I'm here to propose an auto editing app

Hello, my name is Tan Zhong Ming

Just a reminder, It is supposed to be automatic

Let us do a quick recap

The first problem is remove stuttered speech manually

But this time i do a little bit modification

My second problem is improving the accuracy of subtitles. This can be done by using several techniques such as denoising, audio pre-process and deep speech. Audio pre-processing includes pre-emphasis, normalization, amplification and etc. But denoising is not necessary is audio pre-processing, it can be done by using AI to remove the background noise. This has been done by Nvidia, they publish a new feature for their CPU which is able to remove the background noise real time by using the latest RTX series GPU. Next, deep speech has a good potential to become the solution. I will explain why later.

So, the scope also must change, here is my latest scope.[Read slide]

Why we need this app. Earlier, we mentioned removing the stuttered speech by humans is slow and inefficient. The editor is required to remove and import one by one from the library. Apart from that, editors need to denoise, transcribe and process the audio manually to get better results from the subtitle generator. The proposed app has the potential to detect and remove the stuttered speech automatically in batch. Second it has better accuracy to auto generate subtitles. The processing time for the current video editing process is high, but the app is low. That means it is faster. Apart from that the human involvement for the proposed app is significantly reduced. Why do we need this app? From my study, 4 out of 6 of my friends are beginner level for video editing. and Based on my study, all of them hope there is an app like this with the subtitle generator.

Now, let me share what I found on the internet.

The first one of course is comparative study. Dataset is always the most important step in machine learning. No data no talk. There is a lot of different language dataset for stuttered speech such as UCLASS, Indian Regional Language, Polish Language, German Language and etc. The most commonly used in recent years is UCLASS. It is collected by University College London. Full name is University College London’s Archive of Stuttered Speech. They have 2 versions. First version only includes monolog, the second version includes monolog, reading and conversation. The age for the version 1 is between 5years 4month until 47years 0 month. version 2 is 7 years old 10 month until 20 years 7 month. Version 1 includes 120 male and 18 females and version 2 has more options.

Here is the summary of the several previous collections of research work on stuttered speech recognition. We can see in the early age, the database they used is non standard. The size of the sample and speaker are also non fixed. The features they used also are non standardized and required a lot of process on audio. But they still get pretty good results, the accuracy they get is range from 73.25% to 92%. They used ANN, HMM and SVM. This is the summary for research within 10 years. We can see almost everyone uses the UCLASS dataset as their data. This will provide a very good baseline for our project. Most of them use MFCC features and k-NN architecture. We can see the accuracy does improve from previous work. But the downside of MFCC is sensitive to noise.

This is the second literature review, the paper talking about the detection and analysis of stuttered speech. They use MFCC technique to extract the feature so no pre-emphasis is required. Then classify the stuttered speech by using SVM. They obtain 96.67% of accuracy with 10-12 coefficient.

Third paper that I read is speech recognition and correction of stuttered speech. The goal of this paper is to develop a system that is able to identify and correct the stuttered speech. Two algorithms are required to make it work. First is the algorithm to remove prolonged stuttered speech. This algorithm will obtain the maximum amplitude as input for the neural network. Then the neural network will output a proper threshold to remove the prolongation. Second is removing the repetition of stuttered speech. This algorithm will use Text-to-Speech technique to remove repetition of stuttered speech. They chunk the samples into 6 seconds segments for training. The processing time for this system is around 5- 8 seconds and they obtain 86% of accuracy.

Last one is deep speech 2. Also known as DS2. This research is done by Baidu

The goal of deep speech is to predict multiple languages by only using 1 model. This is because the current ASR Automatic Speech Recognition is overcomplicated. Because the current system divides the word into phonemes, syllables and more. This will make the current system face the problem if they need to implement another new language. So, they hope to develop only one engine applicable to all the languages. And they hope to solve other issues too, such as accented speech, noisy environment and more.

This is the architecture of the DS2. Total 11 layers. They have 3 convolution layers, 7 recurrent layers and 1 fully connected layer. The fully connected layer is responsible to calculate the probability distribution and provide output to the CTC layer.

The data they used for English in the experiment is WSJ, Switchboard, Fisher, LibriSpeech, and Baidu. Baidu dataset only for internal usage. So they have a total eleven thousand nine hundred forty hours of speech data. WSJ, Switchboard, Fisher you can get from Linguistic Data Consortium. LibriSpeech is the only free dataset in here. They use 95minutes data per epoch for English and 25 minute per epoch for mandarin. For data augmentation, they added noise to 40% of the dataset.

Next is the most interesting thing. The read speech DS2 outperforms humans. We can see some of the Word Error Rate (WER) is better than humans. But accented speech is close to human performance. This can be improved by adding more data. Unfortunately noisy speech is worse than humans. This is because they use synthetic voices rather than real environment recording.

The methodology that I choose is scrum. Why? Because it is fast and supports incremental deliverable. Next is user story makes me understand the user requirement. Product backlog can ensure the quality of the outcome.

Software, this is important because I faced a problem. This problem is not caused by human error. It is a compatibility issue. The story is when I need to read the audio by using librosa library. It is a python distribution that allows us to access the file in the directory. “pip” and “npm” is famous and professional tools. But after I installed librosa library. I cannot compile my python code. I reinstall over and over again, check through all the github stack overflow changing the version. But still didn’t solve the issues. The moment I wanted to reinstall the whole environment, I saw many articles mentioning the issues when installing the librosa by using pip. So they have to manually install the librosa, download, unzip, and put it to the folder. So I know it is the distribution software problem. Then I download anaconda3, anaconda 3 also is a python distribution. Ahha, then I solve the issues, my code doesn't have any problem, librosa also has no problem. The problem is the python distribution. So I will use anaconda 3 as my development tools. The minimum requirement to run tensorflow in python 3.5 or above. And to support tensorflow lite, you need at least support android 6.0 or API23 above.

Here is the new system workflow. It didn't change too much, just adding a new module. I insert a deep speech module here to generate subtitles for us. Hope it can flip this industry.

This is the activity diagram for traditional methods. It is very similar to the previous video.

Here is the proposed method. It is also very similar to the previous design. But I added an auto subtitle generator by using a deep speech module over here.

This is the use case diagram. Users are able to record, edit video and edit the subtitle. The special about this app is this app has a feature to remove the stuttered speech and auto generate subtitles for us. These features will be included in edit video and edit subtitle.

This is the activity diagram for editing the video. [Explain]

So let us talk about the app design. The app will include three main screens which are home screen, edit video screen and edit subtitle screen. The feature mentioned in above will present in “one button to solve all” way. First is the home screen. The user is able to record directly from the app or choose to import the video for future process. This is the most efficient and fastest way for users.

Here is the extent for the first screen when the user chooses to press import, in this screen user is able to view the gallery and import the video from the gallery.

Third is edit videos. This screen will take most of the time for the user. Users are able to play or pause the video here. The feature will represent indirectly on the app. because this project is more to automate the process. The main goal of this project is to remove the stuttered speech automatically. so the video will be cut into segments and removed automatically, or the user can choose to remove the segment manually.

Next is editing the subtitle. The subtitle will be auto generated and present on this screen. The user is able to adjust the start and end time, edit the subtitle and pause the video. Subtitles will be auto generated on this screen.

Thank you very much.