VERY-LARGE-VOCABULARY MANDARIN VOICE MESSAGE FILE RETRIEVAL USING SPEECH QUERIES

Bo-Ren Bai[1], *Lee-Feng Chien*[2], *Lin-Shan Lee*[1,2]

[1]Dept. of Electrical Engineering, National Taiwan University
[2]Institute of Information Science, Academia Sinica
Taipei, Taiwan, R.O.C.
E-mail: white@speech.ee.ntu.edu.tw

ABSTRACT

In order to solve the problem with the new environment of fast growth of audio resources on the Internet, this paper presents a new approach which is capable of retrieving Mandarin voice message files using queries of unconstrained speech. By properly utilizing the monosyllabic structure of the Chinese language, the proposed approach performs the statistical similarity estimation between the speech queries and the voice message files, and executes the complete matching process directly at the phonetic level using syllable-based statistical information. Based on this approach, some experiments are tested and encouraging results are demonstrated.

1. INTRODUCTION

In order to solve the problem with the new environment of fast growth of audio resources on the Internet, this paper presents a new approach which is capable of retrieving Mandarin voice message files using queries of unconstrained speech. This subject has not been adequately discussed in the literature, although some works of text retrieval of voice files[1] or voice retrieval of text files[2][3] have been reported. By properly utilizing the monosyllabic structure of Chinese language, the proposed approach executes the complete matching process directly at the phonetic level using syllable-based statistical information. Such an approach will be shown to be feasible for Mandarin voice file retrieval.

Although there exist more than 10,000 commonly used Chinese characters, each character is monosyllabic and the total number of phonologically allowed Mandarin syllables is only 1,345. The combination of these monosyllabic characters or 1,345 syllables gives almost unlimited number of monosyllabic or polysyllabic Chinese words. This special monosyllabic feature of the Chinese language makes it possible to compare the voice records and the speech queries directly at syllable level. In this way the high ambiguity caused by the one-to-many mapping relation from syllables to characters in Mandarin speech recognition can be completely bypassed and the computation requirement can be significantly reduced.

Concerning the efficiency and accuracy of the retrieval process, a syllable-based feature vector is first defined for each voice record, which includes the presence information, and acoustic scores for all possible syllable pairs to appear adjacently in the voice record. This can be obtained by first performing syllabic recognition on each voice record, constructing a lattice of possible syllable candidates, and then transforming it into the desired syllable-based feature vector. The same feature vector is then defined and obtained for the input speech query via exactly the same procedure. With these feature vectors, a two-stage searching scheme is developed to perform the retrieval. The first stage is fast with high recall rate, comparing only the acoustic score information for the syllable pairs in the feature vectors of the query and the voice records, with a purpose to delete most irrelevant records. The second stage then performs detailed matching based on a specially designed similarity measure between the input query and the selected records using further information to provide a high precision rate. The block diagram of the proposed approach is shown in Figure 1, where the shadowed part can be off-line processed in advance to get the feature vector database D_v. When given an input query, we first extract its feature vector and then perform the matching processes to measure the similarity between the feature vector of the query and those in D_v. Records with high similarity scores will be retrieved. In the following, in Section 2, we first introduce the feature vectors used in the approach. In Section 3, the retrieving processes are described clearly. Moreover, some experiment results are presented in Section 4. Finally, in Section 5, some conclusions are drawn.

2. FEATURE VECTORS

Before describing the detail of the retrieving process, here we will introduce two types of the feature vectors used for similarity measurement in advance. The first feature vector includes the acoustic score information for all possible syllable pairs appearing adjacently in each voice record or the speech query. The second feature vector indicates the inversion document frequency of each syllable pair appearing within database or speech queries. This feature vector is used as a weighting vector to indicate the importance of each syllable pair.

2.1 Feature Vector with Acoustic Score Information of Syllable Pairs

For each voice record d in database D, we will first extract a feature vector V_d, which includes the acoustic score information of syllable pairs. To get such a feature vector, all possible syllable beginning frames in the voice record will be first obtained by some spectral information. Corresponding to a beginning frame, the possible ending frames for the syllable can be found using the estimated minimum and maximum duration of a syllable. Given all these possible beginning frames and their corresponding ending frames, with a dynamic programming approach several most possible syllable candidates with their acoustic scores can be found for each segment which may include a syllable. After utilizing some verification techniques to reduce the numbers of syllable candidates for each segment, we will have a syllable lattice for each voice record. Searching the syllable lattice, all acoustic scores of syllable pairs appearing adjacently on the syllable lattice can be extracted to form the feature vector V_d. This feature extracting procedure can be performed off-line on all voice records in database D to form a feature vector database D_v, which will be the target database we would like to retrieve. Regarding a speech query, q, we can also obtain such a feature vector V_q, via exactly the same procedure.

Besides, to compare the performances of retrieving by speech query and retrieving by text query, we also define another feature vector T_q, for a given text query. Regarding a text query, we can first get its syllable string and perform an easier search in the syllable string than we have done in the syllable lattice mentioned above. All syllable pairs (s_i,s_j) in the string can be found, but instead of recording acoustic scores information, as we have done for V_q, here we will use frequency count to form the feature vector T_q.

2.2 Feature Vector with Inversion Document Frequency Information

Another feature vector we will define here is the inversion document frequency, idf, which is defined as :

$$idf(s_i,s_j) \equiv \log(N / N_{s_i,s_j})$$

And we can discuss this feature vector from viewpoint of database and that of input query.

To the voice record database, where N is the total number of voice records in the database, and N_{s_i,s_j} is the number of records where the syllable pair appears in their syllable lattices. $idf_D(s_i,s_j)$ indicates the significance of the syllable pair (s_i,s_j) in the database D_v [4]. The syllable pair with smaller $idf_D(s_i,s_j)$ value means to be less discriminative for information retrieval. For example, if a syllable pair (s_i,s_j) appears commonly in many voice records, N_{s_i,s_j} is large for the syllable pair, and we will give the syllable pair a small weight by the definition of idf. This is very reasonable since a commonly used vocabulary will be less important from the viewpoint of information retrieval.

On the other hand, from the viewpoint of input query, N is the total number of queries ever been asked, and N_{s_i,s_j} is the number of queries in whose syllable lattice the syllable pair (s_i,s_j) appears. And $idf_Q(s_i,s_j)$ indicates the significance of the syllable pair (s_i,s_j) in the query. Using the idf_Q value as a weight for a syllable pair will compensate the weakness of not using any semantic information. For example, in a spoken query, we often use "我想要找..." ("I would like to find ...") , "有沒有關於..." ("Is there anything about...") or some phrases like these to express our intention. These phrases are absolutely not the main information we would like to find, but they will seriously affect the retrieving results if we cannot filter them out. After weighted by idf_Q value, the importance of the these commonly used phrase will be reduced. And the more queries the system asked, the better filtering effect will be obtained since idf_Q can be estimated better when more queries are collected.

3. RETRIEVING PROCESS

Given feature vectors database D_v and a speech query, q, the retrieving problem is now a searching process to retrieve the record d^* in the target database D_v which is most related to the query. This searching process can be formalized as :

$$d^* \equiv \arg\max_{d \in D_v} Sim(d,q)$$

Where $Sim(d,q)$ is a measurement to indicate the similarity between a record d and the query q.

The approach we proposed to deal with the problem is a two-stage searching scheme. The first stage is an approximate matching which is fast with high recall rate. In order to delete most irrelevant records, here we just compare the acoustic score information of the syllable pairs in the feature vectors for the query with those for the voice records. The second stage then performs detailed matching based on a specially designed similarity measure between the input query and the selected records by using the accumulated acoustic score and idf information to provide a high precision rate. The detailed operation of each searching stage will be described separately.

3.1 Approximate Matching

Given a query q, we will first get its feature vector V_q for speech query, or T_q for text query. Ranking each syllable pair by their $idf_Q(s_i, s_j)$, we can get K most important syllable pairs. Then for every voice record, d, in feature vector database D_v, one relevance score, $R(d,q)$, can be got by summing up from V_d the acoustic scores of these K syllable pairs. Now, ranking all $R(d,q)$'s and filtering out records with low $R(d,q)$'s, several most relevant records can be chosen. Collecting these most relevant records to form a new target database D_v^* for the next searching stage, we will have a much smaller target database in the second searching stage that will be described in next paragraph. Meanwhile, the feature vector, V_q or T_q, will be used to modify values in idf_Q such that we can have better idf_Q value along with increasing of total number of asked queries.

3.2 Detailed Matching

Since the new target database D_v^* after first searching stage is much smaller than the initial database D_v, it is able to perform a much more detailed matching process in the second searching stage. To estimate the relevance value between the input query and the records in D_v^* precisely, we first modify the feature vectors of the query and voice records individually by combining their acoustic score information and idf information:

for voice records,

$$\begin{aligned} U_d &= diag(V_d \cdot idf_D^T) \qquad , for\ d \in D_v^* \\ &= (V_d(s_1,s_1) \times idf_D(s_1,s_1), \ldots, V_d(s_i,s_j) \times idf_D(s_i,s_j), \ldots, \\ &\quad V_d(s_{1345},s_{1345}) \times idf_D(s_{1345},s_{1345})) \end{aligned}$$

for speech query,

$$\begin{aligned} U_q &= diag(V_q \cdot idf_Q^T) \\ &= (V_q(s_1,s_1) \times idf_Q(s_1,s_1), \ldots, V_q(s_i,s_j) \times idf_Q(s_i,s_j), \ldots, \\ &\quad V_q(s_{1345},s_{1345}) \times idf_Q(s_{1345},s_{1345})) \end{aligned}$$

or for text query,

$$\begin{aligned} T_q^* &= diag(T_q \cdot idf_Q^T) \\ &= (T_q(s_1,s_1) \times idf_Q(s_1,s_1), \ldots, T_q(s_i,s_j) \times idf_Q(s_i,s_j), \ldots, \\ &\quad T_q(s_{1345},s_{1345}) \times idf_Q(s_{1345},s_{1345})) \end{aligned}$$

Then a Cosine measure is used to estimate the relevance score between the speech query and voice records:

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

This measure also works for text query merely replacing U_q with T_q^*.

In this way, the larger the value of $R^*(d,q)$ the more record d is relevant to the query. Records with larger $R^*(d,q)$ values will be selected and ranked as the results.

4. EXPERIMENTS

In this section, we will present two categories of experiments to show the feasibility of our approach. The first category of experiments use simple queries to retrieve voice records, and the second category of experiments use quasi-natural-language queries to retrieve voice records. The test voice database includes about 1,000 voice records read by several speakers.

4.1 Retrieving by Simple Query

In order to understand the feasibility of the proposed approach, the first category of experiments use simple queries to retrieve voice records in the database. For each simple query, only one key information to retrieve is expressed in the query without irrelevant words. And to show the performance degradation due to higher complexity in speech recognition, two different input modes are performed for each query: the speech-input mode and the text-input mode.

The experiment results for the two input modes are listed in Table 1, and the hit rates for retrieved records of top 5, 10, and 20 are listed individually. It shows that there is about 4~6% of performance degradation in using speech query.

# of selected records / Input modes	Top 5	Top10	Top20
Speech Query	75.63	79.38	86.25
Text Query	81.76	85.13	90.54

Table 1: The test results of hit rates using simple queries.

4.2 Retrieving by Quasi-Natural-Language Query

The second category of experiments are designed to evaluate the feasibility of retrieving the voice record database by quasi-natural-language query. Here more than one key information with some irrelevant words are allowed in the query. For understanding the performance of the idf_Q information, the results with and without using idf_Q information are further shown in Table 2. Besides, the results of both using text and speech query are listed together for comparison.

Table 2 shows that there is a larger performance degradation from text-input mode to speech-input mode than that in Table 1, this means that the effect of the complexity of speech recognition in quasi-natural-language query is larger. At the same time, it can be found that the idf_Q information is helpful in reducing the interference of irrelevant words. Especially in the test of text-input mode, the performance of quasi-natural-language query is very close to that of simple query.

# of selected records / Input modes	Top 5	Top10	Top20
Speech query without using idf_Q	53.13	56.36	65.75
Text query without using idf_Q	58.38	63.75	73.12
Speech query with using idf_Q	60.63	66.25	76.25
Text query with using idf_Q	74.32	78.08	87.16

Table 2: The test results of hit rates using quasi-natural-language queries.

5. CONCLUSION AND OUTLOOK

This paper presents a new approach which is capable of retrieving Mandarin voice message files using queries of unconstrained speech. Although his subject has not been adequately discussed in the literature, the experiment results encourage us that the problem of retrieving voice database is feasible to be solved.

6. REFERENCES

[1] U. Glavitsch and P. Schauble, "A System for Retrieval Speech Documents", ACM SIGIR Conference on R&D in Information Retrieval, 1992, p 168-176.

[2] Sung-Chien Lin, Lee-Feng Chien, Keh-Jiann Chen, and Lin-Shan Lee, "An Efficient Voice Retrieval System for Very-Large-Vocabulary Chinese Textual Databases with a Clustered Language Model", ICASSP, 1996, p 287-290.

[3] Sung-Chien Lin, Lee-Geng Chien, Keh-Jiann Chen, and Lin-Shan Lee, "Unconstrained Speech Retrieval for Chinese Document Databases with Very Large Vocabulary and Unlimited Domains", EUROSPEECH, 1995, p 1203-1206.

[4] G. Salton, Introduction to Modern Information Retrieval, NY, McGraw-Hill, 1983.

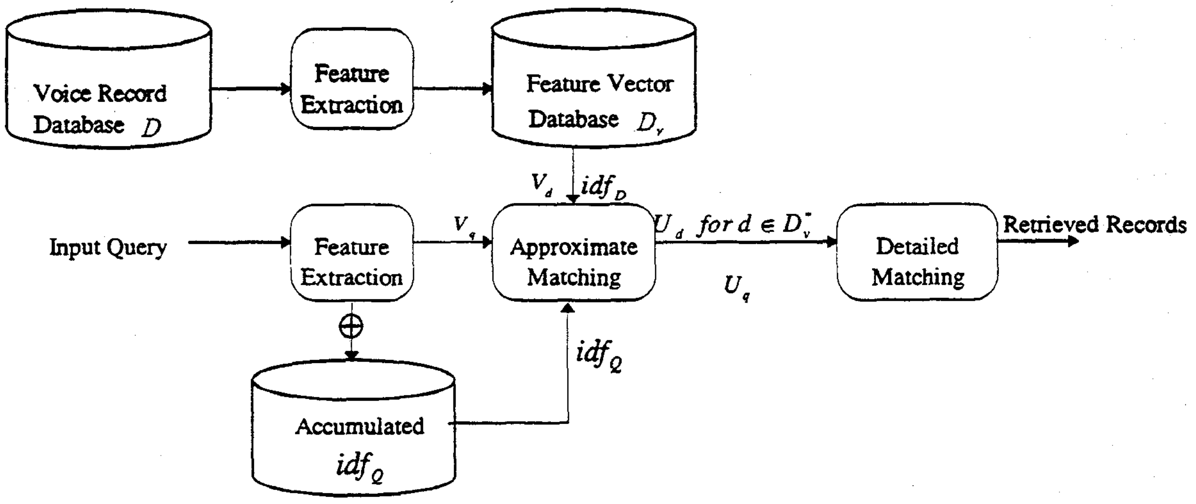

Figure 1: The block diagram of the proposed approach for voice message file retrieval.