A vector-based approach to broadcast audio database indexing and retrieval

Wang, Lei; Li, Haizhou; Chng, Eng Siong

2007

Wang, L., Li, H., & Chng, E. S. (2007). A vector-based approach to broadcast audio database indexing and retrieval. 2007 IEEE International Conference on Multimedia and Expo, pp512-515.

https://hdl.handle.net/10356/97783

https://doi.org/10.1109/ICME.2007.4284699

A VECTOR-BASED APPROACH TO BROADCAST AUDIO DATABASE INDEXING AND RETRIEVAL

Lei Wang[1], *Haizhou Li*[1,2], *and Eng Siong Chng*[1]

[1]School of Computer Engineering, Nanyang Technological University, Singapore 639798
{wang0161, aseschng}@ntu.edu.sg
[2]Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
hli@i2r.a-star.edu.sg

ABSTRACT

This paper proposes a novel framework to index and retrieve audio content from broadcast database that contains both speech and music. In this framework, we model the acoustic events using hidden Markov models, which are then used to decode the audio content. The decoding results in the form of acoustic token sequence and acoustic lattice are used to generate features for indexing and retrieval with the vector space model. Experiments were carried out on the TRECVID database and the results showed that the proposed framework is effective in audio information retrieval. The results also showed that the features generated from the acoustic lattice provide more accurate information than token sequence.

1. INTRODUCTION

The huge amount of multimedia data such as the audio database of radio or music recordings [1] available on the Internet cannot be easily queried and retrieved as these data are unstructured. That is, it is not easy to locate desired segments unless manual transcript is available. This prompts us to examine methods for automatic indexing and retrieval of audio database.

In the late 90's, researchers were more interested in building up search engine for spoken documents on the Web [2]. These systems face two major issues: i) out-of-vocabulary queries and ii) high word error rate. To address the problems, researchers proposed vocabulary-independent approach to speech indexing. In their implementation [3], speech is first transcribed into phonetic sequence and then indexed. Searching is then performed on the phonetic transcription. Although this approach partially addresses the out-of-vocabulary problem, it is not robust due to high phonetic recognition error.

Recently, researchers have designed many effective frameworks for music indexing and retrieval. Examples include the music structure based system [4] and the query-by-humming systems [5]. As these frameworks were developed specifically for music audio signal, they are unsuitable for general multimedia data such as TV broadcast which contains both speech and music. To address this problem, early researchers explored techniques to first separate speech from music and then apply different methods for speech and music [1].

Inspired by the progress in spoken document retrieval, we believe that the acoustic modeling and decoding process can work for audio indexing as well. In speech recognition, acoustic tokens are typically defined by phonetic experts. They can be phones, words or subwords. The training or modeling of these acoustic tokens is carried out in a supervised manner and is therefore language and task dependent. As such, the speech data need to be manually transcribed. Obviously, the automatic speech recognition method cannot be implemented for music transcription directly because definition of acoustic tokens for music is not as straightforward.

For audio transcription, we explore a unified method to model acoustic tokens for both speech and music signals. To circumvent the need of definition of acoustic tokens and manual transcription, we propose a data-driven acoustic modeling technique. A new framework for audio signal tokenization motivated by the Acoustic Segment Model (ASM) [6] will be studied.

The audio signal is subsequently decoded using the ASM. The decoded information can be in two forms: acoustic token sequence and acoustic lattice. In this paper, we examine the performance of the system using these two different types of information. The features used by our system for indexing are vectors containing the n-gram statistics from the decoded acoustic token sequence or lattice. Finally, we examine a query-by-example problem to illustrate the performance of the proposed approach.

Section 2 discusses acoustic modeling for acoustic tokens in speech and music signal. Section 3 proposes a novel audio indexing and retrieval framework. Section 4 reports the experiment results and finally, we conclude in Section 5.

2. DATA-DRIVEN ACOUSTIC MODELS

In this section, we discuss the procedure to create our data-driven musical acoustic models and briefly describe the speech phone models used in our system.

For our system, we used 39 phonemes to model the English language. A set of 39 hidden Markov models (HMMs) that describe the phonemes and 1 additional silence model are trained using the Wall Street Journal database using HTK [7]. Each model consists of three states and each state has 16 Gaussian mixtures.

As broadcast audio normally consists of both speech and musical contents, speech-only phonetic models are insufficient for the system. In order to transcribe music signal, music token models are created using a data-driven approach. To compete with speech phonetic models during the decoding process, we use an equivalent number of music token models. Li *et al.* [8] introduced a bootstrapped ASM training procedure for universal phonetic tokenization. This unsupervised approach does not require transcription of the training data such as musical corpus. Vector quantization is first applied to produce pseudo labels to initialize the training procedure.

The features used to train the musical HMMs are as the same as those used to train the English phonetic HMMs. The features are the MFCC features as well as their first and second order time derivative. The ASM training process is applied to automatically generate the 40 HMMs for music in the following manner:

Step 1) Segment the music signals in the training corpus into short segments of equal length.

Step 2) Cluster the segments into 40 clusters with k-means clustering; Label the segments in the entire corpus with the cluster identity.

Step 3) Create 40 HMM models. Adapt these 40 HMM models using the training corpus with the labeled cluster identity found by Step 2.

Step 4) The trained 40 HMMs are used to decode the training corpus. The recognized tokens whose duration is less than a pre-defined threshold are removed.

Step 5) The HMMs are re-adapted using the new token labels found after Step 4.

Step 6) Repeat 4 -5 until convergence.

The music corpus consists of 50 pieces of pure music played by five different instruments and another 30 pieces of songs in three styles. All the music pieces are converted from MP3 format and down-sampled to 11 kHz to be consistent with the speech training data. With 40 phoneme models and 40 music ASM models, we are now able to convert an audio signal into text-like transcript.

3. AUDIO INDEXING AND RETRIEVAL

One main challenge in audio indexing is to have a representation of the audio signal that is expressive, precise and is still computationally cheap. We consider the vector space modeling (VSM) method to transform a segment of audio transcript into a vector representation which contains the segment's feature *n*-gram statistics. VSM can capture the statistical temporal information of the speech signal and has been used in applications such as language identification. In [8], a method was proposed to extract the *n*-gram statistics of the phonemes in a speech segment to form a *bag-of-sounds* vector. Inspired by this method, we propose using the decoded acoustic sequence or lattice to form an *n*-gram vector to index the audio segment.

The HMM decoder can generate results in different formats, e.g., top-1 token sequence, top-n token sequences, and acoustic lattice. The top-1 token sequence can be easily obtained by the Viterbi search. The acoustic lattice is an intermediate representation of the decoding process [7]. The acoustic lattice is defined as a connected loop-free directed graph with each node representing a time frame and each arc representing a token hypothesis with a likelihood score. At every time frame during decoding, the potential token sequences are updated and the higher ranked tokens ending at that frame are stored. Therefore, the lattice stores multiple token hypotheses for every point throughout the audio segment [9]. In this paper, we study the performance of the retrieval system using both the top-1 token sequence and the acoustic lattice as shown in Figure 1.

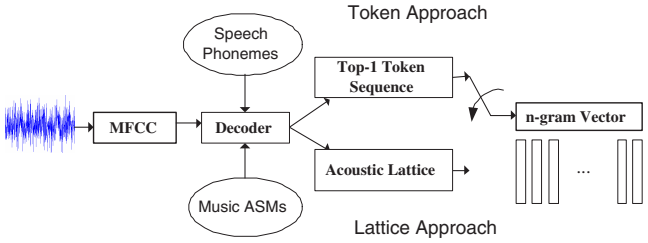


Fig. 1. Two approaches of audio decoding.

3.1. Generating the Features for Indexing and Retrieval

The features used for the HMM are the MFCCs, its delta and double-delta extracted from a 25 millisecond frames of the audio signals with a 10 millisecond hop size. This forms a feature vector of 39 coefficients. These raw MFCC features are not directly used for indexing and retrieval because the performance of the retrieval system would be poor as these features are low-level features which do not capture temporal dynamics.

In our study, the features used for indexing and retrieval are the *n*-gram statistics of the acoustic token sequence and

acoustic lattice decoded using the combination of the 40 English phonetic HMMs and 40 musical HMMs. We named the process to generate these two features as the token approach and the lattice approach respectively.

To generate the feature for the token approach, we used the decoded top-1 token sequence and count the frequencies of each *n*-gram terms. Similarly for the lattice approach, we derive an *n*-gram vector by estimating the expected counts of the frequencies as implemented in the SRILM tool [10]. The expected count of one *n*-gram term is computed using the lattice posterior probabilities of all the arcs corresponding to this term. Details of generating *n*-gram vector from lattice can be found in [11]. If we consider the *n*-gram counting in the top-1 token sequence as the hard-counting, then the process of counting the occurrence of the events in the lattice is considered soft-counting. All the vectors are normalized to unit length. We will report the comparison of these two approaches in Section 4.

We index every t-second window with s-second shift to generate an *n*-gram vector in the entire audio archive.

3.2. Retrieve Audio Segments

Given an audio query, the query is first converted into an *n*-gram vector. We then measure the similarity of the query vector against all the indexed vectors in the archive. Figure 2 illustrates the retrieval process.

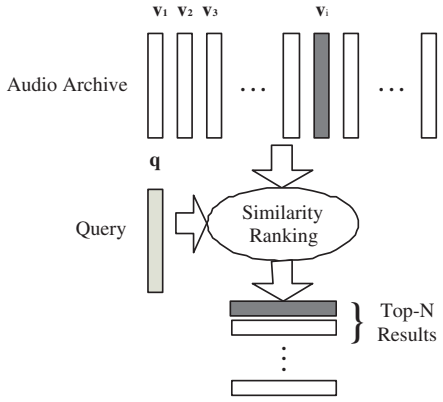


Fig. 2. Audio segment retrieval process. A linear structure is adopted to represent the data archive.

The Pearson correlation coefficient $r \in [-1, 1]$ is used to measure the similarity between the query vector $\mathbf{q}$ and every vector $\mathbf{v_i}$ in the database. Let q_k, and v_{ik} represent the k-th element of $\mathbf{q}$ and $\mathbf{v_i}$ respectively, and the Pearson correlation coefficient between vector $\mathbf{q}$ and $\mathbf{v_i}$ is defined by

$$r_i = \frac{\sum_{k=1}^{n}(q_k - \overline{q})(v_{ik} - \overline{v_i})}{\sqrt{\sum_{k=1}^{n}(q_k - \overline{q})^2 \sum_{k=1}^{n}(v_{ik} - \overline{v_i})^2}}. \quad (1)$$

Using the Person correlation measure, the segments in the entire archive can be ranked in terms of similarity to the query.

4. EXPERIMENTAL RESULTS

Experiments using database extracted from TRECVID 2003 and 2004 were carried out. TRECVID data were recorded from the ABC and CNN network news during the year 1998 and the data are video clips. The audio information from 28 hour video clips in MPEG format were extracted. We manually label all the commercials presented in these 28 hour database and select 100 unique commercials as our query commercials. In this database, there are in total 242 additional instants for the 100 query.

Several experiments using various duration of the query were conducted. For each set of experiments, query segments of the required length are cut from the query commercials. The lengths of the queries (t) are chosen as 3, 5 and 10 seconds in our experiments. We set the hop size s to 0.5 second.

We compare the performances of the token based and lattice based approaches using both unigram (n = 1) and bigram (n = 2) vectors as features. Since there are a total of 80 token models, the unigram vector has 80 dimensions. The bigram vector has 6561 (81x81) dimensions as we consider the starting point and ending point of the sequence as one extra token.

To evaluate the retrieved results, retrieved boundary of the segment within ± 1 second of the actual boundary is considered as correct answer. Because different queries have different numbers of instants present in the database, we retain the top-10 results for evaluation as no query has more than 10 instants.

The recall and Mean Average Precision (MAP) are used to evaluate the system's performance and are defined as follows:

$$\text{Recall} = \frac{Number\ of\ relevant\ results\ retrieved}{Total\ number\ of\ actual\ relevant\ results}, \quad (2)$$

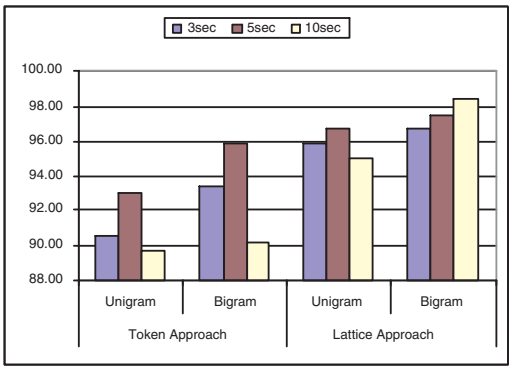
$$\text{MAP} = \frac{1}{M}\sum_{m=1}^{M} AP_m, \quad (3)$$

$$AP_m = \frac{1}{K}\sum_{i=1}^{K}\frac{i}{r_i}, \quad (4)$$

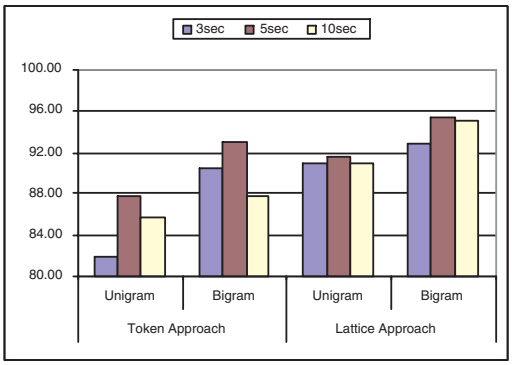
where M is the total number of queries, K is the number of relevant results retrieved for query m, and r_i is the rank of relevant result i.

The recall and MAP of our experimental results are illustrated in Figure 3(a) and 3(b) respectively. The results showed that the lattice approach outperforms the token approach. This can be explained by the fact that soft-counting avoids the imprecision caused by the Viterbi decoding.

As both approaches utilize the temporal information of the signal, the window size t is an essential parameter to the system. Figure 3(a) shows the highest recall 98.35% is achieved



(a) Recall



(b) Mean Average Precision

Fig. 3. Performance comparisons between token approach and lattice approach.

by the 10 second window based on the lattice bigram vectors. Its corresponding MAP is also above 95% which shows our method is not only efficient but also precise. It is obvious that the longer the value of t, the more information the segment will contain. As discussed, the decoded lattice retains more information than top-1 token sequence. Hence it is not surprising that the lattice based approach with unigram count has comparable performance to the token based bigram approach.

5. CONCLUSION

This paper proposed an audio indexing and retrieval framework in which both English phonetic models and music ASMs are used to convert audio segment into sequence of acoustic tokens. We further applied the vector space modeling approach for the audio content indexing and retrieval task. The experiments show that the proposed framework is effective in broadcast audio indexing and retrieval. We also found that lattice approach is more accurate than token approach.

We have only applied the vector space modeling technique to broadcast audio indexing and retrieval. We believe that the same technique can be extended to general purpose audio information retrieval applications as well.

In future research, we will involve reverse indexing so that it can locate query vectors more efficiently. The framework may also incorporate with video indexing system to process multimodal media.

6. REFERENCES

[1] S. Kiranyaz, A. F. Qureshi, and M. Gabbouj, "A generic audio classification and segmentation approach for multimedia indexing and retrieval," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 3, 2006.

[2] D. Abberley, S. Renals, and G. Cook, "Retrieval of broadcast news documents with the THISL system," in *Proc. ICASSP '98*, Seattle, WA, May 1998, pp. 3781–3784.

[3] P. Yu, K. Chen, C. Ma, and F. Seide, "Vocabulary-independent indexing of spontaneous speech," *IEEE Trans. Speech and Audio Processing*, vol. 13, no. 5, 2005.

[4] N. C. Maddage, H. Li, and M. S. Kankanhalli, "Music structure based vector space retrieval," in *Proc. ACM SIGIR '06*, Seattle, Washington, USA, Aug. 2006, pp. 67–74.

[5] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming: Musical information retrieval in an audio database," in *Proc. ACM MM '95*, San Francisco, California, USA, Nov. 1995, pp. 231–236.

[6] C.-H. Lee, F. K. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," in *Proc. ICASSP '88*, New York, NY, 1988, pp. 501–504.

[7] S. Young et al., *The HTK Book (for HTK Version 3.2.1)*, Cambridege University Engineering Department, 2002.

[8] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 1, 2007.

[9] D. A. James and S. J. Young, "A fast lattice-based approach to vocabulary independent wordspotting," in *Proc. ICASSP '94*, Adelaide, Austrailia, Apr. 1994, pp. I/377–I/380.

[10] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. ICSLP '02*, Denver, USA, Sept. 2002.

[11] J.L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," in *Proc. ICSLP '04*, Jeju, South Korea, Oct. 2004.