Amharic Speech Recognition System Using Connectionist Temporal Classification(CTC)

Abstract

This class of applications starts with a clip of spoken audio in Amharic language and extracts the words that were spoken, as text.Out-of-vocabulary (OOV) words are the most challenging problem in automatic speech recognition (ASR), especially for morphologically rich languages. Most end-to-end speech recognition systems are performed at word and character levels of a language. And Amharic is a poorly resourced but morphologically rich language.

1, Introduction

end-to-end ASR has grown to be a popular alterna- tive to simplify the conventional ASR model building process. End-to-end ASR methods depend on paired acoustic and language data, and train the acoustic model with a single end-to-end ASR algorithm. As a result, the approach makes it feasible to construct ASR systems. The end-to-end ASR system directly transcribes an input sequence of acoustic features (F) to an output sequence of probabilities for tokens (p) such as phonemes and characters.

Various types of end-to-end architectures exist for ASR such as connectionist temporal classification (CTC). The CTC method is used to train recurrent neural networks (RNNs) without knowledge of the prior alignment be- tween input and output sequences of different lengths. The CTC model can also make a strong

assumption between labels, and the attention-based model trains a decoder depending on the previous labels.

2 Dataset and Methods

2.1 Dataset and Data Pre-Processing

2.1.1 Dataset

We used the Amharic reading speech database collected for speech recognition purposes in the conventional ASR approaches , and an additional 2 h reading speech con- taining 1000 sentences was used. These reading speech corpora were collected from dif- ferent sources to maintain variety, such as political, economic, sport, health news,and fiction.

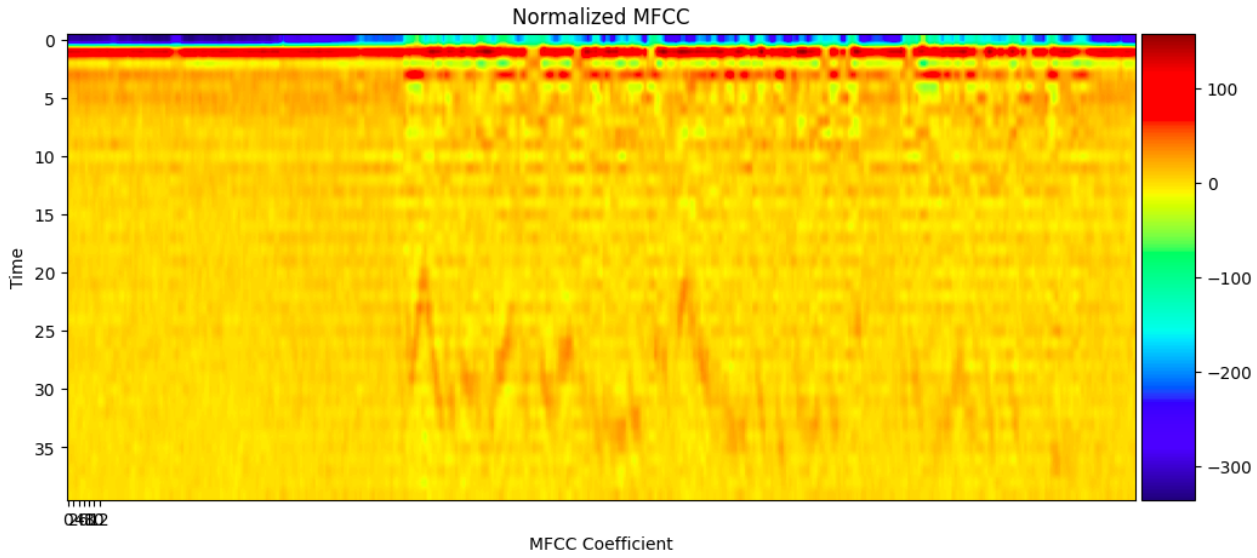
All contents of a corpus were set up using morphological concepts. Reading speech corpora of the corresponding texts were prepared using a 22500 Hz sampling frequency, 22.5 bit sample size and a stereo channel.

Data augmentation is a method of increasing training data in poorly resourced languages . It is used to address the scarcity of resources and to increase the performance of their ASR systems . It is also one of the most effective means of making commutative end-to-end automatic speech recognition (ASR) with a conventional hybrid approach in low-resource tasks .

2.2. Features

Key to any machine learning method is extracting features from data. Features represent data and serve as input to the learner. Speech recognition methods derive features from audio, such as Spectrogram or Mel Frequency Cepstrum (MFCC). Audio is split into small blocks, such as 10 milliseconds, and each block is broken into its constituent frequencies.

We chose to use MFCC over Spectrogram because it produces fewer features. Spectrograms can work even better than MFCC as it gives more features.



3 .Methods

3.1CTC Model

The CTC model is used to map speech input frames into corresponding output labels , which is used to optimize the prediction of a transcription sequence. When the length of the output labels is shorter than the length of the input speech frames, a CTC path is introduced to have an identical length as that of the input speech frames by adding a “blank” symbol as an additional label and allowing repetition of labels to map the label sequence into the CTC path. This forces the output and input sequences to have identical lengths

3.2 Experiment Parameter Setups and Results

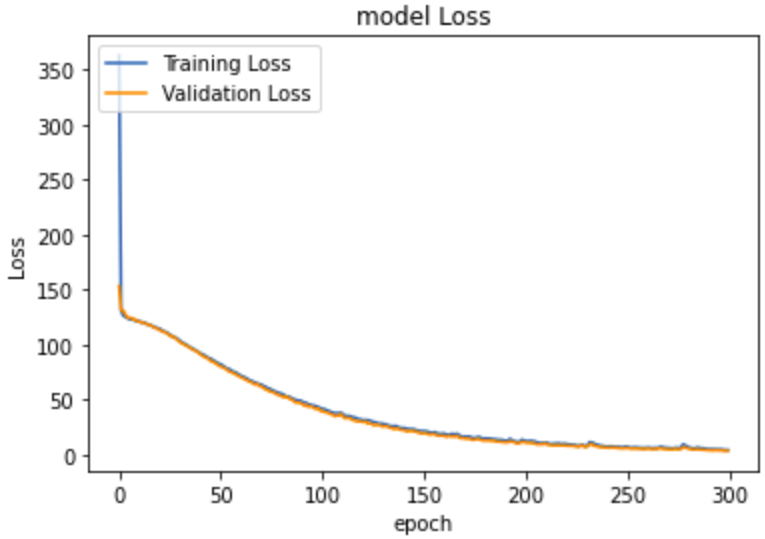
In our experiments, a computer with a GPU (GeForce GTX 1050 ti) and 16 GB memory were used to perform the speech recognition system’s training and testing phases.

The training was performed over 100 epochs using tensorflow modeling with a batch size of 2. In our experiments, a computer with a GPU (GeForce GTX 1050 ti)

and 16 GB memory were used to perform the speech recognition system's training and testing phases.

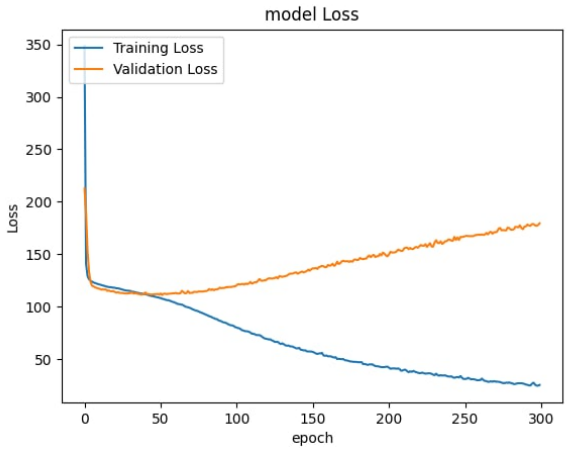
The training was performed over 100 epochs using tensorflow modeling with a batch size of 2. An AWS instance was also used for the training , with batch size of 32 and around 300 epoch for training the 724 audio files. 724 audio files were used because audio files with extreme durations were dropped for the training. The transcriptions were also cleaned and prepared because they contained typos.

After training the LSTM model, with 3 lstm layers containing 100 cells and batch normalization layers, we were able to achieve a validation loss of 3.5202, a training loss of 4.3062 and ler of 0.0107.



Other Experiments were conducted by adding bidirectional layers, which quickly overfit when trained with the small data we had.

After adding the augmented audio files which were around 7000, a deep RNN model was trained on them. Which had the same overfitting problems stated above.



3.3 Character-Based Baseline End-to-End Models

Our proposed CTC-attention end-to-end AASR was evaluated using characters.In our character-based experiment, word-based recurrent neural network language modeling (RNNLM) was used to investigate the recognition performance in various texts.

4. Discussion

We observed that the input-output alignment was appropriately learned. The input-output alignment sequences are shown from the beginning with almost a spectrogram representation of the utterance. When the training extends in different epochs, we observed the gap of alignments. This gap indicates that there were missing characters that can be analyzed in terms of deletion, insertion, and substitution during training.

5. Conclusions and Future Works

As future work, transformer-based end-to-end models will be used to obtain coverage to reduce the errors of the recognition system. A greater corpus size is also required in all end-to-end models; thus, collecting more data to increase the corpus size is a necessary task.

planning to investigate the application of the proposed method in hybrid ASR, machine translation, and speech translation.