A Speech Retrieval System based on Fuzzy logic and Knowledge-base filtering

Malay Singh[1],Uma Shanker Tiwary[2] and Tanveer J. Siddiqui[3]
[1]NUS School of Computing,
Email: malay@comp.nus.edu.sg
[2]Indian Institute of Information Technology-Allahabad,
Email: ust@iiita.ac.in
[3]J K Institute of Applied Physics & Technology,University of Allahabad,
Email: jktanveer@yahoo.com

***Abstract*—The objective of this paper is two-fold. First, to adapt the earlier cognitive interactive framework for speech retrieval application and second, to enhance the accuracy of continuous speech retrieval system using fuzzy word mesh representing linguistic knowledge of users. The proposed method recognizes the audio query and retrieves the audio file(s) in corpus using an information retrieval (IR) Engine. We have used CMUSphinx4 Library for automatic speech recognition after adapting it to Indian accent and lab environment. The IR Engine in the back-end uses Fuzzy Logic based reasoning and knowledge-based filtering to retrieve relevant sentences (transcribed).**

I. Introduction

Huge amount of information is available on the web in the form of Information text, audio, video, etc. In order to utilize this information in the best possible way, we need information retrieval techniques. This paper focuses on audio retrieval. Lectures, talks are generally recorded and stored for future use. A large number of Universities offer free access to their audio lectures by uploading it on their websites. Audio formats are not easily searchable like text. The continuous form of speech makes the retrieval process even more difficult. This paper extends a fuzzy logic based information retrieval algorithm [1] for spoken document.

Various attempts at audio retrieval have been made in previous decades. The problem in speech retrieval contains sub problems of Indexing, Automatic Speech Recognition (ASR) and Information Retrieval (IR). Many papers in speech processing literature have discussed indexing, ASR and classification [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12].

Li, and Zhang have discussed audio retrieval techniques based on meta-database and acoustic features [2]. In [3], a neural network based model for audio classification is proposed. The classification is based on gender, speech/music discrimination, etc. In [10] the authors discussed content - based audio retrieval on large scale using acoustic features. Their dataset contains voice effects, animal voices and natural scenes. [12] provides a detailed survey on the progress made in ASR and audio retrieval in the last two decades. They have also discussed various features that have been used for audio retrieval. In [9], Lu also reviewed indexing and retrieval of audio files. This survey had a good amount of discussion on audio classification techniques.

Foote [11] discussed the various components required for ASR and presented a method for a general audio analysis suitable for a wide range of audio such as music, sound effects. Some open source tools are also available for ASR including CMUSphinx [4], [5] and Julius [6].Speech Indexing based on phonemes has been discussed in [7] and [8].

In[1], the authors proposed a machine-learning based sentence extraction algorithm. They used a word mesh structure to store knowledge and used fuzzy logic for reasoning. A word mesh is defined as a data structure connecting similar words with edge weights representing relevance. The irrelevant words (hence sentences) are filtered out on the basis of weighted edges. After various epochs their method gives ranked sentences. The score of a sentence is defined in terms of the words it contains. To define relationship between different word/group of words fuzzy logic reasoning has been incorporated.

In the present paper, we use their work to propose a new method to rank a given set of audio files (speech corpus) using speech-to-text converted data.We have added the speech to text feature to work on audio files. We use the adapted sentence extraction algorithm to rank audio files [1]. This paper focuses on speech to text conversion and adaptation of work done in [1].

The remaining sections of the paper will discuss about automatic speech recognition, speech retrieval, related theory and experiments. Section II will detail about Automatic Speech Recognition, fuzzy logic reasoning and knowledge base filtering used in the proposed method. Section III details down the procedure of speech retrieval step by step. Sections IV and V discuss about the experiments and their results. Finally Section VI concludes the paper with suggested future work.

II. Automatic Speech Recognition and Reasoning

A. Speech Structure

Speech is an audio stream containing stable state with dynamic state transitions. As the states are stable, one can define a simple classification called sounds or phones. A word is made up of a standard set of phones. However, while speaking the same word the pronunciation changes because of accent, context, speaker etc. This makes recognition

978-1-4799-1205-6/13/$31.00

process difficult. The final audio stream is much different from its canonical (written) representation. Observing this pattern, researchers have discussed using diphone as the dynamic transitions between stable phones [13]. Diphones are parts of phones between two consecutive phones.

The phonetic object used for classification purposes contains three parts. First is transition preceding a stable phoneme, second is the stable phone and the third part is the transition from the stable phone. The CMUSphinx group calls this object a senone [13].The three state outline of senone make it easier to recognize speech based on Hidden Markov Model. A senone depends not only on the left and right context but also it depends on a complex function. The complex function can be defined as per requirement.

Phonemes make subwords known as syllables. Sometimes, syllables are defined as "reduction-stable entities" [13]. For example, when speech becomes fast, phones often change, but syllables remain the same. The change in phone lead to unreliable ASR if it is based on a single phone.

B. Automatic Speech Recognition Models

A speech recognition system uses different types of models: an acoustic model, a phonetic dictionary and a language model. In this work ASR model of CMUSphinx4 [4] has been used.

The acoustic model defines acoustic properties for each senone. Both context-independent and context independent models are used. The context-independent models use most probable feature vectors for each phone while context-dependent are built from senones.

A phonetic dictionary contains mapping from words to phones. The dictionary is not the only variant of mapper from words to phones. It could be done with some complex function generated by a machine learning algorithm.

A language model defines which words are more probable to follow previously recognized words. It helps in restricting the matching process by stripping words that less likely to follow a given word. The speech recognition system by CMUSphinx group uses an acoustic model, phonetic dictionary and language model. These models and dictionary change as per usage.

C. Reasoning through Fuzzy Logic Based Word-Mesh

Once the recorded corpus is converted to text, we perform retrieval steps just like on text corpus. The system performs reasoning by manipulating a set of parameters and weights, which determine the correlation of different units of the text. The idea is to pick up words in the sentences, create a word-mesh containing all the words with edges between synonyms and other related words. Each edge has a weight according to relation between vertices, i.e., words.

To represent these weights, fuzzy values with sigmoid membership function have been used. [1] suggests using fuzzy values for weights to get best possible results. The IR Engine proposed in [1] used the sigmoid function[14]. The membership function is defined as:

$$p_x^{\lambda}(x) = \frac{1}{1 + e^{(-\lambda(x-a))}} \tag{1}$$

Where λ and a are constants and x is independent variable.

The sentence extraction algorithm [1] parses the Word-mesh while ranking the sentences (from audio files). The edge weights represent the similarity/relation between two words. Edge weight may improve or degrade the rank of sentence(audio file). The reader can find more detail about the algorithm in [1].

III. The Proposed Method

Fig. 1 shows the steps in our method. Fig. 1a shows ASR and indexing steps. Automatic Speech recognition is according to a user. These steps are to be done again for different user. Fig. 1b shows Retrieval steps. The method is discussed in detail in following sections.

A. Dictionary Generation & Model Adaptation

To improve the accuracy of ASR component, the dictionary [15] of English in Local (Indian) accent was generated. In the dictionary, as discussed earlier one needs to map the words to the corresponding phonemes. This will include the *.dic* file in Sphinx4 Model. The CMUSphinx group has developed various software to generate the required phonemes. The dictionary for the adaptation process was generated using online tools by CMUSphinx group. The main idea of adaptation is to capture /incorporate accent variation as a phoneme can be spoken in different ways (accents) by different people. The original dictionary provided by CMUSphinx was made for American Accent.

The linguistic knowledge of the user is used for Model Adaptation purposes. This way our method gives more accurate result. Acoustic model was adapted to one of the authors' voice to make recognition good for a particular recording environment, audio transmission channel, and accent. The adaptation process takes transcribed data and improves the model as per the user. It's more robust than training and could lead to good results even if adaptation data is small. As per the documentation of CMUSphinx it's enough to have 5 minutes of speech to significantly improve the dictation accuracy by adaptation to the particular speaker. The first thing one need to do is to create a corpus of adaptation data. This consist of a list of sentences, a dictionary describing the pronunciation of all the words in that list of sentences, and a recording of a person speaking each of those sentences [16]. From the audio WAV recording the acoustic features are generated. Using these features various statistics are generated using SphinxTrain [17]. This adaptation is done in order to improve ASR which may lead to garbage query like "language" can be taken as "guage" or something like that. After model has been adapted for the user, it will guess it as "language". This makes the ASR robust and intelligent and helps in improved Speech Retrieval.

The vocabulary has been restricted as per the usage in a given scenario. If a vocabulary is restricted to some words like for query the database has been adapted for those words only.

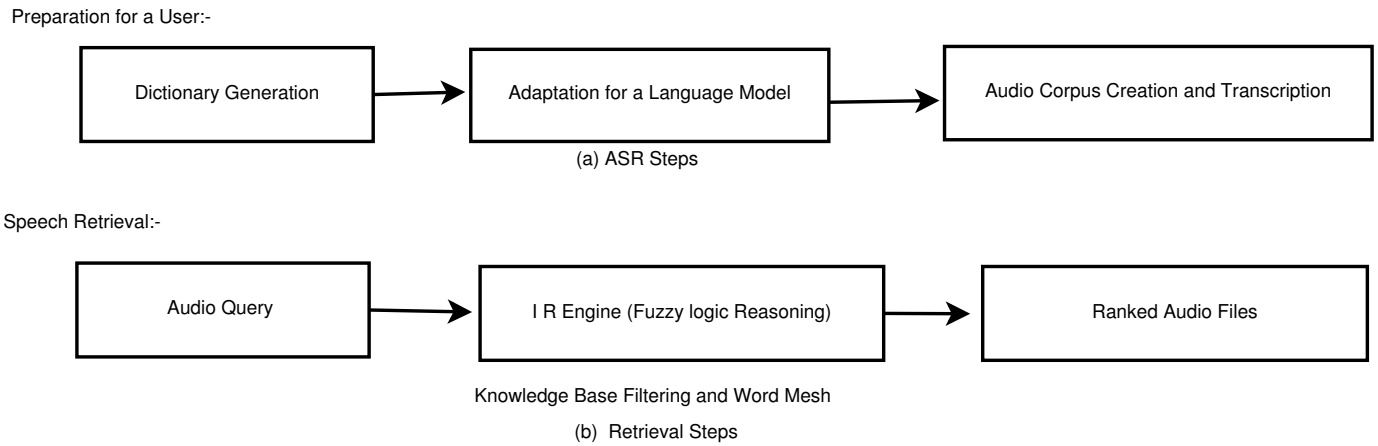

Fig. 1: The Proposed Method

This gave very localized and accurate results. A vocabulary defines what words can be recognized by ASR component of our method accurately. For a word not in vocabulary,the ASR component will get a probable match from its restricted list of words.

B. Retrieval using reasoning and Knowledge Base Filtering

We take the text form of the query and invoke the IR engine. The IR engine parses all text files and creates the word mesh as discussed in [1]. Stanford NLP tools, POS Tagger [18], [19] and Parser are used to differentiate in various sentences in the corpus. WordNet [20] is used to find words with similar meanings when used in different senses. These tools are used by IR engine to build up word mesh, with its edges given different weights. These tools differentiate between the roles of each word in a sentence.

Using these tools weights for each edge between words are calculated. Stanford dependencies are provided by Stanford NLP Group. This tool parses sentences and find grammatical relation (Dependency) between two words in a sentence. We can use Stanford Dependencies [19] for calculating edge weights. For example dependency agent will give different weight to connected words when compared to prep-on dependency. The authors [1] assigned fuzzy weights for different types of dependencies.

Once the weight are calculated the IR Engine [1] uses the Word-mesh while ranking the sentences (from audio files). Edge weight may improve or degrade the rank of sentence(audio file). Score and ranking of the sentences are discussed in detail in [1] .

This helps the IR engine to intelligently generate the final set of ranked sentences.

Score and ranking of the sentences are discussed in detail in [1]. In other words, we are using a fuzzy logic based IR engine on speech corpora to get better performance.

Once the ranked files are shown, user can select rank to play the corresponding file. Different types of corpora and query sets used for experiments are explained in section IV.

IV. EXPERIMENTS

A. Corpus creation and indexing

To check the robustness of all the components of our system namely ASR and IR we have generated four different corpora and two query sets. The corpora contain different number of sentences recorded. The corpora for our experiments had 35, 61,127 and 82 sentences. Also two corpora contain single words while the other two contain sentences.

Corpora A and D are word corpora containing 35 and 82 words respectively. Corpora B and C are sentence corpora containing 61 and 127 sentences respectively.

The size of corpora used may seem small for the retrieval purposes because it was constrained by ASR component. For proper adaptation one need to give as many as possible permutations of a word being used in sentences. With high frequency of word there is more chance of the word being recognized correctly while transcription. By keeping the size of corpus small it improves speech recognition. Corpora A and D comprised of words recorded in WAV Format. Each file contains a single word. Corpora B and C contains a single sentence recorded in WAV Format.

For various experiments as described in following section, two different corpus were made. First type of corpus consist of words recorded in WAV Format. Each file contains a single word. Second type of corpus contains a single sentence recorded in WAV Format. Both corpora are same in essentially the same except for the type of data recorded.

B. Query set creation and retrieval

The two sets comprising of queries were different in the fact that one set had all the queries made up of word which were already in knowledge base of our system. The other set contains queries made up of words not in knowledge base (they were new to the system). Query set P contains known words to the system. Query set Q contains foreign (new) to the system. Table I and II summarize the Corpora and Query Sets used.

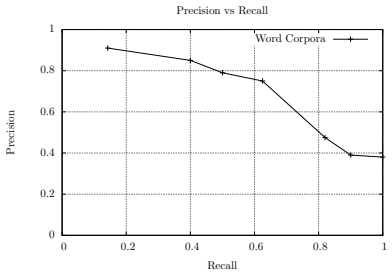

Fig. 2: Precision - Recall for Word Corpora A and D subject to Query Sets P and Q

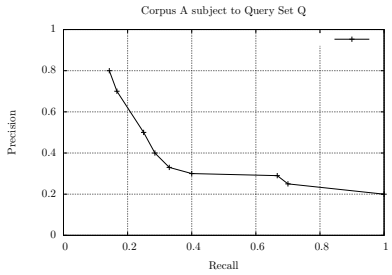

Fig. 3: Precision - Recall for Corpus A subject to Query Set Q

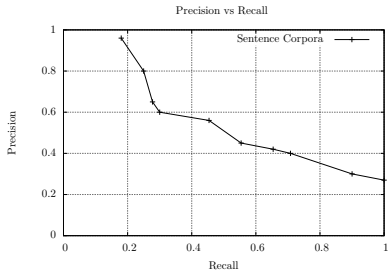

Fig. 4: Precision - Recall for Sentence Corpora B and C subject to Query Sets P and Q

For retrieval purposes, one cannot use a raw WAV file containing a lecture along with its transcribed text file. This raw audio file needs to be segmented in less than one minute sections and transcribed.

For retrieval a user may use any of the following methods to search and retrieve audio files.

- Record the query using Microphone.
- Use a pre-recorded query from file in WAV format.
- Write the query in the form of text.

Once the query is recorded or link to query file is given, our system transcribes it in real-time. Our system has been trained and adapted differently with different training-data-sets for Query Recognition and Corpora Transcription.

TABLE I: Corpora used

Corpus	A	B	C	D
Size	35	61	127	82
Type	Word	Sentence	Sentence	Word

TABLE II: Query sets used

Query Set	P	Q
Type of words	Known	Foreign

V. Results and discussion

A. Results

For corpora A and D, Speech-to-Text Component (ASR) of the system worked with higher accuracy compared to its performance on corpora B and D. 10% of the transcribed files were edited. For example hello was wrongly transcribed to oh. We edited these type of errors. Inaccurate transcription gave incoherent sentences.

For word corpora A and D the IR engine works well when subjected to Query Sets P and Q. One may observe that it approximates the ideal recall-precision curve to much extent as illustrated in Fig. 2. The Precision - Recall curve for experiment with Corpus A subject to Query Set Q is illustrated in Fig. 3.

For word corpora B and C the IR engine works not that good when compared to its performance on word corpora A and D. One may also observe that it approximates the ideal Precision - Recall curve with some aberrations as illustrated in Fig. 4.

There were a few aberrations or outliers in case of experiments with small corpus. As size of corpus was increased these outliers tended to disappear. Also while checking the transcribed file the sentences were not changed in output to a large extent as it would be against the main goal of this paper. In Corpora C and D sentences which were not related to any of the queries were added to induce confusion to check performance of our system under stress.

The Precision - Recall curve for experiment with Corpus B subject to Query Set Q is shown in Fig. 5.

The Precision - Recall curve for experiment with Corpus C subject to Query Set Q is shown in Fig. 6.

B. Discussion

Our method using linguistic knowledge of user, improves the Automatic Speech Recognition which in turn improves

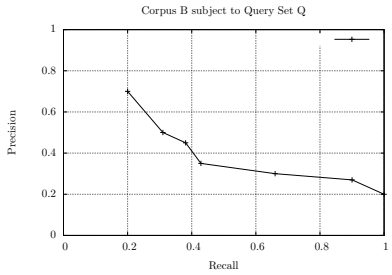

Fig. 5: Precision - Recall for Corpus B subject to Query Set Q

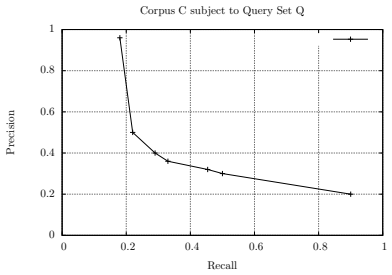

Fig. 6: Precision - Recall for Corpus C subject to Query Set Q

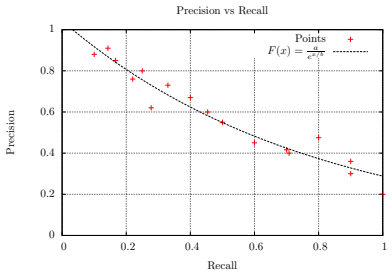

Fig. 7: Precision - Recall curve fitted to $F(x) = \frac{a}{e^{x/b}}$, where $F(x)$ is Precision and x is Recall, $a = 1.0430$ and $b = 0.7781$.

Retrieval. When it fails to recognize query accurately it returns the best probable match. Knowledge Base filtering and word mesh gives a better ranking of the audio files.

The results from all experiments were combined. The precision values for a single recall value were averaged. The observations were fitted to a curve $F(x) = \frac{a}{e^{x/b}}$, where $F(x)$ is Precision and x is Recall given that a and b are numerical constants using gnuplot. Gnuplot uses Marquaralt-Levenburg algorithm. Curve fitting gave the values $a = 1.0430$ and $b = 0.7781$. The approximated curve is illustrated in Fig. 7. This gives one an insight about how one can predict values on the curve. It has been observed that it follows the logarithmic curve approximately.

VI. Conclusion and Future Work

The proposed system uses a more realistic way for reasoning using fuzzy logic. The use of fuzzy logic reasoning improves the system. The performance is quite good for initial purposes. The accuracy of results indicate that we need to increase our training corpus and combinations of sentences for a better system. Errors are generally induced by ASR component. The accuracy of the transcription component is around 70% in worst cases which are explained easily, given that there is always noise in the working system environment.

The future work is suggested as follows.

- Adaptation of acoustic model,such that the accuracy of ASR component improves. A sophisticated training and adaptation corpus such that each occurrence of word gets to be adapted.
- The whole project can be implemented for Indian languages like Hindi,Urdu etc. The developer will need to write the language model for these and do the training of the model accordingly.

Acknowledgment

The authors would like to thank to CMUSphinx Group at CMU for their research in the field of Automatic Speech Recognition. Without their CMUSphinx library it would have been difficult to use IR engine on audio recordings.

References

[1] Anupam Srivastava, Divij Vaidya, Malay Singh, Pranjal Singh and U. S. Tiwary. “A Cognitive Interactive Framework for Multi-Document Summarizer”.Advances in Intelligent Systems and Computing, 1, Volume 179, Proceedings of the Third International Conference on Intelligent Human Computer Interaction (IHCI 2011), Prague, Czech Republic, August, 2011, Part 5, Pages 257-268.

[2] Li, G. H. Wu, D. F. Zhang, J. Concept Framework for Audio Information Retrieval : ARF. Journal of Computer Science and Technology,2003,VOL 18; Part 5, pages 667-673.

[3] H Harb, L. Chen. A General Audio Semantic Classifier based on human perception motivated mode. Multimedia Tools and Applications, Eds. Springer Netherlands, ISSN 1380-7501 (Print) 1573-7721 (Online), http://dx.doi.org/10.1007/s11042-007-0108-9,March 05, 2007.

[4] CMUSphinx Wiki - CMUSphinx Wiki.http://cmusphinx.sourceforge.net/wiki/ Cited 10 January 2013

[5] Willie Walker , Paul Lamere , Philip Kwok , Bhiksha Raj , Rita Singh , Evandro Gouvea , Peter Wolf and Joe Woelfel .‘Sphinx-4:A flexible open source framework for speech recognition’,2004.

[6] Open-Source Large Vocabulary CSR Engine Julius. http://julius.sourceforge.jp/en_index.php Cited 11 January 2013

[7] Martin Wechsler and Peter Schuble.Speech Retrieval Based on Automatic Indexing.Proceedings of the Final Workshop on Multimedia Information Retrieval (MIRO’95), Electronic Workshops in Computing,1995,Springer.

[8] M. G. Brown,J. T.Foote,G.J.F.Jones,I. Sprck Jones and S. J. Young.Open-Vocabulary Speech Indexing for Voice and Video Mail Retrieval,1996.

[9] Goujun Lu. 2001. Indexing and Retrieval of Audio: A Survey. Multimedia Tools Appl. 15, 3 (December 2001), 269-290. DOI=10.1023/A:1012491016871 http://dx.doi.org/10.1023/A:1012491016871.

[10] Gal Chechik, Eugene Ie, Martin Rehn, Samy Bengio, and Dick Lyon. 2008. Large-scale content-based audio retrieval from text queries. In Proceedings of the 1st ACM international conference on Multimedia information retrieval (MIR ’08). ACM, New York, NY, USA, 105-112. DOI=10.1145/1460096.1460115 http://doi.acm.org/10.1145/1460096.1460115

[11] Jonathan Foote. 1999. An overview of audio information retrieval. Multimedia Syst. 7, 1 (January 1999), 2-10. DOI=10.1007/s005300050106 http://dx.doi.org/10.1007/s005300050106

[12] Mitrovic, Dalibor and Zeppelzauer, Matthias and Breiteneder, Christian. “Features for Content-Based Audio Retrieval”,Advances in Computers Volume 78 Improving the Web,2010,pages 71-150.

[13] Basic concepts of speech - CMUSphinx Wiki. http://cmusphinx.sourceforge.net/wiki/tutorialconcepts Cited 20 January 2013.

[14] Jozsef Dombi, Norbert Gyorbiro, Addition of sigmoid-shaped fuzzy intervals using the Dombi operator and infinite sum theorems, Fuzzy Sets and Systems, Volume 157,Issue 7,Pages 952-963,1 April 2006.

[15] Generating a dictionary - CMUSphinx Wiki. http://cmusphinx.sourceforge.net/wiki/tutorialdict Cited 21 February 2013.

[16] Adapting the default acoustic model - CMUSphinx Wiki. http://cmusphinx.sourceforge.net/wiki/tutorialadapt Cited 30 January 2013.

[17] Training Acoustic Model For CMUSphinx - CMUSphinx Wiki. http://cmusphinx.sourceforge.net/wiki/tutorialam?s[]=sphinxtrain Cited 20 January 2013.

[18] Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

[19] Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006.

[20] Princeton University “About WordNet.” WordNet. Princeton University. 2010. http://wordnet.princeton.edu cited 20th April,2013.