# On-The-Fly Student Notes from Video Lecture Using ASR

**Dipali Ramesh Peddawad**[1]

[1]*computer Science and Engineering, Vishwakarma Institute of Technology Maharashtra, India*

***

**Abstract -** In the last decade video lectures has become more popular. The amount of video lecture data on the web is growing rapidly. Video lectures has become a popular storage and exchange medium due to the high speed network, rapid development in recording technology and improved video compression technique Therefore video recordings are used more frequently in e-lecturing systems. A number of lecturers and research institutes are taking the opportunity to record their lectures and publish them online so that students can access them as independent of time and location. Mostly, all video lectures consist of slides with relevant point. In only few video lectures one can find a written transcript of the video. The system will create automatic notes from lecture video so; that it will help students in study also it will help those students who have hearing disability. First of all, we apply Automatic Speech Recognition (ASR) on lecture audio tracks. The Automatic Speech Recognition ASR transcript the lecture video and create a notepad file so that student can use this text file for study. The numerous words consist of the speech are finally noted down in the word processor.

***Key Words*: *Video Lecture, Transcriptions, Automatic Speech Recognition, Speech to Text Conversion.***

## 1. INTRODUCTION

The ASR technology would be especially welcome by automated telephone exchange operators, doctors and lawyers, besides others whose look for freedom from boring and annoying ordinary computer operations using keyboard and the mouse. It is good for computer programs in which computers are used to provide (something commonly done) information and services. The ASR's direct speech to text (when someone writes down what another person says) offers a significant advantage over traditional written versions of spoken words. With further good improvement of the technology in text will become a thing of past. ASR offers a solution to this tiredness-causing procedure by converting speech into text. The ASR technology is now capable of recognition accuracies of 75% - 85 % but only under ideal conditions. The automatic speech recognition technology is still far from perfect in the real world.

### 1.1 Speech Recognition Process

When you speak into a microphone it converts sound waves into electrical signals. The ASR program removes all noise and keeps only the words that you have spoken. The words are broken down into individual sounds, known as phonemes, which are the smallest sound units.

In the next most complex part of the automatic speech recognition is to map the sounds to character groups. The ASR system also has a big dictionary of popular words that exist in the language. Next, each word is matched against the sounds and converted into the appropriate character group. This is where problems begin. To overcome the problems met in this phase, the program uses many methods; first checks and compares words that are almost the same in sound with what they have heard; then it check if the language allows a particular word to appear after another.

The numerous words constituting the speech are finally noted down in the word Processor.

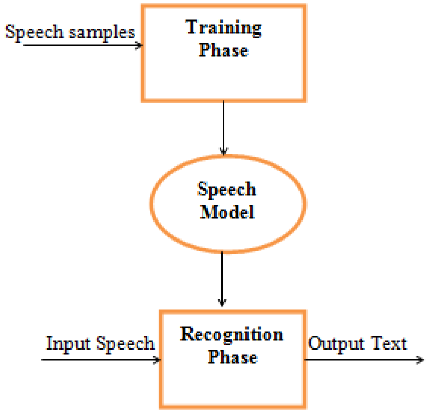

Fig -1: Speech Recognition Process

### 1.2 Motivation

The advances in networking and multimedia technologies have led to the widespread use and availability of digital video lectures. Indeed, many educational institutes use video lectures to improve the effectiveness of teaching in and out of classrooms and to support distance-learning students.

Automatic lecture transcription is an important task for both researches and applications. It is a challenge for speech recognition as, in contrast to broadcast news, lectures typically present in higher changeability in terms of speaking style, language domain and speech fluency.

### 1.3 Objective

- To study the existing Automatic Speech Recognition (ASR) Algorithm.
- Perform comparative study of existing ASR algorithms.
- To extract audio from video lecture and create text file from generated audio file.
- Analysis of the result.

## 2. PROPOSED SYSTEM

In the last decade video lectures has become more popular. The amount of video lecture data on the web is growing rapidly. Video lectures has become a popular storage and exchange medium due to the high speed network, rapid development in recording technology and improved video compression technique Therefore video recordings are used more frequently in e-lecturing systems. A number of lecturers and research institutes are taking the opportunity to record their lectures and publish them online so that students can access them as independent of time and location. Mostly, all video lectures consist of slides with relevant point. In only few video lectures one can find a written transcript of the video. The system will create automatic notes from lecture video so, that it will help those students who have hearing disability.

First of all, we will separate a sound track from video then we apply Automatic Speech Recognition (ASR) on lecture sound tracks. The ASR program removes all noise and retains only the words that you have spoken. In this proposed methodology we will give a separated sound track to the different speech recognition API. In this system we are working with following Speech recognition API:

1. Google API
2. Sphinx API
3. IBM API
4. Houndify API

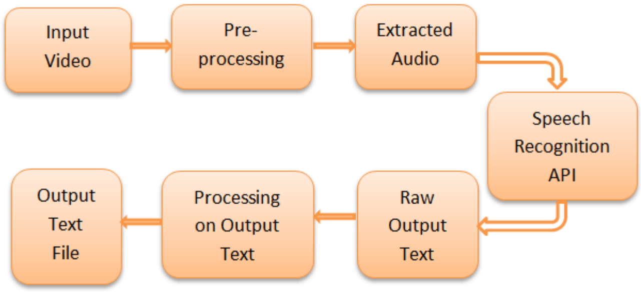

Fig -2: Proposed Methodology

### 1. Google API

Google Cloud Speech-to-Text permits developers to convert sound to text by applying powerful neural network models. The API acknowledges a hundred and twenty languages and versions, to support your worldwide user. Google Speech to text conversion powered by machine learning is offered for brief or long-form audio. You'll be able to send audio knowledge to the Speech-to-Text API, that then returns a text transcription of that audio file. Synchronous speech recognition returns the recognized text for brief audio (less than one minute) within the response as shortly because it is processed. To technique a speech recognition request for long audio, use Asynchronous Speech Recognition. Audio content area typically sent on to Speech-to-Text Cloud.

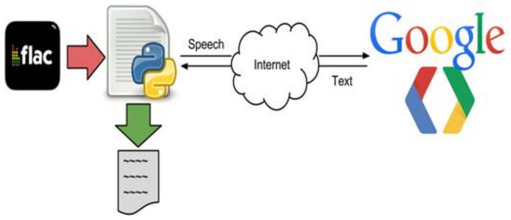

Fig -3: Google API Speech Recognition Process

### 2. Sphinx API

The common way to recognize speech is the following: we take a waveform then split it at vocalizations by silences then attempt to acknowledge what's being said in every utterance. To do that, we would like to take all possible combinations of words and try to match them with the audio.. We decide the most effective matching combination.

- An acoustic model contains acoustic properties for every senone. There are context-independent models that contain properties (the most probable feature vectors for every phone) and context-dependent ones (built from senones with context).

- A phonetic dictionary contains a mapping from words to phones. This mapping is not so effective. As an instance, only two to three pronunciation variants are measured in it. However, it's sensible enough most of the time. The dictionary is not the only technique for mapping words to phones. You may conjointly use some complex machine learning algorithmic rule.

- A language model is employed to limit word search. It defines that word may follow previously recognized words (remember that matching may be a serial method) and helps to considerably prohibit the matching process by stripping words that don't seem to be probable.

### 3. IBM API

The IBM Speech-to-Text service provides application programming interfaces that you will use to feature speech transcription capabilities to your applications. The service leverages machine intelligence to transcribe the human voice accurately. The service combines data concerning descriptive linguistics and language structure with information of the composition of the audio signal. It continuously returns and updates a transcription as additional speech is detected. The service provides numerous interfaces that build it appropriate for any application wherever speech is the input and a transcription text is that the output. The IBM Speech to Text service offers 3 interfaces for speech recognition:

- A WebSocket interface for establishing persistent, full-duplex connections with the service.

- An HTTP REST API interface that supports both sessionless and session-based calls to the service.

- An asynchronous HTTP interface that provides non-blocking calls to the service.

The Speech to Text service converts the speech into the written word. It can be used anywhere there is a need to bridge the gap between the spoken word and their written form. The IBM Speech-to-Text service uses machine intelligence to combine information about grammar and language structure to generate an accurate transcription. Converting speech to text is a difficult problem. Some general things to consider when using the Speech to Text service in your applications follow:

Speech recognition can be very sensitive to input audio quality. When you experiment with a demo application or build an application of your own that uses the service, try to ensure that the input audio quality is as good as possible. Conversion of speech to text may not be perfect. Tremendous progress has been made over the last several years. Today, speech recognition technology is successfully used in many domains and applications. However, in addition to audio quality, speech recognition systems are sensitive to nuances of human speech, such as regional accents and differences in pronunciation, and may not always successfully transcribe audio input.

### 4. Houndify API

Houndify is a platform that enables anyone to feature sensible, voice enabled, informal interfaces to something with an online association. Once you integrate with Houndify, your product can instantly perceive a good type of queries and commands. What makes Houndify distinctive is that instead of being one link in a chain of technologies, Houndify is one-stop destination for all of the technologies required to voice enable anything. If a developer does not want the rich results and only wants to use the recognition feature, he or she should enable the Speech to Text Only domain. This domain should be the only domain enabled for a client that wants to use this domain and returns the raw transcription of the audio as well as the formatted transcription.

### 3. RESULTS

Text transcripts are often used in automatic information retrieval tasks. However, one of the challenges associated with such retrievals is the accuracy of keyword transcription during the ASR process. For this, we have decided to evaluate our proposed system for lecture videos through the transcribed keywords and measured against the manual transcripts. Since no general agreement exists on how to automatically identify such keywords, we used a manually-generated text file.

**Table -1: Processing Time (In Sec)**

| Name of Video | Processing Time of Google API | Processing Time of Sphinx API | Processing Time of IBM API | Processing Time of Houndify API |
|---|---|---|---|---|
| Brain | 4.20 | 4.61 | 11.21 | 12.00 |
| Scope of variable in Python | Not Processed | 128.54 | 96.02 | 32.63 |
| Line charts | Not Processed | 151.58 | 141.51 | 29.33 |
| Number of Arguments | Not Processed | 146.60 | 106.68 | 32.22 |

Table-1 shows the processing time taken by different API for processing different video. Google API is used for the video less than 1 minute. IBM API took less time for processing a video than other API.

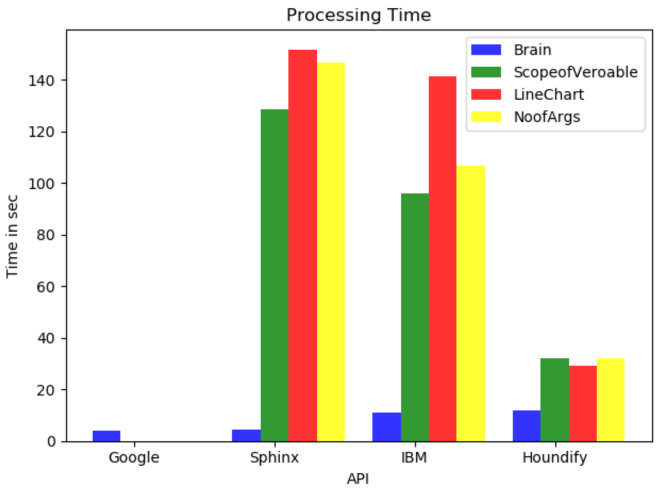

Chart -1: Video Processing time for different API

**Table -2: Accuracy of different API for brain.wav**

| Name of API | Accuracy (In %) |
|---|---|
| Google | 80 |
| Sphinx | 75 |
| IBM | 100 |
| Houndify | 90 |

Table-2 shows the accuracy of the 4 API's for brain.wav.

IBM API gives the highest accuracy as compare to other API and Sphinx API gives the lowest accuracy as compare to other API.

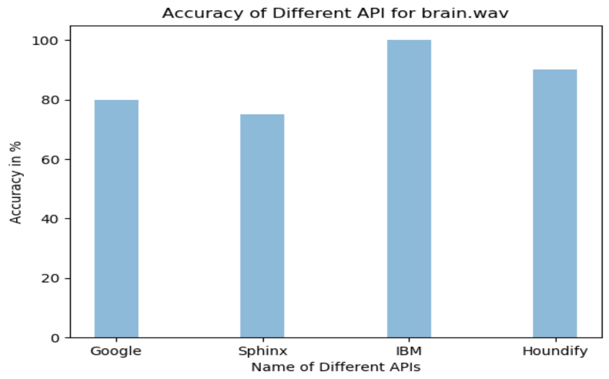

Chart -2: Accuracy of different API for brain.wav

**Table -3: Accuracy of IBM API for different video**

| Name of Video | Accuracy (In %) | Wrong Detection |
|---|---|---|
| Brain | 100 | 0 |
| Scope of variable in Python | 94.88 | 18 |
| Line charts | 81.94 | 82 |
| Number of Arguments | 96.61 | 22 |

Table-3 shows the accuracy of the IBM API's for different videos. If there is more background noise in video then it gives the less accuracy.

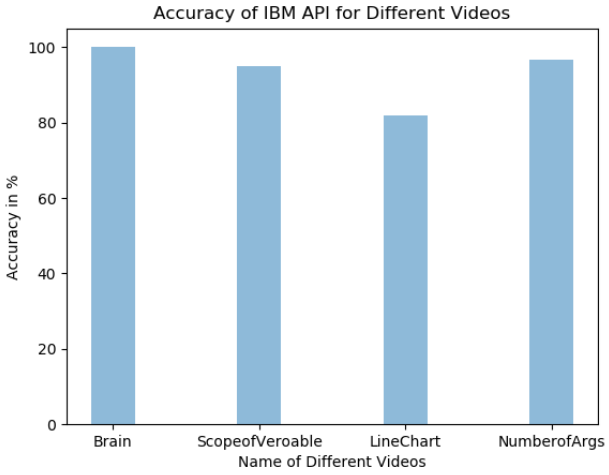

Chart -3: Accuracy of IBM API for different video

## 4. APPLICATIONS

1. Telecommunication and Multimedia

2. Health Care

3. Education System

4. Man-Machine Communication

5. Useful for people with disabilities

## 5. CONCLUSIONS

In a last decade e-lecturing has become more and more popular. Till now, in this area many people have analyzed the different video indexing and video retrieval techniques but Automatic lecture transcription is an important task for both researches and applications. It is a challenge for speech recognition as, in contrast to broadcast news, lectures typically present in higher changeability in terms of speaking style, language domain and speech fluency. So, the proposed system will help students to take their notes. This system will be helpful for the students who have hearing disability.

## REFERENCES

[1] Marco Furini and Silvia Mirri, "On Using On-The-Fly Students' Notes in Video Lecture Indexing". 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC)

[2] Haojin Yang and Christoph Meinel, Member," Content Based Lecture Video Retrieval Using Speech and Video Text Information", IEEE transactions on learning technologies, vol. 7, no. 2, April-june 2014.

[3] Kartiki Gupta, Divya Gupta " An analysis on LPC, RASTA and MFCC techniques in Automatic Speech Recognition System" IEEE2016

[4] Prachi Khilari" Implementation of Speech to Text Conversion" Journal July 2015

[5] Deepa V.Jose, Alfateh Mustafa, Sharan R" A Novel Model for Speech to Text Conversion" Journal January 2014

[6] E. Leeuwis, M. Federico, and M. Cettolo, "Language modeling and transcription of the ted corpus lectures". 0-7803-7663-3/2003 IEEE I - 232

[7] J. Glass, T. J. Hazen, L. Hetherington, and C. Wang, "Analysis and processing of lecture audio Data Preliminary investigations".

[8] C. Munteanu, G. Penn, R. Baecker, and Y. C. Zhang, "Automatic speech recognition for webcasts: How good are good enough and what to do when it isn't".

[9] Itunuoluwa Isewon, Jelili Oyelade, Olufunke Oladipupo "Design and Implementation of Text To Speech Conversion for Visually Impaired People" Volume 7– No. 2, April 2014

[10] André Gustavo Adami, "Automatic Speech Recognition: From the Beginning to the Portuguese Language"

[11] Nuzhat Atiqua Nafis and Md. Safaet Hossain" Speech to Text Conversion in Real-time" International Journal of Innovation and Scientific Research ISSN 2351-8014 Vol. 17 No. 2 Aug. 2015, pp. 271-277

[12] Yan Zhang, Andrew Ng "Speech Recognition Using Deep Learning Algorithms"

[13] James Glass, Timothy J. Hazen, Lee Hetherington, and Chao Wang "Analysis and Processing of Lecture Audio Data: Preliminary Investigations" MIT Computer Science and Artificial Intelligence Laboratory 32 Vassar Street, Cambridge, MA 02139, USA

[14] Ghada AlHarbi, Thomas Hain "Automatic Transcription Of Academic Lectures From Diverse Disciplines" by 978-1-4673-5126-3/12/$31.00 ©2012 IEEE398

[15] Marco Paleari, Benoit Huet, Antony Schutz, Dirk Slock "A Multimodal Approach To Music Transcription" 978-1-4244-1764-3/08/$25.00 ©2008 IEEE