論文 / 著書情報
Article / Book Information

Title	Committee-Based Active Learning for Speech Recognition
Authors	yuzo hamanaka, Koichi Shinoda, Takuya Tsutaoka, SADAOKI FURUI, Tadashi Emori, Takafumi KOSHINAKA
出典 / Citation	IEICE Trans. Inf. & Syst, vol. E94-D, No. 10, pp. 2015-2023
発行日 / Pub. date	2011, 10
URL	http://search.ieice.org/
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 Copyright (c) 2011 Institute of Electronics, Information and Communication Engineers.

PAPER

Committee-Based Active Learning for Speech Recognition

Yuzo HAMANAKA[†*], *Nonmember*, **Koichi SHINODA**[†a)], *Senior Member*, **Takuya TSUTAOKA**[†], *Nonmember*, **Sadaoki FURUI**[†], *Fellow*, **Tadashi EMORI**[††**], *Nonmember*, *and* **Takafumi KOSHINAKA**[††,†], *Member*

SUMMARY We propose a committee-based method of active learning for large vocabulary continuous speech recognition. Multiple recognizers are trained in this approach, and the recognition results obtained from these are used for selecting utterances. Those utterances whose recognition results differ the most among recognizers are selected and transcribed. Progressive alignment and voting entropy are used to measure the degree of disagreement among recognizers on the recognition result. Our method was evaluated by using 191-hour speech data in the Corpus of Spontaneous Japanese. It proved to be significantly better than random selection. It only required 63 h of data to achieve a word accuracy of 74%, while standard training (i.e., random selection) required 103 h of data. It also proved to be significantly better than conventional uncertainty sampling using word posterior probabilities.

key words: *active learning, query by committee, LVCSR, progressive alignment*

1. Introduction

Statistical speech-recognition systems require a large amount of speech data and transcriptions for training speech models. Unfortunately, it is too expensive to transcribe speech data. Semi-supervised learning and active learning have been studied as ways of reducing the costs of such transcriptions. Semi-supervised learning [1] is a learning approach where unlabeled data are used for training as well as labeled data. Active learning is where a learner selects data to be labeled, which are then used for training. Transcription costs in active learning for speech recognition are reduced by selecting and transcribing a small amount of informative data, which is expected to be the most useful for training.

There have been many studies on active learning for speech recognition [2]–[5]. The key issues in active learning are the criteria for selecting useful utterances. Many approaches [2]–[4] have used *uncertainty sampling* based on confidence measures. The initial recognizer in these approaches, which is prepared beforehand, is first used to recognize all the utterances that have recognition results with less confidence and is then selected. The word posterior probabilities (WPPs) for each utterance have often been used as confidence measures [2], [3]. Varadarajan *et al.* [4] used entropy in a word lattice for each utterance produced by a recognizer.

This paper proposes a committee-based active-learning method for large vocabulary continuous speech recognition (LVCSR) [6]. Multiple speech recognizers are prepared beforehand in this approach, and those utterances with a high *degree of disagreement* among the recognition results are selected to be manually transcribed.

A committee-based active-learning approach, called query-by-committee (QBC), was first proposed by Seung *et al.* [7]. It was applied to selective-sampling problems by Freund [8], where the learner examined many unlabeled examples and only selected those samples that were more informative for learning than the others. The learner in this committee-based sampling scheme constructed a *committee* of classifiers using the training data currently available. Each committee member then classified the candidate samples extracted from the unlabeled training data, and the learners measured *the degree of disagreement* among the committee members. Samples with larger degrees of disagreement were selected for labeling.

Early QBC studies by Seung et al. [7] took into consideration their theoretical aspects within the context of binary-classification problems. They defined a version space as a set of concepts that labeled all the training examples correctly, and they developed an algorithm to effectively restrict the version space as the number of examples increased. They proved that it achieved an exponential reduction in the ratio of the number of labeled examples required to attain a necessary classification accuracy to the number needed in the random selection of training samples.

However, it is rather difficult to directly apply the original QBC framework to speech recognition, since our classification problem, i.e., LVCSR, is much more complex than simple binary classification problems. The QBC approach to problems other than such simple ones may not exponentially reduce the number of labeled examples required to achieve a certain accuracy. However, Dagan *et al.* [9] experimentally proved that committee-based active learning performs well in part-of-speech tagging tasks, which is a more complex problem than simple binary-classification problems.

Inspired by this work, we applied committee-based ac-

Manuscript received March 17, 2011.
Manuscript revised June 16, 2011.

[†]The authors are with Tokyo Institute of Technology, Meguro-ku, Tokyo, 152–8552 Japan.
[††]The authors are with NEC Corporation, Kawasaki-shi, 211–8666 Japan.
[*]Presently, the author is with Asahi Kasei Corporation, Chiyoda-ku, Tokyo, 101–8101 Japan.
[**]Presently, the author is with Yahoo Japan Corporation, Minato-ku, Tokyo, 107–6211 Japan.
a) E-mail: shinoda@cs.titech.ac.jp

DOI: 10.1587/transinf.E94.D.2015

tive learning to speech recognition, where utterances with higher degrees of disagreement in their recognition results among several classifiers were selected as informative training data and then transcribed. We first aligned word sequences from the K recognizers using an efficient search technique and then calculated the *voting entropy* of the alignment result, represented by a confusion network (CN).

Our approach is closely related to the entropy-based approach proposed by Varadarajan *et al.* [4]. However, while they measured the entropy of a word lattice produced by a single recognizer, we measured that of a CN produced by the utterances identified by many recognizers.

This paper is organized as follows. Section 2 explains the active method of learning we propose. Section 3 describes how we evaluated it and Sect. 4 presents the results of our analysis of the evaluations. Section 5 concludes the paper.

2. Committee-Based Active Learning

2.1 Algorithm

Figure 1 outlines the flow for our committee-based active-learning method for speech recognition. Let us assume we have training data, T, whose utterances are fully transcribed, and untranscribed training data, U. We determine the number of recognizers, K, for active learning, the amount of data N (h) to be selected in one active learning cycle, and the amount of transcribed data we would like to have, which are all done beforehand.

The active learning is carried out in a five-step process.

1. Divide the training data, T, randomly and equally into K data sets, T_k, $k = 1, \ldots, K$.
2. Train the k-th recognizer, M_k, by using the k-th data set, T_k, for $k = 1, \ldots, K$.
3. Recognize all the utterances in the untranscribed training data, U, with each recognizer, M_k, $k = 1, \ldots, K$, to generate K different recognition results (sentences) for each utterance.
4. Select those utterances with a higher degree of disagreement between K recognizers than the others until the selected utterances reach N (h).
5. Transcribe the selected data from U, move them from U to T, and go to Step 1.

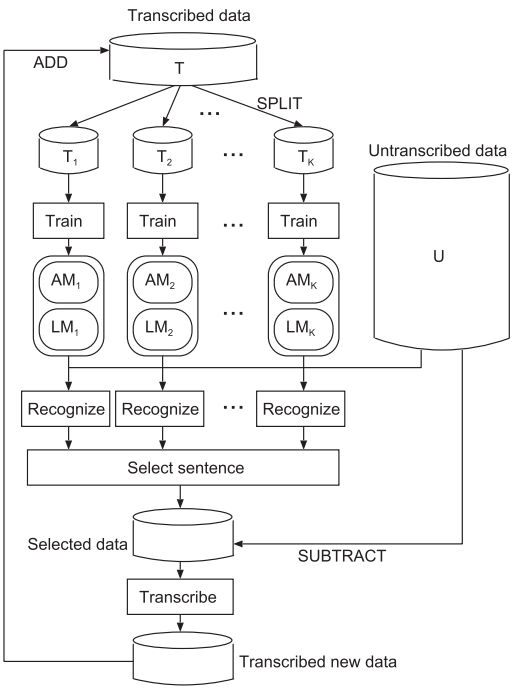

Fig. 1 Committee-based active-learning scheme for speech recognition.

We repeat this active-learning cycle until the amount of transcribed data reaches our predetermined goal. The selection process in Step 4 is explained in detail in Sects. 2.2 and 2.3.

Active learning for speech recognition can be applied not only to an acoustic model but also to a language model. We applied the active learning process previously described in three ways: to both of these and to either of these. A single recognizer trained with the currently available training data, T, was used for recognizing test data in our evaluation.

2.2 Sentence Alignment

Unlike part-of-speech tagging [9], the numbers of recognized words for an utterance differ among recognizers, so it is necessary to align sentences to measure the degree of disagreement. The alignment process for two sentences is called pairwise alignment and that for more than two sentences is called multiple alignment. When the dynamic programming (DP) algorithm, which is used for pairwise alignment, is applied to multiple alignment, its computational complexity increases exponentially in proportion to the number of sentences. Thus, many approximation methods for multiple alignment that are less computationally complex are being studied. The alignment method in ROVER [11] is one such method, in which multiple alignment is done by aligning one sentence and then aligning each of the other sentences one by one. This method, however, does not focus much on the alignment algorithm or accuracy. The results may change according to the order of sentences to be aligned, and the way the order is determined is not mentioned. We employed a progressive search [12], which could be expected to produce more accurate alignment. The progressive search algorithm is as follows:

1. Calculate the degree of similarity between all pairs in a sentence set and construct a guide tree (Sect. 2.2.1).
2. From the first node added to the tree, align child nodes (which may be two sequences, a sequence and an alignment result, or two alignment results). Repeat this for all other nodes in the order that they were added to the tree until all sentences have been aligned.

2.2.1 Construction of Guide Tree

We employed the unweighted pair group method with arith-

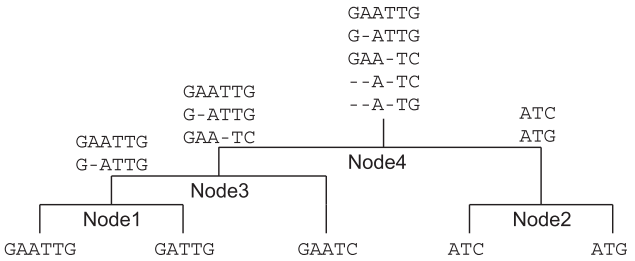

Fig. 2 Example guide tree. Hyphens "–" indicate gaps.

Table 1 Example multiple-alignment result. There are five rows, K, and six columns, C. Each letter of the alphabet is a unique word in the vocabulary. Hyphens "–" indicate gaps.

	1	2	3	4	5	6
1	G	A	A	T	T	G
2	G	–	A	T	T	G
3	G	A	A	–	T	C
4	–	–	A	–	T	C
5	–	–	A	–	T	G

metic mean (UPGMA) to construct a guide tree (Fig. 2). DNA sequences are used in Figs. 2 and 3 and Table 1 to explain progressive alignment, while words in speech recognition results can be applied in the same way.

Initialization Assign a cluster, C_i, to each sequence, s_i. Add all clusters to a cluster group, S. Carry out pairwise alignment for all pairs of sentences (s_i, s_j), and calculate similarity between cluster pair C_i, C_j, which is denoted by d_{ij}. d_{ij} is the mean of match, mismatch, and gap scores. Here they correspond to 1, 0, and 0.

Iteration Combine C_i, C_j with the largest d_{ij} and create a new cluster, C_k. Add C_k to S and delete C_i, C_j from S. The similarity, d_{kl}, between C_k and one of the other clusters C_l is calculated as

$$d_{kl} = \frac{1}{|X||Y|} \sum_{x \in X} \sum_{y \in Y} d_{xy} \tag{1}$$

where X is a group of sentence indexes in cluster C_k and Y is that in C_l. Iterate this process until S has only two clusters.

Termination Construct a guide tree in the order clusters are combined.

Figure 2 has an example of a guide tree. It should be noted that a gap introduced in alignment is treated as a word.

2.2.2 Alignment of Child Nodes

Child nodes are aligned after a guide tree is constructed. *Two sentences* (Node1 and Node2 in Fig. 2) can be aligned by pairwise alignment. *A sentence and an alignment result* (Node3 in Fig. 2) can be aligned and *two alignment results* (Node4 in Fig. 2) can be aligned as in Fig. 3. DP matching is carried out by expanding the DP matrix, maintaining the original alignment relationship in the alignment result.

				j				
				1	2	3	4	5
	n \ n'	1		G	A	A	T	C
	1	2	0	0	0	0	0	0
i	1	G	G	0	6	3	0	0	0
	2	A	–	0	3	6	3	0	−3
	3	A	A	0	3	9	12	9	6
	4	T	T	0	3	9	12	18	15
	5	T	T	0	3	9	12	18	18
	6	G	G	0	6	9	12	18	18

Fig. 3 Example of DP matrix. Hyphens "–" indicate gaps.

Figure 3 is a DP matrix calculated at Node3 in Fig. 2.

We can explain this alignment using an expanded DP matrix. Each node has two objects to be aligned, which are a sentence or an alignment result. Let N be the number of sentences in an object to be aligned, s_n $(1 \le n \le N)$ be each sentence, Q be the number of columns in the sentence, and m_n^q be a word in sentence s_n in the q $(1 \le q \le Q)$-th column. Similarly, let N' be the number of sentences in the other object to be aligned, $s'_{n'}$ $(1 \le n' \le N')$ be each sentence, Q' be the number of columns in the sentence, and $m'^{q'}_{n'}$ be a word in sentence $s'_{n'}$ in the q' $(1 \le q' \le Q')$-th column. $s(a, b)$ is a score which is 2 if words a and b are the same except for a gap, otherwise −1.

$$s(a,b) = \begin{cases} 2 & (a = b \neq \text{gap}), \\ -1 & (a = b = \text{gap or } a \neq b) \end{cases} \tag{2}$$

This value of $s(a, b)$ was the best in the pre-experiment. We express $\text{SPS}(\{a_i\})$ as the sum of all pair scores $s(a, b)$ in a word set, $\{a_i\}$. The score, $S(i, j)$, at point (i, j) in the DP plane is calculated as:

$$S(i, 0) = 0, \tag{3}$$

$$S(0, j) = 0, \tag{4}$$

$$S(i,j) = \max \begin{cases} S(i-1, j-1) + \text{SPS}(\{m_n^i, m'^j_{n'}\}), \\ S(i, j-1) + \text{SPS}(\{N\text{gaps}, m'^j_{n'}\}), \\ S(i-1, j) + \text{SPS}(\{m_n^i, N'\text{gaps}\}), \end{cases} \tag{5}$$

where $1 \le n \le N, 1 \le n' \le N'$. During this recursive process, the previous point selected in Eq. (5) is memorized for every point (i, j). The alignment result is obtained by tracing back after the scores of all points in the DP matrix have been calculated. In the example at Node3 in Fig. 3, $N = 2$, s_1 ="GAATTG", s_2 ="G−ATTG", $Q = 6$, and $N' = 1$, s'_1 ="GAATC", $Q' = 5$. The score of (3,3) in the matrix is, for example,

$$S(3,3) = \max \begin{cases} S(2,2) + \text{SPS}(\{m_1^3, m_2^3, m'^3_1\}) \\ \quad = 6 + s(\text{A},\text{A}) + s(\text{A},\text{A}) + s(\text{A},\text{A}) = 12, \\ S(3,2) + \text{SPS}(\{-, -, m'^3_1\}) \\ \quad = 9 + s(\text{A},-) + s(\text{A},-) + s(-,-) = 6, \\ S(2,3) + \text{SPS}(\{m_1^3, m_2^3, -\}) \\ \quad = 3 + s(\text{A},\text{A}) + s(\text{A},-) + s(\text{A},-) = 3 \end{cases}$$

$$= 12$$

2.3 Voting Entropy

We can measure the degree of disagreement among recognizers by *voting entropy*. The results of multiple alignment described in Sect. 2.2 can be represented by $K \times C$ matrix where K is the number of recognizers and C is the number of columns in the alignment result. An example of this is given in Table 1. Let P be the number of unique words in column c ($1 \leq c \leq C$), w_p ($1 \leq p \leq P$) be a unique word, and $V(w_p, c)$ be the number of w_p in the c-th column. Then, the voting entropy, $\mathrm{VE}(c)$, for the distribution of K words in the c-th column of the alignment result is defined as

$$\mathrm{VE}(c) = -\sum_{p=1}^{P} \frac{V(w_p, c)}{K} \log \frac{V(w_p, c)}{K} \tag{6}$$

The degree of disagreement, D, is defined as the average voting entropies over all the columns:

$$D = \frac{1}{C} \sum_{c=1}^{C} \mathrm{VE}(c) \tag{7}$$

Those utterances with larger D are selected to be transcribed at Step 4 in the process of the active-learning method explained in Sect. 2.

3. Experiment

3.1 Experimental Conditions

We evaluated our method using speech data of academic lectures obtained from male speakers in the Corpus of Spontaneous Japanese (CSJ) [13]. We prepared 224,434 utterances (191 h) from 666 speakers as the untranscribed data for active learning. In our evaluation, an utterance was defined as a speech sample longer than one second, where each shorter speech sample was connected with its successive sample such that their total duration was longer than one second. It should be noted that these utterances were fully transcribed but we assumed that they were untranscribed in the utterance-selection experiments. We used 2328 utterances (1.95 h) from another ten speakers, which were fully transcribed, to test the performance of the proposed method.

The frame period in speech analysis was 10 ms and the frame width was 25 ms. The speech-feature vector was 39 dimensional, consisting of 12-order mel-frequency cepstral coefficients (MFCCs) appended with energy, delta, and delta-delta coefficients. We applied cepstral mean subtraction to all utterances.

The acoustic model for a recognizer was a triphone hidden Markov model with 3000 states, each of which had a Gaussian-mixture probability density function. The number of triphones was 7361. There were 16 mixtures in each state. The structure of the acoustic model remained unchanged throughout all the experiments in this study. We applied a two-pass search for speech recognition. A 2-gram language model was used in the first pass and a 4-gram language model was used in the second. HTK [14] was used in the experiment.

We randomly selected 25 h (29,461 utterances) of data as the initial transcribed training data from the training data, and used them to train the initial acoustic model and the initial 2-gram and 4-gram language models. The other data from the training data were used as untranscribed data for active learning. The amount of data N to be selected at one cycle of the active learning process was set at 25 h.

We carried out three experiments to confirm how effective our proposed method was by (1) investigating how to construct a committee, (2) comparing it with other methods (random selection, a WPP-based method, and a simplified form of Varadarajan's method), (3) optimizing the number of recognizers in a committee, K, and (4) comparing methods of alignment (progressive and ROVER alignment).

3.2 Results

3.2.1 Investigation into How to Construct a Committee

The number of speech recognizers in a committee, K, in this experiment was set to eight. Each recognizer consisted of an acoustic model (AM) and a language model (LM). To investigate what was the most effective combination of AMs and LMs, we trained AMs and LMs in three ways and constructed a committee.

AM8-LM8
: Trained the AM and LM of the k-th recognizer with data T_k ($k = 1, \ldots, 8$),

AM8-LM1
: Trained the AM of the k-th recognizer with data T_k ($k = 1, \ldots, 8$) and trained an LM with the whole data, T, and

AM1-LM8
: Trained the LM of the k-th recognizer with data T_k ($k = 1, \ldots, 8$) and trained an AM with the whole data, T.

Figure 4 plots the recognition results. The best result was obtained when an LM was trained with currently available transcribed data T and shared among all the committee recognizers. This may be because an LM needs more training data to achieve a certain level of performance than an AM. When LMs are trained with a divided data set, the reliability of recognition results by all recognizers may decrease.

3.2.2 Comparison with Other Methods

We compared the proposed approach with three other methods: random selection, a WPP-based method, and a simplified form of Varadarajan's method. The utterances to be transcribed in random selection were arbitrarily chosen. The word-posterior probabilities of words with the WPP-based method [2] in the recognition results of all utterances in untranscribed data U were averaged, and those utterances with

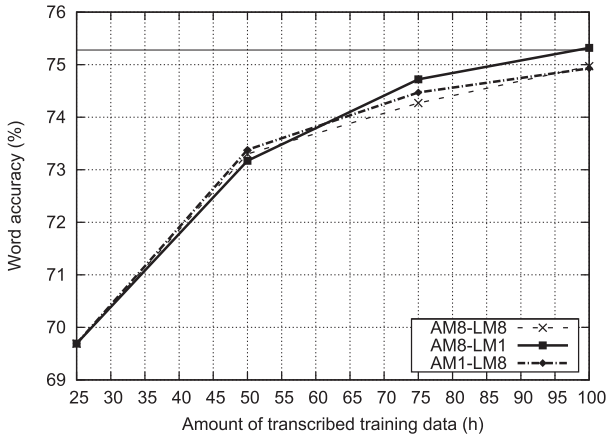

Fig. 4 Recognition results for three ways of constructing a committee. The horizontal solid line plots the recognition result (75.3%) obtained by using all the training data (191 h) we prepared for the experiment.

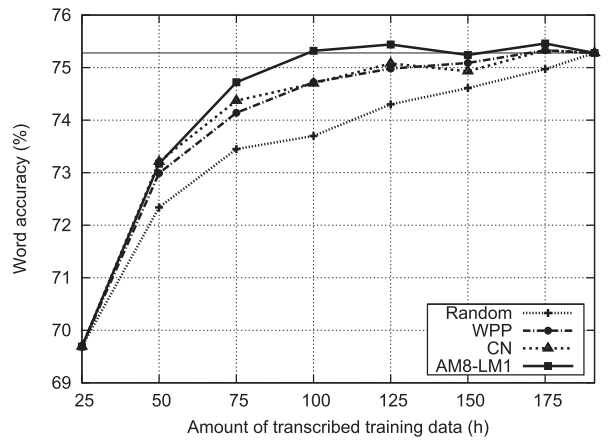

Fig. 5 Recognition results by random selection (Random), WPP-based method (WPP), simplified form of Varadarajan's method (CN), and proposed method (AM8-LM1).

lower word-posterior probabilities were selected. The word-posterior probabilities were calculated by making a confusion network (CN) from a word lattice. A CN is a chain of nodes where two adjacent nodes are connected by several arcs. Each arc represents a word or a gap. An entropy of word-posterior probabilities in a CN was calculated in the simplified form of Varadarajan's method (CN) [4], where first the entropy of word arcs for every adjacent node pair is calculated and then they are averaged over all the adjacent node pairs. Those utterances with the higher averaged entropies were selected [†].

Figure 5 plots the results obtained from the comparisons. Our proposed method had significantly better recognition than random selection. The proposed approach only required 63 h of data to achieve a word accuracy of 74.0% while random selection required 113 h. It also had better recognition than the other two methods, i.e., the WPP-based method and the simplified form of Varadarajan's method. Furthermore, the new approach obtained a word accuracy of 75.3% in just 100 h, which is almost half the amount of data (191 h) needed in standard training (i.e., random selection) to achieve the same accuracy. When the amount of data was 125 h, the recognition accuracy of our proposed method was higher than that obtained using all the data (191 h). This indicates that nearly one-third of the training data did not contribute to increase recognition performance, and our method successfully excluded those data in its selection process.

Table 2 Recognition results (word accuracy %) obtained with different numbers of recognizers, K.

Amount of T (h)	AM4-LM1	AM8-LM1	AM16-LM1
25	69.7	69.7	69.7
50	73.4	73.2	73.6
75	74.7	74.7	74.7
100	75.1	75.3	75.2

We carried out a matched-pair test [15] to investigate whether the difference in recognition accuracy between the proposed method and the others was statistically significant or not. The proposed method was significantly better than the random selection at 0.1% level when the data amounts were 75 h, 100 h, 125 h, and at 1% level when the amounts were 50 h, 150 h, 175 h. It was also significantly better than the WPP-based method at 1% level when the data amount was 100 h and at 5% level when the data amounts were 75 h and 125 h. It was significantly better than the CN-based method at 1% level when the data amount was 100 h. Thus, the effectiveness of the proposed method was confirmed.

In Figs. 4 and 5, the recognition accuracy of the proposed method was slightly better than that obtained by using all the training data. This result indicates that some portion of the training data may degrade the recognition accuracy and that our method can effectively exclude those data.

3.2.3 Optimization of Number of Recognizers in Committee

Table 2 lists the active learning results obtained with various K of 4, 8, and 16. No great differences were observed in this experiment. Taking into account the computational cost in sentence selection, K was sufficiently acceptable at four with our method.

3.2.4 Comparison of Alignment Methods

Table 3 lists the recognition results obtained with two different alignment methods of progressive and ROVER alignment. The word accuracy using progressive alignment was slightly better than that using ROVER alignment except at 50 h in AM8-LM1, while the difference between them was not significant.

[†]In Varadarajan's method [4], entropies of utterances were measured by their word lattices, not by their CNs. Furthermore, they used the distance between each pair of utterances as well as entropies to select utterances, even though we did not use this here.

Table 3 Recognition results (word accuracy %) with different alignment methods: Progressive and ROVER alignment.

	AM8-LM1		AM16-LM1	
Amount of T (h)	Progressive	ROVER	Progressive	ROVER
25	69.7	69.7	69.7	69.7
50	73.2	73.4	73.6	73.2
75	74.7	74.5	74.7	74.7
100	75.3	75.0	75.2	75.2

4. Discussion

4.1 Discussion on Experimental Results

We analyzed the experimental results to find why the proposed method was significantly better than the WPP-based method. First, we investigated which model, an AM or an LM, contributed more to improved recognition accuracy. The test data in Fig. 5 were recognized by a single recognizer with an AM and an LM, which were trained using data T of 25, 50, 75, and 100 h. Here, whenever data T were 25, 50, 75, or 100 h, either model was trained with all the training data (191 h) and the other model was trained with data T available at that time, so that the model trained with all the training data would not affect the recognition results and only improved recognition accuracy obtained with the other model could be observed. Figure 6 plots the LM-only active learning results obtained with an AM trained with all the training data, and Fig. 7 plots the AM-only results obtained with an LM trained with all the training data.

These figures indicate both models with the proposed method were better than those with the WPP-based method. However, there was no significant difference between them either in Fig. 6 or in Fig. 7. This indicates that, while neither AM-only active learning nor LM-only active learning may be effective, their combination is significantly effective. The recognition accuracies in Fig. 6 increase as the amount of data T increases, whereas the accuracies in Fig. 7 become almost constant after 50 h, which means LM training requires more data than AM training does. We further evaluated the *triphone coverage* and *N-gram hit rate* to investigate why the proposed method were better than the WPP-based method.

4.1.1 Triphone Coverage

We measured how many triphones contained in the test data were covered by the selected data T to analyze how much useful utterances for speech recognition were selected. Triphone coverage (%) was calculated by the ratio of the unique triphones in the test data covered by those in the data T.

Fig. 8 plots the triphone coverages for the different methods. There were 3647 unique triphones in the transcription of the test data in this experiment. The proposed approach has higher triphone coverage than the WPP-based method and the random selection method. This high triphone coverage may be one reason that the recognition accuracy with the proposed method was better than that with those methods. However, since triphone coverage is over 98.6% even when T is 25 h, the contribution of this high triphone coverage to the improvement in accuracy may be limited.

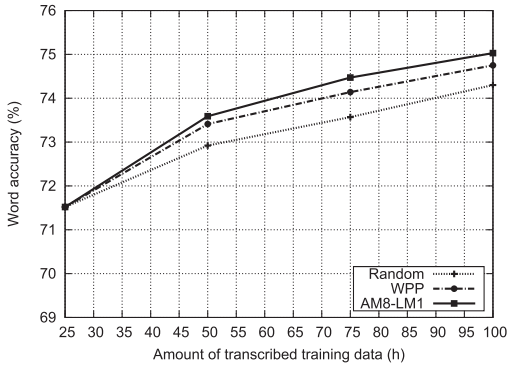

Fig. 6 Recognition results with an AM trained with all training data (191 h) and LM trained with data T selected by random selection (Random), WPP-based method (WPP), and proposed method (AM8-LM1).

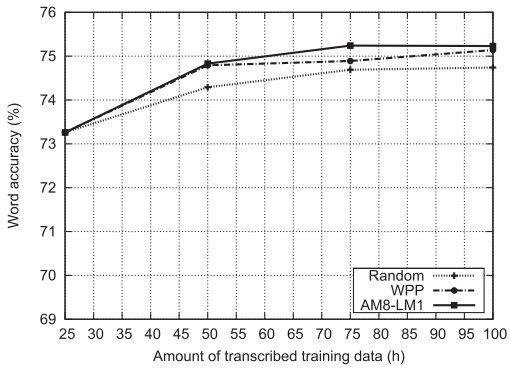

Fig. 7 Recognition results with LM trained with all training data (191 h) and AM trained with data T selected by random selection (Random), WPP-based method, (WPP) and proposed method (AM8-LM1).

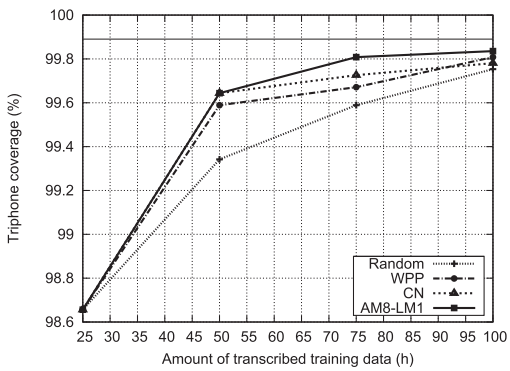

Fig. 8 Triphone coverage by random selection (Random), WPP-based method (WPP), and proposed method (AM8-LM1). The horizontal solid line plots the triphone coverage (99.89%) obtained by using all the training data (191 h) prepared for the experiment.

Table 5 Vocabulary sizes of training data, OOV rates, and test set perplexities for 4-gram.

	Vocabulary size			OOV rates (%)			Test set perplexity		
Amount of T (h)	Random	WPP	AM8-LM1	Random	WPP	AM8-LM1	Random	WPP	AM8-LM1
25	16375			22.5			76.9		
50	23402	28365	27141	18.7	17.0	17.2	72.9	74.5	74.2
75	29008	35414	34771	16.7	14.9	14.9	72.1	72.7	72.4
100	33577	39932	39700	15.2	13.5	13.9	71.0	71.0	70.3

Table 4 2-gram and 4-gram hit rates by random selection (Random), WPP-based method (WPP), and proposed method (AM8-LM1).

	2-gram hits rate (%)			4-gram hits rate (%)		
Amount of T (h)	Random	WPP	AM8-LM1	Random	WPP	AM8-LM1
25	56.1	56.1	56.1	20.4	20.4	20.4
50	63.0	64.0	64.1	25.1	24.4	24.4
75	66.9	67.8	68.0	27.9	27.3	27.1
100	69.0	70.1	70.3	30.0	29.3	29.4

4.1.2 N-Gram Hit Rate

We assessed how many word N-grams included the test data were included in the transcription of data T to analyze how effective the proposed method was from the point of view of the language model. The N-gram hit rate was calculated by the ratio of unique N-grams in the test data covered by those in the data T.

Table 4 lists the 2-gram and 4-gram hits rates with the three methods. We can see that the proposed approach and the WPP-based method had similar tendencies, and they had high 2-gram hit rates compared with random selection, while they had low 4-gram hit rates. It seems that the reason that recognition accuracy with an LM trained with data T with the new method was better than that with the WPP-based method was not related to the N-gram.

4.1.3 Vocabulary Size, OOV Rate, and Perplexity

Finally, we examined the vocabulary size of the training data, the out-of-vocabulary (OOV) rate in the test data, and the test set perplexity, for different amounts of the training data. The results are shown in Table 5. The vocabulary size of the training data for the proposed method and for WPP were both significantly larger than Random, and accordingly, the OOV rates for these two methods were smaller than Random, while there were no significant differences between the proposed method and WPP. The perplexities for the three methods were decreased as the amount of training data increases, but the differences between them were not significant. We believe the reduction of the OOV rates is one major reason for the effectiveness of our method and of WPP method.

4.2 Computational Time for Sentence Selection

Since our method uses multiple recognizers, some might argue that it has too high computational costs to be used in real application. To investigate this point, we compared computational time for the three methods. A computer with a 2.4-GHz Core 2 Duo processor and 3 GB of memory was used in the experiment. The sentence selection processes required by the three methods were as follows:

Random selection

1. Randomly select utterances to be transcribed.

WPP-based method

1. Recognize all utterances in data U, and calculate word-posterior probabilities.
2. Sort the utterances by their word-posterior probability averages, and select utterances with a lower average.

Proposed method

1. Construct a committee by using data T.
2. Generate K recognition-result sentences using the committee.
3. Individually carry out multiple alignments for K recognition-result sentences for all utterances.
4. Sort the utterances by the degree of disagreement D, and select utterances with higher D.

As random selection and sorting did not take much time, we neglected them. Here, computational time was calculated by measuring the time needed to process part of the data extracted randomly from the entire amount of data and then multiplying the time by the amount of data. We calculated the computational time to transcribe up to 100 h of data by only using one CPU.

4.2.1 WPP-Based Method

The recognition and calculation of WPPs for data U of 165, 140, and 115 h (420 h in total) were carried out until 100 h of data were transcribed with the WPP-based method. We recognized one hour of data randomly selected from data U of 165 h using an AM and an LM trained by transcribed data T of 25 h and then calculated WPP. It took 35 minutes. As the number of states and Gaussian mixtures in an AM were constant no matter how much data T there were, the computational times to recognize the same amount of data and calculate WPPs using the pairs of an AM and an LM trained with 25, 50, and 75 h data were considered to be almost the same. Thus, the computational time for the WPP-based method was calculated as

$$420 \times \frac{35}{60} = 245 \text{ (h)}$$

4.2.2 Proposed Method

We calculated the computational time with the proposed method of AM8-LM1, where the recognition and multiple alignment for data U for a total of 420 h were also carried out until 100 h of data were transcribed. First, we took into consideration the computational time in process 1. It took 7.13 h to train eight AMs by using each divided data set T_k when the amount of T was 25 h. It was not necessary to train an LM because it was already made to recognize test data. Training eight AMs with data T of 50 h and 75 h took twice and three times longer than training AMs with data T of 25 h, so the time in process 1 was calculated as

$$(1 + 2 + 3) \times 7.13 \;=\; 42.8 \text{ (h)}$$

Second, we took into consideration the computational time in process 2. We reduced the computational time to recognize data U for K times by cutting off the beam width from 120 to 100 in HDecode. We recognized one hour of data randomly selected from data U using an AM trained with divided transcribed data set T_k and an LM trained with whole transcribed data T, whose amount was 25 h. It took 20 min. The computational time in process 2 was

$$420 \times \frac{20}{60} \times 8 \;=\; 1120 \text{ (h)}$$

Finally, we calculated the computational time in process 3. It took 292 s to carry out progressive alignment for eight recognition-result sentences for the one hour of data selected in process 2, and it took 162 s to do ROVER alignment for the same sentences. The computational time in process 3 with progressive alignment was

$$420 \times \frac{292}{60 \times 60} \;=\; 34.1 \text{ (h)}$$

It took 1,197 h in total to select utterances with the proposed method. The computational time in process 2 dominated the whole computational time. The proposed approach required $K/2$ times longer than the WPP-based one, because the former needed to recognize data U for K as many times as the latter did, while the former reduced the recognition time by around half by cutting off the beam width.

The computational time increased as the amount of data U increased. However, this computational time would not be a practical issue considering the huge amount of data of over 190 h we used in this experiment. Even if it took about 1200 h with one CPU, it would only take 60 h with 20 CPUs.

5. Conclusion

We proposed an active method of learning based on query-by-committee for speech recognition. A committee was constructed by using data sets of divided transcribed data. A degree of disagreement for recognition-result sentences by the committee was calculated by multiple alignment and voting entropy and used for sentence selection.

Our method performed better than did the others in our evaluation where we used speech data of academic lectures by male speakers in CSJ. It only took 63 h to obtain a word accuracy of 74.0% with the proposed method (AM8-LM1), while it took random selection 107 h. It also proved to be significantly better than the WPP-based method and the simplified form of Varadarajan's method. The proposed approach required just 100 h to achieve a word accuracy of 75.3%, which was obtained by using all the training data of 191 h.

We demonstrated AMs should be trained by divided data sets and an LM trained with the entire amount of transcribed data to construct a committee. The optimal number of recognizers K in a committee was four, considering the fact that there were no significant improvements when K was increased to eight and 16 and that the computational time with the proposed method was $K/2$ times longer than that with the WPP-based method. We also found progressive alignment was slightly better than ROVER alignment as a multiple method of alignment, though the difference between them in recognition accuracy was not significant.

The reason our approach was better than the WPP-based method has not yet become clear. We investigated triphone coverage, N-gram hit rates, OOV rates, and perplexities in those methods but found no significant difference in them. Further analysis should be carried out.

In future, we plan to investigate how to construct different recognizers. We are also interested in using numerous word graphs, each of which is generated by one recognizer, in our framework. We also plan to combine the proposed method and others using a confidence measure. Our method is expected to be more effective in different tasks such as call routing in telephone applications, and we plan to apply our method to these.

Acknowledgments

This work was supported by a Grant-in-Aid for Scientific Research (B) 20300063.

References

[1] G. Tur, D. Hakkani-Tur, and R.E. Schapire, "Combining active and semi-supervised learning for spoken language understanding," Proc. Speech Commun., vol.45, pp.171–186, 2005.
[2] D. Hakkani-Tur, G. Riccardi, and A. Gorin, "Active learning for automatic speech recognition," Proc. ICASSP, pp.3904–3907, 2002.
[3] G. Riccardi and D. Hakkani-Tur, "Active learning: Theory and applications to automatic speech recognition," IEEE Trans. Speech Audio Process., vol.13, no.4, pp.504–511, 2005.
[4] B. Varadarajan, D. Yu, L. Deng, and A. Acero, "Maximizing global entropy reduction for active learning in speech recognition," Proc. ICASSP, pp.4721–4724, 2009.
[5] H. Lin and J. Bilmes, "How to select a good training-data subset for transcription: submodular active selection for sequences," Proc. Interspeech, pp.2859–2862, 2009.

[6] Y. Hamanaka, K. Shinoda, S. Furui, T. Emori, and T. Koshinaka, "Speech modeling based on committee-based active learning," Proc. ICASSP, SP-L8.1, Dallas, 2010.
[7] H.S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," Workshop on Comput. Learning Theory, pp.287–294, 1992.
[8] Y. Freund, H.S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," Mach. Learn., vol.28, pp.133–168, 1997.
[9] I. Dagan and S.P. Engelson, "Committee-based sampling for training probabilistic classifiers," Proc. ICML, pp.150–157, 1995.
[10] G. Tur, R. Schapire, and D. Hakkani-Tur, "Active learning for spoken language understanding," Proc. ICASSP, vol.1, 2003.
[11] J.G. Fiscus, "A post-processing system to yield reduced error word rates: Recognizer output voting error reduction (ROVER)," Proc. IEEE Workshop on Automatic Recognition and Understanding, pp.347–354, 1997.
[12] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, Biological Sequence Analysis, Cambridge University Press, 1998.
[13] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," Proc. LREC, vol.2, pp.947–952, 2000.
[14] "The Hidden Markov Model Toolkit," http://htk.eng.cam.ac.uk/
[15] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," Proc. ICASSP, pp.532–535, 1989.

Yuzo Hamanaka received his B.E. in integrated information technology from Aoyama Gakuin University, Kanagawa, Japan, in 2008 and received his M.E. in computer science from the Tokyo Institute of Technology, Tokyo, Japan, in 2010. He is now with Asahi Kasei Corporation.

Koichi Shinoda Koichi Shinoda received his B.S. in 1987 and his M.S. in 1989, both in physics, from the University of Tokyo. He received his D.Eng. in computer science from the Tokyo Institute of Technology in 2001. In 1989, he joined NEC Corporation, Japan, and was involved in research on automatic speech recognition. From 1997 to 1998, he was a visiting scholar with Bell Labs, Lucent Technologies, Murray Hill, NJ. From June 2001 to September 2001, he was a Principal Researcher with Multimedia Research Laboratories, NEC Corporation. From October 2001 to March 2002, he was an Associate Professor with the University of Tokyo. He is currently an Associate Professor with the Tokyo Institute of Technology. His research interests include speech recognition, statistical pattern recognition, and human interfaces. Dr. Shinoda received the Awaya Prize from the Acoustic Society of Japan in 1997 and the Excellent Paper Award from the Institute of Electronics, Information, and Communication Engineers IEICE in 1998. He is an Associate Editor of Computer Speech and Language. He is a member of IEEE, ACM, ASJ, IPSJ, and JSAI.

Takuya Tsutaoka received his B.E. in information science from Tokyo Institute of Technology in 2010. From 2010, he is a Master student in the department of computer science of Tokyo Institute of Technology, Tokyo, Japan.

Sadaoki Furui Sadaoki Furui received his B.S., M.S., and Ph.D. in mathematical engineering and instrumentation physics from Tokyo University, Tokyo, Japan in 1968, 1970, and 1978. He is engaged in a wide range of research on speech analysis, speech recognition, speaker recognition, speech synthesis, and multimodal human-computer interactions and has authored or coauthored over 800 published articles. He has received Paper Awards and Achievement Awards from the IEEE, the IEICE, the ASJ, the ISCA, the Minister of Science and Technology, and the Minister of Education. He has also been a recipient of the prestigious Purple Ribbon Medal from the Japanese Emperor.

Tadashi Emori received his B.E. in physics from Tokyo University of Science, Chiba, Japan, in 1993 and received his M.E. in physics from Keio University, Yokohama, Japan, in 1995. In 1995, he joined NEC Corporation, Japan, and has been involved in research on speech recognition. His current research interests include statistical pattern recognition and machine learning. He is currently with Yahoo Japan Corporation. He is a member of the Acoustical Society of Japan.

Takafumi Koshinaka received his B.E. and M.E. in aeronautical engineering from Kyoto University, Kyoto, Japan, in 1991 and 1993. In 1993, he joined NEC Corporation, Japan, and has been involved in research on image and speech recognition. His current research interests include statistical pattern recognition, machine learning, and neural networks. He received the Young Researcher's Award from the Institute of Electronics, Information, and Communication Engineers (IEICE) in 2000. He is currently a Principal Researcher at Media and Information Research Labs., NEC Corporation, and a member of the Acoustical Society of Japan.