ＨＳＳベースの音声文書検索システム

HSS-based Spoken Document Retrieval System

ヤオジエ ル[*]　ダフェ シ[*]　ユエヤン イン[*]　ジチュアン チョン[*]　リジュン ジョ[*]
YaoJie Lu　DaFei Shi　YueYan Yin　JiChuan Zheng　LiJun Zhao

要　　旨

近年，音声文書検索（SDR）が広く使われるようになってきた．SDRシステムの性能向上のために，ワード，サブワード，あるいは音素に基づくアプローチが採られてきた．しかし，ワードベースでは訓練データの不足が，音素ベースでは精度の不足が，それぞれ性能向上の障害となっている．本論文は，音響的特徴，音素，サブワード，（音声認識結果としての）ワード，およびコンテキストといった，音声文書のさまざまなレベルの特徴を同時に利用するHSS（ホロ符号化音声検索）に基づく検索手法を提案する．本検索手法では異なる音響モデルや言語モデルを統合することもできる．HSSでは，音声文書の分割，評価，ランキングのために，シンボル構造という新しいデータ構造を考案した．シンボル構造に基づいて音声文書のさまざまな階層や粒度から特徴を検索するには，テキスト検索の手法を修正して用いる．詳細な分析により，HSSのアプローチが上述のSDRの問題を緩和できることを検証した．

Abstract

In recent years, Spoken Document Retrieval (SDR) becomes widely used in our life. Word-based, subword-based, and phoneme-based approaches have been used to improve the performance of SDR system. However, the known problems of SDR, e.g. insufficient training of Word-based system, low accuracy of phoneme-based system, limit the performance. This paper presents a HSS (Holo-coded Speech Search) based retrieval approach for SDR systems, which tries to make full use of the evidences of spoken documents on different levels, such as acoustic features, phonemes, sub-words, words (Automatic Speech Recognition Result) and context data. This method can also integrate different acoustic models and language models. In HSS, a novel data structure called Symbol Structure is designed and implemented to segment, score and rank the spoken documents. We introduce an improved text retrieval method to retrieve evidences from different hierarchies and granularities of spoken documents based on Symbol Structure. After a detailed analysis, we can see that the HSS approach is a feasible way to alleviate the problems in SDR. The experimental result shows it can meet the desire of practicability.

[*] リコーソフトウェア研究所（北京）有限公司
Ricoh Software Research Center(Beijing) Co.,Ltd.

1. Introduction

Nowadays, audio contents are continuously growing and filling our computers, networks and daily lives, such as broadcast news, TV shows, podcasts, lectures, videos, voice mails, (contact center or meeting) conversations, etc. How to find the wanted spoken document is a severe problem which we have to face and resolve. The need for intelligent indexing and retrieval of spoken documents is becoming increasingly pressing. With the maturity of the speech recognition, audio event detection and other speech processing techniques, Spoken Document Retrieval (SDR) has become feasible.

IR (information retrieval) is widely applied in text corpus to retrieve the documents that match queries[1). The aim of SDR is to provide similar functionality for spoken document corpus. It would be desirable to be able to retrieve in both text and spoken queries. The spoken documents can be converted into acoustic features, phonemes, sub-words or words, using speech related technologies, in order to be matched against queries[2). After the conversion, IR methods could be introduced to make index on the acoustic features, phonemes, sub-words or words.

The contributions of this paper include:

- Make full use of the evidences (acoustic features, phonemes, sub-words, words and context data) from the spoken documents on different levels. This has reduced several problems when using single evidence in the SDR, including insufficient training when using word evidence, low precision when using phonemes evidences and etc.
- Integrate different Acoustic Models and Language Models in SDR to handle spoken documents of different languages.
- Introduce a novel data structure called Symbol Structure, which contains all the evidences from the spoken documents. With it, the HSS method could be implemented.
- Improve text retrieval method to retrieve evidences from different hierarchies and granularities of spoken documents based on Symbol Structure, thus relatively high performance is achieved.

This paper is organized as follows: In section 2, we give an introduction to related works, section 3 describes a brief overview of the system architecture, section 4 introduces our HSS approaches, Symbol Structure and its usage for HSS, and section 5 gives a sample of implementation and experimental results. We reach our conclusions and discuss about future works in section 6.

2. Related Works

The easiest speech search method decodes continuous speech into text by Automatic Speech Recognition system with a dictionary and then uses common text search algorithms to find terms for obtained files. The main problem of this method is that the limited dictionary cannot recognize Out of Vocabulary (OOV) Words, such as names, acronyms, words from foreign languages.

Then a method that uses phonetic string representation for speech retrieval emerged[4), but it was clear that a large vocabulary recognition system can do better than an all phone recognition system. As a result, the method of combining both word and phone representations performs better than either method alone[5).

The phoneme-based approach is less effective than the word-based one, but is nonetheless effective enough to be used in practice[8), which can handle OOV problem and extend the recall.

Another method is sub-word based approach, which can be regarded as a combination of some continuous phoneme. It's an effective way to alleviate the OOV problem [9) 10).

Other methods are proposed to take possible recognition error into account. They utilize multiple speech recognition output alternatives in addition to the

1-best result. Lattice-based approaches such as position specific posterior lattices (PSPL)[6] and confusion networks (CN) are examples[7].

The acoustic features (eg. Mel-Frequency Cepstral Coefficients) are used for retrieving and classifying music files. Paper[11] proposes and develops a novel index structure for efficient content-based music retrieval, named the CF-tree, which adopts multiple acoustic features.

3. HSS method introduction

3-1 What is HSS?

Usually, we divide evidences of the spoken document into different levels. They are acoustic features, phonemes, sub-words, words (ASR results) and context data. Using evidence alone for SDR will experience merits and drawbacks, as Table 1 shows.

HSS is the abbreviation for Holo-coded Speech Search, in which the prefix 'Holo-' means "whole and entire". In this sense, the HSS method tries to make full use of all the evidences of the spoken documents on different levels and to integrate different Acoustic Models (AM) and Language Models (LM) for SDR.

In practical usage, one evidence would be enough for HSS, though it would then degenerate into a traditional SDR method. Text retrieval methods are introduced to retrieve on different hierarchies and granularities of the spoken documents. In order to achieve such goals, we design and implement a novel data structure called Symbol Structure, which is the unit of segmentation, scoring and ranking in the spoken documents retrieval.

3-2 HSS system overview

Figure 1 shows the main components of the HSS system, which include both Off-line process and On-line retrieval. The Off-line process prepares and extracts the acoustic features, phonemes, sub-words, words and context data of the speech corpus. After segmentation, all the features and evidences are filled in the Symbol Structure with time slots. Based on the Symbol Structure units, the Symbol Structure index is generated. After Off-line process, the On-line retrieval is possible. During the retrieval, the query (text or spoken) is transformed to the same kind of Symbol Structure too. This query Symbol Structure is used for matching, scoring, ranking and retrieval on the Symbol Structures of the spoken document corpus.

In the system, the FrontEnd, Recognizer, Pronunciation Model, Acoustic Model, Language Model and Acoustic Feature Extractor are out of scope of our discussion, though we just utilize the results of the speech recognizer. Indexing and retrieval techniques are introduced to improve the SDR system.

Table 1 The merits and drawbacks when using single evidence.

Evidence	Merits	Drawbacks
Acoustic features (the physical feature of the audio)	Don't require training. There is no language extensibility problem.	Inaccuracy for SDR
Phonetic code (the smallest unit of human voice)	Alleviate the training problem. Have fault-tolerance ability.	Not accurate as using sub-word and words.
Sub-words (the reasonable combination of phonetic code)	Between using the Phonetic code and words.	Require training by Large Corpus.
Words (word or sentence)	Usually, accuracy enough for SDR when the training is sufficient.	Require training by Large Corpus. If the Word Error Rate (WER) is high, the SDR will be greatly influenced.
Context data (manually annotation or materials affiliated to the audio)	The most useful and accurate material for SDR.	Hard to collect and annotate.

4. HSS approach for SDR

4-1 Evidence data generation

We generate acoustic feature, phoneme, sub-word transcription of corpus and word transcription by using speech recognizer. The time information is also generated from the speech recognizer.

Context data such as lecture PPTs, meeting minutes and transcripts are collected and associated with spoken document manually. While they are useful for spoken document retrieval, unlike phonemes, sub-words and words, there is no strict time information with them.

4-2 Segment

Spoken document segmentation is the process of identifying the boundaries between words, syllables, or phonemes in natural spoken languages. In the HSS approach, the spoken retrieval unit is Symbol Structure instead of words, syllables, or phoneme. We segment the spoken document by setting a constant time span, TL, and constant overlap time length, OTL. The length of TL should be larger than one word pronunciation length. With TL and OTL, the spoken document can be segmented. Given that the speech length is T. The first segment's start point is time 0; the end point is the smaller one between TL and T. If T is chosen, the segmentation process is finished. The next segment's start point is:

$$S_n = T_{ps} + TL - OTL$$

Wherein, the Tps means the previous segment's start point.

The end point is:

$$\mathrm{Min}(S_n+TL, T)$$

If the T is chosen, the segmentation process is finished.

Figure 2 shows one example of the segmentation process. The spoken document is divided into 5 segments, from S1 to S5. And the (T2, T1'), (T3, T2'), (T4, T3') and (T5, T4') are the time overlaps.

The context data may not have the fine-granular and precise time information. So the segmentation process is different. If there are event changes (page up, page down, mouse click, and so on) or manual annotation, the time information of such event or annotation is used to segment the context data. If not, the context data don't need to be segmented, and are just associated with the spoken document as a whole.

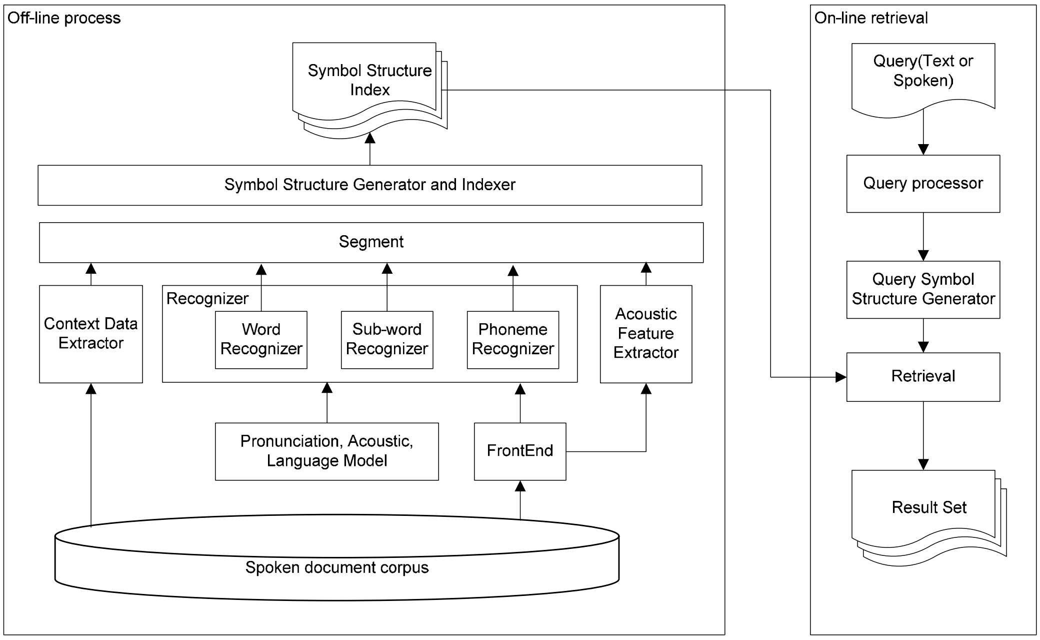

Fig.1 HSS System Overview.

4-3 Symbol structure

Figure 3 shows the sketch of the Symbol Structure. Actually, a Symbol Structure contains segments of acoustic feature codes, phonetic codes, sub-words, words and context data, and more importantly the time slot information of the segment. Symbol Structure is the unit of the retrieval process.

Usually, the Symbol Structure is filled by evidences from one Acoustic Model (AM) and one Language Model (LM). Further, information generated from different AM and LM can be used to fill the Symbol Structure, thus the training problem and language dependency could be alleviated to a certain degree, due to the different training materials of different AM and LM. For example, we can use phonetic code1 from AM1 and LM1 and phonetic code2 from AM2 and LM2 to fill the Symbol Structure.

4-4 Retrieval

In the HSS system, index is separately made on the acoustic feature codes, phonetic codes, sub-words, words or context data. It's improved index, in which the associated Symbol Structure and spoken document could be found.

To process the query (spoken or text), Speech Recognition, Text To Speech (TTS) and Phoneme Dictionary are used to convert the query into the acoustic features, phonetic codes, sub words, words. The query doesn't have any context data, so the word itself is

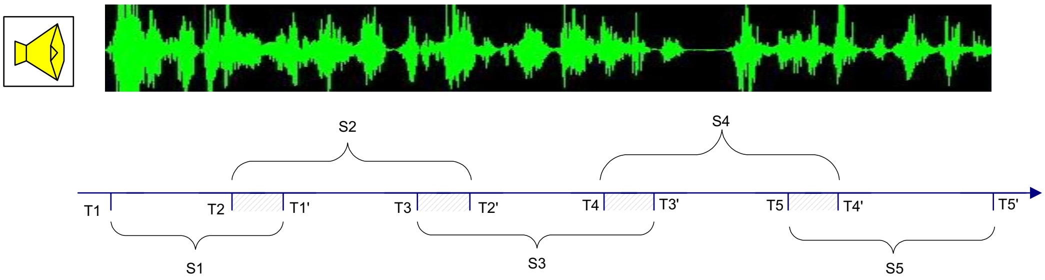

Fig.2 Spoken document segment method.

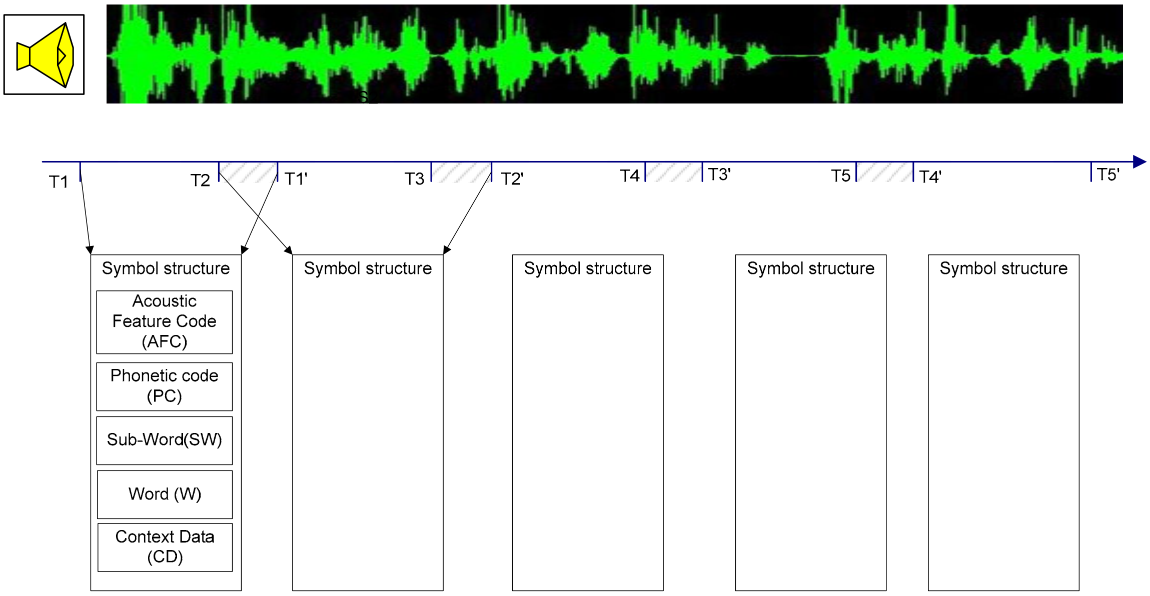

Fig.3 Symbol structure.

used to retrieve context data.

Acoustic features, phonetic codes, sub words, words and context data are used to calculate the matching separately. Each matching on any of the acoustic features, phonetic codes, sub words, words or context data contributes to the final matching between the Symbol Structure of the query and of the spoken document. To determine the matching, a weighted summation is needed. If the summation reaches or exceeds a threshold, the query Symbol Structure matches the certain Symbol Structure of the spoken document.

Figure 4 shows an example of the matching process. If the threshold is set to 2, and all the weights are set to 1, the final units that match the query are Symbol Structure 2 and 3.

The input query contains several terms (q_1, q_2, q_3, … , q_i, … , q_n).

After matching is calculated, we modify the Okapi BM25[15)] to rank the Symbol Structure matching. The Symbol Structure Frequency (SSF) is calculated as follows.

$$\mathrm{SSF}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

$n_{i,j}$ is the number of Symbol Structures in the spoken document (s_j) that match the Symbol Structure of query (q_i). $\sum_k n_{k,j}$ is the sum of the numbers of all the Symbol Structures in spoken document (s_j).

The Inverse Symbol Structure Frequency (ISSOF) in the spoken document corpus is calculated as follows.

$$\mathrm{ISSOF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

N is the total number of spoken documents in the corpus. $n(q_i)$ is the number of spoken documents containing q_i. ("Containing" here means having Symbol Structure(s) that matches the Symbol Structure of query (q_i)).

The score by SSF and ISSOF is calculated as follows.

$$\mathrm{SCORE}(S,Q) = \sum_{i=1}^{n} \mathrm{ISSOF}(q_i) \bullet \frac{\mathrm{SSF}(q_i,S)\cdot(k_1+1)}{\mathrm{SSF}(q_i,S)+k_1(1-b+b\cdot\frac{|S|}{avgSl})}$$

SSF(q_i, S) is q_i's Symbol Structure frequency in spoken document S, n is the number of the query terms.

The letter S in the expression |S| should be italic. (But the vertical bars should not be italic.) is the length of speech S in Symbol Structure, and $avgSl$ is average The letter S in the expression |S| should be italic.(But the vertical bars should not be italic.) in the spoken document corpus. k_1 and b are free parameters, usually chosen as k_1=2.0 and b=0.75.

The speech is ranked by the SCORE(S, Q), thus the retrieval results list is generated. The location of the query in the speech can be found easily, by using the time span information in the Symbol Structure. The retrieval and ranking methods are originated from Okapi BM25 (information retrieval methods).

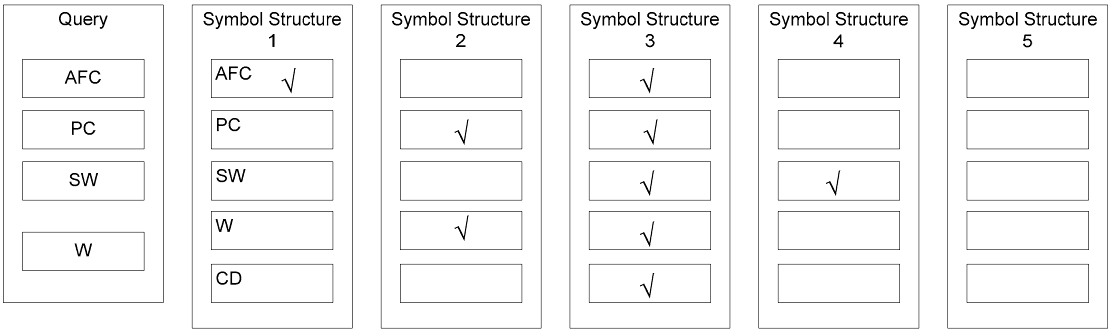

Fig.4 Match using symbol structure.

5. Prototype Implementation

5-1 Spoken document corpus

As Table 2 shows, the corpus used in the experiments is AMI Mic-Array Corpus [16], which is a multi-modal data set consisting of 100 hours of meeting recordings.

Moreover, AMI Head-Set Corpus[16], WSJ[21] and Voxforge[22] are selected for training the Acoustic Models and Language Models.

5-2 The acoustic model and language model

In the prototype implementation, we use several Acoustic Models and Language Models, as Table 3 and Table 4 shows.

5-3 Evidences and symbol structure generation

In the prototype implementation, 5 evidences are generated by Sphinx-4[12] speech recognizer, using the Acoustic Models and Language Models listed in Section 5.2. The 5 evidences are listed in Table 5. The Acoustic Feature Evidence and Context Data Evidence are not added in the prototype system.

As discussed in the Section 4.3, we set TL to 2 seconds. The length of OTL should be smaller than TL/2. Here we set it to 200 milliseconds. With the TL and OTL, the spoken document can be segmented, further, the evidences are segmented too.

Using the evidences segmented, the Symbol Structures are generated.

Table 2 AMI Mic-Array meeting corpus.

Corpus	Amount	Data size	Duration
AMI mic-array	127 meetings	7.87G	73.4h
AMI mic-array for retrieval	37 meetings	2.33G	22.0h
AMI mic-array for training	90 meetings	5.54G	51.4h

Table 3 Different acoustic models.

No	Acoustic Models	Description
1	WSJ Acoustic Model	Trained using Wall Street Journal corpus
2	AMI Mic-array Acoustic Model	Trained using AMI Mic-array corpus
3	AMI Head-set Acoustic Model	Trained using AMI Head-set corpus
4	AMI Head-set & Voxforge Acoustic Model	Trained using AMI Head-set data and Wall Street Journal corpus

Table 4 Different language models.

No	Language Models	Description
1	Word Language Model	Trained using AMI manual annotation and Voxforge manual annotation
2	Sub-word Language Model	Trained using AMI manual annotation (Sub-word format) and Voxforge manual annotation (Sub-word format)
3	Phonetic Code Language Model	Trained using AMI manual annotation (Phonetic Code format) and Voxforge manual annotation (Phonetic Code format)

Table 5 Different evidences.

No	Evidence	Description
1	Evidence1	Word (AMI Mic-array Acoustic Model and Word Language Model)
2	Evidence2	Sub-word (AMI Mic-array Acoustic Model and Sub-word Language Model)
3	Evidence3	Phonetic Code (AMI Mic-array Acoustic Model and Phonetic Code Language Model)
4	Evidence4	Sub-word (AMI Head-set Acoustic Model and Sub-word Language Model)
5	Evidence5	Word (AMI Head-set & Voxforge Acoustic Model and Word Language Model)

5-4 Matching and retrieval

To allow matching of phonemes, and sub-words we treat them as n-grams. Any sequence of symbols can be transformed into a sequence of n-grams; for example, the sequence “ABCDEF” is transformed into a sequence of 3-grams “ABC BCD CDE DEF”[19].

TRMeister (DBMS with high-performance full-text search functions)[20] is used to generate the index, and the retrieval using Symbol Structures is implemented in the application layer.

As for the single evidence, TRMeister is used as the search engine. All the single evidence search result is generated by TRMeister.

5-5 Results

In the experiment, Average MAP@TopN (Mean Average Precision)[20] and Average Precision@TopN are used as the evaluation metric. For the evaluation, 90 queries are selected from the AMI annotations. In the experiment, we aim to compare the evaluation metrics between the HSS method (retrieval using HSS method) and single evidence along (retrieval using single evidence).

Table 6 MAP@Top10.

Evidences	Map@Top10	Map@Top5
Evidence1	0.397163182	0.472764228
Evidence2	0.283414634	0.327560976
Evidence3	0.041799264	0.061869919
Evidence4	0.057433217	0.105284553
Evidence5	0.380979481	0.444796748
HSS	0.501191444	0.59097561

Table 7 Precision@Top10.

Evidences	Precision@Top10	Precision@Top5
Evidence1	0.52439	0.278049
Evidence2	0.419512	0.214634
Evidence3	0.085366	0.046341
Evidence4	0.080488	0.065854
Evidence5	0.507317	0.273171
HSS	0.617073	0.341463

The evaluation results are list below, from Table 6 to 7. And Fig.5～6 compares the MAP and Precision results between the HSS method and single evidence.

The results shows the HSS method improve the MAP and Precision. The MAP@Top10 (HSS) improves 26.1% of the best of the single evidence (Evidence1). The MAP@Top5 (HSS) improves 25.0% of the best of the single evidence (Evidence1). The Precision@Top10 (HSS) improves 17.7% of the best of the single evidence

(Evidence1). The Precision@Top5 improves 22.8% of the best of the single evidence (Evidence1).

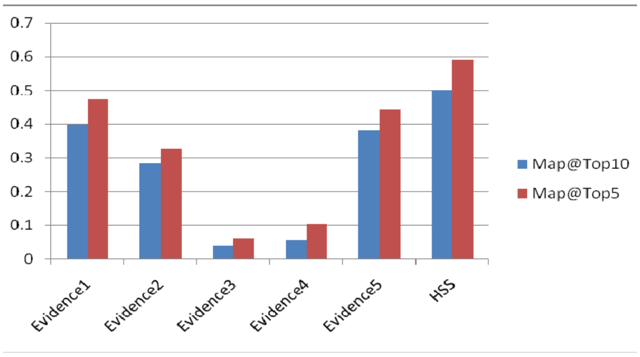

Fig.5 Map evaluation results.

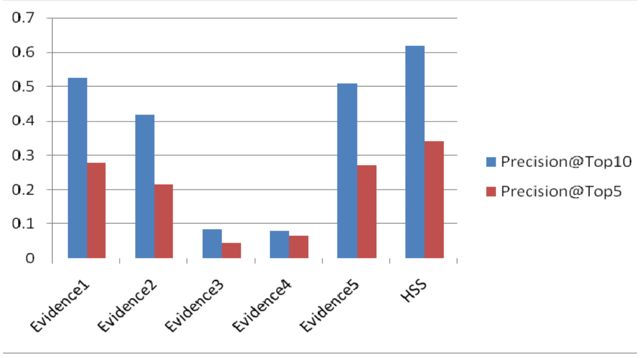

Fig.6 Precision evaluation results.

And we did the speed test of the HSS system as Figure 7 shows. First we enlarge the corpus to 2650 hours, and then deploy the HSS system on the Window XP server with Intel 6300, 1G memory, TRMeister on Linux with Intel Xeon 3.0G ×2, 8G memory.

The result shows that the system can have desirable practicability. There are two reasons why high speed is reached. 1) Due to the index on evidence separately, the Symbol Structure can be located quickly. 2) On every user request, the retrieval on evidences works in parallel.

Sometimes, the query contains most popular Symbol Structures which are linked to many audio time slots. This causes computing burden on hit process. In some cases, the max response time reaches to 10000ms.

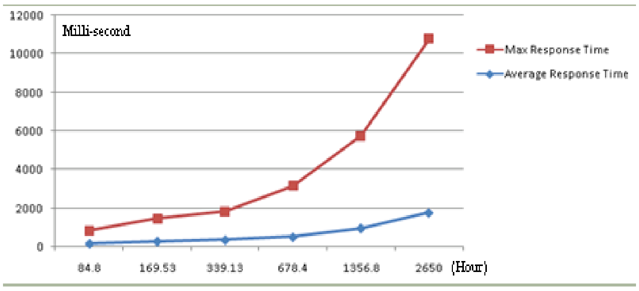

Fig.7 Speed test of the HSS system.

6. Conclusions and Future Work

We have explored the approach of SDR based on HSS code. Using the method, the multi-level evidences from the spoken documents could be taken part in the retrieval. Not only the word evidence (Automatic Speech Recognition result) but also the phonetic codes, and sub-words are used, so the influence of the training problem, Out of Vocabulary Problem of ASR, is reduced to a certain extent. Each of evidences contributes to the final matching between the query and the spoken document, so the retrieval precision is improved compared to single evidence method. The MAP@Top10 (HSS) improves 26.1% of the best of the single evidence (Evidence1). The MAP@Top5 (HSS) improves 25.0% of the best of the single evidence (Evidence1). The Precision@Top10 (HSS) improves 17.7% of the best of the single evidence (Evidence1). The Precision@Top5 improves 22.8% of the best of the single evidence (Evidence1).

The text retrieval method and Symbol Structure are introduced to make the index, matching, scoring and ranking, so the high-speed spoken documents retrieval can be realized. Furthermore, the time slot information in the Symbol Structure makes it easy to locate matching position in the spoken documents. The experiment result has shown it can meet the desire of practicability.

In terms of future work, it's necessary to make a comparison between our approach and other existing systems and methods. Do further experiments to measure the advantages of training, language independence and fault-tolerance. The proposed HSS method should be also evaluated on a larger spoken document corpus.

References

1) G. Salton and M. J. McGill: Introduction to Modern Information Retrieval, 1st edition, McGraw Hill, (1983), pp. 229-237.
2) K. F. Lee: Automatic Speech Recognition: The Development of the SPHINX System, Kluwer Academic Publishers, (1989), pp. 63-90.
3) W. B. Cavnar: Using an n-gram-based document representation with a vector processing retrieval model, In Proceedings of the Third Text Retrieval Conference (TREC-3), (1994), pp 269 -227.
4) Schäuble, P. and Wechsler, M: First Experiences with a System for Context Based Retrieval of Informatin from Speech Recordings, In IJCAI-95, Workshop on Intelligent Multimedia Information Retrieval, Maybury, M.T., (1995), pp 59-69.
5) James, D.: A System for Unrestricted Topic Retrieval from Radio News Broadcasts, In Proceedings of ICASSP-96, Atlanta, GA, May (1996), pp. 279-282.
6) C. Chelba, J. Silva, and A.Acero: Soft indexing of speech content for search in spoken documents, Computer Speech and Language, vol, 21, no. 3, July (2007), pp. 458-478.
7) T. Hori, I.L. Hetherington, T.J. Hazen, and J.R. Glass: Open-vocabulary spoken utterance retrieval using confusion networks, in LCASSP, (2007), pp 73-76.
8) Corinna Ng, Ross Wilkinson, Justin Zobel: Experiments in spoken document retrieval using phoneme n-grams, Special issue on accessing information in spoken audio, vol 32, (2000), pp. 61-77.
9) Ville T. Turunen, Mikko Kurimo: Indexing Confusion Networks for Morph-based Spoken Document Retrieval, SIGIR 2007 Proceedings, July 23?27, (2007), pp. 632-635.
10) K. Ng.: Subword-based Approaches for Spoken Document Retrieval, PhD thesis, Massachusetts Institute of Technology, (2000), pp. 105-130.
11) Bin Cui, Jialie Shen and Gao Cong: Exploring Composite Acoustic Features for Efficient Music Similarity Query, International Multimedia Conference, 2006, pp. 412-420.
12) The CMU Sphinx Group Open Source Speech Recognition Engines, http://cmusphinx.sourceforge.net/html/cmusphinx.php
13) CMU Sphinx Wall Street Journal Acoustic Models, http://www.inference.phy.cam.ac.uk/kv227/sphinx/acoustic_models.html
14) Hub 4: Business Broadcast News, http://www.speech.cs.cmu.edu/air/papers/hub4/proceedings_paper1.html
15) Okapi BM25, http://en.wikipedia.org/wiki/Okapi_BM25
16) AMI Corpus, http://corpus.amiproject.org/
17) Vector Quantization, http://www.mqasem.net/vectorquantization/vq.html
18) Statistical Language Modeling (SLM), http://homepages.inf.ed.ac.uk/lzhang10/slm.html
19) N-gram, http://en.wikipedia.org/wiki/N-gram
20) Tetsuya Ikeda,Hiroko Mano,Hideo Itoh,Hiroshi Takegawa,Takuya Hiraoka,Shiroh Horibe,Yasushi Ogawa: TRMeister: A DBMS with High-Performance Full-Text Search Functions, IEEE Computer Society Washington, DC, USA, 2005, pp 958 – 967.
21) WSJ, Wall Street Journal Corpus, http://www.idiap.ch/mmm/corpora/ami_wsj
22) Voxforge, http://www.voxforge.org/