

Capstone Project 2
Project Proposal
5/26/2020

Title:

Audio Auditor: A Speech to Translation Pipeline using LSTM and seq2seq

The Problem:

Most platforms that allow users to express themselves and share their own content have safeguards in place to monitor for abusive behaviors. For example, they may use machine learning to detect written hate speech or offensive images. However, there are two issues that make content moderation more difficult for these platforms: non-English language content, as well as audio, video, and streaming content. Under these conditions, the platform often has to rely on a great number of human content moderators. This slows the detection process, and the content then has the potential to harm many more people before it is removed.

This project tackles both the issues of non-English and audio content by building a pipeline to translate foreign language audio input into English text.

The Client:

This application is relevant to any social media platform with a global audience that allows users to post audio or video content. In particular, it is useful to companies that employ thousands of human content moderators, such as Facebook, YouTube, and Twitter, because it can save the moderators hours of having to sit through a video to determine if the content is allowable. It is also useful for other video-centric platforms with potentials for abuse, such as TikTok and Twitch. And although content moderation is an obvious implementation of a speech to translation application, there is also promise for use outside of the abuse space, such as achieving wider audiences for foreign language films on video streaming services like Netflix by offering automatic subtitles in a target language, or allowing podcasters to publish translated transcripts of their talks.

The Data:

The data for this project will be a parallel Chinese-English corpus. This means that it will consist of English natural language text data paired sentence-by-sentence with the Chinese translation for each sentence. Open source parallel English-Chinese corpora on various topics can be found here: [Ted Talks](#), [News](#), [UN Resolutions](#), [General Text](#), [Microblog Posts](#).¹

¹ The sample English translations of the Weibo microblog posts do not appear to be very good.

The Approach:

This project consists of two main parts: a speech to text part to render the audio of the input language into written text, then a translation part to translate the input text to the target language.

The first part is rather straightforward and can be done using the Python [SpeechRecognition](#) library. If used with the Google Cloud Speech API, it supports speech recognition in several languages, including Chinese.

The second part involves processing the language data and creating word embeddings for both languages to map them to each other. First, both languages will be preprocessed and tokenized. For English, there are a number of libraries that can do this. For Chinese preprocessing, we can use SpaCy, which has a Chinese language model, [SnowNLP](#), or the [Stanford NLP](#) Chinese Segmenter. Then the input language is encoded, and the encoding is decoded into the output language. This can be done using seq2seq for encoding word embeddings for both languages, or by using pre-trained, multilingual aligned vectors from Google [USE](#) or Facebook [MUSE](#). Training of the neural network can be enabled through TensorFlow or Keras. Finally, depending on the dataset size and training time, a GPU may be necessary, and this can be tackled through accessing a GPU instance through Google Cloud or AWS.

The Deliverables:

The final version of this project will include the complete code and a paper explaining the motivations and processes. In addition, I will include a slide deck to use for presentations.

Bibliography:

<https://paperswithcode.com/paper/neural-machine-translation>

<https://paperswithcode.com/paper/learning-phrase-representations-using-rnn>

<https://paperswithcode.com/paper/classical-structured-prediction-losses-for>

<https://paperswithcode.com/paper/scaling-neural-machine-translation>

<https://paperswithcode.com/paper/tensor2tensor-for-neural-machine-translation>

<https://paperswithcode.com/paper/phrase-based-neural-unsupervised-machine>

<https://paperswithcode.com/paper/bart-denoising-sequence-to-sequence-pre>

<https://paperswithcode.com/paper/attention-is-all-you-need>

<https://paperswithcode.com/paper/achieving-human-parity-on-automatic-chinese>

<https://runder.io/cross-lingual-embeddings/>

<https://pdfs.semanticscholar.org/86b9/4a1e3522f7a09dba6aad7659173e8dd2ab78.pdf>

<https://www.aclweb.org/anthology/N16-1083.pdf>

<https://arxiv.org/pdf/1705.03127.pdf>

Similar projects:

<https://towardsdatascience.com/neural-machine-translation-with-python-c2f0a34f7dd>

<https://medium.com/@dev.elect.iitd/neural-machine-translation-using-word-level-seq2seq-model-47538cba8cd7>

<https://towardsdatascience.com/implementing-neural-machine-translation-using-keras-8312e4844eb8>

[https://github.com/udacity/aind2-nlp-capstone/blob/master/machine translation.ipynb](https://github.com/udacity/aind2-nlp-capstone/blob/master/machine_translation.ipynb)

<https://towardsdatascience.com/word-level-english-to-marathi-neural-machine-translation-using-seq2seq-encoder-decoder-lstm-model-1a913f2dc4a7>

<https://indico.io/blog/sequence-modeling-neuralnets-part1/>