Advancing Speech Recognition With No Speech Or With Noisy Speech

Abstract

In this paper we demonstrate end to end continuous speech recognition (CSR) using electroencephalography (EEG) signals with no speech signal as input. An attention model based automatic speech recognition (ASR) and connectionist temporal classification (CTC) based ASR systems were implemented for performing recognition. We further demonstrate CSR for noisy speech by fusing with EEG features.

Introduction

Electroencephalography (EEG) is a non invasive way of measuring electrical activity of human brain. In BIBREF0 we demonstrated deep learning based automatic speech recognition (ASR) using EEG signals for a limited English vocabulary of four words and five vowels. In this paper we extend our work for a much larger English vocabulary and we use state-of-art end-to-end continuous speech recognition models to perform recognition. In our prior work we predicted isolated words and vowels.

ASR systems forms the front end or back end in many cutting edge voice activated technologies like Amazon Alexa, Apple Siri, Windows Cortana, Samsung Bixby etc. Unfortunately these systems are trained to recognize text only from acoustic features. This limits technology accessibility to people with speaking disabilities and disorders. The research work presented in this paper tries to address this issue by investigating speech recognition using only EEG signals with no acoustic input and also by combining EEG features along with traditional acoustic features to perform recognition. We believe the former will help with speech restoration for people who can not speak at all and the latter will help people who are having speaking disabilities like broken or discontinued speech etc to use voice activated technologies with better user experience there by helping in improving technology accessibility.

ASR performance is degraded in presence of noisy speech and in real life situations most of the speech is noisy. Inspired from the unique robustness to environmental artifacts exhibited by the human auditory cortex BIBREF1 , BIBREF2 we used very noisy speech data for this work and demonstrated lower word error rate (WER) for smaller corpus using EEG features, concatenation of EEG features and acoustic features.

In BIBREF3 authors decode imagined speech from EEG using synthetic EEG data and connectionist temporal classification (CTC) network but in our work we use real EEG data, use EEG data recorded along with acoustics. In BIBREF4 authors perform envisioned speech recognition using random forest classifier but in our case we use end to end state of art models and perform recognition for noisy speech. In BIBREF5 authors demonstrate speech recognition using electrocorticography (ECoG) signals, which are invasive in nature but in our work we use non invasive EEG signals.

This work is mainly motivated by the results explained in BIBREF0 , BIBREF6 , BIBREF7 , BIBREF3 . In BIBREF6 the authors used classification approach for identifying phonological categories in imagined and silent speech but in our work we used continuous speech recognition state of art models and our models were predicting words, characters at each time step. Similarly in BIBREF7 neural network based classification approach was used for predicting phonemes.

Major contribution of this paper is the demonstration of end to end continuous noisy speech recognition using only EEG features and this paper further validates the concepts introduced in BIBREF0 for a much larger English corpus.

Automatic Speech Recognition System Models

An end-to-end ASR model maps input feature vectors to an output sequence of vectors of posterior probabilities of tokens without using separate acoustic model, pronunciation model and language model. In this work we implemented two different types of state of art end to end ASR models used for the task of continuous speech recognition and the input feature vectors can be EEG features or concatenation of acoustic and EEG features. We used Google's tensorflow and keras deep learning libraries for building our ASR models.

Connectionist Temporal Classification (CTC)

The main ideas behind CTC based ASR were first introduced in the following papers BIBREF8 , BIBREF9 . In our work we used a single layer gated recurrent unit (GRU) BIBREF10 with 128 hidden units as encoder for the CTC network. The decoder consists of a combination of a dense layer (fully connected layer) and a softmax activation. Output at every time step of the GRU layer is fed into the decoder network. The number of time steps of the GRU encoder is equal to product of the sampling frequency of the input features and the length of the input sequence. Since different speakers have different rate of speech, we used dynamic recurrent neural network (RNN) cell. There is no fixed value for time steps of the encoder.

Usually the number of time steps of the encoder (T) is greater than the length of output tokens for a continuous speech recognition problem. A RNN based CTC network tries to make length of output tokens equal to T by allowing the repetition of output prediction unit tokens and by introducing a special token called blank token BIBREF8 across all the frames. We used CTC loss function with adam optimizer BIBREF11 and during inference time we used CTC beam search decoder.

We now explain the loss function used in our CTC model. Consider training data set INLINEFORM0 with training examples INLINEFORM1 and the corresponding label set INLINEFORM2 with target vectors

INLINEFORM3 . Consider any training example, label pair (INLINEFORM4 , INLINEFORM5). Let the number of time steps of the RNN encoder for (INLINEFORM6 , INLINEFORM7) is INLINEFORM8 . In case of character based CTC model, the RNN predicts a character at every time step. Whereas in word based CTC model, the RNN predicts a word at every time step. For the sake of simplicity, let us assume that length of target vector INLINEFORM9 is equal to INLINEFORM10 . Let the probability vector output by the RNN at each time step INLINEFORM11 be INLINEFORM12 and let INLINEFORM13 value of INLINEFORM14 be denoted by INLINEFORM15 . The probability that model outputs INLINEFORM16 on input INLINEFORM17 is given by INLINEFORM18 . During the training phase, we would like to maximize the conditional probability INLINEFORM19 , and thereby define the loss function as INLINEFORM20 .

In case when the length of INLINEFORM0 is less than INLINEFORM1 , we extend the target vector INLINEFORM2 by repeating a few of its values and by introducing blank token (INLINEFORM3) to create a target vector of length INLINEFORM4 . Let the possible extensions of INLINEFORM5 be denoted by INLINEFORM6 . For example, when INLINEFORM7 and INLINEFORM8 , the possible extensions are INLINEFORM9 , INLINEFORM10 , INLINEFORM11 , INLINEFORM12 , INLINEFORM13 , INLINEFORM14 and INLINEFORM15 . We then define INLINEFORM16 as INLINEFORM17 .

In our work we used character based CTC ASR model. CTC assumes the conditional independence constraint that output predictions are independent given the entire input sequence.

RNN Encoder-Decoder or Attention model

RNN encoder - decoder ASR model consists of a RNN encoder and a RNN decoder with attention mechanism BIBREF12 , BIBREF13 , BIBREF14 . The number of time steps of the encoder is equal to the product of sampling frequency of the input features and the length of input sequence. There is no fixed value for time steps in our case. We used dynamic RNN cell. We used a single layer GRU with 128 hidden units for both encoder and decoder. A dense layer followed by softmax activation is used after the decoder GRU to get the prediction probabilities. Dense layer performs an affine transformation. The number of time steps of the decoder GRU is same as the number of words present in the sentence for a given training example. Training objective is to maximize the log probability of the ordered conditionals, ie: INLINEFORM0 , where X is input feature vector, INLINEFORM1 's are the labels for the ordered words present in that training example and INLINEFORM2 is the length of the output label sentence for that example. Cross entropy was used as the loss function with adam as the optimizer. We used teacher forcing algorithm BIBREF15 to train the model. During inference time we used beam search decoder.

We now explain the attention mechanism used in our attention model. Consider any training example, label pair (INLINEFORM0 , INLINEFORM1). Let the number of times steps of encoder GRU for that example be INLINEFORM2 . The GRU encoder will transform the input features (INLINEFORM3) into hidden output feature vectors (INLINEFORM4). Let INLINEFORM5 word label in INLINEFORM6 (sentence) be INLINEFORM7 , then to predict INLINEFORM8 at decoder time step INLINEFORM9 , context vector INLINEFORM10 is computed and fed into the decoder GRU. INLINEFORM11 is computed as INLINEFORM12 , where INLINEFORM13 is the attention weight vector satisfying the property INLINEFORM14 .

INLINEFORM0 can be intuitively seen as a measure of how much attention INLINEFORM1 must pay to INLINEFORM2 , INLINEFORM3 . INLINEFORM4 is mathematically defined as INLINEFORM5 , where INLINEFORM6 is hidden state of the decoder GRU at time step INLINEFORM7 .

The way of computing value for INLINEFORM0 depends on the type of attention used. In this work, we used bahdanau's additive style attention BIBREF13 , which defines INLINEFORM1 as INLINEFORM2) where INLINEFORM3 and INLINEFORM4 are learnable parameters during training of the model.

Design of Experiments for building the database

We built two types of simultaneous speech EEG recording databases for this work. For database A five female and five male subjects took part in the experiment. For database B five male and three female subjects took part in the experiment. Except two subjects, rest all were native English speakers for both the databases. All subjects were UT Austin undergraduate,graduate students in their early twenties.

For data set A, the 10 subjects were asked to speak the first 30 sentences from the USC-TIMIT database BIBREF16 and their simultaneous speech and EEG signals were recorded. This data was recorded in presence of background noise of 40 dB (noise generated by room air conditioner fan). We then asked each subject to repeat the same experiment two more times, thus we had 30 speech EEG recording examples for each sentence.

For data set B, the 8 subjects were asked to repeat the same previous experiment but this time we used background music played from our lab computer to generate a background noise of 65 dB. Here we had 24 speech EEG recording examples for each sentence.

We used Brain Vision EEG recording hardware. Our EEG cap had 32 wet EEG electrodes including one electrode as ground as shown in Figure 1. We used EEGLab BIBREF17 to obtain the EEG sensor location mapping. It is based on standard 10-20 EEG sensor placement method for 32 electrodes.

For data set A, we used data from first 8 subjects for training the model, remaining two subjects data for validation and test set respectively.

For data set B, we used data from first 6 subjects for training the model, remaining two subjects data for validation and test set respectively.

EEG and Speech feature extraction details

EEG signals were sampled at 1000Hz and a fourth order IIR band pass filter with cut off frequencies 0.1Hz and 70Hz was applied. A notch filter with cut off frequency 60 Hz was used to remove the power line noise. EEGlab's BIBREF17 Independent component analysis (ICA) toolbox was used to remove other biological signal artifacts like electrocardiography (ECG), electromyography (EMG), electrooculography (EOG) etc from the EEG signals. We extracted five statistical features for EEG, namely root mean square, zero crossing rate,moving window average,kurtosis and power spectral entropy BIBREF0 . So in total we extracted 31(channels) X 5 or 155 features for EEG signals.The EEG features were extracted at a sampling frequency of 100Hz for each EEG channel.

We used spectral entropy because it captures the spectral (frequency domain) and signal complexity information of EEG. It is also a widely used feature in EEG signal analysis BIBREF18 . Similarly zero crossing rate was chosen as it is a commonly used feature both for speech recognition and bio signal analysis. Remaining features were chosen to capture time domain statistical information. We performed lot of experiments to identify this set of features. Initially we used only spectral entropy and zero crossing rate but we noticed that the performance of the ASR system significantly went up by 20 % when we added the remaining additional features.

The recorded speech signal was sampled at 16KHz frequency. We extracted Mel-frequency cepstrum (MFCC) as features for speech signal. We first extracted MFCC 13 features and then computed first and second order differentials (delta and delta-delta) thus having total MFCC 39 features. The MFCC features were also sampled at 100Hz same as the sampling frequency of EEG features to avoid seq2seq problem.

EEG Feature Dimension Reduction Algorithm Details

After extracting EEG and acoustic features as explained in the previous section, we used non linear methods to do feature dimension reduction in order to obtain set of EEG features which are better representation of acoustic features. We reduced the 155 EEG features to a dimension of 30 by applying Kernel Principle Component Analysis (KPCA) BIBREF19 .We plotted cumulative explained variance versus number of components to identify the right feature dimension as shown in Figure 2. We used KPCA with polynomial kernel of degree 3 BIBREF0 . We further computed delta, delta and delta of those 30 EEG features, thus the final feature dimension of EEG was 90 (30 times 3) for both the data sets.

When we used the EEG features for ASR without dimension reduction, the ASR performance went down by 40 %. The non linear dimension reduction of EEG features significantly improved the performance of ASR.

Results

The attention model was predicting a word and CTC model was predicting a character at every time step, hence we used word error rate (WER) as performance metric to evaluate attention model and character error rate (CER) for CTC model for different feature sets as shown below.

Table i@ and ii@ shows the test time results for attention model for both the data sets when trained using EEG features and concatenation of EEG, acoustic features respectively. As seen from the results the attention model gave lower WER when trained and tested on smaller number of sentences. As the vocabulary size increase, the WER also went up. We believe for the attention model to achieve lower WER for larger vocabulary size more number of training examples or larger training data set is required as large number of weights need to be adapted. Figure 3 shows the training loss convergence of our attention model.

Table iv@ and v@ shows the results obtained using CTC model. The error rates for CTC model also went up with the increase in vocabulary size for both the data sets. However the CTC model was trained for 500 epochs compared to 100 epochs for attention model to observe loss convergence and batch size was set to one for CTC model. Thus CTC model training was lot more time consuming than attention model.

In BIBREF0 we have demonstrated that EEG sensors T7 and T8 features contributed most towards ASR performance. Table vi@ shows the CTC model test time results when we trained the model using EEG features from only T7 and T8 sensors on the most noisy data set B. We observed that as vocabulary size increase, error rates were slightly lower than the error rates from Table iv@ where we used EEG features from all 31 sensors with dimension reduction. Table iii@ shows the results for attention model when trained with EEG features from sensors T7 and T8 only on data set B. We observed that error rates were higher in this case compared to the error rates reported in table ii@.

Figures 4 shows the visualization of the attention weights when the attention model was trained and tested using only EEG features for Data set B. The plots shows the EEG feature importance (attention) distribution across time steps for predicting first sentence and it indicates that attention model was not able to attend properly to EEG features, which might be another reason for giving higher WER.

Conclusion and Future work

In this paper we demonstrated the feasibility of using EEG features, concatenation of EEG and acoustic features for performing noisy continuous speech recognition. To our best knowledge this is the first time a continuous noisy speech recognition is demonstrated using only EEG features.

For both attention and CTC model we observed that as the vocabulary size increase, concatenating acoustic features with EEG features will help in reducing the test time error rates.

We further plan to publish our speech EEG data base used in this work to help advancement of research in this area.

For future work, we plan to build a much larger speech EEG data base and also perform experiments with data collected from subjects with speaking disabilities.

We will also investigate whether it is possible to improve the attention model results by tuning hyper parameters to improve the model's ability to condition on the input,improve CTC model results by training with more number of examples and by using external language model during inference time.

Acknowledgement

We would like to thank Kerry Loader and Rezwanul Kabir from Dell, Austin, TX for donating us the GPU to train the models used in this work.