INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Rescoring by Combination of Posteriorgram Score and Subword-Matching Score for Use in Query-by-Example

Masato Obara[1], *Kazunori Kojima*[1], *Kazuyo Tanaka*[2], *Shi-wook Lee*[3] *and Yoshiaki Itoh*[1]

[1]Iwate Prefectural University
[2]University of Tsukuba
[3]National Institute of Advanced Industrial Science and Technology

y-itoh@iwate-pu.ac.jp

Abstract

There has been much discussion recently regarding spoken term detection (STD) in speech processing communities. Query-by-Example (QbE) has also been an important topic in spoken-term detection (STD), where a query is issued using a speech signal. This paper proposes a rescoring method using a posteriorgram, which is a sequence of posterior probabilities obtained by a deep neural network (DNN) to be matched against both a speech signal of a query and spoken documents. Because direct matching between two posteriorgrams requires significant computation time, we first apply a conventional STD method that performs matching at a subword or state level, where the subword denotes an acoustic model, and the state composes a hidden Markov model of the acoustic model. Both the spoken query and the spoken documents are converted to subword sequences, using an automatic speech recognizer. After obtaining scores of candidates by subword/state matching, matching at the frame level using the posteriorgram is performed with continuous dynamic programming (CDP) verification for the top *N* candidates acquired by the subword/state matching. The score of the subword/state matching and the score of the posteriorgram matching are integrated and rescored, using a weighting coefficient. To reduce computation time, the proposed method is restricted to only top candidates for rescoring. Experiments for evaluation have been carried out using open test collections (Spoken-Doc tasks of NTCIR-10 workshops), and the results have demonstrated the effectiveness of the proposed method.

Index Terms: spoken-term detection, deep neural network, posteriorgram, rescoring,

1. Introduction

It has been possible to accommodate a significant quantity of audio and video data using a smart phone via the Internet. Research on spoken-term detection (STD) has been actively carried out in searching for a query consisting of one or more words among a significant number of STD audio signals, referred to as spoken documents. Recently, queries have been entered by voice, in Google voice search and the like; this technology is referred to as "Query by Example (QbE)" in STD research. In STD with a text query, it is trivial to distinguish between whether a query uses in-vocabulary (IV) words or OOV words by merely referring to a lexicon of the speech recognizer. In the case of QbE, it is difficult to distinguish whether a query uses IV words or OOV words. Therefore, it is necessary to assume that a given query uses OOV words, and, in the general approach, a spoken query is converted to a subword sequence by a subword-based speech recognizer. The search process uses the same framework as that of STD for a text query. Then, a search is performed for the subword sequence among subword sequences of spoken documents that have been converted in advance from speech signals. Nonlinear matching at subword/state-level is performed using continuous dynamic programming (CDP) [1], which is a general method of matching between two signals or symbol sequences. The acoustic distance between subwords or states is used as a local distance in CDP. When matching at the subword or state level, acoustic information at the frame level is lost. The output probability of a deep neural network (DNN) is obtained at the frame level, and a posteriorgram, which is a sequence of posterior probabilities obtained by DNN to be matched against both a speech signal of a query and spoken documents, achieves a high accuracy in STD [2,3]. Because direct matching between two posteriorgrams requires significant computation time, we have adopted a two-step method. In the first step, we apply a fast conventional STD method that performs matching at the subword or state level. Here, the subword denotes an acoustic model, and the state composes a hidden Markov model (HMM) of the acoustic model. Spoken documents are converted to subword sequences by using an automatic speech recognizer beforehand. Given a spoken query, the query is also converted to a subword sequence in the same manner as the spoken documents. CDP is performed between a query subword sequence and the spoken documents at the subword/state level, and scores of candidates are obtained for each spoken document. In the second step, matching at frame level using the posteriorgram is performed by CDP for only the top *N* candidates obtained by the first step. A cosine distance is used as a local distance in CDP, requiring a significant degree of computation because the dimensionality of a posteriorgram can be large (e.g., 3,000). To reduce computation time, the proposed method is restricted to the top *N* candidates for rescoring. The score of the subword/state matching and the score of the posteriorgram matching are integrated linearly, and rescored using a weighting coefficient. Evaluation experiments are conducted by using open test collections of Spoken-Doc tasks of NTCIR-10 workshops; the results have demonstrated the effectiveness of the proposed method.

Section 2 describes the details of the proposed method. The evaluation experiments using open test collections of NTCIR-10 workshops [4-6] as evaluation data are described in Section 3. The conclusion is presented in Section 4.

2. Proposed Method

The proposed method uses a two-step approach to improve retrieval accuracy in realistic processing time. The next section describes the first step, which uses a fast conventional STD method to perform matching at the subword or state level. Section 2.2 describes the second step, which performs matching at the frame level, using the posteriorgram generated by using a DNN.

http://dx.doi.org/10.21437/Interspeech.2016-309

2.1. Matching at Subword and State Level

A fast conventional STD method using subwords is described in this section. A flowchart of the conventional method is shown in Figure 1. Subword sequences for spoken documents are prepared by using a subword recognizer (1). For a subword, a monophone, syllable, demiphone or similar have been used in our previous experiments. In this paper, a triphone is used for an acoustic model consisting of a left-to-right HMM with three states. The spoken documents are converted to syllable sequences by using an automatic syllable speech recognizer. All syllable sequences are transformed into subword (triphone) sequences according to conversion rules. These procedures are performed in advance.

When a text query is submitted (2), it is automatically transformed into a subword sequence according to a set of conversion rules (3). (A phone sequence is entered initially and transformed to a word in Japanese; its phone sequence can be used for determining query pronunciation). When a spoken query is submitted, it is also converted to a triphone sequence via a syllable sequence, in the same manner as the spoken documents (4).

Subsequently, a search is performed for the query subword sequence on the spoken documents that have been transcribed into subword sequences in advance. CDP, which is a word-spotting algorithm, is employed as a search method (5). Each candidate has a score, and is ranked according to the score (6). Ranked candidates composed of an utterance are provided to a user (7).

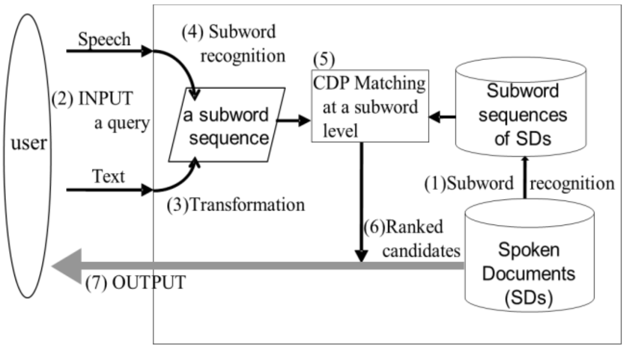


Figure 1: Conventional STD Method Performing Matching at Subword or State Level.

2.1.1. *Subword-level Matching*

Both spoken queries and spoken documents are converted to triphone sequences. We have prepared triphone acoustic distances obtained from the statistics of triphone HMMs [7,8]. The triphone acoustic distances are used for local distances in CDP.

2.1.2. *State-level Matching*

In matching at the state level, each triphone is converted to three states composing its triphone HMM. As shown in Figure 1, for both queries and spoken documents, each triphone, is divided into three states (area within dotted line). S1, S2, and so on denote the states in a triphone HMM. Because both the length of a query and the spoken documents are multiplied by three, the CDP lattice becomes more detailed by a factor of nine, and the retrieval time is also increased by a factor of nine.

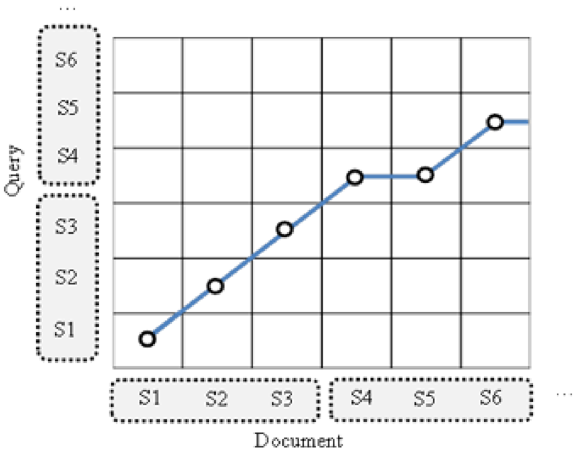
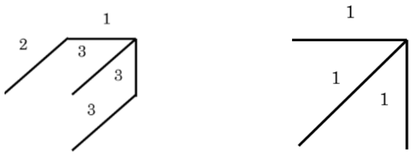


Figure 2: Matching at State Level

Acoustic distances between any two states are determined from the statistics of two HMM states in the same manner as acoustic distances between subwords.

2.1.3. *DP Local Path*

In the subword-based STD system described above, a DP local path and slope weight with restrictions are often used, as shown in the left panel of Figure 3. The matching length is restricted from half to twice the query length, and weights with value 3 are added when going up by 1 in the DP lattice. As a result of the restrictions, all the paths have equal weight at the last node of the query. Conversely, the path in the right panel denotes a path without restrictions. This paper investigates more efficient local paths in CDP at the subword and state levels, and at the frame level, as described in the next section.



(1) Path with Restrictions (2) Path without Restrictions

Figure 3: Lack of Restrictions and Weight Paths

2.2. Proposed Rescoring

2.2.1. *Matching at Frame Level Using a Posteriorgram*

Although matching at the state level is more detailed by a factor of nine than matching at the subword level in a DP lattice, acoustic information at the frame level is lost. Output probabilities are generated by the DNN at each frame, and the probabilities are regarded as speaker-independent features. A posteriorgram, which is a sequence of output probabilities, is considered to be a promising feature [9], and is used for a feature vector at the frame level in the second step.

In the second step, CDP is performed for only the top *N* candidates that have been obtained in the first step. In CDP, the local distance is obtained by using a posteriorgram at the frame level. A product is obtained for two feature vectors in the posteriorgram and transformed to a local distance in CDP by computing the negative logarithmic value, as shown in Equation 1. The product showed the better results, compared with a Cosine similarity and an Euclidean distance to the other.

$$D(P_i,Q_j) = -\log(\sum_{k=0}^{\dim} P_i(k)Q_j(k)) \tag{1}$$

Here, P_i and Q_j are the *i*th and *j*th feature vectors in a posteriorgram of spoken documents and a query, respectively. $P_i(k)$ and $Q_j(k)$ denote output probabilities of the *k*th state of the feature vectors P_i and Q_j, respectively. The product of the two vectors corresponds to the similarity between the two vectors.

The computation of the product in Equation 1 and the generation of output probabilities by a DNN become expensive because the dimensionality of the feature vector of the posteriorgram rises 3,000, corresponding to the number of states in the HMMs when graphics processing units are used for computation. To reduce the computation time, the proposed method is restricted to only the top *N* candidates for rescoring, although a significant number of candidates are generated in the first step. Other candidates ranked worse than *N* are used in this step.

2.2.2. Rescoring by Integrating Two Scores

The score of the subword/state matching and the score of the posteriorgram matching are integrated linearly, and rescored according to Equation 2.

$$Dnew = (1-\alpha)D_s + \alpha D_d \tag{2}$$

The parameter α represents a weighting coefficient. D_S and D_d denote the distance at the state/subword level and the distance at the frame-level matchings, respectively. The weighting coefficient α is sought experimentally.

3. Evaluation Experiments

3.1. Open Test Collection

An open test collection of NTCIR-10 workshops, as shown in Table 1, is used for the evaluation experiments. Presentation speeches of a Spoken Document Processing Workshop (SDPWS) containing 104 lectures (approximately 28.6 h of speech) are used for spoken documents. A query set includes 100 query terms. We regard all query terms as OOV queries; results of syllable recognition, therefore, are used for both spoken queries and spoken documents. An NTCIR-10 workshop provides queries in text. For the QbE experiments, we record queries spoken by 10 people (5 men and 5 women).

Table 1: Open Test Collection

Spoken Documents	SDPWS: 104 presentations, 28.6 hours, 40746 utterances
Number of Queries	Formal run: 100 queries

We use the mean average precision (MAP) as a measure of STD accuracy. We use mean average precision (MAP) as an evaluation measure of performance, also employed in NTCIR-9 and 10. Average precision (AP) for a query is obtained by averaging the precisions at every correct occurrence, and the MAP is then the averaged AP of all queries. The processing time was then obtained using a personal computer (CPU: Intel Core i7 3770k processor; RAM: 16 GB; Operating system: Linux).

3.2. Experimental Conditions

A Corpus of Spontaneous Japanese (CSJ) [10,11] that includes 2,702 presentation speeches is used for the evaluation. 177 speeches are used for test data among 2,702 speeches, and the other 2,525 speeches are used for training acoustic models and language models of a speech recognizer. We use a triphone acoustic model composed of a left-to-right HMM with three states, and we use tied-state triphone models with 3009 states and 32 Gaussian in a state. For a feature vector frame, 120-dimensional filter banks are used. The input feature vectors were extracted under the conditions shown in Table 1. Five-frame feature vectors were added before and after the current frame as feature vectors for the DNN. The DNN was trained using a 1,320-dimensional feature vector under the conditions shown in Table 2.

Table 3 shows learning conditions for the DNN that are generally used in a DNN-HMM speech recognizer [12]. The DNN is composed of seven layers, including an input layer, five hidden layers, and an output layer. The input layer receives a 1,320-dimensional feature vector, and the output generates output probabilities of 3,009 states of HMMs.

Table 2: Conditions for Feature Extraction

Feature Parameters	1320-dimension (dim) FBANK (40 dim) + Delta-FBANK (40 dim) + Delta-Delta-FBANK (40 dim)
Analysis Window	Hamming window
Window Length	25 ms
Frame Shift	10 ms

Table 3: *Conditions for DNN*

Number of Nodes		Input layer: 1320 Output layer: 3009
Number of Hidden Layers	Layers	5 layers
	Nodes	2048 nodes
RBM	Learning Rate	0.004
	Momentum	0.9
	Mini-batch Size	256
	Epochs	10
DNN	Learning Size	0.007 (when recognition rate is lower than in previous epoch, it is reduced by half)
	Mini-batch Size	256
	Epochs	30

3.3. Results

In the proposed method, there are three elements affecting the retrieval accuracy of STD. The first one is the restriction of a local path in CDP, as described in 2.1.3. The second one is a parameter α, which is a weighting coefficient in Equation 1. The third one is a parameter *K*, which is the number of candidates for which the proposed rescoring is applied, as described in 2.2.

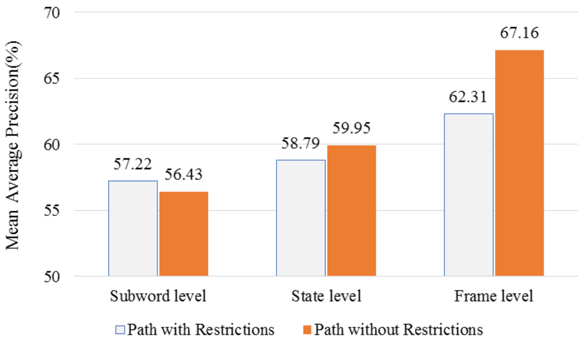


Figure 4: Retrieval Accuracy (MAP %) By Restrictions for Local Path in CDP

We investigated the effect of these three elements through the following experiments.

First, we investigated the effect of two restrictions for a local path in CDP, as shown in Figure 3. The results are shown in Figure 4. In matching at the frame level, *K* and α were set to 100 and 1.0, respectively. The same tendency was obtained with other settings for *K*.

The path with restriction was only effective for matching at the subword level. Contrary to our expectations, the path without restrictions provided better retrieval accuracy in matching at the state level and frame level. A flexible path in a DP lattice is considered to be necessary for detailed matching. The path without restrictions was used in the following experiments.

Second, we investigated the number of candidates *K* for which the proposed rescoring method was applied. Cases of *K* as 10, 100, 1,000, 10,000 and *K=ALL* were evaluated. *K=ALL* corresponds to the result when applying the proposed rescoring method for all spoken documents. The results are shown in Table 4. The parameter α was set to 0.5 in the experiment. The results of subword-level matching have been omitted here.

Baseline retrieval accuracy (MAP) was 59.95 % when using a conventional method at the state level. When *K* was increased, retrieval accuracy improved for both methods. When applying only state-level matching using a posteriorgram (posterior matching) for all spoken documents, MAP amounted to approximately 75%, improved by 15 points. Retrieval time, however, required 2,629 s (over 40 minutes) for a query. When *K* was equal to 10, MAP improved by over 2 points in a realistic retrieval time. Retrieval accuracy for the proposed rescoring method integrating the score of state matching and the score of the posteriorgram matching was 75.80%, which was approximately 1.5 points higher than the posteriorgram matching when *K =ALL*. When *K=10*, it was also 0.34 points higher. These results demonstrated the effectiveness of the proposed rescoring method. Reducing retrieval time is still the most crucial problem, as indicated by these results.

Figure 5 shows retrieval accuracy (MAP) when changing values of *K* and α. The case when α is equal to 0.0 indicates the result using a conventional method at the state level, the baseline, with 59.95% in MAP. The case when α is equal to 1.0 indicates the result using posteriorgram matching.

Table 4: Results of NTCIR-10 Formal Run When α=0.5

	Using only a Posteriorgram		Proposed Rescoring (Integration)		Retrieval Time (in sec.)
	MAP	Increase (in points)	MAP	Increase (in points)	
State-level CDP	59.95	0.00	59.95	0.00	0.2
K= 10	62.06	+2.11	62.40	+2.45	0.9
K= 100	67.16	+7.21	68.37	+8.42	8.0
K= 1,000	71.47	+11.52	72.98	+13.03	79.8
K=10,000	74.26	+14.31	75.80	+15.85	770.9
K= All	75.02	+15.07	76.49	+16.54	2,629.1

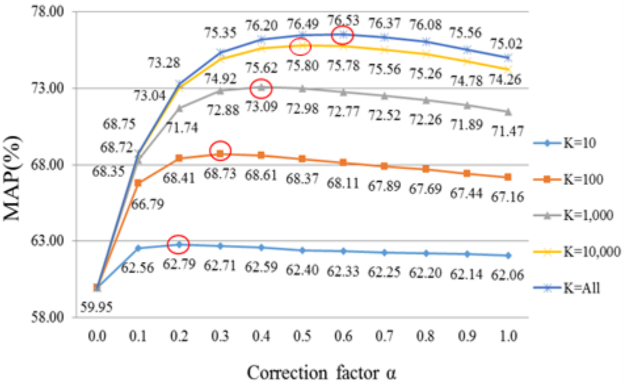


Figure 5: Results of Changing Correction Coefficient Values

When *K* = 10, MAP was highest at α = 0.2, and was 2.72 and 0.71 points higher than for the state-level matching and for the posteriorgram matching, respectively. When *K* = All, MAP was highest at α = 0.6, and was 16.58 points and 1.51 points higher than for the state-level matching and for the posteriorgram matching, respectively. When *K* was small, better results were obtained for smaller values of α, such as 0.2 or 0.3. Results of matchings at the state level were much more significant for such small α.

The reason is that the top candidates obtained by the state-level matchings were reliable, evidenced by higher precision rates. When *K* was large, better results were obtained at larger values of α, such as 0.6, and the results of the posteriorgram matching were much more significant.

4. Conclusion

This paper has proposed a two-step method for Query-by-Example. The first step used a fast conventional STD that performed matching at the subword or state level. The second step conducted a posteriorgram matching generated by a DNN at the frame level. Both the scores of the matching for subword/state matching and posteriorgram matching were integrated and rescored. Experimental results using open test collections of an NTCIR-10 workshop showed that retrieval accuracy improved a number of points in comparison with the conventional method and the posteriorgram matching, and demonstrated the effectiveness of the proposed method. MAP improved 16.58 points when *K* = All, and the retrieval time required 2629 seconds per query. Reduction of retrieval time for large *K* is a topic for the future.

5. Acknowledgements

This work was supported by JSPS (C), KAKENHI, Grant Number 15K00241.

6. References

[1] K. Iwata, Y. Itoh, K. Kojima, M. Ishigame, K. Tanaka, and S. Lee, Open-Vocabulary Spoken Document Retrieval based on new subword models and subword phonetic similarity, INTERSPEECH, 2006.

[2] P. Motlicek, F. Valente, and PN. Garner, “English Spoken Term Detection in Multilingual Recordings", INTERSPEECH 2010, pp.206-209, 2010.

[3] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, Exploiting diversity for spoken term detection, in Proc. ICASSP, 2013.

[4] Tomoyosi Akiba *et al*, Overview of the NTCIR-10 SpokenDoc-2 Task, NTCIR-10 Workshop Meeting, pp.573-587, 2013.

[5] K. Kon'no, H. Saito, S. Narumi, K. Sugawara, K. Kamata, M. Kon'no, J. Takahashi and Y. Itoh, An STD System for OOV Query Terms Integrating Multiple STD Results of Various Subword units, Proceedings of the 10th NTCIR Conference, 2013.

[6] Tomoyosi Akiba et al., Overview of the IR for Spoken Documents Task in NTCIR-9 Workshop, NTCIR-9 Workshop Meeting, pp. 223-235, 2011.

[7] Jinki Takahashi *et al*, Improving Retrieval Accuracy by Applying Various Methods to Spoken Term Detection, *Journal of Information Processing Society of Japan*,2013.

[8] Roy Wallace, et al, “A Phonetic Search Approach to the 2006 NIST Spoken Term Detection Evaluation”, INTERSPEECH 2007, pp2385-2388, 2007.

[9] Yaodong Zhang *et a*l, Unsupervised Spoken Keyword Spotting via Segmental DTW on Gaussian Posteriorgrams, ASRU 2009, pp. 398-403, 2009.

[10] K. Maekawa, Corpus of Spontaneous Japanese: Its design and evaluation Proceeding of the ISCA & IEEE Workshop on Spontaneous Speech Processing Association Annual Summit and Conference (APSIPA ASC), 2009.

[11] National Institute for Japanese Language and Linguistics, Corpus of Spontaneous Japanese, http://julius.sourceforge.jp/

[12] R. Konno, K. Kojima, K. Tanaka, S. Lee, and Y. Itoh, A rescoring method for STD using output probability of DNN, Proceedings of APSIPA Annual Summit and Conference, 2015.

[13] Timothy J. Hazen, Wade Shen, and Christopher White, Query-By-Example Spoken Term Detection Using Phonetic Posteriorgram Templates, ASRU 2009, pp. 421-426, 2009