STD based on Hough Transform and SDR using STD results: Experiments at NTCIR-9 SpokenDoc

Taisuke Kaneko, Tomoko Takigami and Tomoyosi Akiba
Toyohashi University of Technology
{kaneko | takigami | akiba}@nlp.cs.tut.ac.jp

ABSTRACT

In this paper, we report our experiments at NTCIR-9 IR for Spoken Documents (SpokenDoc) task. We participated both the STD and SDR subtasks of SpokenDoc. For STD subtask, we applied novel indexing method, called metric subspace indexing, previously proposed by us. One of the distinctive advantages of the method was that it could output the detection results in increasing order of distance without using any predefined threshold for the distance. The experimental results showed that the proposed method was very fast but there were rooms for improvement in the detection accuracy. For SDR subtask, two kinds of approaches were applied to both the lecture and passage level. The first approach used the conventional word-based IR methods based on the language modeling IR models. The second approach used the STD method for detecting the terms in the query from the spoken documents and then applied the IR methods using the detection as the term's appearances. The experimental results showed that, though the performance of the STD-based method was lower than the word-based approaches in total, it could improve the performance if the query topic included the out-of-vocabulary words.

Keywords

spoken term detection, spoken document retrieval, metric subspace indexing, query expansion, relevance models, [AKBL] [Spoken Term Detection] [Spoken Document Retrieval] [Japanese]

1. INTRODUCTION

In this paper, we report our experiments at NTCIR-9 IR for Spoken Documents (SpokenDoc) task[1], where our previously proposed methods for STD and SDR are applied. We submitted nine runs in total, which were two runs for the CORE set of the STD subtask, four runs for the lecture retrieval task of the SDR subtask, and three runs for the passage retrieval task of the SDR subtask.

For STD subtask, we applied novel indexing method, called metric subspace indexing, previously proposed by us [2]. The proposed method can be considered as using metric space indexing for approximate string matching problem, where the distance is defined between an indexing unit, e.g. a phoneme or a syllable, and a position in the target spoken document. The proposed method can also be considered as applying the Hough transform, an algorithm for detecting straight lines in a given visual image, to the STD task. The most attractive advantage of the proposed method is that it does not need a threshold for the distance used to make the decision about whether the detected term is adopted or not. It can simply output the detection results in increasing order of distance. We applied the vanilla implementation and it's extension for multiple recognition candidates for submitting our runs.

For the SDR subtask, two kinds of approaches were applied to both the lecture and passage level, which were the conventional word-based approach and the STD-based approach. The word based approach uses the word-based speech recognition results for indexing the spoken documents. A text based IR method can be applied to the transcribed text obtained by the speech recognition. The most major problems of this approach lies in that the out-of-vocabulary words of the word-based speech recognition never appear in the text and thus cannot become clues for retrieval. To overcome the problem, we applied relevance models, a query expansion method, for the spoken document passage retrieval task [3]. In this paper, we applied the method to both lecture and passage retrieval of the SpokenDoc SDR subtask.

The second approach, STD-based SDR method, is based on the syllable-based speech recognition results. In the first step, a STD method is applied to the spoken documents, where each term in the given query topic is searched against the syllable sequence obtained by the speech recognition. From the detection results, we can obtain the statistics of the term frequencies for each document, to which we can apply any conventional document retrieval method. The advantage of this method is that it is not affected by the OOV terms in the query topics.

The remainder of this paper is organized as follows. Section 2 describes our STD method used for the STD subtask and its experimental evaluation. Section 3 describes our two approaches for the SDR subtask and their evaluation. In Section 4, we conclude our experiments at SpokenDoc.

2. STD SUBTASK

Most conventional methods for STD use Large Vocabulary Continuous Speech Recognition (LVCSR) to transcribe the target spoken document into textual form and then apply a text-based search method to find the positions in the text where the query term appears. Within this framework, many previous STD works simply use Dynamic Time Warping algorithm for the search, while some others apply text-based indexing method to improve time efficiency. However,

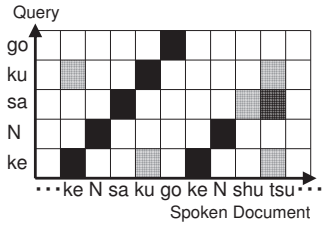

Figure 1: *STD as straight line detection*

most of the previous indexing methods for STD use binary indexing, which expresses only appearance or nonappearance, so that such methods need to calculate distances to filter out implausible results either during or after using the indexes for searching.

We previously proposed a novel indexing method for STD that was not those used for text-based indexing. The proposed method can be considered as using metric space indexing for approximate string matching problem [4], where the distance is defined between a phoneme and a position in the target spoken document. The proposed method can also be considered as applying the Hough transform, an algorithm for detecting straight lines in a given visual image, to the STD task.

The most attractive advantage of the proposed method is that it does not need a threshold for the distance used to make the decision about whether the detected term is adopted or not. It can simply output the detection results in increasing order of distance. Another advantage is that it can deal naturally with the multiple recognition candidates obtained by ASR, because it indexes the distance between a phoneme and a position instead of between phonemes.

In the next subsection (Section 2.1), we recast our metric subspace indexing method for STD. In Section 2.2, our experimental evaluation at the NTCIR-9 SpokenDoc task is described.

2.1 STD by Metric Subspace Indexing

Consider a plane where the x and y axes correspond to the syllable sequence of the spoken document obtained by using ASR and the syllable sequence of the input query, respectively (see Fig. 1). For each grid point on the plane, the distance between the syllable in the document at x and the syllable in the query at y is defined. The distance at a grid point is analogous to the pixel density at an image data point. Our proposed method divided into indexing process and detection process.

2.1.1 *Indexing Process*

Let I be the query length (number of syllables in the query), let J be the spoken document length (number of syllables in the spoken document), and let $D_{i,j} (0 \leq i < I, 0 \leq j < J)$ be the syllable distances defined at the grid point (i, j) on the plane. Then the STD problem can be recognized as the line detection problem on the plane and can be formulated as detecting the position j that has the minimum cumulative distance T_j , defined as follows.

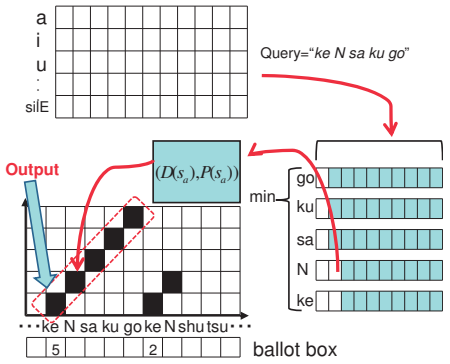

Figure 2: *Detection Process of our proposed method*

$$T_j = D_{0,j} + D_{1,j+1} + \cdots + D_{I-1,j+I-1} = \sum_{i=0}^{I-1} D_{i,i+j}$$

In the case of line detection in image data, the pixel densities can be processed only at detection time, because the target image data are not known in advance. In the case of STD, however, the distances $D_{i,j}$ can be processed beforehand, because the target spoken document is known in advance. Let $D(a)_j$ be the distance between a syllable a and the syllable that appears at position j in the target spoken document. Then, for each a, the syllable distance vector $[D(a)_0, D(a)_1, \ldots, D(a)_j, D(a)_{J-1}]$ can be calculated in advance. When a query is supplied, we can easily construct the xy plane by arranging the distance vectors along the y-axis according to the syllable sequence of the query, so that $D_{i,j} = D(a_i)_j$ for the query $a_0 a_1 \ldots a_{I-1}$. Here the metric space defined between the query string and every substrings in the target document is divided into the metric subspaces, each of which is defined between each syllable in the query and every positions in the document.

Furthermore, the syllable distance vector can be sorted in advance. We pair distance $D(a)_j$ with position j and make a vector $[(D(a)_0, 0), (D(a)_1, 1), \ldots, (D(a)_j, j), (D(a)_{J-1}, J-1)]$. Then we sort this vector according to the distance (the first item of each pair). We call this vector Sorted Distance Vector (SDV). Let S_a be the SDV of the syllable a. We handle S_a as a stack, where the stack top is the leftmost element of the vector. Using $S_a (a \in V)$ for the set of syllables V as the index, we can obtain a fast STD algorithm that can output the detection results in increasing order of distance.

2.1.2 *Detection Process*

Let $s_a = (D(s_a), P(s_a))$ be the top element of SDV S_a, where $D(s_a)$ and $P(s_a)$ are the distance and the position recorded in the paired element s_a, respectively. The detection process is as follows (Fig. 2).

1. According to the query syllable sequence $a_0, a_1, \ldots, a_i, \ldots, a_{i-1}$, prepare the SDVs $S_{a_0}, S_{a_1}, \ldots, S_{a_i}, \ldots, S_{a_{i-1}}$. Initialize the counter (ballot box) $C[j] = 0 (0 \leq j < n)$ and the candidate set $U = \phi$.

2. Pop the top element s_{a_i} of S_{a_i}, that has the minimum distance $min_i\{D(s_{a_i})\}$ from the set that comprises the top elements $s_{a_0}, s_{a_1}, \ldots, s_{a_i} \ldots, s_{a_{i-1}}$ of the SDVs $S_{a_0}, S_{a_1}, \ldots, S_{a_i}, \ldots, S_{a_{i-1}}$. Let $j = P(s_{a_i}) - i$ and add 1 to $C[j]$ (voting).

3. If $C[j] \geq k$, then add the position j to the set U.

4. Output the subset V of U that satisfies a certain condition. Let $U \leftarrow U - V$.

5. Repeat Steps 2,3 and 4 until a certain condition is satisfied.

The simplest version of the above algorithm is to set $k = I$(the query length) at Step 3 and to impose no condition at Step 4. Note that this algorithm does not need a threshold for distances. It outputs the detection results approximately in the order of smaller to larger distances [1]. We could use different conditions at Step 5, such as "until it finds the first result", "until it finds the N-best results", "until it passes a certain period", and, of course, "until the distance exceeds a certain threshold."

Step 2 needs definitely $I - 1$ comparisons, which are not efficient. Therefore, in this paper, we refine our algorithm by inserting the following Step 2.5 between Step 2 and 3.

2.5 Let S be the set of the elements in S_{a_i} that have the same distance as that of the element s_{a_i} popped at the last step (i.e. $S = \{s | s \in S_{a_i} \wedge D(s) = D(s_{a_i})\}$). For each $s \in S$, let $j = P(s) - i$ and add 1 to $C[j]$. Let $S_{a_i} \leftarrow S_{a_i} - S$.

Notice that such elements S are at the top of the sorted stack S_{a_i} if they exist, so we can efficiently select S from S_{a_i}.

2.1.3 Dealing with Multiple Recognition Candidate

The recall of the detection can be improved by considering the multiple candidates from the speech recognition [5]. The proposed method can cope easily with the sausage-like representation of multiple recognition candidates, similarly to the Confusion Network [6] and TALE [7], by redefining the syllable distance $D(a)_j$ as follows.

$$D(a)_j = \min_{b \in B_j} \{D_{max} - w(b)(D_{max} - d(a,b))\} \qquad (1)$$

where B_j is the set of multiple recognition candidates at the position j in the target spoken document, $w(b)$ is the confidence weight of the candidate $b \in B_j$, and $d(a,b)$ is the distance between syllable a and b.

2.2 Experiments

We submitted two runs for the CORE set of the STD subtask. In this subsection, we describe the results at NTCIR-9 SokenDoc STD subtask formal run for CORE set.

[1]This simplest algorithm does not guarantee to output the results precisely in order of distance. However, the more complex version of the algorithm, which sets $k = 1$ at Step 3 and imposes $V = \{j | T_j < \sum_{i=0}^{I-1} D(s_{a_i})\}$ at Step 4, can output the results precisely in order of distance.

1best:	a	b	ϵ	ϵ	ϵ	c	d		a	b	c	d
2best:	ϵ	e	f	g	h	i	j	$\rightarrow$	k	e	i	j
3best:	k		l	m		n	ϵ				n	

Figure 3: *Example of Confusion Network Transformation*

2.2.1 Processing Unit and Distance Measure

We use the syllable based transcription provided by SpokenDoc task organizers. Syllable is used as the processing unit and Bhattacharyya distance between acoustic models are used as then distance measure. Bhattacharyya distance measures the dissimilarity between two probability distributions. In this work, we use syllable HMM for our acoustic model. The distance between two HMMs a and b is defined as follows based on Bhattacharyya distance.

$$d(a,b) = \frac{1}{M} \sum_{\alpha=1}^{M} \min_{\beta,\gamma} BD\{P_a(S_a^\alpha, \beta), P_a(S_a^\alpha, \gamma)\} \qquad (2)$$

$$BD(P_a, P_b) = \frac{1}{8}(\mu_a - \mu_b)^T \{\frac{\Sigma_a + \Sigma_b}{2}\}^{-1}(\mu_a - \mu_b) + \frac{1}{2}\ln(\frac{|(\Sigma_a + \Sigma_b)/2|}{|\Sigma_a|^{\frac{1}{2}}|\Sigma_b|^{\frac{1}{2}}}) \qquad (3)$$

where S_a^α is α-th state of the HMM of the syllable a, $P(S_a^\alpha, \beta)$ is β-th Gaussian distribution of S_a^α, μ_a is mean vector of a, Σ_a is covariance matrix of a, M is number of state, and $BD(P_a, P_b)$ is the Bhattacharyya distance between two Gaussians.

2.2.2 Baseline Method

As the baseline, we refer the system provided by NTCIR-9 SpokenDoc subtask. The baseline system uses a dynamic programming (DP) based word spotting. The score between a query term and an IPU is calculated based on phoneme-based edit distance.

2.2.3 Proposed Method

As the proposed method, we implemented the simplest version of the algorithm (referred to as Basic) described in Section 2.1.1, 2.1.2, and extended method dealing with multiple recognition candidate in Section 2.1.3(referred to as Extended).

In this work, we use confusion network for dealing with multiple recognition candidates. Because confusion network include ϵ transition, we modify the network by applying the following transformation rules (Fig.3).

1. If the best recognition candidate is ϵ, remove all the candidates at the position and shrink the network.

2. If there is ϵ transition other than the best candidate, remove it.

We set the confidence weight $w(b) = 1$ for all the candidates in Eq. 1.

2.2.4 Results

Table 1: *Experimental results when distance threshold is zero.*

method	baseline	basic	extended
recall	0.218	0.232	0.271
precision	0.907	0.912	0.713
processing time[ms/50queries]	-	66.990	84.987

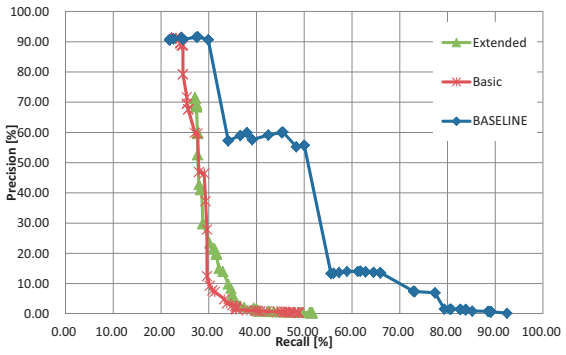

Figure 4: *Recall-Precision Curve by varying the distance threshold*

Firstly, in order to inspect the time efficiency, we set the threshold of the distance to zero. Table 1 shows the micro-averaged recall and precision, and average processing time over the queries. It indicates that the proposed method promises a fast STD, as it boosts the search efficiency without losing detection performance. Looking at the comparison between Basic and Extended, Extended achieves higher recall while it increases the processing time. Note that, viewing from algorithmic point of view, the time efficiencies are same between them. The differences come only from the differences of the number of the detection points by using the different measures of the distance, as using multiple candidates loosen the distance measure.

Fig. 4 shows the relation between recall and precision obtained by varying the distance threshold. It shows that Extended improves the recall a little. However, as the baseline performs much better, there is room for improvement especially in terms of recall.

3. SDR SUBTASK

We participated both the lecture retrieval and the passage retrieval tasks of SpokenDoc SDR subtask. In Section 3.1, we will explains our retrieval methods specifically designed for the passage retrieval task. Then, we will explain our two approaches for SDR, both of which are applied to both the lecture and passage retrieval tasks. Section 3.2 explains the word-based approach, which Section 3.3 explains the STD-based approach. In Section 3.4, we will show the experimental evaluation of these proposed methods in the SpokenDoc task.

3.1 Methods for Passage Retrieval

The SpokenDoc passage retrieval task differs from a conventional document retrieval task in that the segments of passages are not predefined in advance and that it is required both to determine the boundary of the passage in the collection and to rank them according to their relevancy to the query topic. Therefore, we extended the conventional document retrieval method with the two specific methods designed for the passage retrieval.

3.1.1 *Using the Neighboring Context to Index the Passage*

Passages from the same lecture may be related to each other in the passage retrieval task, whereas the target documents are considered to be independent of each other in a conventional document retrieval task. In particular, the neighboring context of a target passage should contain related information. It would seem appropriate for the passage retrieval task to use the neighboring context to index the target passage [8]. A similar method was applied in TREC SDR TRACK [9].

Normally, a passage D is indexed by its own term frequencies $TF(t, D)$ of the terms $t \in D$. This can be extended to use the neighboring context for indexing. For the context $context_n(D)$, the preceding n utterances and the following n utterances are used. Therefore, we use

$$TF_{ext}(t, D) = \beta TF(t, D) + TF(t, context_n(D)), \qquad (4)$$

where β is introduced to specify the relative importance of D and $context_n(D)$.

In our implementation, an utterance is used for D, and n and β are set to 7 and 5 respectively through the preliminary experiments. We refer this method as to *context indexing*.

3.1.2 *Penalizing Neighboring Retrieval Results*

In applying context indexing, neighboring passages are liable to be retrieved at the same time as they share the same indexing words. This is not adequate from the perspective of retrieval systems because such systems output many redundant results.

For this reason, we penalize a retrieval result that is neighbor to another result that has been output previously. In practice, the retrieved passage is discarded from the output list, if there are other results already retrieved within an n-utterances neighborhood of it.

3.2 Word-based Approach

The word based approach for SDR uses the word-based speech recognition results for indexing the spoken documents. In the SpokenDoc task, we used the word-based reference transcription released by the organizers. Once the transcripts of the spoken documents are obtained, any text based IR method can be applied. The most major problems of this approach lies in that the out-of-vocabulary words of the word-based speech recognition never appear in the text and thus cannot become clues for retrieval. To overcome the problem, we applied relevance models, a query expansion method, for the spoken document passage retrieval task [3].

3.2.1 *Relevance Models*

Levrenko and Croft [10] proposed *relevance models* as an information retrieval model. They define the relevance class R to be the subset of documents in a collection $\mathcal{C}$, which are

relevant to some particular information need, i.e. $R \subset \mathcal{C}$. A relevance model is the probability distribution $P(w|R)$, where $w \in V$ is a word in a vocabulary V. $P(w|R)$ is estimated from a given query Q as follows.

$$P(w|R) \approx P(w|Q) = \frac{P(w,Q)}{P(Q)} \tag{5}$$

Suppose that Q consists of a sequence of words $q_1 \cdots q_k$ and that both $q_1 \cdots q_k$ and w are sampled identically and independently from a unigram distribution $P(w|R)$. Assuming a sampling process where a document D is sampled from $\mathcal{C}$ at first, then words are sampled from D, $P(w,Q)$ is obtained as follows.

$$P(w,Q) = \sum_{D \in \mathcal{C}} P(D)P(w,Q|D) \tag{6}$$

Because we assume that w and $q_1 \cdots q_k$ are sampled independently and identically, the joint probability $P(w,Q|D)$ can be expressed as follows:

$$P(w,Q|D) = P(w|D)\prod_{i=1}^{|Q|} P(q_i|D). \tag{7}$$

By substituting equation (7) into equation (6), the following estimate is obtained:

$$P(w,Q) = \sum_{D \in \mathcal{C}} P(D)P(w|D)\prod_{i=1}^{|Q|} P(q_i|D). \tag{8}$$

Suppose that $P(D)$ is distributed uniformly, $P(w|R)$ is estimated as follows:

$$P(w|R) = \frac{1}{P(Q)} \sum_{D \in \mathcal{C}} P(w|D)\prod_{i=1}^{|Q|} P(q_i|D), \tag{9}$$

where $P(Q)$ is constant with respect to Q.

Then, $P(w|R)$ is used to rank the documents $D \subset \mathcal{C}$ by using the Kullback–Leibler divergence between the distributions $P(w|R)$ and $P(w|D)$:

$$H(R||D) = -\sum_{w \in V} P(w|R) log P(w|D). \tag{10}$$

Relevance models can be seen as an implementation of pseudo relevance feedback, which is a sort of query-expansion technique using the target document collection, i.e. the query Q is expanded with the related words in the collection $\mathcal{C}$ through the estimation of the relevance model $P(w|R)$.

3.2.2 *Extending Relevance Models to Context Indexing*

Applying relevance models directly to our passage retrieval, specifically the context-indexing method described in Section 3.1.1, is problematic. Because context indexing uses neighboring utterances to index a document (an utterance), several neighboring documents share the same index words. This makes the estimated $P(w|R)$ inaccurate.

In order to deal with this problem, no context-expanded documents, i.e. a set of utterances, are used in the estimation of $P(w|R)$, but then context-expanded documents are ranked using $P(w|R)$. Namely, $P(w|R)$ is estimated as follows:

$$P(w|R) = \sum_{D \in \mathcal{C}} P(w|D_{nc})\prod_{i=1}^{|Q|} P(q_i|D), \tag{11}$$

where D and $\mathcal{C}$ are an utterance and a set of utterances, respectively. Then, the context-expanded documents $\tilde{D} \subset \tilde{\mathcal{C}}$ are ranked by the following equation:

$$H(R||\tilde{D}) = -\sum_{w \in V} P(w|R) log P(w|\tilde{D}). \tag{12}$$

3.2.3 *Query Likelihood Model*

We also applied the query likelihood model as our retrieval model for SDR. For document re-ranking, we use the probability $P(Q|D)$ that a query Q is constructed from a relevant document D:

$$P(Q|D) = \prod_{q \in Q} P(q|D). \tag{13}$$

$P(q|D)$ is estimated by

$$P(q|D) = (1-\gamma)\frac{TF(q,D)}{\sum_t TF(t,D)} + \gamma\frac{TF(q)}{\sum_t TF(t)}, \tag{14}$$

where $TF(q)$ is the global term frequency of a query term q calculated from the target document collection C by

$$TF(q) = \sum_{D \in C} TF(q,D). \tag{15}$$

The $P(Q|D)$ is used to rank the document $D \in \mathcal{C}$. In this paper, the context-expanded document $\tilde{D} \in \tilde{\mathcal{C}}$ is used instead of D.

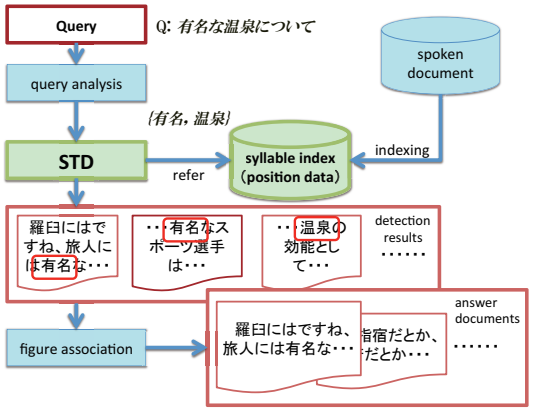

Figure 5: *STD-SDR system*

3.3 STD-based approach

The conventional SDR methods use the word-based speech recognition to obtain the transcription of the spoken documents, then the text based document retrieval is applied on the transcription. However, the out of vocabulary (OOV) words of the word-based speech recognition and the miss-recognized words can never be used as the clues for the document retrieval, which results in the degradation on the re-

Table 2: *Retrieval performance (pwMAP) on the dry run data using manual transcription. (CI: context indexing, NP: neighborhood penalty)*

retrieval model	BASE	+CI	+NP
vector space model	0.144	0.137	0.162
query likelihood model	0.161	0.126	0.176
relevance model	0.170	0.166	0.183

Table 3: *Results by using automatic transcription (REF-WORD).*

retrieval model	dry run	formal run
vector space model	0.121	-
query likelihood model	0.120	0.144
relevance model	0.136	0.158

trieval performance. In order to deal with such a problem, we have proposed the STD-based approach for SDR.

Firstly, the keywords are extracted from the query topics and are converted to the subword sequences. In this paper, we used nouns as the keywords and syllable as the subword unit. Then, each syllable sequence is searched against the syllable-based transcription of spoken documents, which have been obtained by using the syllable-based speech recognition. For this STD method, we used the Dynamic Time Warping (DTW) algorithm according to the following equation.

$$D_{i,j} = \min\{D_{i,j-1}, D_{i-1,j-1}, D_{i-1,j}\} + d_{i,j} \qquad (16)$$

where $d_{i,j}$ is a distance between syllables at the position i in the spoken documents and at j in the keyword, and $D_{i,j}$ is a cumulative distance. The cumulative distances at the tail of the keyword is normalized by its length and the detection is made if the normalized distance is below some predefined threshold. From the STD results, we can obtain the keyword frequency for each document in the collection. We repeat the process for all keywords, then we can obtain the vector of the keywords frequency for each document. Finally, the vectors are compared with the query vector to select the retrieval results, which is equal to applying the conventional vector space model for IR. Figure 5 shows the configuration of our STD-based system.

3.4 Experiments

3.4.1 Word-based Approach

We investigated the results for the passage retrieval task for the word-based approaches, where we used pwMAP as the evaluation metric.

The retrieval performances on the dry run data compared among the retrieval models using manual transcription of the target spoken documents are shown in Table 2. The baseline indexing methods index just the utterance, which corresponds to the BASE column in the table. The results show that the two language modeling retrieval models outperform the traditional vector space model with TF–IDF term weighting. It also shows that the retrieval model using query expansion (relevance model) outperforms the model without it (query likelihood model).

In Section 3.1, we introduced the retrieval methods for passage retrieval, i.e. context indexing (Section 3.1.1) and neighborhood penalty (Section 3.1.2). The two methods are incrementally applied in this order to the BASE indexing method. The results are shown at the column labeled +CI (applying only context indexing) and +PI (applying both context indexing and neighborhood penalty) in Figure 2. The results show that applying only context indexing decreases performance. This is because context indexing favors outputting neighboring passages at the same time, which results in decreasing the retrieval performance. However, the results also show that applying both context indexing and the neighborhood penalty at the same time successfully overcomes the harmful influence resulting in the method outperforming the BASE method. These results are consistent among the compared retrieval methods.

Table 3 shows the results for both the dry run and the formal run queries by using the automatic transcription instead of the manual transcription. Here, we also applied both the context indexing and the neighborhood penalty at the same time. The used automatic transcription was REF-WORD, which was provided by the SpokenDoc task organizers. It shows that the results are consistent among manual and automatic transcriptions used as target documents.

3.4.2 STD-based Approach

We investigated the results for the lecture retrieval task using the dry run data of SpokenDoc task, where we used MAP as the evaluation metric.

The proposed STD-based approach used the reference syllable-based automatic transcription (REF-SYLLABLE) provided by the task organizers, where only 1-best result was used. The STD method was the Dynamic Time Warping (DTW) algorithm, in which the Battacharyya distance between the acoustic models was used as the distance between two syllables. We only extracted the keywords longer than two syllables in order to restrain the insertion errors. Moreover, only exact matching was considered for the short keyword consisted of two syllables at the detection. The detection threshold was selected so as to maximize the MAP on the dry run data. For the document retrieval, we used the vector space model with TF-IDF term weighting.

We compared our STD-based approach with the conventional SDR approach as an baseline. The baseline system used the reference word-based automatic transcription (REF-WORD), which was also provided by the task organizers, as a textual representation of spoken documents and applied the word-based textual retrieval method with TF-IDF term weighting, which was same as those used at the SDR part of our STD-based approach.

For analysis purposes, the performances on the manual transcription were also investigated for both the proposed and the baseline systems. The syllable sequence extracted from the manual transcription was used for the proposed system, while the word sequence itself was used for the baseline system.

Table 4 shows the results. The line labeled “Conventional SDR” corresponds to the baseline system, while the line la-

Table 4: *SDR results on manual and automatic transcription in terms of MAP.*

	manual transcription			automatic transcription		
	ALL	IV	OOV	ALL	IV	OOV
Conventional SDR (using word transcription)	0.37	0.35	0.55	0.26	0.27	0.15
STD-SDR (using syllable transcription)	0.31	0.29	0.48	0.24	0.24	0.26

beled "STD-SDR" corresponds to the proposed STD-based system. The column labeled ALL corresponds to the MAP averaged over all the query topics, while the column labeled OOV (or IV) corresponds to the MAP averaged over only those including at least one OOV term (or only IV terms).

The result in total (ALL) shows the proposed STD-based approach performs worse than the baseline. It seems because the STD-based approach cannot benefit from the word information, which is much informative for those languages using Kanji representation like Japanese. However, the OOV results shows that the STD-based approach is effective for the query topics including OOV terms. It also shows the comparison between the ALL and OOV column in the manual transcription reveals that the OOV terms tend to be effective clues for IR. Those indicates that the STD-based approach is tolerant for the errors introduced by the speech recognition and it is promising for SDR as it can benefit much from the effective clues brought by the OOV terms.

4. CONCLUSION

We participated both subtasks in the NTCIR-9 SpokenDoc task.

For the STD subtask, novel metric subspace indexing was investigated. It had the distinctive advantage of its unnecessaries of pre-determined threshold as it can detect the results in increasing order of distance. The experimental results showed that the proposed method was very fast but there were rooms for improvement in the detection accuracy.

For the SDR subtask, two approaches were investigated. The first approach was the conventional word-based IR methods based on the language modeling IR models, which were extended to fit the SpokenDoc passage retrieval tasks. The experimental results showed that the relevance model was best performed among the methods compared.

The second approach was the STD-based, in which the STD was applied for detecting the terms in the query from the spoken documents and then applied the IR methods using the detection as the term's appearances. The experimental results showed that, though the performance of the STD-based method was lower than the word-based approaches in total, it could improve the performance if the query topic included the out-of-vocabulary words.

5. REFERENCES

[1] Tomoyosi Akiba, Hiromitsu Nishizaki, Kiyoaki Aikawa, Tatsuya Kawahara, and Tomoko Matsui. Overview of the ir for spoken documents task in ntcir-9 workshop. In *Proceedings of the 9th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-lingual Information Access*, 2011.

[2] Taisuke Kaneko and Tomoyosi Akiba. Metric subspace indexing for fast spoken term detection. In *Proc. of INTERSPEECH*, pages 689–692, 2010.

[3] Tomoyosi Akiba and Koichiro Honda. Effects of query expansion for spoken document passage retrieval. In *Proc. of INTERSPEECH*, pages 2137–2140, 2011.

[4] Gonzalo Navarro, Ricardo Baeza-Yates, Erkki Sutinen, and Jorma Tarhio. Indexing methods for approximate string matching. In *IEEE Data Engineering Bulletin*, volume 24, pages 12–27, 2000.

[5] John S. Garofolo, Cedric G. P. Auzanne, and Ellen M. Voorhees. The TREC spoken document retrieval track: A success story. In *Proc. of TREC-9*, pages 107–129, 1999.

[6] Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: Word error minimization and other applications of confusion network. *Computer Speech and Language*, 14(4):373–400, October 2000.

[7] Peng Yu, Yu Shi, and Frank Seide. Approximate word-lattice indexing with text indexers: Time-anchored lattice expansion. In *Proc. of ICASSP*, pages 5248–5251, 2008.

[8] Koichiro Honda and Tomoyosi Akiba. Language modeling approach for retrieving passages in lecture audio data. In *Proceedings of International Conference on Language Resources and Evaluation*, pages 1525–1530, 2010.

[9] S.E. Johnson, P. Jourlin, K.sparck Jones, and P.C. Woodland. Spoken document retrieval for TREC-9 at cambridge university. In *Proceedings of TREC-9*, 1999.

[10] V.Lavrenko and W. B. Croft. Relevance models in information retrieval. *In Language Modeling for Information Retrieval*, pages 11–56, 2003.