

# Automatic Subtitle Generation in Real-Time

Prof. SmitaJawale<sup>1</sup>, Ankit D. Singh<sup>2</sup>, Pritesh C. Mane<sup>3</sup>, Abhishek V. Thakur<sup>4</sup>

(Computer Science, Vidyavardhini's College of Engineering and Technology, Vasai)

## Abstract:

The last ten years have been the witnesses of the emergence of any kind of video content. Moreover, the appearance of dedicated websites for this phenomenon has increased the importance the public gives to it. In the same time, certain individuals are deaf and occasionally cannot understand the meanings of these videos. They don't have any readable transcribed manuscripts to refer. So it becomes difficult to such people to interpret these videos. So we put the sample video through a 3-step process. i.e. Audio Extraction, where audio is extracted from sample video. Speech Recognition, where audio is transcribed and Subtitle Generation, where the generated text is timestamped with sample video and displayed in the Graphical User Interface (GUI) player. Accuracy rate of this attempt is not up to satisfaction. Parallel decoding, punctuation insertion are some of the issues to be worked on.

**Keywords**—transcribed manuscripts, audio extraction, speech recognition, subtitle generation, parallel decoding, punctuation insertion.

## I. INTRODUCTION

Over the exceeding years of technology we have come across various methods of interacting and manipulating with videos more efficiently. Speech-To-Text display is one of them i.e. subtitles that we use in order to elaborate movie dialogs for the purpose of understanding, for the sake of deaf individuals, etc. So video subtitles (speech-to-text) interaction plays a vital role in social aspect of society. Till date this is done by various subtitle editors that exists that involve excess of pre-run coding. We in this project are trying to make it real-time i.e. instantaneous generation of subtitles. [5]

## II. LITERATURE REVIEW

Various subtitle editors that are available till date are as follows :-

### IMTOO DVD RIPPER:

- Convert Home DVD and convert to almost all video or audio formats
- HD Home DVD converter, convert Home DVD to HD videos
- Numerous video effects to choose from
- Add subtitles and soundtracks to movies
- Video-editing features: Clip, Merge, Split and Crop
- Convert files compatible with most portable players, newly support Apple New iPad and Google Android-based phones
- ImTOO DVD to Video supports multithreading converting

- Fast converting speed and high quality conversion
- Can't rip/convert DVD with copyright. [1]

### XILISOFT DVD RIPPER

- Convert Home DVD to more than 160 video and audio formats.
- Convert Home DVD to multimedia devices like iPhone/iPad/iPod, Galaxy.
- Convert Home DVD to HD videos like H.264/MPEG-4 AVC, HD WMV.
- 5X Speed Promotion with NVIDIA CUDA & ATI APP technology.
- Can't rip/convert DVD with copyright. [2]

The above softwares are used for audio extraction purposes. i.e converting any video into any suitable file format.

### SUBTITLE EDITOR

- Convert Home DVD to more than 160 video and audio formats.
- Convert Home DVD to multimedia devices like iPhone/iPad/iPod, Galaxy.
- Convert Home DVD to HD videos like H.264/MPEG-4 AVC, HD WMV.
- 5X Speed Promotion with NVIDIA CUDA & ATI APP technology.

- Can't rip/convert DVD with copyright. [3]

**SUBTITLE WORKSHOP**

- Subtitle Workshop is a free application for creating, editing, and converting text-based subtitle files.
- User-friendly, multi-language, customizable interface (including customizable shortcuts and translation mode).
- Comprehensive customizable system for automatically or manually detecting, marking, and fixing various timing and text subtitle errors.
- Supports style tags (bold, italic, underline) and color tags in the subtitle text -- including full tags support for multiple tags in a single subtitle.
- Supports displaying and manipulating the timing pauses (the time gap between two subtitles) and offers a CpS (Characters per Second) system.
- Multi-level Undo-Redo system.
- Integrated video player with customizable subtitles preview and full screen mode that would play any video or audio format the system has codecs installed for.
- Offers various information about the subtitle file, including custom information based on user-defined rules about the timing or the text.
- Supports the usage of external Pascal scripts, as well as find-and-replace text scripts (called OCR Scripts). [4]

These softwares are used for pre-run subtitle editing purpose. A time dedicated hardwork and editing enables us to make avail the subtitles that are to be used when the video is played. The main drawback to all these editors are that they are not real-time. This drawback id the purpose of our system.

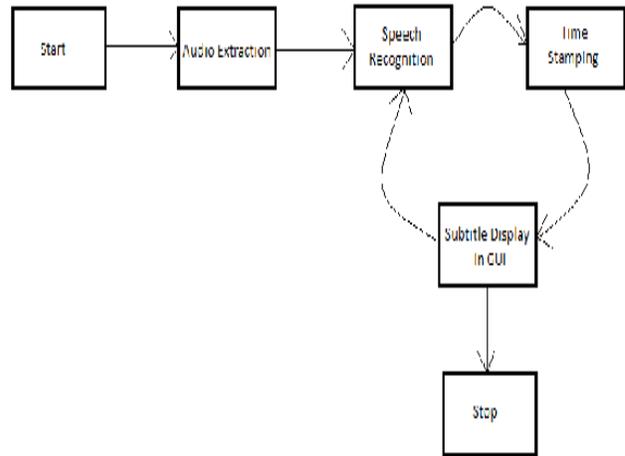
**III. EXISTING ALGORITHMS**

As described earlier only subtitle editors are available till date. Their description and features are given in the preceding point. But no 'subtitle generator' as such exists.

**IV. IMPLEMENTATION**

There are three steps in implementation of this project.

- 1.Audio Extraction
- 2.Speech Recognition
- 3.Subtitle Generation.



**1.Audio Extraction :-**

A media movie is generally composed of a video track and an audio track. During the montage, the tracks are gathered together and the final artefact is a single file. In this section, we study the feasibility to isolate the audio track from the rest of the file in order to solely process sound. A google research on the Internet for 'extract audio from movie' provides multiple links to software but few tutorial or tip to implement it on his own, especially in Java for this thesis concern. We are using a custom library in Java named JAVA AUDIO VIDEO ENCODER(JAVE). It is developed by Sauronsoftwares a UK based IT firm.

The JAVE (Java Audio Video Encoder)[6] library is Java wrapper on the [ffmpeg](http://ffmpeg.org/) project. Developers can take advantage of JAVE to **transcode audio and video files** from a format to another. In example you can **transcode an AVI file to a MPEG** one, you can change a **DivX video stream into a (youtube like) Flash FLV** one, you can **convert a WAV audio file to a MP3 or a OggVorbis** one, you can **separate and transcode audio and video tracks**, you can **resize videos**, changing their sizes and proportions and so on. Many other formats, containers and operations are supported by JAVE.

It works on the principle of Audio/Video encoding. The most important JAVE class is *it.sauronsoftware.jave.Encoder*. *Encoder* objects expose many methods for multimedia transcoding. In order to use JAVE, you always have to create an *Encoder* instance:

```
Encoder encoder = new Encoder();
```

Once the instance has been created, you can start transcoding calling the *encode()* method:

```
public void encode(java.io.File source,  
java.io.File target,  
it.sauronsoftware.java.EncodingAttributes attributes)  
throws java.lang.IllegalArgumentException,  
it.sauronsoftware.java.InputFormatException,  
it.sauronsoftware.java.EncoderException
```

The first parameter, *source*, represents the source file to decode. The second parameter, *target*, is the target file that will be created and encoded. The *attributes* parameter, whose type is *it.sauronsoftware.java.EncodingAttributes*, is a data structure containing any information needed by the encoder. In order for further processing we need output in .wav format. So ffmpeg code of the respective format i.e. 'pcm\_s16le' is used to get the desired output.

```
AudioAttributes audio = new AudioAttributes();
```

```
audio.setCodec("pcm_s16le");
```

## 2.Speech Recognition :-

Speech Recognition is done by using an open source software called CMU SPHINX-4. [7] The extracted audio in .wav format is given as input to the **transcriber program**. The programming is done in java. Netbeans is used as IDE. Sphinx provides developer with three elements.

1. Acoustic Model
2. Dictionary
3. Language Model

Acoustic model deals with acoustic configuration of the sample speech input provided to the transcriber. It analyses the audio sample rate, frequency and compatible format(.wav). Sampling rate must be less than 16000 Hz, mono, Microsoft pcm.

Language dictionary deals with all words and alphabets storage database needed for a particular language. There is different storage for different languages. Transcriber searches for all the word content in language dictionary by analyzing the sample audio and tries to match the found words.

Language model is the required grammar for the language under use. The sample audio is analyzed for grammatical errors and corrected.

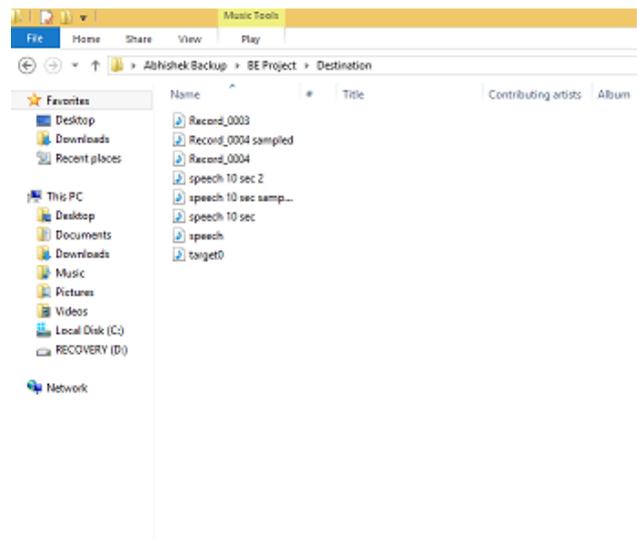
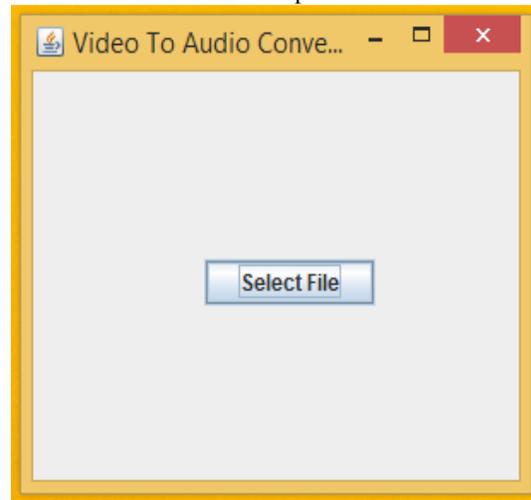
## 3.Subtitle Generation:-

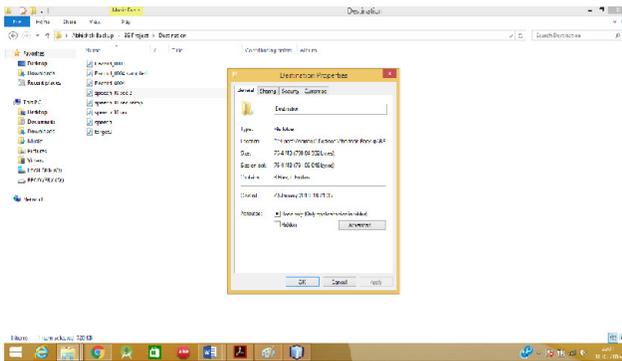
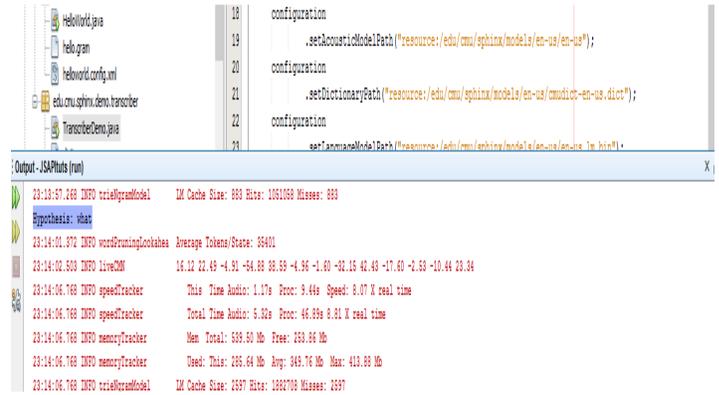
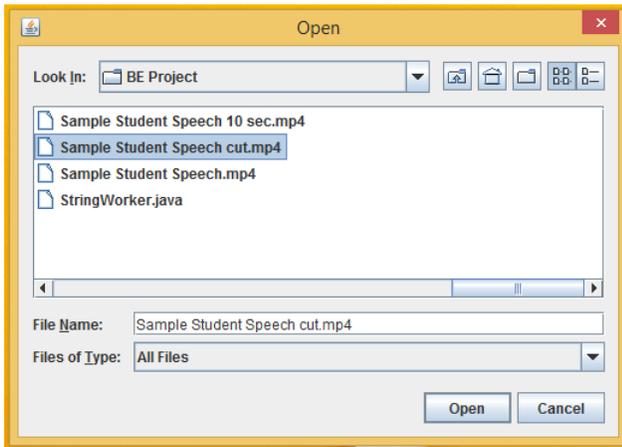
The module is expected to get a list of words and their respective speech time from the speech recognition module and then to produce a SRT subtitle file. To do so, the module must look at the list of words and use silence(SIL) utterances as delimitation for two consecutive sentences. [8] The list of words transcribed is then timestamped to introduce the synchronization. However, we face up to some limitations.

Indeed, it will not be able to define punctuation in our system since it involves much more speech analysis and deeper design. Orientations about this problem will be discussed in the conclusion.

## V. RESULT

In the first step of audio extraction we get extracted audio in .wav format from the sample video.





Audio is extracted from the sample video here and shown as output.

Transcriber Demo :-

```
public static void main(String[] args) throws Exception {
    Configuration configuration = new Configuration();

    configuration
        .setAcousticModelPath("resource:/edu/cmu/sphinx/models/en-us/en-us");
    configuration
        .setDictionaryPath("resource:/edu/cmu/sphinx/models/en-us/cmudict-en-us.dict");
    configuration
        .setLanguageModelPath("resource:/edu/cmu/sphinx/models/en-us/en-us.lm.bin");

    StreamSpeechRecognizer recognizer = new StreamSpeechRecognizer(
        configuration);
    InputStream stream = new FileInputStream(new File("C:\\Users\\Abhishek\\Desktop\\Abhishek Backup\\
    recognizer.startRecognition(stream);
    SpeechResult result;
    while ((result = recognizer.getResult()) != null) {
        System.out.format("Hypothesis: %s\n", result.getHypothesis());
    }
    recognizer.stopRecognition();
}
```

Audio output in previous step is given as input for transcription.

## VI. CONCLUSIONS

We proposed a way to generate subtitles for sound videos. A complete system including the three required modules introduced in section 1 could not be realized since the audio conversion needed more resources. VLC gave an appropriate solution but a custom component coded in Java is expected in further work so that portability and installation of the system is rendered uncomplicated. Nonetheless, the expected output for each phase has been reached. The audio extraction module provides a suitable audio format to be used by the speech recognition module. This one generates a list of recognized words and their corresponding time in the audio although the accuracy is not guaranteed. The former list is used by the subtitle generation module to create standard subtitle file readable by the most common media players available.

## VII. ACKNOWLEDGMENT

It gives us a great pleasure to express our deep sense of gratitude and indebtedness to our guide Prof.SmitaJhawale for his valuable support and encouraging mentality throughout our ongoing project. We are highly obliged to him for providing us this opportunity to carry out new ideas and work during our project period and helping us to gain the successful progress of our Project. Our special thanks is going to Head of the Department of Computer Engineering of our college, Dr.SwapnaBordeand to all of the faculties for extending their helping hand and encouraging us constantly to work hard in this project. We are highly grateful to the Honourable Principal of VCET,Dr. A. V. Bhonsalefor giving us this golden opportunity to be a part of this organization for this period.

**VIII. REFERENCES**

- [1] Boris Guenebaut, "AUTOMATIC SUBTITLE GENERATION FOR SOUND IN VIDEOS", Software Engineering Institution of Economics and IT Hogskolan Väst, 461 86 Trollhättan
- [2] Imtoodvd subtitle ripper. URL <http://www.imtoo.com/dvd-subtitle-ripper.html>.
- [3] Xilisoftdvd subtitle ripper. URL <http://www.xilisoft.com/dvd-subtitle-ripper.html>.
- [4] Subtitle editor, 2005-2009. URL <http://home.gna.org/subtitleeditor/>.
- [5] Subtitle workshop, 2006. URL <http://subtitle-workshop.en.softonic.com/>.
- [6] JAVE- Java Audio Video Encoder e Transcoder. URL <http://www.sauronsoftware.it/projects/jave>
- [7] CMU SPHINX URL <http://cmusphinx.sourceforge.net>