An approach towards voice-based video content search

Tanmay Bhowmik[1], Aman Rai[2], Soumya Pandey[3], Prudhviraj Boddu[4], Nilay Patel[5], Veni S[6], Vignatha Manchala[7]

[1] Department of Computer Science Engineering, Bennet University, Greater Noida, India
tanmaybhowmik@gmail.com

[2] National Institute of Technology Silchar
726amanrai@gmail.com

[3] Bhilai Institute of Technology-Durg
somi1511p@gmail.com

[4] Siddarth Institute of Engineering & Technology-Puttur
prudhvirajboddu@gmail.com

[5] Sarvajanik College of Engineering & Technology-Surat
patel000nilay@gmail.com

[6] Saintgits College of Engineering-Kottayam
veni.s1721@gmail.com

[7] VNR Vignana jyothi Institute of Engineering & Technology-Hyderabad
vignatha.99.rs@gmail.com

Abstract. This paper describes challenges and solutions for building a successful voice-based video content search system. People have been typing to search queries online for the past several decades. But it is easier and more convenient to use voice search than typing the whole text. Voice based search is dominating the market since last few years. With the expanding dominance of smartphones this voice search also steadily growing. We have implemented voice search instead of web search, specifically it is voice-based video content search. A particular word has been provided as voice input in a video file and it is tried move directly to that point of the video. The ffmpeg tool has been used to convert the video file into corresponding file format. Speech recognition module has been used to convert audio file into text. The required attributes from speech recognition module provide us all the possible words similar to the given audio input, so that we won't miss any possible word or sentence. The texts are divided and saved in different locations in an array for every interval of 3 seconds. To access the voice input word, the index value has been multiplied to the duration to get the timestamp where the input word lies. Then the user can select the timestamp shown on the screen and move to the intended location of the video.

Keywords: Speech recognition, voice search, transcription, timestamp, precision, accuracy.

1 Introduction

The ability to perform web searches simply by speaking them to your mobile phone, first appeared around 2008 on iPhone and Android phones in US English [1]. Soon after it was obvious that developing voice search for other languages, especially ones where keyboard input is more difficult, provided even greater benefits than for English. Speech is one of the most convenient way of medium for communication. As technology evolves, human is going to be more dependent on machines. Speech based communication is the easiest one to communicate between human and machine. During last few years, speech-based control on system is gaining popularity heavily. Speech-based searching technique emerged as one of the easiest searching techniques for the users. This searching technique can be improvised in speech-based or voice-based searching in video content. Speech recognition technology is there in behind of voice-based search which enables users to just voice search their queries in internet [2]. Speech-based or Natural Language based searching has very good performance under Deep Learning based framework [5]. It became more convenient to search with voice instead of typing the whole sentence or word in the search engine. Now these technologies are being implemented for voice search in videos. This is what the paper is about, voice-based content search in video files. Using this technique, a person will not require to search his desired video location in any particular video file by repeated mouse click. The most challenging part was getting the accurate timestamps. At first, we tried using several available APIs of Google cloud [3], IBM Watson etc. But including them in code led to more and more errors also not much accurate transcription for words, so we did that manually by breaking the entire video's text into chunks of small duration fixed time, here it is 3 seconds, and stored them into an array. Time interval of 3 seconds has been taken to ensure the accuracy and precision of timestamps. If we take time interval of 5 seconds, then we could only get better accuracy but poor precision. And if we take time interval less than 3 seconds, the precision could be good, but accuracy will get affected. After generating array, to access the particular content of voice inputted word, we just multiplied the index value to duration to get the timestamp after checking if the input word lies in that particular chunk's text. Then, the user can select any of the timestamps displayed on the screen to play the video from that point in the video. Thus, this paper contributed in making an API on our own for voice based voice search in video files without relying on existing APIs and problems while integrating them into our code.

2. Related Work

2.1. Google speech-to-text API

Google is essentially one of the top-most API developers in the world. Google Speech-To-Text was introduced in the year 2018 [6]. The error rate with Google Speech-To-Text is comparatively lesser than other APIs. One of the main success behind this API is its impressive accuracy. It recognizes over 120 languages and users can choose respective model from many machine learning models. Also, automatic language recognition is also provided by this API.

The Google Speech-to-text is not free or can be used for video transcriptions. It may cost $0.006 per 15 seconds for videos up to 1-hour length. For videos longer than 1-hour costs $0.012 per 15 seconds. And it has limited custom vocabulary builder. The main problem we faced was getting errors in the timestamps from it. Video file length was another disadvantage of Google Speech-To-Text API.

2.2. IBM Watson Speech to Text API

The IBM Watson Speech to Text is another tool which provides Speech to text conversion [7]. IBM Watson is good at generating natural language patterns which are needed for most developers. It supports both audio and video file transcriptions. Also, able to differentiate between multiple speakers. After transcription we could get words or sentences along with their timestamps and corresponding confident values. Various API reference manuals of the same is provided by IBM in their documentation page.

Watson Speech to Text takes more time for transcription of videos of about 50 minutes. Also. it transcribes audio files effectively within short span of time. Taking longer time is a problem. It supports only limited number of languages like 11. The words or sentences after transcription were not accurate enough. We were not getting accuracy as expected with Watson.

2.3. Using python code

Speech recognition can be implemented in simple python code by importing speech recognition module in the code snippet. Also, with help of tools like ffmpeg, pyglet, cv2 we have implemented the same in the python program. While executing the program we are asked about to input the video file and our voice input. After then a window will pop-up with corresponding timestamps of the word or sentence. User can choose from it and move to that particular part of the video. The ffmpeg is a tool that converts video into audio formats. It can encode in real-time. And it converted the input video into a audio of wav format. We used speech recognition module to convert audio into text. This show all attribute of the speech recognition module is set to true to provide us all the possible words similar to the given audio so that we don't miss any possible word or sentence due to accent variation or due to noise or other errors. Also. it allows us to search a word even if there is some spelling error occurs. Then we made the duration of the audio get stored in a variable. pyglet library used here is for object-oriented application programming interface. And cv2 library supports various operations on images, videos. Here, it serves the purpose of opening the video at a particular timestamp.

The video will get played with a bit of error of 2-3 seconds. We have used the trim function to trim the video up to timestamp and piglet function to play from that point. Percentage error will be different for different videos and type of input given whether it is a word or sentence. Less error with word as voice input. We will get different accuracy with different inputs but more or less video gets played with a bit of error.

Voice Based search is dominating the market since the start of the decade. And according to recent surveys more than 50% people prefer voice searching [7]. We have implemented voice search in a bit different way instead of web search as it is voice-based voice search in video content.

3. Methodology

3.1. Proposed Method

The basic block diagram for the proposed method has been depicted in Figure 1.

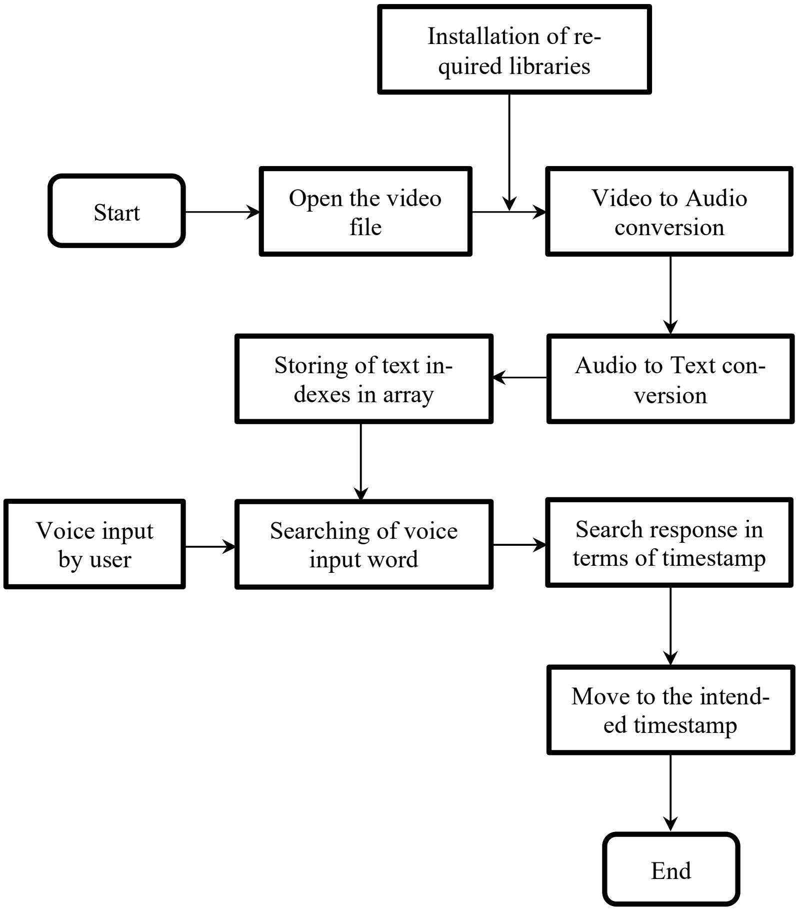


Figure 1: Basic block diagram of voice-based voice content search in video file

In the basic block diagram, all the individual steps are clearly shown. According to the diagram, after opening the video file the task is being performed to convert the video file into audio format. The converted audio is of type 'mp3'. Some required libraries need to be installed to do this. Then other steps are followed. The steps are described here.

3.2 Video to Audio conversion

In this work the input video file is converted into audio format. The FFMpeg tool has been used for this task. The FFmpeg is an online free and open-source project consisting of a vast and simple software suite of libraries and programs for handling video, audio, and all other type of multimedia files and streams [8]. At its core there is the FFmpeg program itself, designed for processing of video and audio files, and widely used for basic editing (trimming and concatenation), format transcoding, video scaling, video post-production effects, and standards compliance. FFmpeg is part of the workflow of hundreds of other software projects, and its libraries are a core part of software media players such as Windows media player or VLC player, and has been included in core processing for YouTube videos. Codecs are used for the encoding and/or decoding of most of all known audio and video file formats is included, making it highly useful for the transcoding of the media files into a single common format.

Through this library the input video has been converted into audio as per requirements of the task performed.

3.3 Audio to Text conversion

Speech-to-text software is a type of software that takes audio content and transcribes it into text in a word processor or other display source. This type of speech recognition software is highly valuable to anyone who needs to generate a lot of (text) written content without a heavy load of manual typing. Also for advance speech recognition as in [5] it is useful for people with disabilities who used to face difficulties to use a keyboard or a writing tool.

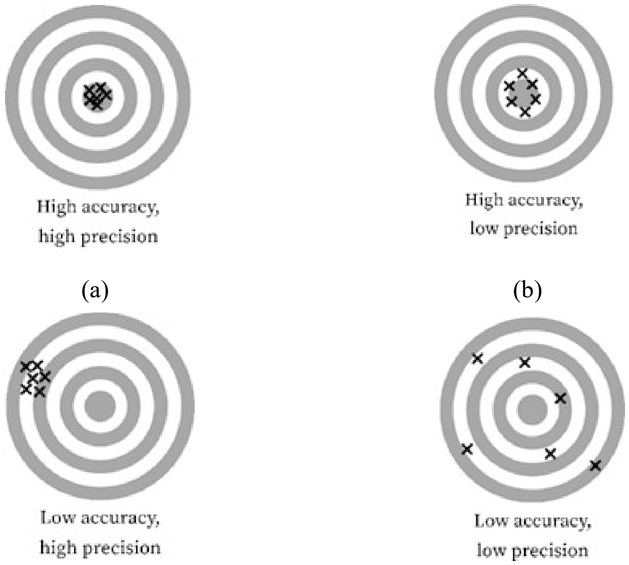
Speech-to-text software can also be called as voice recognition software or speech recognition software. This software can also be used for voice queries in mobile device as seen in the google Arabic text [3].

The next step is to store the converted texts. During conversion of an audio into text format, the texts are stored into an array with consecutive indexes with an interval of 3 seconds. Then we search the word or the input command in the array in order to get the required timestamps.

In the next step, the user will give the voice input. This voice input is basically is given to search the intended content in the initial video file. The input word will be searched in the array of text-indexes. This search response will be in terms of timestamps. System will provide the available timestamps with that specific word. Then the user can move to any of the timestamps to play the video from that time zone.

4. Experimental Results

For every voice input we get time stamps of the word or sentence searched by user and the video plays from the time selected by the user. Time Interval of 3 seconds for storing the text into an array has been taken on an experimental basis to ensure the accuracy and precision of the timestamp. If we would take the time interval of 5 seconds, then we would get a better accuracy but poor precision. And if we would take a time interval of less than 3 seconds let say 1 seconds then precision would be good, but accuracy will be affected.



(a) (b) (c) (d)

Figure 2: Variation of Precision and Accuracy

In the above-mentioned figures, all the variations of precision and accuracy has been depicted from Figure 2(a) to Figure 2(d). In case of taking 5 second time interval, the situation will match Figure (b). In case of time interval has been taken as 1 second, situation will be similar as Figure (c). All the other options where the timestamps were considered as more than 5 seconds are ended with a situation like Figure 2(d). That is why to maintain an ideal combination of accuracy and precision, time interval of 3 seconds has been chosen based on various trial and error methods. The situation with optimum combination of precision and accuracy is found in Figure 2(a).

A comparative analysis of the results has been depicted also with a bar chart representation. In Figure 3, the bar chart is shown where L informs about timestamps. It is clearly visible that when L = 1 and 5 that is in case of 1 second and 5 second timestamp the variation of precision and accuracy is too much. Graphically, an optimum combination is found in case of 3 second timestamp.

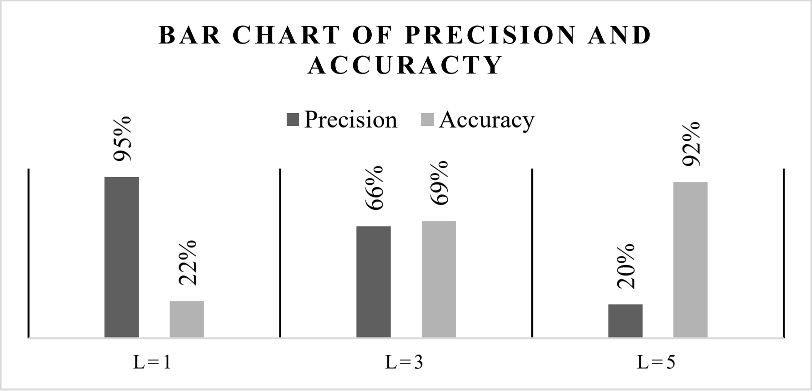


Figure 3: Bar Chart of Precision and Accuracy

A further analysis of the result will highlight the point that when the duration of each timestamp is being decreased gradually, the result will be much precise. This is because when timestamp is considered of less duration the text searching is more compact. So, the result becomes more precise. But less duration of timestamp means a greater number of timestamps. So, text index searching will be done among higher number of timestamps. As a result, accuracy is decreased.

As a contrary of above-mentioned situation, when the timestamp is considered as 5 second or more, a smaller number of timestamps will be generated. So, text index searching procedure will be executed among a smaller number of timestamps. So, there is always a chance of getting better accuracy. However, in a single higher duration timestamp, a greater number of text indexes will be found. So, it is obvious that the result will be less precise.

In the situation when the duration of timestamp is 3 second, the precision is less than the situation with 1 second timestamp and accuracy is less than 5 second timestamp. But it is an optimum combination of accuracy and precision.

5. Conclusion

In this study, voice-based voice search system has been built. While developing the systems we found that the simplified process described above based on modeling can help the user to watch a video at particular time of interest which in a way help in time saving. The proposed model takes voice input for a video and gives the time stamps to the user. Then the video will play from the time selected by the user. The accuracy differs based on the video used by the user. The video may play with a bit error in the time. Voice based voice search can be implemented in different applications which contains video contents.

Lots of difficulties are still there in developing this model. The primary concern associated with this work is the precision and accuracy. In optimum condition, the precision and accuracy has been observed as 66% and 69% respectively. Which needs to be improved.

Another problem is the searching time to identify the text indexes. When the voice input of the user is started to search in the array of text indexes, sometimes it is taking longer time to complete the searching. The searching time should be less. Extensive work is going on to improve the precision and accuracy along with a target to minimize the searching time.

6. References

1. Hurst-Hiller, O., & Farago, J. (2010). *U.S. Patent No. 7,672,931*. Washington, DC: U.S. Patent and Trademark Office.
2. Ju, Y. C., & Wang, Y. Y. (2013). *U.S. Patent No. 8,589,157*. Washington, DC: U.S. Patent and Trademark Office.
3. Guy, I. (2016, July). Searching by talking: Analysis of voice queries on mobile web search. In Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 35-44).
4. Biadsy, F., Moreno, P. J., & Jansche, M. (2012, March). Google's cross-dialect Arabic voice search. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4441-4444). IEEE.
5. Schalkwyk, J., Beeferman, D., Beaufays, F., Byrne, B., Chelba, C., Cohen, M., Kamvar, M. and Strope, B (2010). “Your word is my command”: Google search by voice: A case study. In Advances in speech recognition (pp. 61-90). Springer, Boston, MA.
6. Shakhovska, N., Basystiuk, O., & Shakhovska, K. (2019). Development of the Speech-to-Text Chatbot Interface Based on Google API. In MoMLeT (pp. 212-221).Santiago, F., Singh, P., & Sri, L. (2017). Building Cognitive Applications with IBM Watson Services: Volume 6 Speech to Text and Text to Speech. IBM Redbooks.
7. Defourny, J., & Nyssens, M. (2008). Social enterprise in Europe: recent trends and developments. Social enterprise journal.
8. Tomar, S. (2006). Converting video formats with FFmpeg. Linux Journal, 2006(146), 10.