International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 9 (2015) pp. 24741-24746

http://www.ripublication.com

Automated Subtitle Generation

N. Radha and R. Pradeep

Dept. IT, SSNCE, Chennai, India

Abstract:

In this paper, we present an automated subtitle generation system using speech recognition. Subtitles are generated by extracting the audio signal data from the input videos. The audio data is then converted into a suitable format and automatic speech recognition is performed. Finally a subtitle film or video is generated from the recognition results. The subtitles for the input video is generated in English language, given that no more than one person talks at a time and there are no major pauses in the speech signals.

Keywords: Mel frequency cepstral coefficients, Hidden markov model, Automatic speech recognition

1. INTRODUCTION:

With the invention of films and technology, the need for conveying the dialogues of actors to audience also increased. In recent days many video- on-demand outlets requires subtitles and closed captions which helps in human-machine communications. Generally videos are the most popular multimedia artefacts and the sound role in a video holds a very important place. Many research shows that three percent people in the US are "functionally deaf" and around seventeen percentage of Americans report some sort of hearing impairment defects, this percentage is high or low reflected worldwide. Hence, it is essential to make the understanding of the speech in a video comprehensive for people with auditory problems as well as for people with gaps in the spoken language.

Focusing this problem, the natural way lies in the use of subtitle generation [1]. Consequently, the subtitles are displayed to give a good experience for the viewer, making it easy to understand the language without obstructing the video experience. There are some existing software provides support for the manual creation of subtitles, but it is time-consuming and tedious. Hence, we have proposed a technique for automated subtitle generation system using speech recognition.

Fig1. shows an representation of automated subtitle generation which contains the following steps audio extraction as an first step, secondly speech recognition, time

synchronization is an third module, and finally subtitle generation.

The audio extraction and speech recognition modules are discussed in section 2. Section 3 briefly explains about time synchronization, and subtitle generation are discussed. Section 4 explains about the experimental study of subtitle generation with an examples. Section 5 summarizes the work.

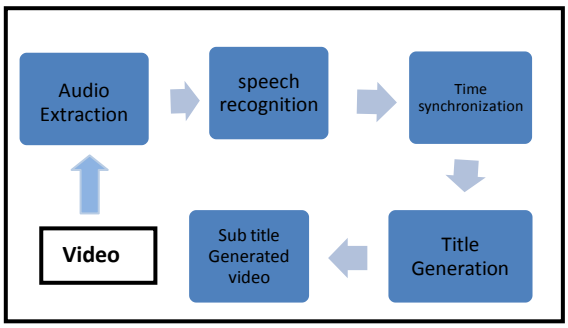

Fig1. Block diagram of subtitle generation system

2. SYSTEM DESIGN:

This section presents the system design of our work. The audio extraction and speech recognition is discussed in the first session. In the second section the implementation phase in which various processing steps like feature extraction, modeling and the recognition of customized modules are discussed.

2.1 Audio Extraction:

Audio extraction is done by using *ffmpeg*. *ffmpeg* is a very fast video and audio converter that can also grab from a live audio as well as the video source. Conversion between arbitrary sample rates and resize video on the fly with a high quality poly phase filter is also can done by *ffmpeg*. The audio extraction module is automated to operate upon the click of a button and extracts the audio file (.mp3) from the input video. The commands to perform these actions are executed using a windows "batch file." Further audio files are processed by speech recognition systems. The steps involved in speech recognition process is discussed in the next sub section.

2.2. Speech recognition:

Automatic speech recognition is a process used to recognize speech uttered by a speaker shown in Fig 2. Speech parameter extraction is the next step. Various methods are proposed for efficient extraction of speech parameter for recognition, the MFCC method with recognition method such as HMM is more dominantly used. The speech data is processed to extract 39 dimensional MFCC for every 15 msec frame, shifted by 5 msec

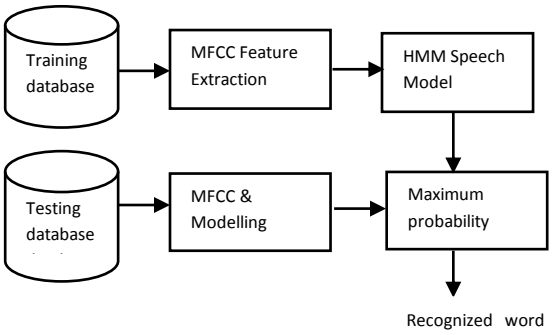

Fig 2. Block diagram of ASR system

An initial HMM model is used to begin the training process. The initial model can be randomly chosen or selected based on a priori knowledge of the model parameters. In this work, the models are re-estimated using the viterbi algorithm which maximizes the likelihood of model M for having generated the observed sequences. The HMMs are trained the resultant output of recognition units determines the class to which it belong in dataset. The best state transition path are calculated during classification process, and determines the likelihood value using the viterbi search algorithm[3] for recognition. The compared likelihood value of each HMM and determines the input word as the index of HMM with the maximum likelihood value. The speech is modeled by a 5-state left-right HMM, of which only three states are emitting. The recognition is tested for different states of HMM. Special attention is paid to the silence models since in the case of audio data even in the case when nothing is uttered the signal's energy can still be very high. Generation of text file with the recognized text as an output. Subtitle generation using this text file is discussed in the next section.

3. SUBTITLEGENERATION:

The number of words displayed on the screen in a particular time period should not exceed a certain limit, so that the text does not obstruct the video. The subtitle file is generated with a number of six second intervals. The total number of characters in the text file (sample1.txt) is calculated using a simple code in java. The duration of the video in seconds is extracted using delimiter from the details contained within the ffmpeg library. The total number of six second intervals is given by: (Duration /6).The number of characters to be displayed during each six second interval is calculated using the formula:

Total No: of characters / (Duration /6).

Since the characters are printed into the file from a buffer it is essential that at the end of an interval the word that is printed is complete. This is done by checking whether the character in the buffer is a blank space before proceeding to the next - time synchronization done using the formula develops an offset over time. To avoid

this we alternate between two blocks in which the second block “rounds off” to the previous blank space. The total number of characters in the text file containing the recorded speech is calculated using code written in Java. The file handling functions available provide adequate support for this purpose. The SRT file should generate in its standard format with the:

1. Serial number: The line number of the text displayed
2. Time code: This specifies the start time and the end time of the text data displayed on the user screen.
3. Subtitle text: This is the actual subtitle to be displayed

4. EXPERIMENTS:

This section presents the experimental system to be setup. The experimentation has been carried out by training the speaker dependent system for 5 speakers (2 Male and 3 Female), and for a minimum of 50 utterances. The subtitle generation test has been conducted for the above videos with the language specific English. The dataset is also collected from 5 speakers consist of sequence signals and tested for the same case.

Fig 3. Input sample video

The sample video with normal speech is depicted in fig 3. The audio extraction carried out using FFMPEG. Audio extraction samples are represented in fig 4. The saved audio file further processed by speech recognition system. In speech recognition system, normal speech data is processed to extract 39-dimensional MFCC for every 15 msec frame, shifted by 5 msec.

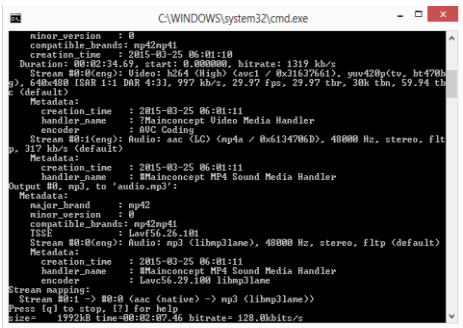

Fig 4. Audio extraction using FFMPEG

The SRT file generated in its standard format with the time synchronization is shown in fig 5. The file consists of serial number of video, time code, and with the subtitle representation.

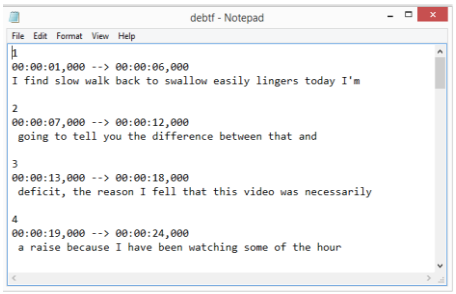

Fig 5. Time Synchronization

To measure the performance of our system, percentage accuracy was calculated for each of the 5 speakers for each module using the following formula. The performance evaluation is shown in Table 1.

$$Percentage\ Accuracy\ (\%)\ =\ \frac{No.of\ test\ cases\ identified\ correctly}{Total\ number\ of\ test\ cases}$$

Table 1: Word accuracy for overlapped speech

SlNo	Time	Words	WA%
1	00:55:23	240	81.25
2	00:52:35	230	75.4
3	00:49:35	210	84.7
4	00:51:64	225	77.6
5	00:59:45	265	79.2

For the overlapped speech the word accuracy is low when compared with non-overlapped speech. This system works better for the case of speech with non overlapping conditions. The recognition performance of non-overlapped speech results are listed in table 2.

Table 2: Word accuracy non-overlapped speech

SlNo	Time	Words	WA%
1	00:55:23	240	97.9
2	00:52:35	230	96.7
3	00:49:35	210	96.1
4	00:51:64	225	93.7
5	00:59:45	265	94.7

The subtitles are generated from the speech recognition modules. Subtitles then automatically embedded to video and it is properly synched. The subtitle generated video is represented in fig 6.

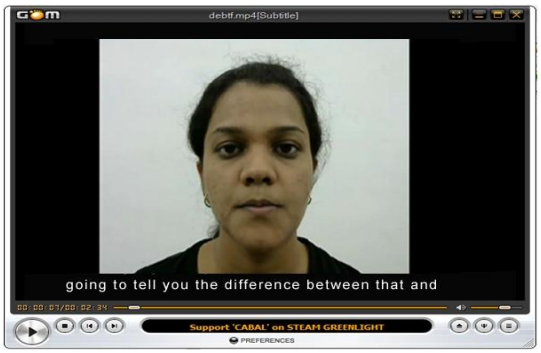

Fig 6. Output video with subtitles

5. CONCLUSION:

In this paper we proposed system for generation of subtitles for English. ASR is implemented using MFCC for feature extraction with HMM for recognition. It provides good accuracy when the system is tested with the speaker dependent recognition. The future work that can be seen from this is to design a system for generation of closed captions along with subtitles.

REFERENCES:

[1]. Boris Guenebaut, "Automatic Subtitle Generation for Sound in Videos", Master Thesis, Department of Economics and IT, University West, Report Nunber: TAPRO 02, 2009.

[2]. Rabiner, L. and Juang, B.-H., Fundamentals of speech recognition, PTR Prentice Hall. 1993.

[3]. L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) ,pp.136–257,1989.