REPORT

Transcribo

By

Avdhut Kamble

Meerali Naseet

ABSTRACT

Due to the Pandemic, the majority of Businesses and Educational institutes have switched over to online teaching and online meetings. Due to Network issues, device compatibility and server-side problems the students attending online classes are affected very badly, moreover the business meetings are also not attended by many professionals, and due to this they face many problems.

'Transcribo' project is created to solve this problem. It transcribes the Online video lectures/meetings to produce their summary/report resp., and also adds the highlights/topics to the content present, with features which add to the productivity of the peoples working online!

This project was created during the Codebreak2.0 Hackathon, using Python as a Backend Language and HTML & CSS is used for Frontend Development.

Table of Contents

Contents	Page No.	
Abstract	i	
Table of Contents	ii	
Chapter 1:	**Introduction**	**1**
Chapter 2:	**Objectives**	**2**
Chapter 3:	**System Design**	**3**
Chapter 4:	**System Implementation**	**4**
Chapter 5:	**Applications**	**5**
Chapter 6:	**Conclusion and Future Scope**	**6**
	References	**7**

Chapter 1 : INTRODUCTION

The project ‘Transcribo’ was developed during CodeBreak 2.0 Hackathon, which was conducted online.

This project is developed to solve and reduce the problems to some extent, faced by the students and all online medium users during Meetings or online teachings.

With the help of this project anyone can get the transcribed file which is not limited to summary only, but also highlights the main key points of the long transcription.

It uses the audio or video format file as an input and produces the transcript file as an output. For further, clarity and wider applicability and use we have given it a form of a Web application, so that it doesn’t get a limited to computer or Mobile users.

Chapter 2 : OBJECTIVES

The objective of this project developed is to provide the full transcripted file of audio/video along with the highlighted key words, so as to grasp the transcription in very short amount of time and henceforth help all the online platform users and specially able's too, to get more soother experience of online meeting & classroom platforms.

Chapter 3: SYSTEM DESIGN

The System design consists of Python as backend, HTML/CSS for frontend and FLASK as web framework.

The Following Pythons libraries were used:-

- Flask
- NLTK
- SCIKIT-LEARN
- GENSIM
- MOVIEPY
- SPEECH-RECOGNITION
- WAVE
- MATH
- CONTEXTLIB

The following technologies were used:

- TfidfVectorizer
- Gensim’s Text Summarization tool for keywords
- Speech to text By Google
- Natural Language Processing for the whole block

Chapter 4: SYSTEM IMPLEMENTATION

Firstly the project is based on Flask, so all the data dealing is done by the framework smoothly. The first real task of the program is to convert the video file provided to an .wav audio file for transcription. We did this using a package called moviepy. Then the output audio clip was gone through Google's Speech recognition API in a loop to cover the whole audio. The transcription from the API was saved in a local .txt file for better use. We tried to append the whole transcription to a single string, but we got some runtime issues (Basically out of RAM). Though Google API is one of the best speech Recognition API out there, it doesn't punctuate the text in the output. Thus, we came across a pre-trained punctuation model based on theano and a package called "punctuator". We punctuated our text with the model itself. The summarization process: For getting the summary out of the text, we went through several options, and chose Scikit learn's tfidfvectorizer. The process seems to be complicated but was quite easy. First, dividing the text with sentences with the help of punctuations, then tokenize each word with the help of nltk's word_tokenise. Then getting the average of frequency of words and removing stopwords/regular words from the text. Then calculating the importance or Accuracy of the words in the sentence. Then with that accuracy, calculate which sentences are more valuable. Lastly, adding the sentences with the highest threshold to the cut of final summary. And all the data we obtained from the following system is given out as html with the help of Flask!

Chapter 5: APPLICATIONS

This project has wide range of applications, some of which are listed below:-

1. For School/College Students, to get notes, if they missed any class by chance.
2. For School/College Students, to get notes with highlighted keywords, to revise the class during examinations.
3. For Business professionals to get the minutes of meeting.
4. For all the online events and workshops, which happens online, and after that report is to be prepared.
5. To know the name of medicines and brief of call, when the doctor consults the patients online using video/audio call.
6. To store the financial based startup’s (like policy bazaar) call data in transcripted files, for future evidence and reference.
7. For all Specially abled who can’t hear the audio/video can see the transcription.

Chapter 6 : CONCLUSION & FUTURE SCOPE

This project has many applications and benefits to humans across the globe with its specially designed features and platform independent service, as it is a web application.

The future of the world is Digital in nature, and this project can prove to be the one of the most important and needed to all the users who are digitally connected to the virtual world, while providing virtual services, can use it without any technical knowledge, due to its simple and easy to use operating features.

Within a few years, most of the Corporate Business will get shifted to online platforms, where this project will play a very key role in each and every meeting of theirs.

It will not get limited to the Corporate world, but also prove to be efficient in Educational Institutes as they will also start to use digital platforms for teaching.

REFERENCES

- W3Schools Online Web Tutorials
- Scikit-Learn’s Official Documentation
- Gensim’s Text Summarization
- Speech recognition Guide realpy