV; otherwise, set to 0 |
| | F06$*$ | The matching cost of the query | $cost_q$ |
| | F07 | The duration time of the query | $t'_q - t_q$ |
| Characteristics of the competitive keyword set | F08 | The number of morae of a query keyword | |
| | F09$*$ | Character type of a competitive keyword | Kanji, Hiragana, Katakana, mix of Kanji and Hiragana, and alphanumeric characters |
| | F10 | Part-of-speech (POS) of a competitive keyword | POS of 14 types |
| | F11 | The DTW-based matching cost of a competitive keyword | $cost$ |
| | F12$*$ | The duration time of a competitive keyword | $t' - t$ |
| | F13 | The overlapped duration between a competitive keyword and the query | $t_e - t_b$ (see Figure 3) |
| | F14$*$ | The overlapped rate between a competitive keyword and the query | $(t_e - t_b)/(t'_q - t_q)$ |
| | F15 | The difference between the matching costs of a competitive keyword and the query | $cost_q - cost$ |
| | F16 | The distance between a competitive keyword and the query | An edit distance at the phoneme level |

For example, in Figure 2, if $\Delta = 0.10$, then $C(\Delta = 0.10)$ is represented as follows:

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

Finally, from $C(\Delta)$ we select the quadruple $\langle w_{ld}, t, t', cost_{ld}\rangle$, that has the longest duration ($ld$) when the duration of a detected keyword is defined as $t' - t$. The keyword $w_{ld}$ is outputted as the STD result and used as a pre-indexing word for spoken documents. In the example shown in Figure 2, the following quadruple is the final output:

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

### 3.3. The second STD process

The pre-index can be represented as a sequence of $m$ quadruples $\langle w_{ld}, t, t', cost_{ld}\rangle$ as follows:

$$\langle w_{ld}, t, t', cost_{ld}\rangle_1, \ldots, \langle w_{ld}, t, t', cost_{ld}\rangle_m.$$

These intervals $([t, t']_1, \ldots, [t, t']_m)$ do not overlap each other.

When a query keyword $w_q$ is inputted to the same STD engine as the one used for making the pre-index, the STD engine outputs detected candidates. We define a detection candidate as a quadruple as follows and denote it as $q$:

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

The interval $[t_q, t'_q]$ of $q$ overlaps at one or more pre-indexed keywords as follows:

$$\langle w_{ld}, t, t', cost_{ld}\rangle_i, \ldots, \langle w_{ld}, t, t', cost_{ld}\rangle_j.$$

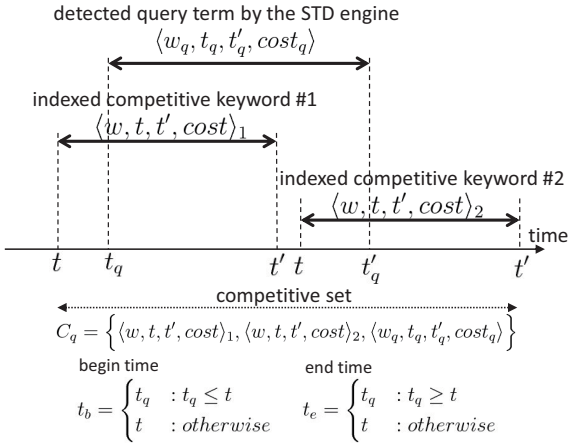


Figure 3: Calculation of the overlapped duration and rate between the detected query and the competitive keywords in a competitive set $C_q$.

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

$$C_q = \Big\{\langle w_{ld}, t, t', cost_{ld}\rangle_i, \ldots, \langle w_{ld}, t, t', cost_{ld}\rangle_j, \langle w_q, t_q, t'_q, cost_q\rangle\Big\}.$$

The final decision process of the STD verifies whether $q$ is confident or not by an SVM classifier. This process is performed for all the detected candidates $q$.

### 3.4. SVM classifier for verification

We prepared 16 types of features for training the SVM classifier that verifies $q$. Table 1 shows a feature list. All the features are related to a competitive set, competitive keywords registered in a pre-index, and a query. The eight types of features having IDs with an asterisk ($*$) are the effective features. They are selected by analyzing the evaluation results of all the combination of features examined in this paper. The evaluation results show that those eight features achieved higher performance than when all the features were used. We used the LIBSVM tool [20] as an SVM 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 on a five-fold cross validation framework using the target speech data. In this study, we tried two kinds of cross validations, query-based and presentation speech-based cross validations, which were used during the evaluation of an STD task. All the feature values were scaled from 0 to 1 using the "svm-scale" tool of the LIBSVM tool. The SVM classifier can output the result of verifying $q$ with a confidence score. The detected candidates, which are determined as "confident," are sorted in descending order of confidence scores. We can draw a recall-precision curve by changing the threshold for the confidence score.

## 4. Evaluation

### 4.1. Experimental setup

We used the moderate-size STD task used in NTCIR-10 SpokenDoc-2 [19] as the STD task for evaluation. The evaluation speech data was from the Corpus of Spoken Document Processing Workshop. It consisted of the recordings of the first to sixth annual Spoken Document Processing Workshop, comprising 104 real oral presentations (28.6 h). The number of query terms was 100, where 47 of the all query terms were INV queries that were included in the ASR vocabulary of the word-based trigram model and the other 53 queries were OOV. The occurrences of the INV and OOV queries in the set were 444 and 456, respectively.

We compared the results of four types of STD systems to show the effectiveness of the proposed two-stage STD system: **"baseline"** was used as the STD engine described in Section 2; **"QueryCV: all features"** was one of the proposed methods using all the features for SVM, the parameters of which were tuned by the query-based[1] five-fold cross validation; **"QueryCV: selected features"** used the SVM classifier with the eight effective features; and **"PresenCV: selected features"** used the SVM classifier tuned with the presentation-based[2] five-fold cross validation. The STD cost range $\Delta$ was determined as 0.20 in our study by optimizing it using held-out data. The evaluation metrics used in this study are recall, precision, and F-measure. These measurements are frequently used to evaluate information retrieval performance. F-measure values for the optimal balance of recall and precision values are denoted "maximum F-measure."

### 4.2. Experimental result

Figure 4 shows recall-precision curves for four of the STD systems. All the proposed STD systems achieved better STD performance compared with the baseline. The proposed methods drastically improved the precision rate in the lower half of the recall rate in particular. The SVM classifier tuned by the presentation-based cross validation demonstrated the best performance among all the systems because it implemented the closed condition for the query set. In the open condition for the query set, "QueryCV: selected features" prevailed against the one trained by all types of features.

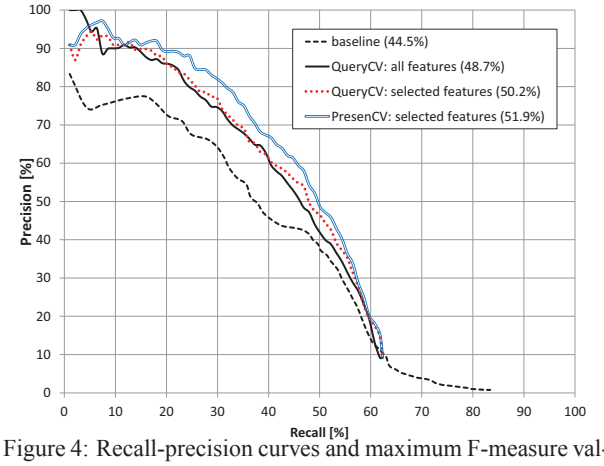

Figure 4: Recall-precision curves and maximum F-measure values for each STD system. The numbers in parentheses show maximum F-measure values.

The results show that the features that are not related to acoustic and ASR-related parameters but related to a competitive set, a query, and competitive keywords, effectively verified the detected candidates on the SVM classification framework. This SVM-based decision process does not necessarily require the use of the STD engine to generate the pre-index of the spoken documents. It is acceptable to use a simple ASR system for making the pre-index. An analysis of the effectiveness for each feature, however, showed that the DTW-based matching cost worked well on this SVM-based decision process, in particular. Therefore, it could be said that it effectively prepared the pre-index with richer information when using the STD engine. In addition, specification of the character type of keywords was useful because most query terms comprised Kanji, Katakana, and alphanumeric characters. This is not uncommon for a Japanese STD task. These character types usually compose trendy phrases and buzzwords in Japanese.

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

## 5. Conclusion

This paper described a two-step STD framework that used the same STD engine twice: the first STD process with a keyword list from an ASR result was performed to generate a pre-index of the spoken documents, and the second STD process searched for an inputted query from the target documents. Candidates detected by the second STD process were verified by an SVM classifier trained with features related to information of the pre-index and the query. The experimental results showed that the proposed STD framework was very effective in drastically reducing the number of falsely detected candidates in the lower half of the recall rate in the verification process. As future studies, we plan to evaluate the proposed STD system on the test collection-free framework, in which the classifier is trained with features from another dataset. We also plan to integrate the SVM features from all the keywords in a competitive interval.

[1] 100 queries were divided into five groups. Each group had 20 queries.

[2] 104 presentations were divided into five groups. Each group had about 21 presentations.

## 6. References


[1] C. Yonekura, Y. Furuya, S. Natori, H. Nishizaki, and Y. Sekiguchi, "Evaluation of the usefulness of spoken term detection in an electronic note-taking support system," in *Proceedings of the 5th Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2013)*, 2013, pp. 1–4.

[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," in *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH2007)*. ISCA, 2007, pp. 2393–2396.

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

[4] R. Prabhavalkar, K. Livescu, E. Fosler-Lussier, and J. Keshet, "Discriminative articulatory models for spoken term detection in low-resource conversational settings," in *Proc. of The IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2013)*, 2013, pp. 8287–8291.

[5] X. Wang, T. Li, Y. Xiao, J. Pan, and Y. Yan, "Improved Mandarin spoken term detection by using deep neural network for keyword verification," in *Prceedings of the 10th International Conference on Natural Computation (ICNC)*, 2014, pp. 144–148.

[6] D. Wang, S. King, J. Frankel, and P. Bell, "Term-dependent confidence for out-of-vocabulary term detection," in *Proceedings of the 10th Annual Conference of the International Speech Communication Association (INTERSPEECH2009)*. ISCA, 2009, pp. 2139–2142.

[7] J. Tejedor, A. Echeverria, and D. Wang, "An evolutionary confidence measurement for spoken term detection," in *Proc. of the 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2011, pp. 151–156.

[8] 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," in *Proc. of the IEEE Internationl Workshop on Automatic Speech Recognition and Undersitanding (ASRU2011)*, 2011, pp. 383–388.

[9] N. Sawada, S. Natori, and H. Nishizaki, "Re-ranking of spoken term detections using CRF-based triphone detection models," in *Proceedings of the 6th Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2014)*, 2014, pp. 1–4.

[10] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-examble spoken term detection," in *Proceedings of The IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2012)*, 2012, pp. 5157–5160.

[11] S.-R. Shiang, P.-W. Chou, and L.-C. Yu, "Spoken term detection and spoken content retrieval: Evaluations on NTCIR-11 SpokenQuery&Doc Task," in *Proceedings of the 11th NTCIR Conference*, 2014, pp. 371–375.

[12] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, "Exploiting diversity for spoken term detection," in *Proceedings of The IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2013)*, 2013, pp. 8282–8286.

[13] 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," in *Proceedings of The IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2011)*, 2011, pp. 5644–5647.

[14] D. Wang, S. King, J. rankel, 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, 2012, pp. 16:1–16:34.

[15] 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, 2013, pp. 176–185.

[16] A. Lee and T. Kawahara, "Recent development of open-source speech recognition engine Julius," in *Proceedings of the 1st Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC2009)*. APSIPA, 2009, pp. 1–6.

[17] K. Maekawa, "Corpus of spontaneous Japanese: Its design and evaluation," in *Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003, pp. 1–8.

[18] T. Akiba, H. Nishizaki, K. Aikawa, T. Kawahara, and T. Matsui, "Overview of the IR for spoken documents task in NTCIR-9 workshop," in *Proceedings of the 9th NTCIR Workshop Meeting*. NTCIR, 2011, pp. 223–235.

[19] 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," in *Proceedings of the 10th NTCIR Conference*. NTCIR, 2013, pp. 573–587.

[20] 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, 2011, pp. 27:1–27:27.