COMP 576 Introduction to Deep Learning

Final Project Proposal

Isolated Word Recognition & Profanity Censor

Group Members:

Kyle Manning (ksm9)
Ben Harris (bbh3)

Introduction

Since its formation in 1934, the Federal Communication Commission has been actively censoring American media for “obscene, indecent, and profane” actions. Radio and television stations use ineffective and outdated methods of keeping their audio content clean; in most cases humans manually bleep on a 10 to 15 second delay (as is the case for Rice’s own KTRU).

For our project, we would like to explore using neural networks to isolate and bleep profanity. A perfect product would run on live audio and quickly output bleeped, safe-for-air content. To stay within the realm of possibility given our time frame and resources, we aim to build an isolated word recognizer and use cutting edge keyword spotting methods to isolate timestamp a specific word in a sentence.

Context

In the past 50 years, automated speech recognition has progressed in leaps and bounds. Traditionally recognizers have been built with Hidden Markov Models, although in the past few years, neural networks have finally surpassed the HMM and probabilistic analysis.

Given that the task at hand is to filter out a few select words, an artificial neural network is the best approach. An HMM would require every word to be labelled and is computationally expensive. Using a neural network to recognize and timestamp a certain instance of a word is the most robust, flexible, and simple approach to the problem.

Goals

We first would like to build an isolated word recognizer that will correctly classify the curse words at above 80% accuracy. This should not be an extremely difficult task and is intended to introduce us to the problem of speech recognition.

Next, we would like to modify existing architectures for keyword spotting to timestamp the use of curse words in dialogue; the timestamp will allow an easy bleep replacement. We are not sure what accuracy we will be able to get with this.

We will also explore different implementation techniques that might enable the network to censor a live stream of audio with only a slight delay till broadcast.

Since most datasets refuse to incorporate profanity, the words we hope to identify and bleep are “cat”, “dog”, “taste”, “Ben”, and “Kyle.”

Data

We will use two datasets in this project. The classification words are “cat”, “dog”, “taste”, “Ben”, and “Kyle.” Both datasets will be built from Mozilla’s Common Voice Project.

The first dataset will isolate the audio of just the words from the sentences. To do this, we will use a forced aligner -- such as Gentle -- to timestamp the words. Forced aligners types of programs use the Viterbi algorithm for recognition and thus require enough words to do NLP (making them unsuited for our project proposal even if they do very similar work).

The second dataset will keep the mp3s in sentence format. We will try to have a mixed number of sentences with a keyword in it and ones without. Timestamping will once again be done with Gentle. We will have to preprocess this data to make it an even size.

Model and Methods

For simple isolated word recognition, we will use a variety of simple RNN models. We plan to test them and will record the results from each.

For keyword spotting, there have been a wide variety of models used, from HMM’s and simple RNN’s to more complex CNN-RNN hybrid models. Very recently, a model using DenseNet and BiLSTM was used to achieve never-before-seen levels of accuracy. We propose using this model, pretrained, and simply fine-tuning to fit our task. If the desired results are not achieved, we may look into other techniques.

Feasibility

One possible issue we might run into is that of training. The DenseNet-BiLSTM hybrid model is extremely large; at around 223k parameters, in one of the architectures proposed by Zeng et. al. As mentioned above, to combat this we’ll use a pretrained model and simply fine-tune.

We also are hampered in our ability to accomplish our desired task by Mozilla’s anti-profanity policy. With more time and resources, we would like to explore revamping the dataset by merging the sentences with an instance of a curse word. Given the

logistical challenges of building the cursing dataset for that task, we have decided to focus on identifying words already included in the Common Voice data.

References

A time-delay neural network architecture for isolated word recognition -- Kevin Lang, Alex Waibel, Jeffrey Hinton

Temporal Feedback Convolutional Recurrent Neural Networks for Keyword Spotting -- Taejun Kim, Juhan Nam

Effective Combination of DenseNet and BiLSTM for Keyword Spotting -- Mengjun Zeng, Nanfeng Xiao