Building an Automatic Speech Recognition(character level) Model using Swahili Language Audio Dataset

Author: Clemencia Siro

1 Introduction

In this project we are going to build an Automatic Speech Recognition Model at character level using the data we recorded in Swahili from the Lig-aikuma app, which is a native language for over 100 million people in sub-saharan Africa. Before the recording data was preprocessed to the format acceptable by the Lig-aikuma app. The recorded audio data is loaded onto the pre-trained Facebook CPC_Audio model and then the phone and character error rate of our model are computed.

2 Methodology

This section is dedicated for the loading of data and various preprocessing methods we did on our data into a format acceptable by the CPC model.

2.1 Data collection

The data used to train this model is part of the class project of recording audio of 2h in our native languages. The application used was Lig-aikuma which allowed us to record by elicitation of text. Our text data used for recording was scrubed from a pdf version of the Old Testament Bible more so from the book of Joshua. Lig-aikuma only accepts text in .txt format so the scrubbed data was converted into a text file. Considering sometimes the Bible comprises of long sentences and this would have affected our recording with the app, most of the sentences were split to a maximum length of 15 words each and preceeded with double hash tags as required by the lig-aikuma app.

2.2 Data Preprocessing

The audio data recorded with the Lig-aikuma app is in the format of a .wav file. Each recorded session was saved in a folder together with the linker file and json file. Before loading our data to the model a file with character text and indices mapping the characters was generated using the session file and linker files, a format required for training in the CPC_ audio Model.

The data was later split into train, validation and test splits. The train split comprised of 41 minutes of recording, the validation had 21 minutes while the test split had 1 hour of recording. The train split was used to train our model while the validation set was used to evaluate the model.

2.3 Problems Encoutered

During scrubing of text data to use for recording the first idea was to use news from Swahili BBC and Taifa Leo a kenyan newspaper. Butbecause of patent and copyright rights we shelved the idea and sort to use the Bible.

Some problems were encountered during recording as the lig-aikuma app could hang and restart at times which led to sudden breaking in the recording. This is also led to some text being missed while recording and also corrupted files. The broken audios and corrupted files were deleted during data preprocessing to avoid error during data loading.

3 Architecture

As an inspiration for this this project is the implementation of the CPC_Audio model by Facebook Research group for automatic spech recognition. A function to get the phone eror rate and character error rate was implemented and run on our data.

The model was first trained using the CPC loss with crossentropy as the classification loss. later it was fine-tuned using the ctc loss on unalligned text. Due to limitation of data, the same dataset was used in fine tuning as in prior training.

4 Results

This section is dedicated to the results obtained from our model.

4.1

We trained the model and fine tuned on unalligned phonemes using the same swahili dataset.A slight decrease in the loss was noticed. And the results are shown in the figure below for the validation dataset.

	WINDOWSIZE	Loss	Avg_loss	Avg_accuracy
CPC_Model	20483	cpc_loss	4.91794	0.0
PER	20483	CTCloss		0.974947
PER without allignment	20483	CTCloss	4.6308	
CER	20483	CTCloss	26.58459	0.96743

5 Conclusion

From this project I have learnt how to collect data and preprocessing it for use in building machine learning models. Also I have gained skills on how to build an ASR using the CPC Model.

Due to the constraint of time some of the additional steps were not done and this leave room for improvement of this model Future directions should include:

- Data augmentation for the fine tuning (you can use noise datasets like MUSAN)
- Multilingual pretraining (using CPC) with these datasets
- Add a Language model and compute WER
- Study the effect of language proximity

Adding a language modeling is a very pomising step as this will allow us to see the perfomance of our model in text transcription and compute the word error rate. Fine tuning on a different dataset might improve the overall perfomance of the model as our model was fine tuned on the same training dataset.