

Spoken Document Retrieval (SDR) for Broadcast News in Indian Languages

Chirag Shah

Dept. of CSE

IIT Madras

Chennai - 600036

Tamilnadu, India.

chirag@speech.iitm.ernet.in

A. Nayeemulla Khan

Dept. of CSE

IIT Madras

Chennai - 600036

Tamilnadu, India.

nayeem@speech.iitm.ernet.in

Abstract

Audio data carries much more information than text and are widely used in many applications like broadcast news. But there is a lack of good techniques for retrieving some information from such data. We are trying to address this problem, called the Spoken Document Retrieval (SDR) for Indian languages, by matching speech-segments without actual recognition. The objective is to index the audio data in suitable format so that we can search in this indexed database for a given query. The output can be some audio file(s) or segment(s). At present, the experiments are done on labeled data and text query. The system is interactive and works based on probabilistic searching. Further work is being done on partially labeled as well as unlabeled speech data and speech query. The techniques being explored involve hypothesizing the word boundaries, developing phoneme-level Hidden Markov Models (HMMs), phoneme-boundary marking using segmental k -means algorithm, deriving syllable-level models, and Vector Quantization. We provide the outline and various issues related to both these scenarios: labeled and transcribed data, and unlabeled and non-transcribed data. The implementation details of the former case is given and the latter case is analyzed. We show that we could achieve reasonably well precision and recall for speech documents, too.

1 Introduction

Today a database does not mean a collection of textual data only, but it incorporates a variety of data including audio and video. With the rapidly growing use of multi-media information, an exponentially increasing number of spoken document archives, such as broadcast radio, television programs, digital libraries etc. are accumulated and made available [min Wang, 2000]. However, most of them are simply stored, and are difficult to be reused further because of lack of some efficient retrieval technology. Development of mechanism for retrieving speech information has thus become more and more important.

In general, we have data in three formats: text, audio, and video. Text data is easy to retrieve, but conveys information in a limited sense. Video data contains more information, but is very difficult to store and retrieve. Whereas, audio data falls in between these two extremes both in terms of carrying information and ease of retrieval. Moreover, techniques used for Spoken Document Retrieval (SDR) may also be extended for video data retrieval.

In this paper we describe a system to retrieve spoken document(s) from a set of spoken documents (mainly news broadcast in some Indian Language) based on the query given by the user. We analyze the problem considering various possible scenarios. Some of these ideas are implemented and the others are in the progress. The rest of the paper is organized as follows. Section 2 clarifies the problem by specifying various scenarios and assumptions. We divide SDR task in two categories: SDR with labeled and transcribed data (section 3), and SDR with unlabeled and non-transcribed data (section 4). The paper is concluded in section 5.

2 Problem Specifications

Spoken documents, which are broadcast news in some Indian language, are available. We need to index these documents and store that information in a format so as to facilitate their retrieval later by a query given by the user.

The query for retrieval can be in either text or speech form. We need to compare the terms of the query with the indexed documents. We then find the document(s) in which matching term(s) occurs, and then present it to the user. Upon request, the system should also be able to further prune the search and do matching among some subset of the results.

The SDR task can be divided in two categories depending upon the situation:

1. Labeled and transcribed data
2. Unlabeled and non-transcribed data

First category is the case when spoken documents are manually transcribed and annotated or they could be the output of a Large Vocabulary Continuous Speech Recognizer (LVCSR) [Stolcke *et al.*, 2000; Zheng *et al.*, 2002]. Second category is the case when only spoken documents are available

without any labeling or transcription. The following sections describe these two cases in detail and outline the problems and their possible solutions. Some of the implementation details are also given.

3 SDR with Labeled and Transcribed Data

This is the case when we have labeled spoken documents as well as their transcription. Although this case does not seem important by practical aspect, the motivation behind doing some experiments with this data is to understand various issues of SDR assuming that we have an LVCSR. Moreover, we also want to study problems related to storage and searching in speech databases.

In order to perform efficient retrieval of speech documents all the stories generated from the database needs to be properly indexed using appropriate indexing methods and stored in relational databases enabling SQL type query retrieval. To each of the transcribed story of the speech document is associated a time index of the original speech file. On locating a news story we can get the corresponding speech file.

In speech queries we generally say *I would like to, Is there any information about* etc. Such parts of speech are not essential for our retrieval process and they have to be ignored to obtain better results. These are called stop-words [Yang and Wilbur, 1996]. Hence stop-word removal from the query should be performed before any query processing is done.

The document and query can be represented in terms of vectors where each component of the vector is an indexing term. The weight of each term i in the vector for document d is popularly found as

$$d[i] = 1 + \log(f_d[i]) \tag{1}$$

and the weight of term i in the vector for query q is

$$q[i] = (1 + \log(f_q[i])) \cdot IDF(i) \tag{2}$$

where

$f_d[i]$ = frequency of the term i in the document d

$f_q[i]$ = frequency of the term i in the query q

$d[i]$ = term frequency in document d

Inverse Document Frequency (IDF) of a term i is defined as

$$IDF(i) = \log\left(\frac{N_D}{N_{Di}}\right) \tag{3}$$

where N_D = total number of documents in the database

N_{Di} = number of documents in which term i occurs

The intuition behind Inverse Document Frequency (IDF) is the following. The more the number of documents in which a term is occurring, the less will be its IDF, and so the less its importance, and vice versa.

Once represented in terms of vectors, we can retrieve a document based on the similarity measure which is the normalized inner dot product between the document and the query vectors. The similarity between the query q and the document d is given by

$$Sim(q, d) = \frac{q \cdot d}{\|q\| \|d\|} \quad (4)$$

$$= \sum_{i \in q} \frac{q[i] d[i]}{\|q\| \|d\|} \quad (5)$$

Another simple vector space matching technique is using subword (say syllable) approach. This requires finding a feature vector V_d , for each spoken document d in the database D , which includes the acoustic score information of syllable pairs appearing adjacently in each spoken document. To obtain scores of syllables we need to do syllable segmentation and recognition of syllable hypothesis using some dynamic programming techniques [Held and Karp, 1961]. Similarly, we need to repeat the same procedure to obtain a feature vector V_q for the speech query q .

If the query is in text form then the feature vector T_q is formed by using frequency counts for syllable pairs instead of acoustic scores.

IDF for syllable pairs can be defined as

$$IDF(s_i, s_j) = \log \left(\frac{N}{N_{s_i, s_j}} \right) \quad (6)$$

where

N = the total number of speech documents

N_{s_i, s_j} = the number of documents where the syllable pair appears

As explained above about the significance of IDF, the syllable pair with smaller $IDF(s_i, s_j)$ value means it is of less discriminative for information retrieval.

As an initial search for a given query q , a feature vector V_q for the speech query and T_q for the text query is formed. Using the $IDF_q(s_i, s_j)$ we can get k most important syllable pairs for the query. Then for every feature vector d the relevant score is computed by summing up the acoustic scores of these k syllable pairs, and keeping only the top n . Now, from this reduced set of documents D_v^* the relevance between the input query and database is found as follows.

For speech document,

$$U_d = \text{diag}(V_d \cdot IDF_d^T), \quad \text{for } d \in D_v^* \quad (7)$$

$$= (V_d(s_1, s_1) \times IDF_d(s_1, s_1), \dots, V_d(s_i, s_j) \times IDF_d(s_i, s_j), \dots, \\ V_d(s_n, s_n) \times IDF_d(s_n, s_n)) \quad \text{n, number of syllables} \quad (8)$$

For speech query,

$$U_q = \text{diag}(V_q \cdot IDF_q^T), \quad (9)$$

$$= (V_q(s_1, s_1) \times IDF_q(s_1, s_1), \dots, V_q(s_i, s_j) \times IDF_q(s_i, s_j), \dots, \\ V_q(s_n, s_n) \times IDF_q(s_n, s_n)) \quad \text{n, number of syllables} \quad (10)$$

For text query,

$$T_q = \text{diag}(T_q \cdot IDF_q^T), \quad (11)$$

$$= (T_q(s_1, s_1) \times IDF_q(s_1, s_1), \dots, T_q(s_i, s_j) \times IDF_q(s_i, s_j), \dots, \\ T_q(s_n, s_n) \times IDF_q(s_n, s_n)) \quad \text{n, number of syllables} \quad (12)$$

The cosine measure for finding relevance score is given by

$$R(d, q) = \cos(U_d, U_q) = \frac{U_d \cdot U_q}{|U_d||U_q|} \quad \text{for } d \in D_v^* \quad (13)$$

This method can be extended for syllable triples and above. Any search technique employed will result in a number of results. Therefore, we need to determine thresholds based on which n best results will be retained.

If the spoken document is represented in terms of syllables or by any other subword unit, then the factors to be considered in using syllable or phoneme n -grams of larger width are: if they are very long then the possibility of this combination occurring across documents reduces, whereas if the size of the syllable string is small say two or less, then there are more number of matches, some of which may be irrelevant. Hence it needs to be experimentally determined which is the optimum length of the syllable string to search for.

Another issue in the subword representation or retrieval is that the syllable string we search may span words, in case it will throw up irrelevant results. This also needs to be addressed.

4 SDR with Unlabeled and Non-Transcribed Data

In case of unlabeled data, we can divide the SDR task in terms of five subtasks. Table 1 lists these subtasks. It also gives comparison between text and speech for all these subtasks.

We describe these subtasks in the following subsections.

Subtask	Text	Speech
Segmentation	Already segmented	Fixed/Varying
Representation	Words	Co-efficient
Indexing	Alphabetically	VQ index
Matching	Strings	Distance measures
Retrieval	Display	Play

Table 1: Subtasks of SDR with comparison between text and speech

4.1 Segmentation

First, we apply the Word Boundary Hypothesization technique [Rajendran and Yegnanarayana, 1996] for getting word level marking in the given spoken document. This technique uses the knowledge of f_0 patterns of *Hindi* and thus, it can work only with *Hindi* and similar Indian languages. Using some language dependent rules, it hypothesizes the word boundary with more than 70% accuracy.

One experiment showed that for a spoken document containing 1316 words, the technique could detect 1227 boundaries.

4.2 Representation

After getting segments, we extract some features like Linear Prediction Cepstrals (LPC) [Makhoul, 1975] from every segment. Thus, each segment is represented in terms of feature vectors.

4.3 Indexing and storage

Once the spoken documents are converted in some suitable format as discussed in the previous subsection, the next step is to index the terms. For this particular application - SDR, the well-accepted and the most used format for storing and indexing the terms is Inverted File Structure (IFS) [Jeong and Omiecinski, 1995; CACHED and Vina, 2002]. The basic format of this structure is as follows:

$$\langle t, (d_1; p_1, p_2, \dots), (d_2; p_1, p_2, \dots), \dots, (d_i; p_1, p_2, \dots) \rangle \quad (14)$$

where t = term (represented as discussed in the previous section)

d_i = document i

p_j = position j in that document

If there are too many positions where the term is occurring, then it will increase the space overhead. In that case, instead of storing positions, we can divide the document in blocks of some terms and use the block number to represent the position of a term. While searching, we can reach to the block directly, but we will have to do linear search in that block. Thus, this block-level approach is a trade-off between space and time.

4.3.1 Matching

After previous three off-line steps, we have stored the spoken documents in the database. Now, our next step is to take a query and perform searching in the database for this query on-line. Here, problem is to address the variability in duration of utterances. Dynamic Time Warping (DTW) [Chang, 1972] is a popular technique to take care of the variations in speaker rate, and variations in durations of the same utterance, that is, it will perform time alignment and normalization.

4.4 Retrieval

Depending on the relevance score, best k results are retrieved, where k is generally some value between 5 and 20. These results are then played for the user. The experiments that we have done over nearly 100 queries could achieve 58% of recall and 73% of precision considering the first 10 results ($k = 10$). Increasing k can increase the recall, but lower the precision.

5 Conclusion

Direct searching in some spoken document is not possible due to the fact that two audio signals cannot be matched in a straight forward way. Therefore, we require some mechanism to represent the spoken document in some other format than waveform to facilitate the matching. In this paper we discussed several issues related to such problem.

We showed that once we get a sequence of numbers for every utterance, we can index that segment. This indexed database can now be used for various application like searching, retrieval etc. For searching application, we take the query in spoken form and apply the same procedure used for indexing and get the corresponding sequence of numbers for that query. This can now be compared to the indexed database. For retrieval application, we should store some additional information with the index like position of that utterance, term frequency etc. We highlighted several issues of SDR and proposed some novel techniques for dealing with some of them. Process for solving many other problems is in the progress.

References

- [Cacheda and Vina, 2002] Fidel Cacheda and Angel Vina. Inverted files and dynamic signature files for optimization of Web directories. In *The Eleventh International World Wide Web (WWW) Conference*. 2002.
- [Chang, 1972] C. Y. Chang. Dynamic programming as applied to feature subset selection in a pattern recognition system. In *Proceedings of the ACM annual conference*, pages 94–103, 1972.
- [Held and Karp, 1961] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. In *Proceedings of the 16th ACM national meeting*, pages 71.201–71.204, 1961.
- [Jeong and Omiecinski, 1995] B. S. Jeong and E. Omiecinski. Inverted file partitioning schemes in multiple disk systems. *IEEE Transactions on Parallel and Distributed Systems*, 6, February 1995.

- [Makhoul, 1975] John Makhoul. Linear prediction: A tutorial review. *Proceedings of IEEE*, 63:561–580, 1975.
- [min Wang, 2000] H. min Wang. Experiments in syllable-based retrieval of broadcast news speech in Mandarin Chinese. *Speech Communication*, 32:49–60, 2000.
- [Rajendran and Yegnanarayana, 1996] S. Rajendran and B. Yegnanarayana. Word boundary hypothesization for continuous speech in Hindi based on f_0 patterns. *Speech Communication*, 18:21–46, 1996.
- [Stolcke *et al.*, 2000] A. Stolcke, H. Bratt, J. Butzberger, H. Franco, V. R. Rao Gadde, M. Plauche, C. Richey, E. Shriberg, K. Sonmez, F. Weng, and J. Zheng. The sri march 2000 hub-5 conversational speech transcription system. In *Proceedings of NIST Speech Transcription Workshop*, College Park, MD, 2000.
- [Yang and Wilbur, 1996] Yiming Yang and John Wilbur. Using corpus statistics to remove redundant words in text categorization. *Journal of the American Society of Information Science*, 47(5), 1996.
- [Zheng *et al.*, 2002] J. Zheng, H. Franco, and A. Stolcke. Modeling word-level rate-of-speech variation in large vocabulary conversational speech recognition. *Speech Communication*, 2002.