STD and SCR Techniques and Their Evaluations on the NTCIR-10 SpokenDoc-2 task

Yuto Furuya
University of Yamanashi
4-3-11 Takeda, Kofu
Yamanashi, 400-8511, Japan
furuya@alps-lab.org

Daiki Nakagomi
University of Yamanashi
4-3-11 Takeda, Kofu
Yamanashi, 400-8511, Japan
daiki@alps-lab.org

Satoshi Natori
University of Yamanashi
4-3-11 Takeda, Kofu
Yamanashi, 400-8511, Japan
natori@alps-lab.org

Hiromitsu Nishizaki
University of Yamanashi
4-3-11 Takeda, Kofu
Yamanashi, 400-8511, Japan
hnishi@yamanashi.ac.jp

Yoshihiro Sekiguchi
University of Yamanashi
4-3-11 Takeda, Kofu
Yamanashi, 400-8511, Japan
sekiguti@yamanashi.ac.jp

ABSTRACT

This paper describes spoken term detection (STD) and spoken contents retrieval (SCR) techniques and their evaluations at the NTCIR-10 SpokenDoc-2 task. First of all, we describes our STD technique using a phoneme transition network (PTN) derived from multiple speech recognizers' outputs and its evaluations at the STD and the iSTD (inexistent STD) tasks. Next, we introduce our SCR technique using Web documents for expanding the target spoken documents. It is evaluated on the two SCR tasks.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Performance

Keywords

Multiple recognizers, phoneme transition network, spoken conetent retrieval, spoken term detection, spoken document segmentation

Term name: [**ALPS**]
Subtask: [**Spoken Term Detection**], [**Spoken Contents Retrieval**]
Language: [**Japanese**]

1. INTRODUCTION

Recently, information technology environments have evolved in which numerous audio and multimedia archives, such as video archives, and digital libraries, can be easily used. In particular, a rapidly increasing number of spoken documents, such as broadcast programs, spoken lectures, and recordings of meetings, are archived, with some of them accessible through the Internet. Although the need to retrieve such spoken information is growing, an effective retrieval technique is currently not available; thus, the development of technology for retrieving such information has become increasingly important.

In the Text REtrieval Conference (TREC) Spoken Document Retrieval (SDR) track hosted by the National Institute of Standards and Technology (NIST) and the Defense Advanced Research Projects Agency (DARPA) in the second half of the 1990s, many studies of SDR were presented using English and Mandarin broadcast news documents [5]. The TREC SDR is an ad-hoc retrieval task that retrieves spoken documents which are highly relevant to a user query.

In 2006, NIST initiated the Spoken Term Detection (STD) project with a pilot evaluation and workshop [12]. The aim of STD is to find the position of spoken terms selected for evaluation from an audio archive.

The difficulty in STD lies in the search for terms in a vocabulary-free framework, because search terms are not known a prior to the speech recognizer being used. Many studies dealing STD tasks have been proposed, such as [16, 9]. Most STD studies focus on the out-of-vocabulary (OOV) and speech recognition error problems. For example, STD techniques that use entities such as sub-word lattice and confusion network (CN) have been proposed.

This paper describes our STD and SCR (spoken content retrieval) evaluation on the NTCIR-10 SpokenDoc-2 task. The former part of the paper introduces our STD and iSTD techniques and these evaluations. The other part of the paper shows the SCR technique and its evaluation results.

Our STD technique uses a phoneme trasition network (PTN)-formed index derived from multiple speech recognizers' 1-best hypothesis and a dynamic time warping (DTW) framework with false control parameters applied at the term searching. PTN-based indexing originates from the idea of the CN being generated from a speech recognizer. CN-based indexing for STD is a powerful indexing method. The multiple speech recognizers cangenerate the PTN-formed index by combining sub-word (phoneme) sequences from the output of these recognizers into a single CN. This study uses 10 types of speech recognition systems with the same decoder used for all. Two types of acoustic models (triphone-based and syllable-based Hidden Markov Models (HMMs)) and five types of language models (word-based or sub-word based) were prepared.

The use of the 10 recognizers and their output is very effective in improving speech recognition performance. For example, Fiscus [4] proposed the ROVER method that adopts a word voting scheme. Utsuro et al. [15] developed a technique for combining multiple recognizers' output by using a support vector machine (SVM) to improve speech recognition performance. Application of the characteristics of the word (or sub-word) sequence output by recognizers may increase STD performance since these characteristics are different for each speech recognizer. The PTN-based on multiple speech recognizers' output can cover more sub-word sequences of spoken terms. Thus, multiple speech recognizers may improve STD performance compared to that of a single recognizer's output.

We evaluated the PTN-formed index derived from the 10 recognizers' output. The experimental result for the Japanese STD test collection [6] showed that using the PTN formed index was effective in improving STD performance compared to that of the CN-formed index, which was derived from the phoneme-based CN made up to the 10-best phoneme sequence outputs from a single speech recognizer [10, 11].

However, many false detection errors occurred because the PTN-formed index had redundant phonemes that were mistakenly recognized by some speech recognizers. Using more speech recognizers can achieve a higher speech recognition performance but more errors may occur. Therefore, we adopted a majority voting parameter and a measure of ambiguity, which are easily derived from the PTN, into the term search engine.

We installed the voting and ambiguity parameters of the PTN to our term search engine to prevent false detections. Furthermore, we refined the search engine, where the DTW cost used for calculating the distance is dynamically changed depending on length of a query term (the number of phonemes consisting of a query term).

We also challanged the SCR on the lecture retrieval and the phrase retrieval task. Our SCR technique uses a spoken document expansion using WEB documents. There are two advantages for using document expansion using WEB on an SCR task. One of advantages is robust for speech recognition errors and OOV. The other is to supplement keywords that are not uttered in the spoken document, which a user wishes to look for. For solving the speech recognition errors and OOV problems, we prepared two kinds of indexes: One is made from transcriptions of spoken documents by an LVCSR system; the other is made from WEB pages related to the target spoken documents. In our approach, WEB pages are retrieved by a search engine that uses WEB search queries automatically composed from transcriptions of the target spoken documents.

When document expansion techniques are used in the retrieval of these documents, WEB pages may be the most suitable, because the Internet has a wide variety of topics. Consequently, in this paper, we investigate the effectiveness of document expansion by using WEB pages on SCR.

The problem of document expansion using WEB for SCR is to how to collect WEB pages whoes contents are similar to the spoken documents. We performed WEB pages collection by human power, then, the SDR performance was drastically improved compared with the performance without the document expansion. However, it is hard to collect suitable WEB pages. So, we propose an WEB collection method using an automatic spoken document segmentation method [14]. Most of the spoken documents have some topics. Collecting WEB page for each topic may improvement the quarity of WEB pages.

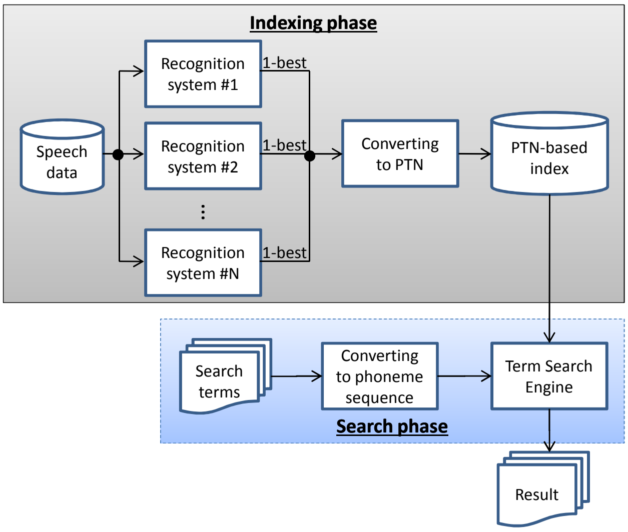


Figure 1: Overview of the STD framework.

This paper introduces the our STD, iSTD and SCR systems and reports these systems' performances on the NTCIR-10 SpokenDoc-2 tasks.

2. STD TASK

2.1 Outline

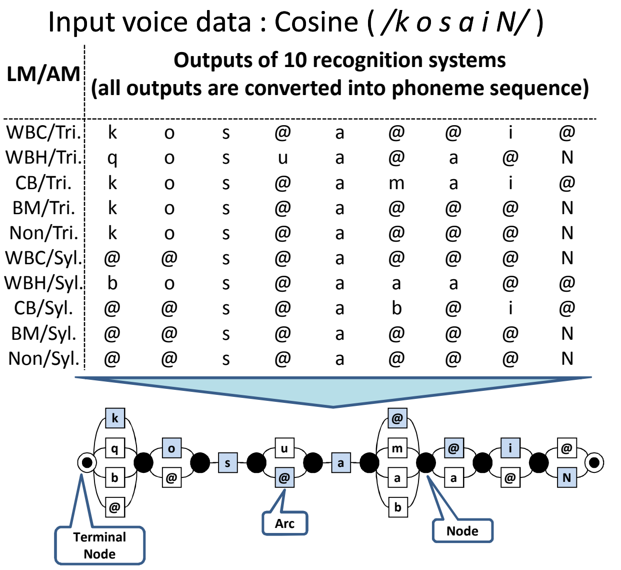
Figure 1 represents an outline of the STD framework in this paper.

In the indexing phase, speech data is performed by speech recognition and the recognition outputs (word or sub-word sequences) are converted into the PTN index for STD. In the search phase, the word-formed query is converted into the phoneme sequence, then the phoneme-formed query is input to the term search engine. In the case of treating English queries, we have to consider the variety of pronunciations of the queries. There are some reports fighting the pronunciation problem[17]. In this paper, however, we handle Japanese STD. Most of Japanese words can be completely translated to phoneme sequence (pronunciation). Therefore, we do not consider the pronunciation problem in this paper.

The term search engine searches the input query term from the index in phoneme level using the DTW framework.

2.2 PTN-based indexing

Figure 2 shows an example of making part of PTN-formed index of an utterance "*cosine*" (Japanese pronunciation is /k o s a i N/) by performing the alignment process of 10 of phoneme sequences from 1-best hypotheses of the ASRs. We used 10 types of ASRs for making PTN-based index. In Fig. 2, the utterance is recognized by the 10 ASRs, and then, the 10 hypotheses are obtained. They are converted into the phoneme sequences. Next, we can get "aligned sequences" by performing a dynamic programming (DP) scheme as well as making the CN-formed index. Finally, the PTN is gotten by converting the aligned sequences. "@" in Fig. 2 indicates



Input voice data : Cosine (/k o s a i N/)

LM/AM	Outputs of 10 recognition systems (all outputs are converted into phoneme sequence)								
WBC/Tri.	k	o	s	@	a	@	@	i	@
WBH/Tri.	q	o	s	u	a	@	a	@	N
CB/Tri.	k	o	s	@	a	m	a	i	@
BM/Tri.	k	o	s	@	a	@	@	@	N
Non/Tri.	k	o	s	@	a	@	@	@	N
WBC/Syl.	@	@	s	@	a	@	@	@	N
WBH/Syl.	b	o	s	@	a	a	a	@	@
CB/Syl.	@	@	s	@	a	b	@	i	@
BM/Syl.	@	@	s	@	a	@	@	@	N
Non/Syl.	@	@	s	@	a	@	@	@	N

Figure 2: *Making PTN-based index by performing alignment using on DP and converting to PTN.*

a null transition. Arcs between nodes in the PTN have some phonemes and null transitions with an occurrence probability. However, in this paper, we do not use any phoneme occurrence probabilities.

2.3 Term search engine with false detection control

The term search engine uses the DTW-based word-spotting method. Figure 3 represents an example of the DTW framework between the search term "k o s a i N" (cosine) and for the PTN-formed index. The PTN has multiple arcs between adjoining two nodes. These arcs are compared to one of phoneme labels of a query term.

We use edit distance as cost on the DTW paths, and the cost value for substitution, insertion and deletion errors is commonly set to 1.0 when the number of phonemes including a query is N or larger than N. On the other hand, each cost is commonly set to 1.5 when the number of phonemes is less than N to avoid false term detections in query terms, having less number of phonemes. This cost (=1.5) is optimized using a development query set.

The total cost $D(i,j)$ at the grid point (i,j) ($i = \{0, ..., I\}$, $j = \{0, ..., J\}$, where I and J are the number of the set of arcs in an index and a query term respectively) on the DTW lattice is calculated by following equations:

$$D(i,j) = \min \begin{cases} D(i,j-1) + Del \\ D(i-1,j) + Null(i) \\ D(i-1,j-1) + \\ \quad Match(i,j) + Vot(i,j) + Acw(i) \end{cases} \tag{1}$$

$$Match(i,j) = \begin{cases} 0.0 : Query(j) \in PTN(i) \\ 1.0 : Query(j) \notin PTN(i), J \geq N \\ 1.5 : Query(j) \notin PTN(i), J < N \end{cases} \tag{2}$$

$$Del = \begin{cases} 1.0 : J \geq N \\ 1.5 : J < N \end{cases} \tag{3}$$

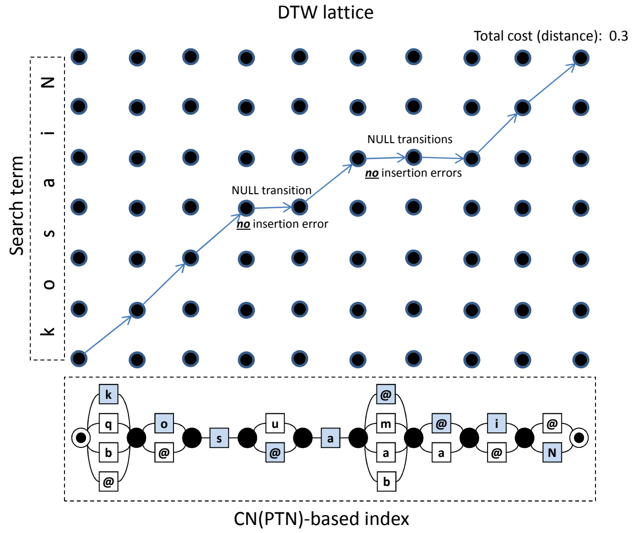


Figure 3: Example of term search on network based index.

$$Null(i) = \begin{cases} \frac{\alpha}{Voting(@)} : NULL \in PTN(i), J \geq N \\ \frac{\beta}{Voting(@)} : NULL \in PTN(i), J < N \\ 1.0 : NULL \notin PTN(i), J \geq N \\ 1.5 : NULL \notin PTN(i), J < N \end{cases} \tag{4}$$

,where $PTN(i)$ is the set of phoneme labels of the arcs at the i-th node in the PTN, and $Query(j)$ indicates the j-th phoneme label in the query term. We allow a null transition between two nodes in the PTN-formed index with the cost defined in Eq.(4). When the query term matches to null (@) in the PTN, a transition cost is dynamically set as shown in Eq.(4). $Voting(@)$ means the number of ASRs outputting NULL at the same arc. We call it "null voting." α and β are hyper parameter, which is optimized using the development set. The appropriate null cost achieves increasing term detection and decreasing false detections.

"$Vot(i,j)$" and "$Acw(i)$" in Eq.(1) are related to the false detection control parameters and calculated as follows:

$$Vot(i,j) = \begin{cases} \frac{\gamma}{Voting(p)} : \\ \qquad \exists p \in PTN(i), p = Query(j) \\ 0.0 : Query(j) \notin PTN(i) \end{cases} \tag{5}$$

$$Acw(i) = \delta \cdot ArcWidth(i) \tag{6}$$

We provide two types of parameters to control false detection as follows:

- "$Voting(p)$" is the number of ASRs outputting the same phoneme p at the same arc. More value of $Voting(p)$ makes reliability of phoneme p better.
- "$ArcWidth(i)$" is the number of arcs (phoneme labels) at $PTN(i)$. Less value of $ArcWidth(i)$ makes also reliability of phonemes at $PTN(i)$ better.

γ and δ are also hyper parameters and set to 0.5 and 0.01 respectively, optimized by the development query set.

In advance searching the query term, the term search engine initializes $D(i,0) = 0$, and then, calculates $D(i,j)$ using Eq.1 ($i = \{0, ..., I\}$, $j = \{1, ..., J\}$). Furthermore, $D(i,J)$ are normalized by length of the DTW path.

After finishing the calculation, the engine outputs the detection candidates which have normalized cost $D(i, J)$ below a threshold θ. Changing the θ value enables us to control the recall and precision rates on STD performance.

2.4 STD experiment

2.4.1 *Speech Recognition*

As shown in Figure 1, the speech data is recognized by the 10 ASRs. Julius ver. 4.1.3 [7], an open source decoder for LVCSR, is used in all the systems.

We prepared two types of acoustic models (AMs) and five types of language models (LMs) for constructing the PTN. The AMs are triphone based (Tri.) and syllable based HMMs (Syl.), where both types of HMMs were trained from the spoken lectures in the Corpus of Spontaneous Japanese (CSJ) [8].

All the LMs are word and character based trigrams as follows:

WBC : word based trigram in which words are represented by a mix of Chinese characters, Japanese Hiragana and Katakana.

WBH : word based trigram in which all words are represented only by Japanese Hiragana. The words composed of Chinese characters and Katakana are converted into Hiragana sequences.

CB : character based trigram in which all characters are represented by Hiragana.

BM : character sequence based trigram in which the unit of language modeling is two of Hiragana characters.

Non : No LM is used. Speech recognition without any LM is equivalent to phoneme (or syllable) recognition.

Each model is trained from the many transcriptions in the CSJ under the open for the speech data of STD.

Finally, the ten combinations, comprising two AMs and five LMs, are formed. The condition is completely the same as the description in the overview paper [1].

2.4.2 *Query set of the STD sub-tasks*

The NTCIR-10 SpokenDoc organizers provided two types of query sets; one, called as "moderate-size" , is for the academic lectures of the spoken document processing workshop (SDPWS) and the other, called as "large-size" is for the all 2702 lectures in CSJ[1]. The moderate-size query set consists of 200 terms, which is the same query set for the iSTD task. Half terms among the query set are not included in the SDPWS lecture speeches.

2.5 Experimental result of the STD tasks

Figure 5 and Figure 4 shows the recall-precision curves that show the STD performance for the formal-run moderate-size and large-size query sets, respectively. "BL-1", "BL-2" and "BL-3" show the baseline results provided by the SpokenDoc-2 organizers. The baseline systems have a dynamic programming (DP)-based word spotting. The score between a query term and an IPU is calculated based on phoneme-based edit distance[1].

Table 1 also indicates the maximum F-measure and MAP values on the same test sets. The decision point for culculating "spec. F" was decided by the result of the NTCIR-9 SpokenDoc STD formal-run query set. We adjusted the threshold to be the best F-measure value on the formal-run set which was used as a development set.

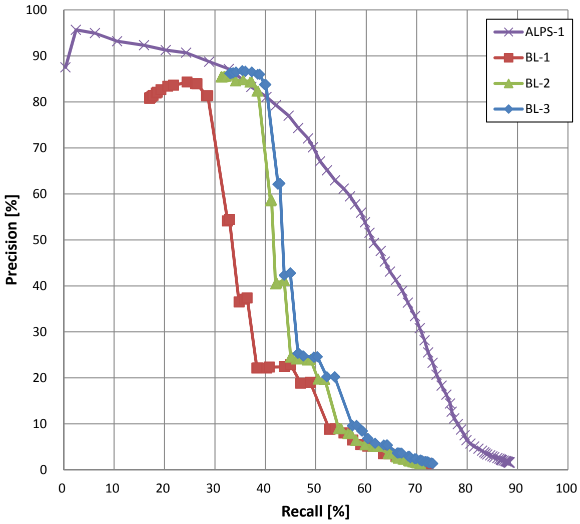


Figure 4: Recall-precision curves for the large-size query set.

Our STD system got the better performances rather than the other participants' runs[1]. Using multiple LVCSRs' outputs was very powerful for improving STD performance. And, the false detection control parameters worked well to reduce detection error candidates.

2.6 Time consumption of the STD processing

Our technique has high performance on a precision-recall curve and a MAP value but needs too much time to search a term from the large scale speech data. It takes 13.5 seconds to search a term from the CSJ CORE speeches (39 hours data) on a computer which has "Intel Core i7 975 3.33GHz" CPU.

3. ISTD TASK

3.1 Task description and test collection

In the iSTD task, we inspect whether a queried term is existent or inexistent in the SDPWS lecture speeches. The query set is the same as the STD moderate-size task, which has 200 queried terms. The half of the terms are inexistent in the SDPWS speeches.

3.2 iSTD Engine

Our iSTD engine is almost the same as the engine of the STD task. We additionally uses entropy value of a detected candidate for ranking the query terms.

An entropy value of an arbitrary interval in a PTN is calculated using the number of phonemes and a posterior probability of a phoneme. A posterior probability at any position of a PTN is calculated based on the number of ASRs that output the phoneme.

Detection entropy (DE_t) of a detected candidate t for a

Table 1: STD performances for the each query set.

set	run	micro ave.		macro ave.			index	search
		max. F [%]	spec. F [%]	max. F [%]	spec. F [%]	MAP	size [MB]	speed [s]
large-size	BL-1	42.32	40.71	43.91	36.70	0.500	58	560
	BL-2	52.52	48.22	47.13	42.21	0.507	58	560
	BL-3	54.25	50.46	46.79	43.95	0.532	116	560
	ALPS-1	58.19	57.38	62.24	50.39	0.717	60	226.4
moderate-size	BL-1	25.08	24.70	25.72	20.07	0.317	3.3	30.8
	BL-2	37.58	37.46	31.43	30.42	0.358	3.3	31.9
	BL-3	39.36	39.16	33.73	32.46	0.393	6.6	30.8
	ALPS-1	46.33	42.83	52.33	39.20	0.606	45	6.06

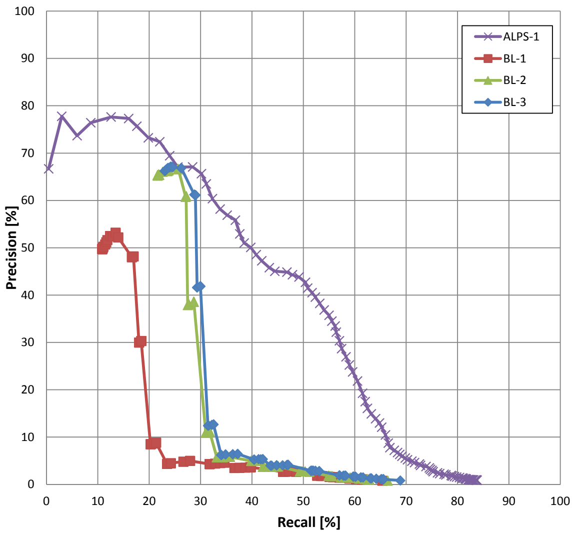


Figure 5: Recall-precision curves for the moderate-size query set.

queried term is calculated as following equations:

$$VE_i = -\sum_{j=1}^{J_i} \frac{Voting(p_{ij})}{R} \log_2 \frac{Voting(p_{ij})}{R} \quad (7)$$

$$DE_t = \frac{1}{T} \sum_{i=t_s}^{t_e-1} VE_i \quad (8)$$

VE_i is a voting entropy between i-th and $(i+1)$-th nodes in a PTN. p_{ij} represents the j-th phoneme at the arcs between i-th and $(i+1)$-th nodes in the PTN, and J_i is the number of phonemes (arcs) between the i-th and $(i+i)$-th nodes. $Voting(p_{ij})$ shows the number of ASRs that output the phoneme p_{ij}. R is the number of all ASRs for making a PTN. In this paper, R is 10.

DE_t is calculated by Eqn. (8). t_s and t_e are the first node and the last node of a candidate t in a PTN, respectively. T is the number of nodes between t_s and t_e.

The highest normarized DTW cost of the candidate of a query term and its entropy value are linealy-combined. Finally, the 200 terms are ranked (ordered) according to combined (DTW cost + entropy) score.

3.3 Evaluation metric

Evaluation metric we used in this task are as follows:

- Recall-Precision curve,
- Maximum F-measure (= the balanced point on Recall-Precision curve),
- Recall, precision and F-measure at the specific rank N (the end point of the curve),
- Recall and precision limiting the terms which have detection="no."

Recall and Precision rates for terms positioned rank r and more than r are calculated as following functions:

$$Recall_r = \frac{T_{\notin,r}}{N_{\notin}} \times 100(\%)$$

$$Precision_r = \frac{T_{\notin,r}}{r} \times 100(\%)$$

, where $T_{\notin,r}$ means the number of $\notin$ terms positioned rank r and more than r, $N_{\notin}$ is the total number of terms belong to class $\notin$. By changing r from 1 to N, a recall-precision curve can be drawn. A maximum F-measure that is from the best balanced point in the curve will also be used for evaluation.

3.4 Result

Figure 6 and Table 2 show the our iSTD performance. "ALPS-1" used entropy for ranking the query terms, and "ALPS-2" only used DTW cost. Our STD engine outperformed the baseline systems.

Our system was robust for false detection errors, although more transcriptions were used. This is because that the false detection control parameters and the entropy of candidate worked well and they are effective to reduce false detections.

4. SCR TASK

4.1 Spoken Document Retrieval Using Web Document Expansion

Figure 7 shows the outline of our proposed technique.

First, spoken documents are automatically transcribed by an LVCSR system. The document index is built by removing stop words from the transcriptions. In this paper, we call it "Recognition-Index" (**RI**).

Next, the other index is made from WEB pages as follows:

1. Each the transcription is automatically divided into some segments depending on topic.
2. Queries for WEB searches are composed from the transcriptions of segmented spoken documents. For each

Table 2: iSTD performances. (*1) Recall, precision and F-measure rates calculated by top-100-ranked outputs. (*2) Recall, precision and F-measure rates calculated by using outputs with "detection=no" tag which is specified by each participant. (*3) Recall, precision and F-measure rates calculated by top-N-ranked outputs. N is set to obtain the muximum F-measure.

run	Rank 100*1			Specified*2				Maximum*3			
	R [%]	P [%]	F [%]	R [%]	P [%]	F [%]	rank	R [%]	P [%]	F [%]	rank
BL-1	73.00	73.00	73.00	81.00	65.85	72.65	123	73.00	76.04	74.49	96
BL-2	74.00	74.00	74.00	81.00	71.05	75.70	114	88.00	69.84	77.88	126
BL-3	75.00	75.00	75.00	81.00	70.43	75.35	115	90.00	68.18	77.59	132
ALPS-1	82.00	82.00	82.00	82.00	82.00	82.00	100	85.00	80.19	82.52	106
ALPS-2	79.00	79.00	79.00	79.00	79.00	79.00	100	84.00	78.50	81.16	107

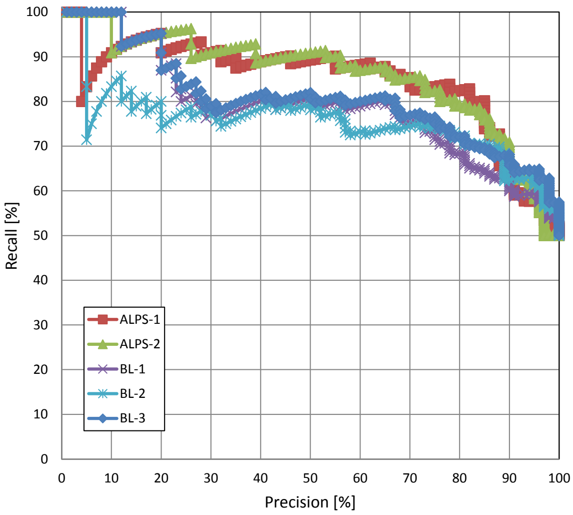


Figure 6: Recall-precision curves at the iSTD task.

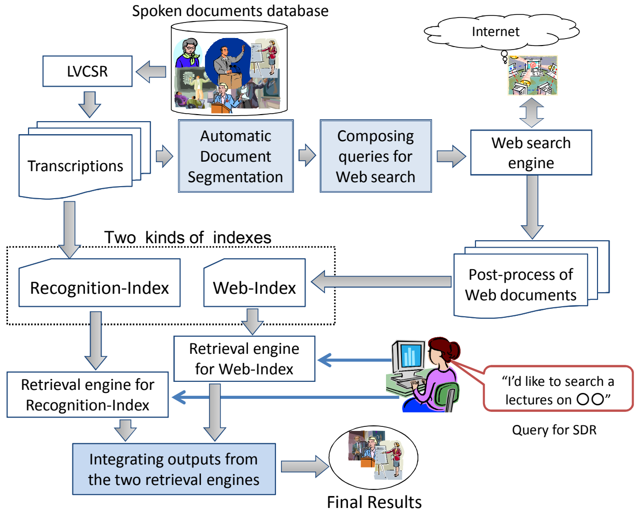

Spoken documents database
Internet
LVCSR
Transcriptions
Automatic Document Segmentation
Composing queries for Web search
Web search engine
Two kinds of indexes
Recognition-Index
Web-Index
Post-process of Web documents
Retrieval engine for Web-Index
Retrieval engine for Recognition-Index
"I'd like to search a lectures on ○○"
Query for SDR
Integrating outputs from the two retrieval engines
Final Results

Figure 7: A framework of spoken document retrieval by using document expansion using WEB.

spoken document, multiple queries, which depend on the number of segments of a spoken document, are prepared.

3. For each segment, an WEB search engine collects WEB pages from the Internet using the query. Most of the collected WEB pages may be related to the part of spoken document from which the search query is made.

4. Stop words are removed from the collected WEB pages. The WEB-based index, which we call the "WEB-Index" (**WI**), is made from them.

The performance of retrieval using the WI depends on how the WEB search query is composed. In addition, the quality of the query relies on speech recognition performance during transcription of the spoken document. However, the object of this study is to investigate whether the WI is effective in spoken document retrieval. Therefore, we adopt a very simple query composition method, as described in Section 4.3.

In the spoken document retrieval process, two retrieval engines search the spoken documents, each using one of the indexes. A query is input to each retrieval engine; the final retrieval results are obtained by integrating their two outputs based on a retrieval score attached to each retrieval document.

4.2 Automatic Document Segmentation

To use WEB pages in spoken document retrieval, queries for an WEB search engine are composed by extracting keywords from transcriptions derived by recognizing spoken documents. However, the transcriptions include a number of words unsuitable for documents on a specific topic. Therefore, it is difficult to extract keywords that well-represent the topic of a spoken document.

To solve this problem, first, the spoken documents are automatically divided into some segments, then, queries are composed from each segment. Most of spoken documents have several kinds of topics. The segmentation of spoken documents makes WEB search queries to be better for collecting WEB pages.

We use the text segmentation algorithm proposed by Utiyama et al.[14] for spoken document segmentation. Although the algorithm was adopted to word sequences in [14], we extended the algorithm to accept sentence sequences.

First, we put nodes on the beginning of the document, the end of the document, and between all two successive sentences. The cost value c_{ij} for a segment S_{ij} between two arbitrary nodes i and j is defined as follows:

$$c_{ij} = \sum_{l=1}^{length} \log \frac{length + k}{tf_l} + penalty \times \log W \qquad (9)$$

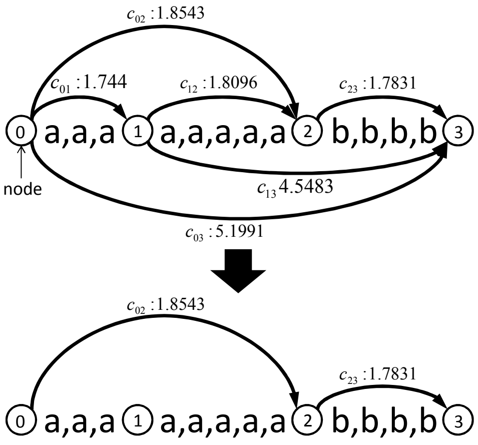


Figure 8: ***Example of automatic document segmentation.***

where *length* is the total number of words contained in a segment S_{ij} and k is the number of kinds of words in the whole document. tf_l means term frequency of word w_l occured in S_{ij} and W is the total number of words in the whole document. *penalty* is a hyper-parameter that can control the number of segments. The dynamic programming (DP) scheme using the cost can indicate some segmentation nodes by minimizing the total DP costs.

Figure 8 shows an example of document segmentation. There are three sentences: S_{01} =“*aaa*”, S_{12} =“*aaaaa*”, and S_{23} =“*bbbb*” and the *penalty* parameter is set to 1.0 in Figure 8. In this case, k and W are 2 and 12, respectively. For example, the cost of the segment S_{03} can be calculated by using Equation(9) and its value is $c_{03} = 5.1991$. Finally, the document is divided into two segments: S_{02} and S_{23} because the total cost is $c_{02} + c_{23} = 3.6374$, which is smaller than c_{03}.

4.3 Query Composition

The query composition method we adopted was very simple. Queries for a spoken document are composed of word-based N-grams that occur with very high frequent in the document’s transcription. The procedure contains only five steps, as follows:

1. A transcription of a spoken document is automatically divided into M segments by the automatic segmentation scheme explained in Section 4.2.
2. Word-based N-grams are extracted from M segmented transcription. An N-gram is denoted as a sequence of specific part of speech, namely, a noun and postpositional particle. The length of N is not limited.
3. Some of the N-grams do not exist in the real world because these are extracted from a transcription containing many speech recognition errors. These are filtered out by using the corpus of “WEB Japanese N-gram 1st edition,” which contains N-grams made from WEB data collected by Google Japan, Inc. In this paper, we call the corpus “Google N-gram”.
4. Stop words that are included in the N-grams are removed. Then, the five most frequent N-grams are extracted from each segment.
5. Finally, M N-gram query sets (each set has five N-grams) is used to search WEB pages. Up to maximum 50 WEB pages are collected for each spoken document. The number of WEB pages collected by a query from a segment depends on the number of sentences in the segment. If a segment has 50% of sentences in a document, the query from the segment can get 25 WEB pages.

4.4 Collecting WEB Documents and Making WEB-Index

For each query, WEB pages are collected by an WEB search engine using the query made from the transcription of the spoken document. We used “Yahoo! WEB search API” as the search engine in this study.

Stop words are removed from the collected WEB pages. WEB pages collected by a query set from a segment are integrated into a file. Each file corresponds to one specific segment. Finally, WI is made of them.

4.5 Spoken Document Retrieval Engine

We used the “Generic Engine for Transposable Association” (GETA) [3] as the spoken document retrieval engine. The GETA can realize fast computation of similarity between a query vector and document vectors.

In this research, word-unit indexes (RI and WI) needed to retrieve spoken documents are constructed from only the content words of transcriptions of spoken documents (RI) or WEB documents (WI) by removing stop words. Nouns, verbs, and adjectives are adopted as content words.

Queries for spoken document retrieval are sentences in the form of “List some World Heritage sites,” for example, so morphological analysis is performed on the query to segment it into a word sequence. After removing stop words from the sequence, the query is input to the GETA engine.

The computation of similarity between a query vector and document vectors is done by the SMART method [13], which is based on cosine similarity, like TF-IDF and is available in the GETA. Each indexed word is weighted by the TF-IDF method, in which the TF value of each word is normalized on document length (number of words).

4.6 Integrating Outputs from Two Retrieval Engines

The final retrieval results are obtained by integrating the outputs from two retrieval engines. One retrieval engine uses RI, and the other uses WI. Each engine outputs a list of spoken documents, in order of similarity score.

Therefore, we can get the final retrieval results (a list of documents) by combining the two similarity scores from RI and WI. The final combined similarity score $sim(d)$ for a spoken document from the two engines is calculated as follows:

$$sim(d) = (1 - \alpha) \times sim(d|r) + \alpha \times \max_{s \in S} sim(d_s|w) \quad (10)$$

where $sim(d|r)$ is the score from the RI engine. Suppose that $S = \{d_1, d_2, \cdots, d_M\}$ is a segment set when a document d is divided into M segments. So, $sim(d_s|w)$ is the score of segment s for document d from the WI engine.

In retrieval experiments, α is empirically set to 0.2 for the all queries. In addition to this, we also tried to estimate α automatically for each query using the development set that is the NTCIR-9 SpokenDoc SDR task data [2]. To do it, we used a multiple linear regression analysis in which an OOV

Table 3: Evaluation results for the lecture retrieval task.

run	MAP
baseline-1	0.268
baseline-2	0.231
ALPS-1	0.384
ALPS-2	0.381

Table 4: Evaluation results for the passage retrieval task.

run	uMAP	pwMAP	fMAP
baseline-1	0.133	0.100	0.087
baseline-2	0.092	0.082	0.068
ALPS-1	0.075	0.046	0.033
ALPS-2	0.075	0.046	0.033

rate of a query and retrieval score (SMART) were used as explanatory variable.

4.7 Experimental result

Table 3 and Table 4 show the retrieval results for the lecture retrieval task and the passage retrieval task, respectively. In the "ALPS-1" system, α was automatically estimated for each query. And "ALPS-2" used the same α value of 0.2 for the all query.

The web documents expansion with the spoken document segmentation was effective for improving the SCR performance at the lecture retrieval task. However, our proposed method did not worked well at the passage retrieval task.

5. CONCLUSION

This paper described the STD, iSTD and SCR techniques for the NTCIR-10 SpokenDoc-2.

First, we introduced the PTN-based indexing, which is essentially a phoneme-based CN, derived from multiple speech recognizers' outputs. One of the aims of this study was to use multiple outputs of LVCSRs for constructing the PTN-formed index for STD, which is different from the sub-word based approaches proposed earlier. Furthermore, we installed the false detection control parameters, majority voting and the width of the arc in the PTN, to the DTW framework. The results indicate that this was very effective in controlling false detection in the query term set.

Second, we challenged the iSTD task using almost the same STD engine used on the STD task. The false detection control parameters also worked well on the iSTD task. In addition to this, entropy value of a detected candidate was useful for filtering out false detection candidates.

Finally, we have shown the SCR technique that is the document expansion using WEB pages collected by the document segmenataion. Our technique worked well at the lecture retrieval task. However, it was not effective on the passage retrieval task.

In future work, we intend to develop a fast STD algorithm under the DTW framework. The processing speed of our engine is too slow for practical use.

6. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant-in-Aid for Young Scientists (B) Grant Number 23700111 and Grant-in-Aid for Scientific Research (C) Grant Number 24500225.

7. REFERENCES

[1] T. Akiba, et al. Overview of the ntcir-10 spokendoc-2 task. In *Proc. of the NTCIR-10 Conference*, 10 pages, 2013.

[2] T. Akiba, et al. Overview of the ir for spoken documents task in ntcir-9 workshop. In *Proc. of the NTCIR-9 Workshop*, 8 pages, 2011.

[3] A. T. et al. Generic engine for transposable association (geta). In *http://nii.ac.jp/geta/english.html.*

[4] J. G. Fiscus. A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER). In *Proc. of ASRU'97*, pp. 347–354, 1997.

[5] J. S. Garofolo, et al. The trec spoken document retrieval track: A success story. In *Proc. of the Text Retrieval Conference (TREC) 8*, pp.16–19, 2000.

[6] Y. Itoh, et al. Constructing japanese test collections for spoken term detection. In *Proc. of INTERSPEECH2010*, pp. 677–680, 2010.

[7] A. Lee and T. Kawahara. Recent development of open-source speech recognition engine julius. In *Proc. of APSIPA ASC 2009*, 6 pages, 2009.

[8] K. Maekawa. Corpus of Spontaneous Japanese: Its design and evaluation. In *Proceedings of the ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR2003)*, pages 7–12, 2003.

[9] S. Meng, et al. Addressing the out-of-vocabulary problem for large-scale chinese spoken term detection. In *Proc. of INTERSPEECH2008*, pages 2146–2149, 2008.

[10] S. Natori, et al. Japanese spoken term detection using syllable transition network derived from multiple speech recognizers' outputs. In *Proc. of INTERSPEECH2010*, pp.681–684, 2010.

[11] S. Natori, et al. Network-formed index from multiple speech recognizers' outputs on spoken term detection. In *Proc. of APSIPA ASC 2010 (student symposium)*, page 1, 2010.

[12] The spoken term detection (STD) 2006 evaluation plan, 2006. `http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf`

[13] A. Singhal, et al. Pivoted document length normalization. In *Proc. of ACM SIGIR' 96*, pp. 21–29, 1996.

[14] M. Utiyama and H. Isahara. A Statistical Model for Domain-Independent Text Segmentation. In *Proc.of the 9th ECACL*, pp. 491–498, 2001.

[15] T. Utsuro, et al. An empirical study on multiple LVCSR model combination by machine learning. In *Proc. of HLT-NAACL 2004*, pp.13–16, 2004.

[16] D. Vergyri, et al. The SRI/OGI 2006 spoken term detection system. In *Proc. of INTERSPEECH2007*, pp.2393–2396, 2007.

[17] D. Wang, et al. Stochastic pronunciation modelling for out-of-vocabulary spoken term detection. *IEEE Trans. on Audio, Speech, and Language Processing*, 19(4):688–698, 2011.