Spoken Content Retrieval Using Distance Combination and Spoken Term Detection Using Hash Function for NTCIR10 SpokenDoc2 task

Satoru Tsuge
Daido University
10-3 Takiharu-cho, Minami-ku,
Nagoya, Aichi 457-8539 Japan
tsuge@daido-it.ac.jp

Ken Ichikawa
Nagoya University
Furo-cho, Chikusa-ku,
Nagoya, Aichi 464-8603 Japan
ichikawa.ken@g.sp.m.is.nagoya-u.ac.jp

Norihide Kitaoka
Nagoya University
Furo-cho, Chikusa-ku,
Nagoya, Aichi 464-8603 Japan
kitaoka@nagoya-u.jp

Kazuya Takeda
Nagoya University
Furo-cho, Chikusa-ku,
Nagoya, Aichi 464-8603 Japan
kazuya.takeda@nagoya-u.jp

Kenji Kita
The University of Tokushima
2-1 Minamijosanjima-cho,
Tokushima, Tokushima
770-8506 Japan
kita@is.tokushima-u.ac.jp

ABSTRACT

In this paper we describe a spoken content retrieval (SCR) and a spoken term detection (STD) which were used in the 2nd round of the IR (Information Retrieval) for Spoken Documents (SpokenDoc2) task. Our SCR method maps the target documents into multiple vector spaces, which include a word-based vector space for word-based speech recognition results and a syllable-based vector space for syllable-based speech recognition results. The syllable-based space is spanned by axes extracted using latent semantic indexing (LSI). We also apply query expansion and morpheme weighting to the word-based space. Finally, the distance between the query and the documents in each vector space are combined and ranked for retrieving the documents. On the other hand, our STD method extracts sub-sequences from the target documents and converts them into bit sequences using the hash function. The query is also converted into a bit sequence in the same way. Candidates are detected by calculating the hamming distance between the bit sequence of the query and that of the target documents. Then, our method calculates the distances between the query and the candidates using DP (Dynamic Programming) matching. To evaluate the proposed methods, we conducted spoken document retrieval experiments using the SpokenDoc task from the NTCIR-9 meeting. Using these experimental results to set our parameters, we submitted the results for the SpokenDoc2 task at NTCIR-10.

Team Name

Team Big Four Dragons (TBFD)

Subtasks

Spoken Content Retrieval and Spoken Term Detection

Keywords

NTCIR-10, Spoken Document Retrieval

1. INTRODUCTION

There are many kinds of media data, such as pictures, movies, music, speech, and so on, on the Internet, and opportunities to retrieve such data are increasing. Spoken document retrieval methods have become an essential technique for information retrieval. In this paper, we focus on the speech data which are contained in most media data. Typical information retrieval methods for speech data first transcribe the target speech data into word or sub-word sequences using an automatic speech recognizer (ASR). We call these text documents transcribed by ASRs "spoken documents". By using text retrieval techniques, the target information can be retrieved from the spoken documents. In this paper, we propose two kinds of spoken document retrieval methods: spoken term detection (STD) and spoken content retrieval (SCR). Using these methods, our group, whose name is "Team Big Four Dragons (TBFD)", participated in both the STD subtask and the SCR subtask of the 2nd round of the IR for Spoken Documents (SpokenDoc2) at NTCIR-10.

Our SCR method represents both the target documents and the queries as vectors, based on the vector space model (VSM). The VSM maps the target documents and queries into a vector space spanned by the index terms. Generally, the vectors mapped by VSM, especially in the syllable-based vector space, are high-dimensional and sparse. Hence, they are sensitive to noise such as recognition errors, and it is also difficult to capture underlying semantic structures. Latent semantic indexing (LSI) is one way to overcome these problems[1][2]. Our SCR method applies LSI to the vector space for spoken document retrieval. Because there are some recognition errors and out-of-vocabulary (OOV) terms in spoken documents which are transcribed by speech recognizers, it is difficult to retrieve documents using only traditional text retrieval methods. Hence, some additional methods have been proposed for spoken document retrieval. One method is to use sub-word units instead of words for

the indexes[3]. This method avoids the OOV problem. In addition, a method which combines phoneme-based recognition results with word-based recognition results has been proposed for spoken term detection[4]. To deal with recognition errors, indexes have been constructed using words in a lattice/confusion matrix[5]. We use the continuous word recognition results and the syllable recognition results for the target documents in our method. Hence, the proposed method maps a target document to multiple vector spaces, because it is possible to choose multiple types of index terms, such as word- and syllable-based index terms, for constructing vector spaces. Because the proposed method represents a document in different ways in multiple spaces, we can calculate the distance between the document vector and the query vector in each space. As we see it, each vector space represents different information about the documents. If these different kinds of information are combined efficiently, retrieval performance can be improved. Therefore, in this paper, we propose a distance combination method to leverage complementary information.

However, it is difficult to retrieve relevant documents using the query vector if the number of index terms in the query is small. To increase the number of index terms in the query, query expansion methods have been proposed[6]. Query expansion methods often collect related words from the Internet and construct a new query vector using the original query vector and the collected word vector (which we call an "expansion vector") using linear combination. In this paper, we propose a method to model the target documents as a hyperplane. This means that the weight parameter arbitrarily changes to fix a target document model.

Generally, morpheme information are not used for the weighting index because the weight of an index is calculated individually for each index. We assume that the priority of morphemes is different for each query. Hence, we propose a morpheme weighting method for spoken document retrieval.

In the STD method, the sub-sequences extracted from the target document are converted into the bit sequences using the hash function. A query is also converted into a bit sequence in the same way. Candidates are detected by calculating hamming distance between the bit sequence of the query and that of the target documents. Then, the STD method calculates the distances between the query and the candidates using DP matching.

Using the SpokenDoc task from NTCIR-9, we conducted STD and SCR experiments to determine the optimal parameters for our method. Using these parameters, we submitted the results for the SpokenDoc2 tasks at NTCIR-10.

2. SCR METHOD USING DISTANCE COMBINATION

With our SCR method, first the target documents are mapped into the multiple vector spaces, one of which is based on continuous-word speech recognition results and the other on continuous-syllable speech recognition results, using a VSM. We then apply LSI to the syllable-based vector space to reduce the dimensionality of the space. The proposed method also applies query expansion and morpheme weighting to the word-based document space. Finally, our method combines the distances between the query and target documents, calculated in both vector spaces, and compares these total distances to rank the documents. In the following sections, we describe the details of our method.

2.1 Mapping documents to multiple vector spaces

A VSM maps the target documents and query to vector spaces constructed using the index terms. In this paper, we use multiple kinds of index terms (both word-based and syllable-based) because the target documents are speech recognition results, and thus they are noisy due to speech recognition errors. Multiple levels of index terms are expected to have complementary effects on retrieval. We use two methods for index term weighting:

- TF-IDF weight

  Term Frequency - Inverse Document Frequency (TF-IDF) weight is calculated using the following equation:

  $$d_{ij} = \log\left(\frac{D}{D_j}\right)\sum_{n=1}^{N}\frac{T_{ij}^n}{n}, \quad (1)$$

  where d_{ij} indicates the value of the jth index term in the ith document. T_{ij}^n is the term frequency of the jth index term in the ith document which is the nth speech recognition candidate. D and D_j indicate the total number of documents and the number of documents in which the jth term index appears.

- Binary weight

  Binary weight of index terms is calculated using the following equations:

  $$d_{ij} = 1 \text{ if } \sum_{n=1}^{N} T_{ij}^n > 0, \quad (2)$$

  $$d_{ij} = 0 \text{ if } \sum_{n=1}^{N} T_{ij}^n = 0. \quad (3)$$

  With this weighting method, the index term weight is 1 when the term appears in a document. If the term does not appear in a document, the term weight is 0.

By using these index term weights for constructing the vector spaces, the target document can be mapped on multiple vector spaces using a VSM.

2.2 Latent Semantic Indexing in syllable-based vector space

The target document is represented as a vector in a vector space consisting of index terms. Target document vectors are high-dimensional and sparse because the number of index terms is large, therefore they are sensitive to noise, and it is also difficult to capture the underlying semantic structure. Hence, we apply LSI to the vector spaces to overcome these problems. LSI finds a latent semantic space (a concept space) of low-dimension by performing singular value decomposition on the original vector space, and then retrieves documents in the latent semantic space[1][2].In this paper, we apply LSI not to the word-based document space, but to the syllable-based space. Syllable sequences themselves do not have meanings, but LSI is expected to distill the implicit semantics from the co-occurrences of sequences.

2.3 Query Expansion

Generally, it is difficult to retrieve relevant documents if the number of index terms in the query is small. Therefore,

query expansion is often used in order to increase the number of index terms in the query. To expand the query, first our method performs a morphological analysis of the query sentence and selects nouns from the morphological analysis results. Then, using the selected nouns, we search for web pages from the Internet using a search engine, and collect web pages related to the query. Then, using the collected web pages, an expansion vector is constructed in the same manner as the document vector.

Expanded query vector $\hat{\boldsymbol{q}}$ used for retrieval is calculated using the following equation:

$$\hat{\boldsymbol{q}} = (1-\alpha)\boldsymbol{q}_o + \alpha \boldsymbol{q}_e, \tag{4}$$

where $\boldsymbol{q}_o$ and $\boldsymbol{q}_e$ are the original query vector constructed from the query sentence, and the expansion vector constructed from the collected web pages, respectively. α is the weight parameter between the original query vector and the expansion vector.

Although Eq. (4) is similar to the Rocchio-based relevant feedback method[6], our query expansion method uses information from web pages which are searched using a query instead of utilizing user feedback. In the Rocchio-based feedback method, it is important to determine the weight parameter in order to improve retrieval performance. Usually, the value used for α is fixed a priori for all the target documents. This means that the typical target document is modeled using a vector which is a linear sum of $\boldsymbol{q}_o$ and $\boldsymbol{q}_e$ using a fixed weight. This model, however, seems to be too constrained. To relax the constraint, we propose to model the typical target document set as a hyperplane spanned by $\boldsymbol{q}_o$ and $\boldsymbol{q}_e$. Using this model, we evaluate the relevance of document vector $\boldsymbol{d}_i$ by measuring the angle between $\boldsymbol{d}_i$ and the hyperplane. This approach is illustrated in Fig. 1. This method determines the weight parameter by minimizing the distance between an expanded query vector on the hyperplane and each document vector. To obtain the expanded vector, the weight parameter of query expansion $\alpha_{(\hat{\boldsymbol{q}}_m, \boldsymbol{d}_i)}$ is calculated using the following formulas:

$$\alpha_{\{\hat{\boldsymbol{q}}, \boldsymbol{d}_i\}} = \mathrm{argmax}_\alpha \left(\cos(\theta_{\{\hat{\boldsymbol{q}}, \boldsymbol{d}_i\}}) \right) \tag{5}$$

$$= \mathrm{argmax}_\alpha \left(\frac{\hat{\boldsymbol{q}}^T \boldsymbol{d}_i}{|\hat{\boldsymbol{q}}||\boldsymbol{d}_i|} \right) \tag{6}$$

$$= \mathrm{argmax}_\alpha \left(\frac{((1-\alpha)\boldsymbol{q}_o + \alpha \boldsymbol{q}_e)^T \boldsymbol{d}_i}{|((1-\alpha)\boldsymbol{q}_o + \alpha \boldsymbol{q}_e)||\boldsymbol{d}_i|} \right), \tag{7}$$

$\alpha_{(\hat{\boldsymbol{q}}_m, \boldsymbol{d}_i)}$ is calculated using following equation:

$$\begin{aligned} \alpha_{(\hat{\boldsymbol{q}}_m, \boldsymbol{d}_i)} = & -(2((\boldsymbol{q}_o \cdot \boldsymbol{q}_o)(\boldsymbol{d}_i \cdot \boldsymbol{d}_i))((\boldsymbol{q}_e - \boldsymbol{q}_o) \cdot \boldsymbol{d}_i) - \\ & (2(\boldsymbol{q}_o \cdot (\boldsymbol{q}_e - \boldsymbol{q}_o))(\boldsymbol{d}_i \cdot \boldsymbol{d}_i))(\boldsymbol{q}_o \cdot \boldsymbol{d}_i)) \\ & (2(\boldsymbol{q}_o \cdot (\boldsymbol{q}_e - \boldsymbol{q}_o))(\boldsymbol{d}_i \cdot \boldsymbol{d}_i)(\boldsymbol{q}_e - \boldsymbol{q}_o) \cdot \boldsymbol{d}_i - \\ & (\boldsymbol{q}_e - \boldsymbol{q}_o) \cdot (\boldsymbol{q}_e - \boldsymbol{q}_o)(\boldsymbol{d}_i \cdot \boldsymbol{d}_i)(\boldsymbol{q}_o \cdot \boldsymbol{d}_i)))^{-1}. \end{aligned} \tag{8}$$

2.4 Morpheme weighting method

In general, the term weight of a document vector is calculated using the TF-IDF method, and so on, described above. However, these term weighting methods do not consider morpheme information. We believe that the appropriate weight of morphemes depends on the type of query, therefore we propose a morpheme weighting method in this paper. The proposed method calculates the weight for each morpheme using training data. Then, using these weights, the distance between query $\boldsymbol{q}$ and document $\boldsymbol{d}_i$ is calculated using the following formula:

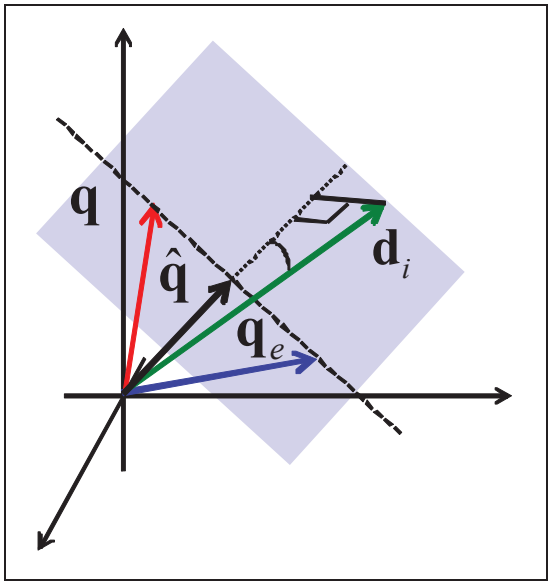

Figure 1: Relevant document modeling

$$S_{\{\boldsymbol{d}_i, \boldsymbol{q}\}} = \frac{\left(\sum_k w_k \boldsymbol{q}_k\right) \cdot \left(\sum_k w_k \boldsymbol{d}_{ik}\right)}{|\sum_k \boldsymbol{w}_k \boldsymbol{q}||\sum_k \boldsymbol{w}_k \boldsymbol{d}_k|}, \tag{9}$$

where w_k indicates the weight of the kth morpheme, and $\boldsymbol{q}_k$ and $\boldsymbol{d}_{ik}$ are the query vector of the kth morpheme, and the document vector of the kth morpheme, respectively.

2.5 Combination of distance

In our SCR method, there are different types of VSMs (word-level VSMs, with or without query expansion, and syllable-level VSMs), and different types of term weighting methods (TF-IDF and Binary Weight). The documents retrieved using each method are different because each method uses different information from the documents. Therefore, we combine these methods in order to retrieve more relevant documents. In this paper, we specifically propose a spoken document retrieval method combining query expansion with continuous syllable recognition.

The proposed method combines the distances calculated by the VSMs as shown in the following equation:

$$S_{\{\boldsymbol{d}_i, \boldsymbol{q}\}} = \sum_k \beta_k \cos_k(\theta_{\{\boldsymbol{d}_i, \boldsymbol{q}\}}), \tag{10}$$

where $\cos_k()$ indicates the cosine distance of VSM$_k$, and β_k is the weight parameter of each cosine distance. Using the distance calculated with Eq. (10), the proposed method ranks and retrieves the documents. We believe that the proposed method retrieves more relevant documents than conventional methods because the proposed method uses a larger amount of information, namely word and syllable recognition results, for spoken document retrieval.

2.6 Experiments using NTCIR-9 SpokenDoc task

2.6.1 Experimental conditions

In order to evaluate the proposed method, we conducted a spoken content retrieval experiment under the conditions of the SpokenDoc task of NTCIR-9[7]. For the target documents in this experiment, we used the speech recognition results provided by the NTCIR-9 organizer, and for the queries we used the dry run queries from NTCIR-9.

As index terms for the word-based VSM, we chose words whose morphemes are nouns, alphabet sequences and katakana sequences from a vocabulary list used in a continuous word recognizer. The number of these index terms was 14,716. As index terms for the syllable-based VSM, we used syllable 3-grams obtained from continuous syllable recognition results. The number of these index terms was 169,363. Binary weighting and TF-IDF were used as the index term weighting methods for the word-based VSM, and TF-IDF was used as the index term weighting method for the syllable-based VSM. For the syllable-based VSM, the queries were translated into Japanese katakana sequences. For query expansion, using the nouns from each query we obtained web pages using Google's search engine. The index terms selected from these web pages, using Google JSON/Atom Custom Search API[8], were used for calculating the expansion vector ($\boldsymbol{q}_e$ in Eq. (4)). For morpheme weighting, we used common noun, proper noun, and numeral. We used Mean Average Precision (MAP) as the measure for evaluation[7].

2.6.2 Evaluation of query expansion

First, we evaluate the effectiveness of query expansion. Figure 2 shows the MAP scores with different values for the weight parameter of query expansion. In this figure, "α(Val)" indicates the MAP score of our query expansion method, which calculates weight parameter α using Eq. (8), under the condition that the number of web pages for query expansion and the number of speech recognition candidates for the VSM are 17 and 5, respectively. "α(Fix)" indicates the MAP score of a conventional query expansion method, which uses fixed weight parameters for all documents, under the condition that the number of web pages for query expansion and the number of speech recognition candidates for VSM are 26 and 2, respectively. The heading "w/o QE" indicates the MAP score without the query expansion method, under the condition that the number of speech recognition candidates for the VSM is 1. These parameters showed the highest MAP scores in our preliminary experiment.

Fig. 2 shows that both of the query expansion methods, "α(Fix)" and "α(Val)", improve the MAP score of "w/o QE". Hence, we demonstrate that query expansion is useful for spoken document retrieval. Comparing "α(Fix)" and "α(Val)" in this figure, we can see that the MAP score of the query expansion method using a fixed weight parameter is higher than that of the query expansion method using the variable weight parameter. The advantage of the query expansion method with the variable weight parameter, however, is that the weight parameter does not need to be calculated using training data. Hence, we believe that this method is flexible and able to support variable conditions. We will investigate the details of these results in the future.

2.6.3 Evaluation of LSI

Figure 3 shows the results of applying LSI to the syllable-based VSM. In this experiment, we used the best speech recognition candidate for constructing the syllable-based VSM.

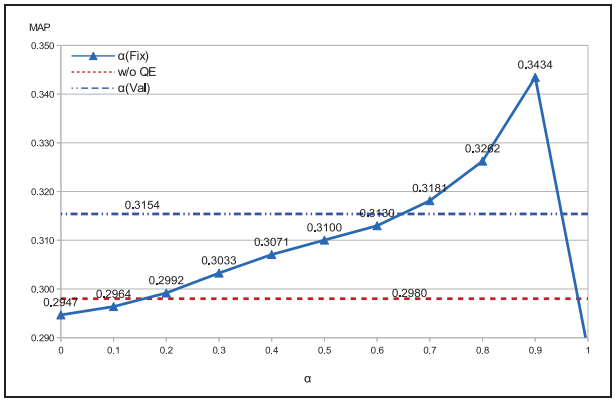

Figure 2: Results of query expansion

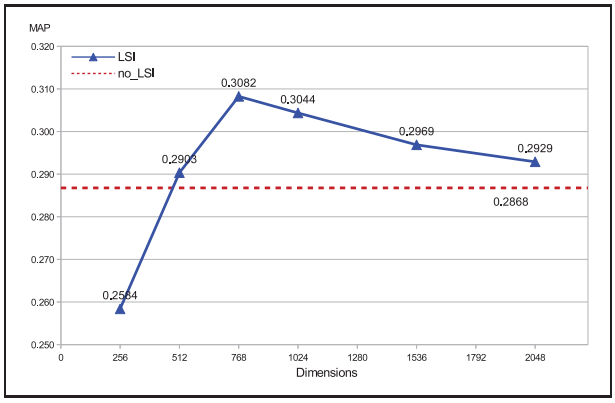

Figure 3: Results of LSI

From this figure, we can see that LSI improved retrieval performance by reducing the dimensions of the document vector. The number of dimensions of the original syllable-based vector space was 169,363.

Because the document vector constructed using the syllable-based VSM is high dimensional and sparse, there was the possibility of the documents containing noise. Therefore, dimension reduction using LSI reduced the influence of this noise and improved retrieval performance.

2.6.4 Evaluation of distance combination

This section describes experimental results regarding our distance combination method. From our NTCIR-9 results[9], distance combination was shown to be valuable for improving retrieval performance. Hence, we combine the distances calculated in the binary-weighted, word-based VSM; the TF-IDF-weighted, word-based VSM with query expansion ($\alpha = 0.9$); and the TF-IDF-weighted, syllable-based VSM. The combined distances were calculated using the following equation:

$$S_{\{\boldsymbol{q},\boldsymbol{d}_i\}} = (1-\beta_1-\beta_2)s_b(\boldsymbol{q}^b,\boldsymbol{d}_i^b) + \beta_1 s_e(\boldsymbol{q}^e,\boldsymbol{d}_i^e) + \beta_2 s_s(\boldsymbol{q}^s,\boldsymbol{d}_i^s), \quad (11)$$

where $s_b()$, $s_e()$ and $s_s()$ indicate the cosine similarity calculated in the binary-weighted, word-based VSM; the TF-

Table 1: Results for distance combination

method	$\beta 1$	$\beta 2$	MAP
Word-based VSM with QE	1.0	0.0	0.343
Distance combination w/o LSI	0.3	0.2	0.389
Distance combination w LSI	0.2	0.1	0.394

IDF-weighted, word-based VSM with query expansion; and the TF-IDF-weighted, syllable-based VSM, respectively. In the previous section, we discovered that LSI improved retrieval performance in the syllable-based document space. Hence, we used two types of distances, with and without LSI, calculated from the syllable-based VSM.

Experimental results are shown in Table 1, which shows the best MAP scores for various values of β_1 and β_2. From this table we can see that distance combination improves retrieval performance.

2.7 FormalRun results for NTCIR-10 SpokenDoc2 task

In this section, we describe the SCR method we used for the to SCR subtask of SpokenDoc2 at NTCIR-10. The experimental conditions are the same as those described in Section **2.6**. The details of the experimental conditions and the queries are described in the organizer's overview paper[10].

In this experiment, we evaluated nine methods which combine distances calculated using three methods. These three methods are the TF-IDF-weighted, syllable-based VSM; the binary-weighted, word-based VSM; the TF-IDF-weighted, word-based VSM with query expansion. The distances used for retrieving the documents are calculated using Eq. (11). The details of the parameters and MAP scores for each method are shown in Table 2. In this table "LSI" means that LSI is applied to the TF-IDF-weighted, syllable-based VSM. After LSI was applied, the number of dimensions of the vector spaces was 768. "α(Val)" indicates the query expansion method which calculates the weight parameter using Eq. (8), and "$\alpha = 0.9$" represents the query expansion method which uses a fixed weight parameter ($\alpha = 0.9$). "MS" indicates the morpheme weighting method described in Section **2.4**. "MS(Fix)" is the morpheme weighting method which uses fixed weighting parameters for the morphemes (common noun, proper noun, numeral = 0.5, 0.4, 0.1, respectively). Finally, "MS-LSI" is the morpheme weighting method which calculates weighting parameters in a low-dimensional vector space reduced by LSI. In addition, this table contains the MAP scores of the baseline method provided by the NTCIR-10 organizer.

From this table we can see that all of our methods significantly improved MAP scores compared to the baseline method.

3. SPOKEN TERM DETECTION

3.1 STD using hash function

Figure 4 illustrates the proposed STD method. In our STD method, the sub-sequences extracted from target documents are converted into bit sequences by using the hash function. Hence, the target documents can be represented by a compact bit sequence. Because the length of the sub-sequence corresponds to the length of the query, we construct the multiple bit sequences from a sub-sequence the same length as the query. Queries are also converted into bit sequences in the same way. Candidates are detected by calculating the hamming distance between the bit sequence of a query and that of the target documents. Finally, the proposed method calculates the distances between the query and the candidates using DP matching and detects the results. To calculate the detection score, we use the following formula[11]:

$$score = \frac{1}{T/l^{3/2} + 1}, \tag{12}$$

where l and T indicate the length of the keyword, and the threshold, respectively. Because the proposed method only uses two kinds of bit operations in the detection process (which are XOR and popcount), we can obtain the detection results quickly.

Table 2: Retrieval results of formal run query set

Priority	VSM syllable	VSM word	β_1	β_2	MAP
1	LSI	$\alpha = 0.9$	0.3	0.1	0.3681
2	LSI	α(Val)	0.3	0.1	0.3681
3	nonLSI	$\alpha = 0.9$	0.4	0.2	0.3920
4	nonLSI	α(Val)	0.3	0.3	0.3721
5	nonLSI	MW	0.3	0.3	0.3754
6	nonLSI	MW(Fix)	0.3	0.3	0.3700
7	noLSI	MW-LSI	0.3	0.3	0.3625
8	nonLSI	$\alpha = 0.9$	0.4	0.1	0.3834
9	LSI	$\alpha = 0.9$	0.4	0.1	0.3815
Baseline (SMART)					0.2679
Baseline (TF-IDF)					0.2310

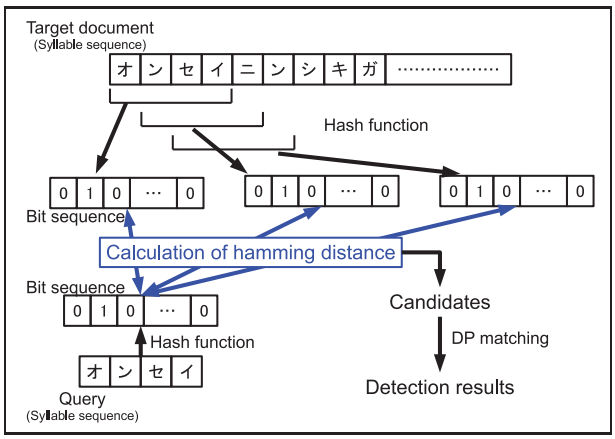

Figure 4: The proposed STD method

3.2 Experiments with NTCIR-9 SpokenDoc task

In order to evaluate the proposed method, we conducted a spoken term detection experiment under the conditions of the SpokenDoc task of NTCIR-9[7]. For the target documents we used the speech recognition results provided by the NTCIR-9 organizer. For the queries in this experiment we used ALL query sets from the dry run of the SpokenDoc task at NTCIR-9.

Experimental results are shown in Table 3. In this table "#. syllable" and "#. word" are the number of syllable recognition candidates, and the number of word recogni-

Table 3: STD results of ALL query sets at NTCIR-9 (F-measure)

	#. syllable		
#. word	1	2	3
0	46.2	47.7	47.7
1	56.1	56.2	55.6
2	**56.6**	56.6	56.0
3	56.4	56.4	55.8
4	56.4	56.4	55.8
5	56.4	56.4	55.8
6	56.4	56.4	55.8
7	56.3	56.3	55.7
8	56.3	56.3	55.7
9	56.3	56.3	55.7
10	56.4	56.4	55.8

tion candidates, respectively. This table shows that STD performance is improved by using both word and syllable recognition results.

3.3 FormalRun results for NTCIR-10 SpokenDoc2 task

We participated in three STD tasks of SpokenDoc2 at NTCIR-10; the STD large-size task, the STD moderate-size task, and the iSTD task. In this section, we describe the STD methods used for these tasks. The details of the experimental conditions and the queries are described in the organizer's overview paper[10]. In this section, we compared our methods with the baseline method described in [10].

3.3.1 Experimental results of STD large-size task

Table 4 shows the parameters of our STD methods. "Pri." is priority number of each method whose results were submitted to NTCIR-10. "#. cand." indicates the number of speech recognition candidates used for the construction of the bit sequences. "M." and "U." represent the "Matched" and "Unmatched", respectively. "Thresh." indicates the threshold of proposed method. Table 5 shows the experimental results of the formal run for SpokenDoc2 at NTCIR-10. This table also contains the experimental results of the baseline method provided by the NTCIR-10 organizer.

From Table 5 we can see that "Pri. 9" is our best performing method. This method used our own transcription, which was not speech recognition results, but the correct answers for the queries from NTCIR-9, which are same as those for the SpokenDoc2 task at NTCIR-10. Therefore, its performance was the highest.

Excepting the "Pri. 9" method with special information, we can see that the methods which use all of the transcriptions provided by the organizer (Priority 1, 2, and 3) have improved performance compared to the baseline methods. From these results we can see that information from multiple transcriptions is useful for STD. The "max. F." score of "Pri. 2" is the highest of the methods submitted for the large-size task, excepting "Pri. 9".

3.3.2 Experimental results of STD moderate-size task

Table 6 shows the parameters of the methods we used for the STD moderate-size task. The abbreviations in this table are the same as in Table 4. The experimental results for the STD moderate-size task of SpokenDoc2 at NTCIR-10 are

Table 4: Parameter on STD large-size task

	Transcription							
	Syllable		Word			#. cand.		
Pri.	M.	U.	M.	U.	OWN	Syl.	Word	Thresh.
1	O	O	O	O	X	2	2	0.890
2	O	O	O	O	X	2	2	0.910
3	O	O	O	O	X	1	1	0.890
4	O	X	X	X	X	1	0	0.890
5	O	X	O	X	X	1	1	0.890
6	O	X	O	X	X	2	2	0.890
7	O	O	O	O	X	0	1	0.890
8	O	X	O	X	X	0	10	0.890
9	O	X	O	X	O	2	2	0.890

Table 5: Experimental results on STD large-size task

	macro ave.		search
Pri.	max. F [%]	MAP	speech [s]
1	60.33	0.553	0.0848
2	63.63	0.551	0.0881
3	60.43	0.548	0.0439
4	41.38	0.324	0.0128
5	47.27	0.391	0.0131
6	48.07	0.408	0.0283
7	42.26	0.357	0.000791
8	28.06	0.224	0.00154
9	85.39	0.690	0.0164
BL-1	43.91	0.500	560
BL-2	47.13	0.507	560
BL-3	46.79	0.532	506

shown in Table 7. This table shows that the F-measures of the methods which use whole transcriptions (Priority 1, 2, and 3) are high. From these results, we conclude that the combination of multiple transcriptions is useful for spoken term detection. However, comparing "Pri. 1, 2, and 3", we can see that detection performance is not influenced by the use of multiple speech recognition candidates.

3.3.3 Experimental results of iSTD task

Table 8 and Table 9 show the parameters of the methods we used and their experimental results for the iSTD task at NTCIR-10. The abbreviations in these tables are the same as in Table 4. Table 9 shows that differences in the threshold slightly affect iSTD performance. We can also see that the F-measures of our method are higher than those of the baseline methods, although Recall and Precision are almost the same as those of baseline methods.

4. SUMMARY

In this paper we described spoken content retrieval (SCR) and spoken term detection (STD) methods which were used in the 2nd round of the IR for Spoken Documents (SpokenDoc2) task at NTCIR-10. Our SCR method extracted the semantic axes by applying latent semantic indexing (LSI) to reduce the dimensions in the syllable-based space. On the other hand, in the word-based space, our SRC method applied query expansion and morpheme weighting. Finally, the distance between a query and a document was calculated by combining the distances calculated in each vector space. To determine the parameters of our SCR method,

Table 6: Parameter on STD moderate-size task

	Transcription							
	Syllable		Word			#. cand.		
Pri.	M.	U.	M.	U.	OWN	Syl.	Word	Thre.
1	O	O	O	O	X	2	2	0.890
2	O	O	O	O	X	2	2	0.910
3	O	O	O	O	X	1	1	0.890
4	O	X	X	X	X	1	0	0.890
5	O	X	O	X	X	1	1	0.890
6	O	X	O	X	X	2	2	0.890
7	X	O	X	O	X	2	2	0.890
8	O	X	O	X	X	-	10	0.890

Table 7: Experimental results on STD moderate-size task

	macro ave.		search
Pri.	max. F [%]	MAP	speech [s]
1	40.70	0.336	0.0425
2	39.11	0.318	0.0430
3	39.14	0.321	0.0218
4	23.23	0.170	0.0087
5	33.27	0.264	0.0090
6	34.11	0.273	0.0179
7	30.53	0.239	0.0175
8	24.23	0.183	0.0010
BL-1	25.72	0.317	30.8
BL-2	31.43	0.358	31.9
BL-3	33.73	0.393	30.8

we conducted spoken document retrieval experiments using the dry run queries from the SpokenDoc task at NTCIR-9. Using these parameters, we submitted our results for the SpokenDoc2 task at NTCIR-10. From the formal run results of the SCR task of SpokenDoc2 at NTCIR-10, which were provided by the NTCIR-10 organizer, our proposed method improved the mean average precision score compared to the baseline method.

We also described our STD method, which converted subsequences extracted from the target document and the query into bit sequences using a hash function. The candidate locations of the demanded terms were detected by calculating the hamming distance between the bit sequence of the query and those of the target documents. Finally, our method calculated accurate distances among candidates using DP-matching. To evaluate our STD method and determine the best performing parameters, we conducted spoken term detection experiments using the STD task from SpokenDoc at NTCIR-9. Using these parameters, we submitted our results for three STD sub-tasks of SpokenDoc task for NTCIR-10. Our method achieved the highest performance for the STD large-size task.

In future work, we will investigate the details of our experimental results and use this information to improve our methods.

Table 8: Parameter on iSTD task

	Transcription							
	Syllable		Word			#. cand.		
Pri.	M.	U.	M.	U.	OWN	Syl.	Word	Thre.
1	O	X	O	X	X	2	2	0.890
2	O	X	O	X	X	2	2	0.900
3	O	X	O	X	X	2	2	0.910
4	O	X	O	X	X	2	2	0.882
5	O	X	O	X	X	2	2	0.920
6	O	X	O	X	X	2	2	0.880
7	X	X	O	X	X	2	2	0.870
8	O	X	O	X	X	2	2	0.860
9	O	X	O	X	X	2	2	0.840

Table 9: Experimental results on iSTD task

	Maximum		
Pri.	Recall	Precision	F-measure
1	88.00	73.33	80.00
2	88.00	73.33	80.00
3	88.00	73.33	80.00
4	88.00	73.33	80.00
5	90.00	70.31	78.95
6	88.00	73.33	80.00
7	88.00	73.33	80.00
8	88.00	73.33	80.00
9	88.00	73.33	80.00
BL-1	73.00	76.04	74.49
BL-2	88.00	69.84	77.88
BL-3	90.00	68.18	77.59

5. REFERENCES

[1] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[2] T. Landauer and S. Dumais. A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. 104:211–240, 1997.

[3] V. T. Turnen. Reducing the effect of OOV query words by using morph-based spoken document retrieval. *Proc. of Interspeech*, pages 2158–2161, 2008.

[4] K. Iwata, K. Shinoda, and S. Furui. Robust spoken term detection using combination of phone-based and word-based recognition. *Proc. of Interspeech*, pages 2195–2198, 2008.

[5] M. Saraclar and R. Sproat. Lattice-based search for spoken utterance retrieval. *HLT-NAACL*, pages 129–136, 2004.

[6] J. Rocchio. Relevance feedback in information retrieval. *Salton G. (Ed.), The SMART Retrieval System. Englewood Cliffs, N.J.: Prentice Hall*, pages 313–323, 1971.

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

[8] JSON/atom custom search API. http://code.google.com/apis/customsearch/v1/overview.html.

[9] S. Tsuge, H. Ohashi, N. Kitaoka, K. Takeda, and K. Kita. Spoken document retrieval method combining query expansion with continuous syllable recognition for NTCIR-spokendoc. *Proceedings of the 9th NTCIR Workshop Meeting*, pages 249–256, 2011.

[10] T. Akiba, H. Nishizaki, K. Aikawa, X. Hu, Y. Itoh, T. Kawahara, S. Nakagawa, H. Nanjo, and Y. Yamashita. Overview of the NTCIR-10 SpokenDoc-2 task. *Proceedings of the 10th NTCIR Workshop Meeting*.

[11] K. Katsurada, K. Katsuura, Y. Iribe, and T. Nitta. Utilization of suffix array for quick std and its evaluation on the NTCIR-9 spokendoc task. *Proceedings of the 9th NTCIR Workshop Meeting*, pages 271–274, 2011.