Rescoring by a Deep Neural Network for Spoken Term Detection

Ryota Konno†, Kazunori Kojima†, Kazuyo Tanaka††, Shi-wook Lee††† and Yoshiaki Itoh†
†Iwate Prefectural University, Japan
E-mail: y-itoh@iwate-pu.ac.jp Tel: +81-01-96942556
††University of Tsukuba, Japan
†††National Institute of Advanced Industrial Science and Technology, Japan

Abstract— **In spoken-term detection (STD), the detection of out-of-vocabulary (OOV) query terms is crucial because query terms are likely to be OOV terms. This paper proposes a rescoring method that uses the posterior probabilities output by a deep neural network (DNN) to improve detection accuracy for OOV query terms. Conventional STD methods for OOV query terms search a query subword sequence for subword sequences of speech data by using an automatic speech recognizer. A detailed matching in the proposed method is performed by using the probabilities output by the DNN. A pseudo query at the frame or state level is generated so as to align the obtained probability at the frame level. To reduce the computational burden on the DNN, we apply the proposed method to only top candidate utterances, which can be quickly found by a conventional STD method. Experiments were conducted to evaluate the performance of the proposed method, using the open test collections for the SpokenDoc tasks of the NTCIR-9 and NTCIR-10 workshops as benchmarks. The proposed method improved the mean average precision between 5 and 20 points, surpassing the best accuracy obtained at the workshops. These results demonstrated the effectiveness of the proposed method.**

I. INTRODUCTION

Research on spoken-document retrieval (SDR) and spoken-term detection (STD) is actively conducted in an effort to enable efficient searching of the vast quantities of audiovisual data [1–3] that have been accumulated following the recent and rapid increase in the capacity of recording media such as hard disks. STD is the task of locating matches in spoken documents to a query consisting of one or more words. Query terms are often out-of-vocabulary (OOV) words, such as technical terms, geographical names, personal names, and neologisms. Therefore, OOV query terms must be retrievable through STD methods. To enable that, subword recognition using monophone, triphone and so on is performed in advance for all spoken documents. When query terms are given to the system, the system converts the query terms to sequences of subwords and then searches for the query's subword sequences among the documents' subword sequences. Matching between a query subword sequence and subword sequences of spoken documents is conducted by a continuous dynamic programming (CDP) algorithm that performs the DTW (dynamic time warping) algorithm continuously[4]. Because matching at a state level improved the STD accuracy[5], more detailed matching such as at a frame level is expected to lead to the better STD accuracy. Therefore the proposed method performs detailed matching at a state level and at a frame level using more sophisticated local distances generated by DNN. The acoustic distances are used as the local distance between any two subwords during CDP to enable approximate matching. The acoustic distances between symbolized subwords are independent of differences between the two audio signals. The difference between posterior probabilities of the two audio signals are obtained by using a deep neural network (DNN) [6,7]. We introduce the probabilities output by a DNN to calculate local distances in CDP. STD accuracy from using the DNN to compare query terms with different audio signals is expected to be higher than that obtained by using conventional acoustic distances. The accuracy of STD depends on the accuracy of speech recognition. The better recognition results obtained by DNN are supposed to lead to the better accuracy in STD. Furthermore, the more precise information at a state level or at a frame level is also expected to lead the better accuracy.

Although, the contents of spoken documents have been recognized using DNN in related works [8,9], but the probabilities from a DNN are not used directly for local distances of CDP between query terms and spoken documents in STD with the aim of improving accuracy.

Section II describes the proposed rescoring method in detail. The open test collections for the SpokenDoc in the NTCIR-9 and NTCIR-10 workshops are used as benchmarks to evaluate the performance of the proposed method in Section III, and a conclusion is presented in Section IV.

II. CONVENTIONAL METHOD AND PROPOSED METHOD

The proposed method performs CDP by using the posterior probability output by a DNN to improve STD accuracy. The DNN uses a graphics processing unit (GPU) to accelerate computation. Even with this improvement, it is not practical to use the DNN for searching all spoken documents after query terms are given. Therefore, in the proposed method, only the top candidate utterances, which are obtained by using our conventional STD method, are rescored. This improves

the STD accuracy while maintaining a reasonable processing time.

A. Conventional STD method using subword acoustic distances

The diagram of our conventional STD method is illustrated in Figure 1. The method searches spoken documents, processed by an automatic speech recognizer, for query subwords. CDP performs matching between a query subword sequence and subword sequences of spoken documents. Subword acoustic distances are used by CDP for local distances in our method. They are obtained as statistic of hidden Markov models (HMMs) of subwords. We have previously demonstrated the effectiveness of using acoustic distances [4,5].

B. Rescoring according to probabilities from a DNN

In this section, we propose using probabilities output by a DNN as local distances in CDP matching. It is computationally impractical to use the DNN to compute probabilities for all spoken documents. To overcome this, we propose two methods for reducing the computational load from the DNN.

a. DNNs

A combination of an HMM and a DNN (hereinafter, DNN-HMM) is used for a speech recognizer. The input data are a frame level of feature vectors of spoken data. The output data are the posterior probabilities of each state from the HMM. Feature vectors include several frames before and after the current frame, and may be high-dimensional in many cases. Each output node in the output layer is associated in advance with a state of the HMM. When a feature vector of one frame is given to the DNN, each output node generates a probability, which is taken as the posterior probability of the associated HMM state.

b. CDP using the probability from the DNN

In conventional methods, both the query and the spoken documents consist of subword sequences, and CDP is performed at the subword level. In contrast, the probabilities output by the DNN are obtained at the frame level because the input data to DNNs are frame-based feature vectors. To perform CDP between the query subword sequence and these frame-based probabilities, we accord the output sequences of

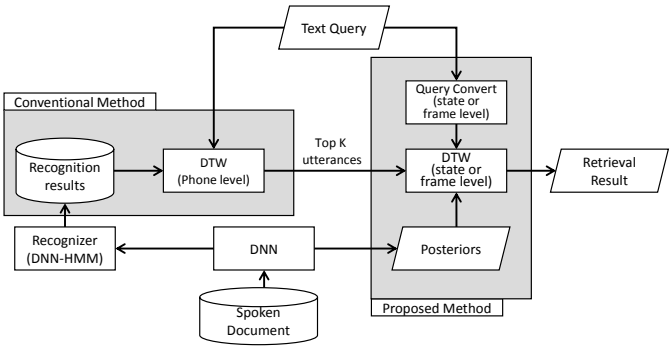

Fig. 1 System diagram

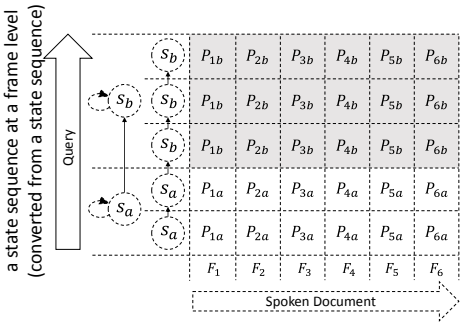

Fig. 2 Frame-level matching

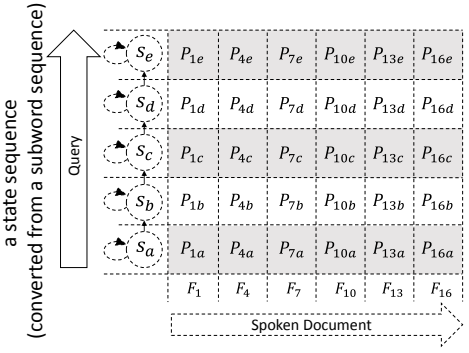

Fig. 3 State-level matching

Fig. 4 Local DP path

the DNN and the subword sequence of query terms by either of the following two methods.

I. Convert a query subword sequence to a frame-level sequence (frame-level matching)
II. Convert the frame-level probability from the DNN to state–level probabilities (state-level matching)

In method I, a query subword sequence is converted to a frame-level sequence, and CDP is performed at the frame level. A query subword sequence is initially converted into a state sequence. First, the average number of frames remaining at each state are computed from training data. Then, for each state in a query state sequence, the same states are stacked to the amount of the average number on the query axis. An example is shown in Fig. 2 Here, the average remaining numbers of state S_a and S_b are 2 and 3, respectively, shown in the vertical axis. F_l denotes the lth frame. P_{ij} denotes the posterior probability of state S_j from frame i. The posterior probability is the same in the same state and the same frame, as shown in the Fig. 2.

In method II, the frame-level probability from the DNN is transformed to state–level probabilities. The number of frames for each state, computed from training data, was about 2.7 frames on average. We considered the probability from the DNN at every 3 frames, which roughly corresponds to the state level. The probability from the DNN at every 3 frames is

provided to the HMM; this method is called state-level matching in this paper. An example is shown in Fig. 3. A state sequence of query terms is listed on the vertical axis. Each input feature vector is provided every three frames, as shown on the horizontal axis. By this method, the processing time for producing probabilities in the DNN is theoretically reduced to one-third of calculating at each frame.

Fig.4 is shows the Local DP path. It used to DTW in method I and method II.

c. Reducing computation by the DNN

In the proposed method, we reduce the processing time by the following two methods.

A) The proposed rescoring by DNN is applied to only top candidate utterances, as rated by our conventional STD method. Here, processing time largely depends on the number of candidate utterances, *K*.

B) Each candidate utterance takes the start and end frame number of the corresponding section with query terms when the corresponding section showed the highest score in the utterance. CDP using DNN is applied for only this section. To prevent the loss of audio information, 10 frames of feature vectors are added before and after the section.

III. Evaluation Experiments

A. Experimental conditions

For training acoustic models and language models, we used 1255 presentation speeches (about 287 hours of audio; about 14 minutes per speech.) included in the Corpus of Spontaneous Japanese (CSJ) [10,11]. We excluded 177 presentation speeches from the CSJ, and used these as testing data. A triphone acoustic model was chosen, composed of a left-to-right HMM with three states, and we used tied-state triphone extraction models with 3009 states and 32 mixtures. The DNN was trained under the conditions shown in Table I. The alignments between speech signals and each state were obtained from the results of forced alignment with a Gaussian mixture model. Forward-syllable bigrams and backward-syllable trigrams [12] were used for language models. We used the open source large vocabulary continuous speech recognition engine Julius [12] for syllable recognition.

The machine used for measuring processing time had an Intel Core i7-4770 CPU, NVIDIA GeForce GTX TITAN GPU, and 16 GB of random access memory.

B. Open Test Collections

To evaluate performance, we used the open test collections that were used in NTCIR-9 Workshop [13] and NTCIR-10 Workshop [14]. As shown in Table II, the test collection of NTCIR-9 contains 44 hours of CSJ presentation speeches (excluding training data) and two query sets. Each query set includes 50 query terms. The test collection of NTCIR-10 contains 29 hours of spoken documents, distinct from CSJ, and two query sets. Formal run is the test set for formal evaluation in the NTCIR Workshop, and dry run is the practice test set before formal run. Four kinds of test collections were used. We regarded all query terms as OOV queries. We used the mean average precision (MAP) [13] as the measure of STD accuracy.

TABLE I
CONDITIONS FOR DNN

Feature parameter		38 dimensions (dim) MFCC (12 dim) + Delta-MFCC (12 dim) + Delta-Delta-MFCC (12 dim)
Number of nodes		Input layer: 418 Hidden layer: 2048 Output layer: 3009
Number of hidden layers		3 layers
RBM	Learning rate	0.004
	Momentum	0.9
	Mini-batch size	256
	Epochs	10
DNN	Learning rate	0.007 (when recognition rate is lower than in the previous epoch, it is reduced by half)
	Mini-batch size	256
	Epochs	30

TABLE II
OPEN TEST COLLECTIONS

	NTCIR-9	NTCIR-10
Spoken documents	CSJ, 177 presentations, 44 hours, 53,892 utterances	SDPWS, 104 presentations, 29 hours, 40,746 utterances
Number of queries	Formal run: 50 Dry run: 50	Formal run: 100 Dry run: 32

C. Results

We conducted experiments with different values of *K* (the number of candidate utterances for the proposed DNN rescoring, as described in Section II.*B.c*). The cases of *K* = 50, 100, 500, 1,000, 2,000, and ALL were evaluated, where ALL denotes the case where the proposed DNN rescoring is used for all utterances in the spoken documents. Tables III to VI show the experimental results for the four test collections. Baseline in the tables is the result of the conventional sub-word level STD method (Syllable recognition is performed using a DHH-HMM recognizer for spoken documents). Shaded cells in the tables indicate the best MAP for each test collection. In all four test collections, the STD accuracy improved according to the increase of *K*. The maximum MAP improvements were 11.52 points in NTCIR-9 Formal run, 14.05 points in NTCIR-9 Dry run, 24.00 points in NTCIR-10 Formal run, and 20.41 points in NTCIR-10 Dry run, obtained with *K* = ALL in all test collections.

The processing time increased linearly in proportion to K for both frame-level and state-level matching. Although the processing time for state-level matching is theoretically one-third that of frame-level matching, the ratio of processing times was 0.57 because of computations expect for the probability in the DNN.

When the number of utterances to be rescored is small, such as with *K* = 100, MAP improved between 7.33 and 15.09

TABLE III
RESULTS OF NTCIR-9 FORMAL RUN

	Frame-level matching			State-level matching		
	MAP	Change (points)	Time (s)	MAP	Change (points)	Time (s)
Baseline	83.63	+0.00	0.04	83.63	+0.00	0.04
K=50	90.42	+6.79	0.17	90.78	+7.15	0.10
K=100	90.96	+7.33	0.33	91.32	+7.69	0.23
K=500	92.35	+8.72	1.68	92.73	+9.10	1.03
K=1000	93.64	+10.01	3.48	94.04	+10.41	1.95
K=2000	94.09	+10.46	7.02	94.31	+10.68	3.91
K=ALL	94.92	+11.29	182.44	95.15	+11.52	107.89
Best MAP in NTCIR-9 workshop : 83.7 (13.50 s)						

TABLE IV
RESULTS OF NTCIR-9 DRY RUN

	Frame-level matching			State-level matching		
	MAP	Change (points)	Time (s)	MAP	Change (points)	Time (s)
Baseline	74.42	+0.00	0.04	74.42	+0.00	0.04
K=50	80.68	+6.26	0.17	80.22	+5.80	0.10
K=100	82.67	+8.25	0.34	82.22	+7.81	0.19
K=500	85.84	+11.42	1.73	85.23	+10.81	0.96
K=1000	87.11	+12.69	3.42	86.44	+12.02	1.96
K=2000	87.64	+13.23	7.14	86.93	+12.52	4.08
K=ALL	88.47	+14.05	181.15	87.58	+13.17	108.19

TABLE V
RESULTS OF NTCIR-10 FORMAL RUN

	Frame-level matching			State-level matching		
	MAP	Change (points)	Time (s)	MAP	Change (points)	Time (s)
Baseline	57.60	+0.00	0.04	57.60	+0.00	0.04
K=50	69.34	+11.73	0.16	69.05	+11.45	0.09
K=100	72.70	+15.09	0.32	72.45	+14.85	0.19
K=500	76.48	+18.88	1.70	75.96	+18.36	0.94
K=1000	78.20	+20.60	3.53	77.33	+19.73	1.98
K=2000	79.59	+21.98	7.04	78.76	+21.16	4.00
K=ALL	81.60	+24.00	138.59	80.58	+22.97	82.43
Best MAP in NTCIR-10 workshop : 67.5 (2.0 s)						

TABLE VI
RESULTS OF NTCIR-10 DRY RUN

	Frame-level matching			State-level matching		
	MAP	Change (points)	Time (s)	MAP	Change (points)	Time (s)
Baseline	69.73	+0.00	0.04	69.73	+0.00	0.04
K=50	76.46	+6.74	0.18	76.24	+6.52	0.10
K=100	81.53	+11.80	0.35	81.16	+11.43	0.19
K=500	86.00	+16.27	1.81	85.62	+15.89	0.97
K=1000	87.11	+17.38	3.57	86.34	+16.62	1.95
K=2000	87.88	+18.15	7.55	87.05	+17.32	4.05
K=ALL	90.13	+20.41	148.59	88.84	+19.11	82.24

points, and processing time was 0.35 s for frame-level matching and 0.23 s for state-level matching. The best MAP was 83.7%, with 13.5 s processing time for the formal run submitted for NTCIR-9 workshop [15], as shown in the bottom of Table III; it was 67.5 % with 2.0 s processing for the formal run submitted for NTCIR-10 workshop [16], as shown in the bottom of Table V. The proposed method was able to improve the best MAP while maintaining a practical processing time.

When K = 2,000, MAP improved between 10.46 and 21.98 points, with 7.19 s processing for frame-level matching and 4.01 s for state-level matching. Compared with the case that of K = 100, the MAP was much improved. However, processing took longer. The reduction of processing time is thought to be important.

The best STD accuracy was obtained by frame-level matching and applying the proposed method to all utterances (K = ALL) in spoken documents. This is because more detailed matching was performed by frame-level matching. State-level matching was faster than frame-level matching, running at about 1.76 times the speed for the same K. When constraining by processing time, the MAP from state-level matching was nearly the same as from frame-level matching.

IV. CONCLUSIONS

In this paper, we proposed a rescoring method that uses the probabilities output by a DNN to rescore the candidate utterances scored most highly by our conventional STD results. This restriction reduces the processing time. We evaluated two proposed methods: frame-level matching and state-level matching. Experimental results demonstrated that the proposed method is more accurate than the best method submitted for the NTCIR-9 and NTCIR-10 workshops and still maintains a practical processing time. As future work, we aim to reduce the processing time that accompanies increasing values of K and to improve the rescoring method.

ACKNOWLEDGMENT

This work was supported by JSPS (C), KAKENHI, Grant Number 24500124.

REFERENCES

[1] C. Auzanne, JS. Garofolo, JG. Fiscus, and WM Fisher,"Automatic Language Model Adaptation for Spoken Document Retrieval," B1, 2000TREC-9 SDR Track, 2000.

[2] A. Fujii, and K. itou, Evaluating Speech-Driven IR in the NTCIR-3Web Retrieval Task, Third NTCIR Workshop, 2003.

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

[4] 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, pp.325-328, 2006.

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

[6] G. Hinton, S. Osindero, and Y. Teh, A fast learning algorithm for deep belief nets, Neural Computation, vol. 18, pp. 1527–1554, 2006.

[7] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, Deep Neural Networks for Acoustic Modeling in

Speech Recognition, IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82-97, 2012.
[8] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, Exploiting diversity for spoken term detection, in Proc. ICASSP, pp. 8282-8286, 2013.
[9] B. Kingsbury, J. Cui, X. Cui, M. J. F. Gales, K. Knill, J. Mamou, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schluter, A. Sethy, and P. C. Woodland, A high performance Cantonese keyword search system, in Proc. ICASSP, pp.8277-8281, 2013.
[10] K. Maekawa, Corpus of Spontaneous Japanese: Its design and evaluation. Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR2003), Tokyo, 2003.
[11] National Institute for Japanese Language and Linguistics, Corpus of Spontaneous Japanese, http://www.ninjal.ac.jp/corpus_center/csj/
[12] A. Lee and T. Kawahara, Recent Development of Open-Source Speech Recognition Engine Julius, Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2009.
[13] Tomoyosi Akiba et al., Overview of the IR for Spoken Documents Task in NTCIR-9 Workshop, NTCIR-9 Workshop Meeting, pp. 223-235, 2011.
[14] Tomoyosi Akiba et al., Overview of the NTCIR-10 SpokenDoc-2 Task, NTCIR-10 Workshop Meeting, pp. 573-587, 2013.
[15] H. Nishizaki, H. Furuya, S. Natori, and Y. Sekiguchi, Spoken term detection using multiple speech recognizers' outputs at NTCIR-9 SpokenDoc STD subtask. In Proceedings of the Ninth NTCIR Workshop Meeting, 2011.
[16] 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.