האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

Automatic Speech Recognition

Final Project For Introduction to Speech and Audio (67355)

Alon Ziv
Amit Sheffi
Amit Roth

Abstract

The goal of the project is to build the best ASR system we can using the tools we have learned in the course. There are many diverse ways to approach this problem: HMM, DTW, NN, etc. In this paper we will detail about the research and the experiments we have conducted towards developing the best ASR model we can given the limited training dataset defined for the project.

Data

We will train the model on the AN4 dataset. This dataset is pretty small and consists of people spelling out addresses, names, letters, etc. We will utilize the size of the vocabulary in our decoding techniques.

Our Approach

Text Representation

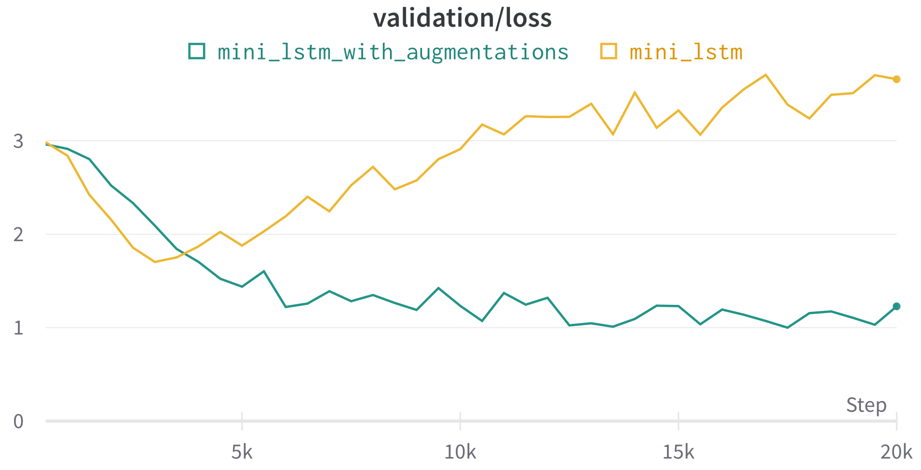
We represented text as per-character tokens. The character range we chose was [ascii_min, ascii_max] where ascii_min and ascii_max were calculated based on the values appeared in the training set. In addition a special token (token 0) was allocated for the ϵ null token required for CTC.

Audio Augmentation

Observing rapid overfitting caused by limited data, we introduced data augmentations to mitigate this issue and subsequently trained the models on the augmented dataset. Evaluating the models on both original and augmented datasets revealed significant and remarkable improvements.

We used three different audio augmentations in order to reduce overfit to the training data:

- Random shift of up to 100ms
- SpecAugment (https://arxiv.org/pdf/1904.08779.pdf)
- Sign flip - phase inversion w.p. 0.5

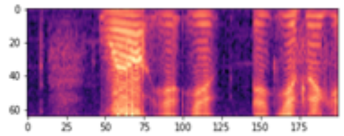


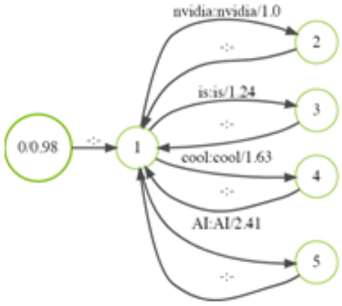
Model

We decided developing an End-to-End system constructed from an acoustic model and a CTC decoder. We train the acoustic model on the data with CTC-loss and later infer the sentence using the decoder.

The acoustic model gets the recording as a Mel spectrogram, where we use 128 Mel bands, and an FFT size of 400, and outputs for each time step a vector, which represents the probability for each token to appear. With this probability we infer the sentence using the decoder.

We started with a super naively one linear layer acoustic model and a greedy CTC decoder, and developed both acoustic model and decoder from this baseline. Our ASR system consists of two distinct components (acoustic model and decoder) operating independently, prompting us to focus on enhancing each of these components individually. We developed additional 4 acoustic models and 2 CTC decoders. During the model evaluation, we have the flexibility to test each component in conjunction with the other and analyze the results. The research process proved fascinating and drove us into developing different aspects of the model - including the acoustic model, the decoder, and even augmenting data in order to train the acoustic model better, finally leading into our final model.





Acoustic Models

We will list all of the acoustic models we used in order, starting from the easiest model and infer it with each iteration.

1. linear layer:

   The simplest model. 1 linear layer with size (n_mels, n_tokens) which is (128, 60) $(n - mels, n - tokens)$

2. MiniLSTM
   - Input embedding - linear layer
   - 2 layer LSTM with hidden dimension 128
   - Readout - linear layer
3. DeepSpeech Toy - Based on the architecture of DeepSpeech Toy:
   - 3 Conv layers
   - 1 RNN layer with hidden dim 256
   - 2 Linear layers
4. DeepSpeech Small - Based on the architecture of DeepSpeech Small:
   - 2 Conv layers
   - 2 RNN layers with hidden dim 512
   - 2 Linear layers
5. DeepSpeech Large - Based on the architecture of DeepSpeech large:
   - 3 Conv layers
   - 4 RNN layers with hidden dim 800
   - 2 linear layers

Decoders

1. Greedy CTC - selects the most probable output label at each time step, without considering the context.
2. Beam Search Lexicon - This method explores multiple hypotheses simultaneously by considering a set of potential output labels at each time step and selecting the most likely paths based on a lexicon of all words

available in the training data. The rationale of this model is that word that appeared in the data are more likely to reappear.

3. Beam Search LM - as the method mentioned above, but considering at the scoring mechanism also an external language model (KenLM librispeech-4-gram, which have trained on different English data). This method has a large potential, but we will see that because of the differences in the data, the outcomes are not outstanding.

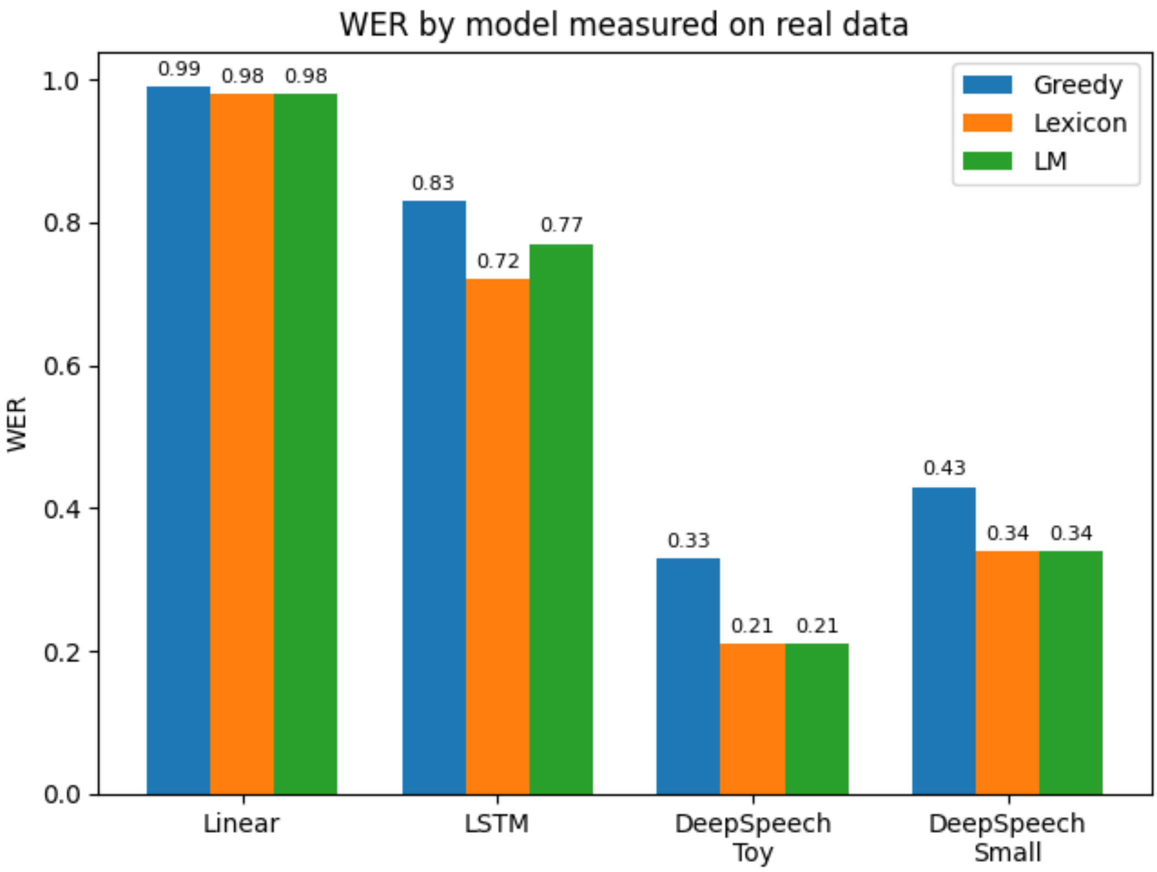
Benchmarks

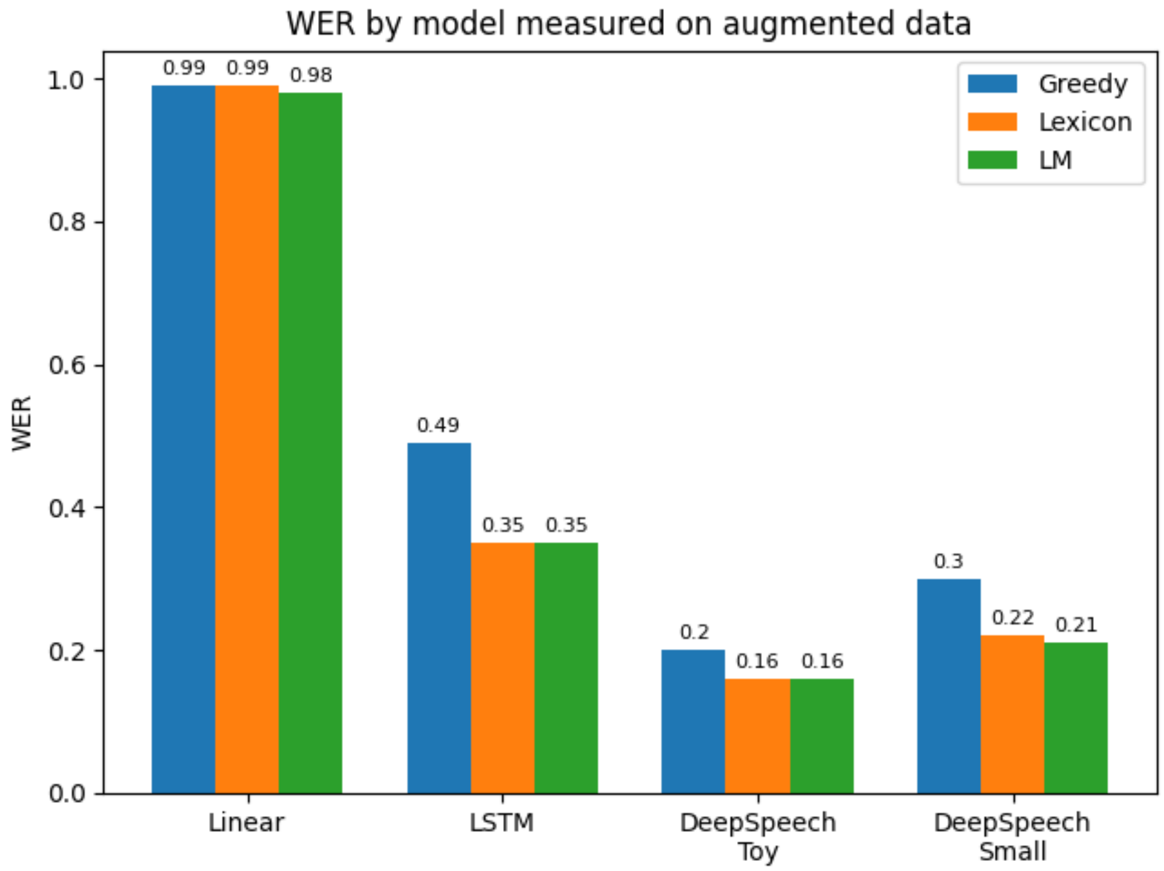
In the following the table, we detail the WER values for all model combinations, data augmentations on different datasets: Train, Test and Val. We didn't manage to train DeepSpeech Large correctly hence left empty.

Acoustic Model / Decoder			Greedy	Lexicon	LM
Name	Parameters	Augmented Data			
Linear	8K	NO	0.9878 0.9923 0.9916	0.9878 0.9930 0.9897	0.9842 0.9893 0.9832
		YES	0.9915 0.9940 0.9902	0.9905 0.9934 0.9863	0.9820 0.9919 0.984
LSTM	288K	NO	0.7144 0.8168 0.8288	0.5814 0.6981 0.7195	0.6302 0.7365 0.7699
		YES	0.16777 0.4262 0.4885	0.0962 0.3252 0.3475	0.1115 0.3254 0.3472
DeepSpeech Toy	4M	NO	0.0272 0.2911 0.3285	0.0043 0.2081 0.2054	0.0065 0.2 0.2132
		YES	0.0075 0.1869 0.2047	0.0086 0.1686 0.1625	0.0114 0.1862 0.1555

Acoustic Model / Decoder			Greedy	Lexicon	LM
Name	Parameters	Augmented Data			
DeepSpeech Small	10.8M	NO	0.0188 0.4130 0.4247	0.0154 0.2932 0.338	0.0179 0.2964 0.343
		YES	0.0410 0.2999 0.3001	0.01431 0.1968 0.2216	0.0159 0.1971 0.2136
DeepSpeech Big	57.8M	NO			
		YES			

The graphs that let you see the differences between the models. Evaluated on the validation data.





Results

- We can see that in most cases the language model does not improve our results and sometimes even make them worse. We explain this phenomenon due to the difference of our dataset from regular English. The dataset contains sequence of words without any grammatical relation hence the language model cannot infer the next word based on the previous words with good probability.
- Simple augmentations helped significantly in the training process due to the small size of the dataset
- Our Best Model is DeepSpeech Toy, trained on the augmented dataset and decoded using the lexicon method or using the language model.

Future Work

Theoretically, the best model should be DeepSpeech Large trained on augmented data and decoded using the language model. We have not achieved this because several reasons

1. **Language Model -** We used a language model from (librispeech-4-gram) that trained on common English, and not on our data which is not similar to common English. An improvement is to build a corpus from our data and to train a language model on it.
2. **Training -** The DeepSpeech Small and Large did not converged and remained with a relatively high loss. With further training and modification of parameters we should achieve better results.

Appendix

Git Repo - https://github.com/MajoRoth/ASR

KenLM Librispeech - https://pytorch.org/audio/stable/tutorials/asr_inference_with_ctc_decoder_tutorial.html#kenlm