Content based Multimedia Retrieval using Automatic Speech Recognition

Neelam Purswani
Computer Department
V.E.S. Institute of Technology
Chembur, Mumbai-400 074

Reshma Ramrakhyani
Computer Department
V.E.S. Institute of Technology
Chembur, Mumbai- 400 074

Meghana Makhija
Computer Department
V.E.S. Institute of Technology
Chembur, Mumbai-400074

ABSTRACT
Our system deals with retrieval of suitable video based on the user input and voice commands. The microphone takes user voice as input and processes it to convert it into the text. It further checks the repository database of videos to find a video that matches the spoken keyword by the user. A list of keywords is available for the user in order to aid him in searching and querying process. After the relevant video is played as per user's wish, it is further given an option if the user wants to navigate to particular topic within the video. The system can do the same for the user. Our system basically encompasses and covers application in lecture video domain. The videos in system are based on a wide variety of topics. In general, navigation in videos is too time consuming as it performed by trial and error. However, with the help of this system, searching becomes faster and response time increases. Proper indexed query handling in database makes navigation easier and efficient. The system is now restricted to lecture videos but can be extended to various different domains too in industry.

General Terms
Fast retrieval, content based retrieval, repository of videos

Keywords
CMU Sphinx, PocketSphinx, ffmpeg

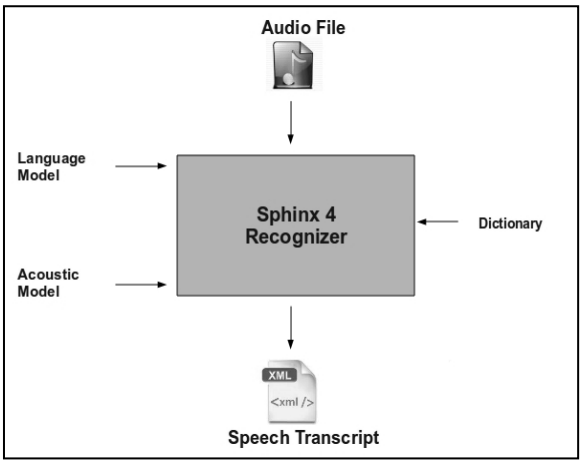
1. INTRODUCTION
In this paper, first the problem has been defined and then an approach has been designed to tackle the problem. With the increasing use of technology and faster access to the huge content available on the Internet, there must a solution to access it using faster and efficient querying methods so that less time is wasted in searching the content and more in understanding it, this applies generally to all the domains but in particular, we focus on educational videos in this paper.

1.1 Motivation
AI (Artificial Intelligence) is a new technological science to research and develop theory, methods, technologies and applications for simulation, extension and expansion of human intelligence. Artificial intelligence is a branch of computer science. It attempts to understand the substance of intelligence and produce a new intelligent machine which could response in a similar manner as human intelligence, in the areas of research of AI include robotics, speech recognition, image recognition, natural language processing and expert systems, etc. Inspired by using AI along with speech recognition, we propose a system that allows users to search the spoken content of multimedia files rather than their associated meta-information and let them navigate to the right portion where queried words are spoken by facilitating within-medium searches of multimedia content through a bag-of-words. The amount of video content available on the internet has increased dramatically over the past few years. More than half of the information on the Internet is formed by spoken documents that cover excessive amounts of academic, technical and news worthy information, which should be universally accessible. The retrieval of video streams and finding their relevant segments within them has gained importance, as these processes would help users to reduce their time and retrieve bandwidth costs. Thus, we thought of developing a system that can relevant video as per the keyword given by user through speech on the relevant topic.

1.2 Problem Scenario
If we look at the current scenario, usually a lot of time is spent on searching the video and then also searching a specific part of the video which the user is interested in watching so the objective is to: develop a system for any repository of videos, if given a set of search keywords through speech returns a list of videos that contain the keywords, highlight the portion of each video where the keyword occurs, allow the user to navigate directly to any highlighted portion of video.

1.3 Overview of the System
Overall objective of this project is to simplify the navigation process associated with multimedia files using open source tools like PocketSphinx or CMU Sphinx and ffmpeg to adjust the bitrate of the multimedia files and enable the user to give voice commands to the system and the system is then expected to translate the voice commands given by the user to its corresponding text representation and search the repository of videos to return the required video and also while watching a video a user must be able to navigate to a particular timestamp based on the keyword the user inputs to the system using voice commands.

2. DETAILS ABOUT TOOLS USED



CMUSphinx project is an open source speech recognition project developed at Carnegie Mellon University, which consists of various tools use to build speech applications:

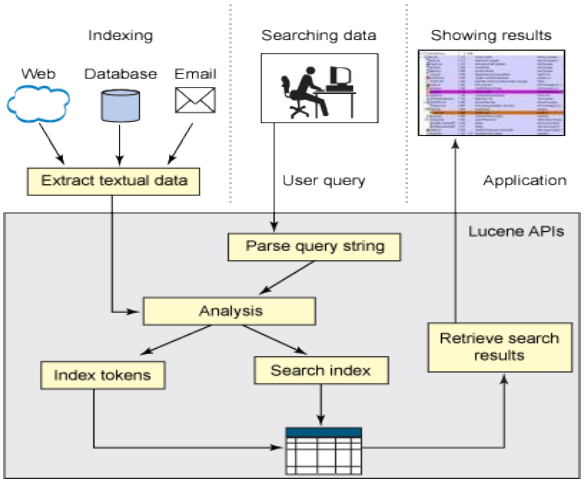
Sphinx4: Speech recognition written in the Java.

This component takes audio file generated in the audio extraction unit as input and creates

a speech transcript. The speech transcript contains individual spoken words in textual form along with their time of occurrence in the lecture video.

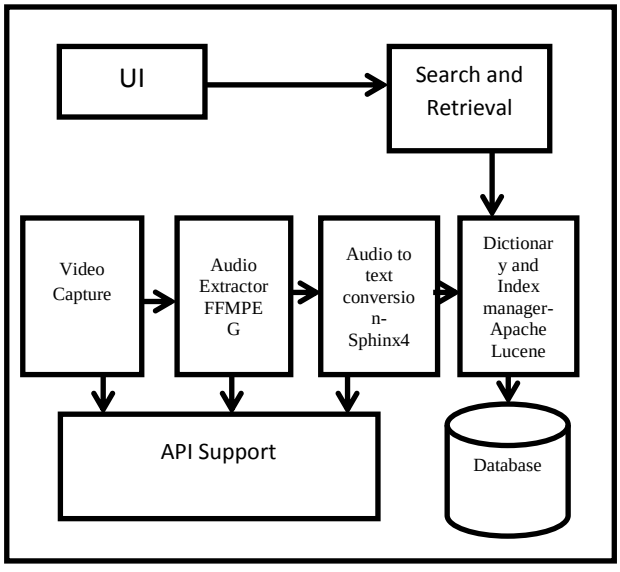
A speech recognition engine uses *acoustic model* and *language model* for automatically converting spoken words into text. It also requires a *pronunciation dictionary* that maps each word to its phoneme representation.[1]

Apache Lucene: Indexing is the next important step in process. Indexing is essential to optimize speed and performance in retrieving relevant documents of search query. We will be using Apache Lucene for the same. Lucene's powerful APIs focus mainly on text indexing and searching. It can be used to build search capabilities for applications such as e-mail clients, mailing lists, Web searches, database search, etc. Web sites like Wikipedia, TheServerSide, jGuru, and LinkedIn have been powered by Lucene. [2]



ffmpeg is a command-line tool to convert one video file format to another. It can also grab and encode in real-time from a TV card.[3]

3. ARCHITECTURE OF THE PROPOSED SYSTEM



The system architecture comprises of Database of stored videos, a user interface where the user can enter the voice query, Speech recognition engine and indexing.

1. **Repository of videos**: It consists of all the videos a user has in the system. As and when new videos are added to the repository, the indexing of the metadata of videos can be done using Apache Lucene.
2. **User interface**: A simple user interface via which user can give voice commands to interact with the sytem.
3. **Indexing**: Apache Lucene is used for indexing incrementally the metadata of newly added videos.
4. **Speech Recognition engine:** It converts user's voice commands to text for querying purposes.

3.1 STEPS INVOLVED IN SYSTEM CREATION

1. Making a repository of videos

We have downloaded various lecture videos from the different sources and have saved it all together. This collection of videos is repository for our system.

2. Converting video to .wav file

We need to convert any kind of video to .wav file format because the speech recognizer does not accept any other form of multimedia files. For this we have used ffmpeg tool. This is an open source tool that works in cmd.

Following is the command that we have used to obtain proper file format:

ffmpeg -i input_video.mp4 output_audio.wav

Accepting speech from microphone and converting it to text.

For this we have used Sphinx4 which is an open source tool completely coded in java. We modified this according to our requirement.

3. Converting the .wav file into text

For this section too we have used Sphinx4. It uses libraries in Sphinx4 to achieve real time conversion of speech to text.

4. Creating gram files

For step 3 and 4 both standard language and acoustic models of Sphinx4 are used. In order to have proper conversion we had to create a gram file of our own as per the requirement of videos.

A sample gram file is as follows:

#JSGF V1.0;

grammar digits;

public <numbers> = (database | environment| I | oh | zero | one | two | three | four | five | six | seven | eight | nine | a | and | ham | with | want | olive | green | onions | tomatoes | pizza | small | analytical | needs | how | do | we | design |for | not | scaled | processing | transform | using | start | this | our | of |understand | was | it | to | need | what | did | then) * ;

5. Obtaining time stamping for videos

Time stamping of videos was very essential without it would have not been possible to allow users to navigate to a particular required section of video. We have thus obtained

time stamp per second i.e word occurred in every second can be identified.
We used the following java method for the same:

getTimeBestResult(true,true)

6. Indexing the videos.
We have used Apache Lucene for indexing the metadata of videos. The index of videos had been stored in the IndexWriter of Lucene along with the speech transcripts including their time stamp.

A sample of indexing file is as follows:

addDoc(w, our want a small pizza a with tomatoes green olive small onions and ham | <sil>(0.45,0.48) our(0.48,0.63) want(0.63,0.99) a(0.99,1.09) small(1.09,1.5) pizza(1.5,1.9) a(1.9,2.04) with(2.04,2.2) tomatoes(2.2,2.92) green(2.92,3.22) olive(3.22,3.65) small(3.65,3.91) onions(3.91,4.42) and(4.42,4.53) ham(4.53,5.12) |,"air_000");

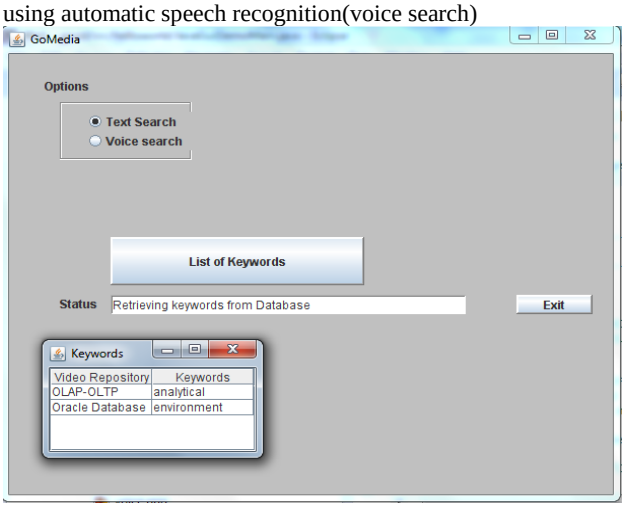
Here "air_000" is name of the video and is used as an index for the same.

7. Playing the required video.
After the indexing of videos, we can play a particular required by searching it with a help of keyword. This keyword acts as a query for the indexed file. On matching with a particular video, its speech transcript along with time stamp is displayed, at the same time, the required video plays in VLC player.

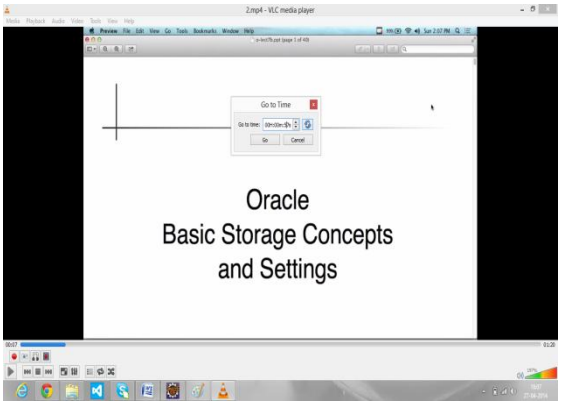
8. Navigating within a video.
When the video plays, and if user wishes to navigate to a particular topic, we provide him with the timestamp of that particular topic. Using this timestamp user can directly jump to that time by using VLC inbuilt function 'Jump

3.2 SNAPSHOTS OF THE IMPLEMENTATION

1. This is a simple user interface where the user has the option to search a video by conventional method (text) or by using automatic speech recognition(voice search)



2. Output of the system



4. CONCLUSION AND FUTURE SCOPE

4.1 Conclusion

The project successfully detects the words spoken and retrieves corresponding video and allows the user to navigate to desired location.The accuracy of the system is subject to accuracy of the speech recognition engine which is highly dependent on language and acoustic models. Also, environmental noise plays a role while taking input from the user.The system is accurate enough for a given set of repository of videos. Scaling up the system would require the need to build better acoustic models and low word probability error while pairing the acoustics. The most frequently used words or keyword of the videos which are stored in the repository are indexed using Lucene, an indexing and searching tool, to enhance the response time of the system. The system returns the timestamp of the occurrence of the keyword in a video and allows the user to jump to the timestamp and view the video.

4.2 Future Scope

1. The current working model is based on lecture video repository. The system can be extended for various different categories for example Advertisement videos, Tourist videos, Vodcast , podcast etc

2. Accuracy of the system can be increased by building Acoustic and language models and dictionary specific to system. The system can be deployed as a web application and made available to the entire world via web.

5. ACKNOWLEDGEMENT

Under the guidance of Prof. Sujata Khedkar, Associate Professor, Department of Computer Engineering, VESIT, Chembur, Mumbai-400 074

6. REFERENCES

[1] http://cmusphinx.sourceforge.net/

[2] http://www.ibm.com/developerworks/library/os-apache-lucenesearch/

[3] http://www.ffmpeg.org/download.html

[4] Browsing within Lecture Videos Based on the Chain Index of Speech Transcription Stephan Repp, Andreas Groß, and Christoph Meinel, Member, IEEE.

[5] Free video lectures : http://freevideolectures.com

[6] http://ant.apache.org/bindownload.cgi

[7] Browsing within Lecture Videos Based on the Chain Index of Speech Transcription Stephan Repp, AndreasGroß, and Christoph Meinel, Member, IEEE