CS505 Project Proposal & Milestones

Team members
Macy So, Lazaro Solorzano, Jared Chou

GitHub

Problem Statement
Closed captioning displays the audio portion of a television program as text on the TV screen, providing a critical link to news, entertainment and information for individuals who are deaf or hard-of-hearing. The importance of this project is to enhance the educational learning experience for students and others with disabilities. There should be no restrictions for simply trying to watch a video or anything related to the sorts. With subtitles/closed captions it will help eliminate this issue by being able to provide it to services that don't already provide their our closed captions. This relates to work that we have done in the class because we will be developing an AI to generate subtitles for audio, developed using natural language processing techniques such as feature engineering, Mel Frequency Cepstral Coefficient, Discrete Cosine Transform and convolutional neural networks.

List of References (2-3 blogs, at least one research paper)
Blogs:

- Netflix Automated Subtitles[1]
- Live Transcribing with Python[2]
- Automatic Subtitle Synchronization through Machine Learning[3]
- How to Perform Real-Time Speech Recognition with Python[4]
- How to Create Subtitles for any Video with Python[5]
- Adding closed captions and subtitles[6]
- Automated Audio Captioning[7]
- Github Challenges Example[8]
- Wave2Vec[9]
- Speech Recognition using Transformers in Python [10]

---

[1] https://ottverse.com/netflix-automated-subtitling-using-ai-nlp/
[2] https://towardsdatascience.com/how-to-build-a-real-time-transcription-app-in-python-7939c7b02614
[3] https://medium.com/@asabater/automatic-subtitle-synchronization-e188a9275617
[4] https://towardsdatascience.com/real-time-speech-recognition-python-assemblyai-13d35eeed226
[5] https://picovoice.ai/blog/how-to-create-subtitles-for-any-video-with-python/
[6] https://cloud.google.com/transcoder/docs/how-to/captions-and-subtitles
[7] https://dcase.community/challenge2021/task-automatic-audio-captioning
[8] https://github.com/paniquex/Automated_Audio_Captioning_DCASE2020
[9] https://ai.facebook.com/blog/wav2vec-20-learning-the-structure-of-speech-from-raw-audio/
[10] https://www.thepythoncode.com/article/speech-recognition-using-huggingface-transformers-in-python
[11] https://arxiv.org/pdf/1904.09740.pdf

Research Papers:

- NLP Driven Ensemble Based Automatic Subtitle Generation and Semantic Video Summarization Technique [11]

Data Sources and plan for Data Wrangling (Method) (Jared)

- https://commonvoice.mozilla.org/en/datasets
- http://www.openslr.org/12

Models/Algorithms/Platforms to use (Laz & Macy)

Below, we have listed some of the technologies that we plan to use in our project. We plan on utilizing pre-built models and utilizing them to streamline our project. We will use OpenSubtitles as the audio subtitle database that we are pulling from in order to further build on pre-trained models utilizing methods highlighted in several of the different blogs we have as resources.

- Wave2Vec
- streamlit: the web framework that we'll use to house all the input and output widgets
- websocket: allow the app to interact with the AssemblyAI API
- asyncio: allow the app to perform all of the various audio input and output in a concurrent manner
- base64: allow the app to encode and decode the audio signal before it is sent to the AssemblyAI API
- json: reads in the audio output (i.e. the transcribed text) generated by the AssemblyAI API
- pyaudio: performs processing of the audio input via the Port audio library
- os and pathlib: used for navigating through the various folders of the project and for performing file processing and handling

Model Accuracy Measure

The metric that we will use for our project will be transcription accuracy based on the partitioned test data set. Our baseline aim will be for at least 50% transcription accuracy on testing data for success, looking to push for much higher accuracy if we have time to do so.

Project Plan:

The model will be run on our own computers due to access to good GPUs with Cuda acceleration, and later moved to SCC if needed.

Milestones

- ☑ ~~Find training dataset (Jared)~~ ~~**due Nov 22nd**~~
- ☑ ~~Find research papers (for additional support) (Macy)~~ ~~**due Nov 22nd**~~
- ☑ ~~Find Blogs (Macy)(Laz)~~ ~~**due Nov 22nd**~~
- ☑ ~~Write project proposal report (all)~~ ~~**due Dec 1st**~~
- ☐ Train model **due Dec 6th**

- [ ] Preprocessing of the data (Week of 11/28 - 12/03)
- [ ] Create initial model

- [ ] Write report
  - [ ] Abstract
  - [ ] Background
  - [ ] Method
  - [ ] Result
  - [ ] Analysis
  - [ ] Future direction
- [ ] Submit project **due Dec 8th**

CS 505 Website
https://www.cs.bu.edu/fac/snyder/cs505/final_assignments.html

Group Meet Time
11/30 from 7-10, location(Mugar 203)

Resources
Link for beeing able to download videos for free from youtube

https://www.geeksforgeeks.org/pytube-python-library-download-youtube-videos/

A python package for music and audio analysis

https://librosa.org/doc/latest/index.html

Code for reference
https://github.com/agermanidis/autosub/tree/master/autosub

Speech Recognition using Transformers in Python
https://www.thepythoncode.com/article/speech-recognition-using-huggingface-transformers-in-python

How to Convert Speech to Text in Python
https://www.thepythoncode.com/article/using-speech-recognition-to-convert-speech-to-text-python

https://commonvoice.mozilla.org/en/datasets (dataset for English audio to text)

Base idea
Use wave