Recep Oğuz Araz
Prof. Engin Erzin
ELEC390
26.01.2021

Report X: Testing Various CTC Decoding Systems

In order to increase the model performance, we investigate more complex decoding systems on the CTC output.

We fix the Neural Network(NN) with the lowest validation loss (10% of the training set, separate), and test the effects of the decoding systems on the unseen test set.

CTC Decoding with Character and Word Level Language Models

Several corpora have been formed for Language Modeling.

Corpus Forming

1) Wikipedia Dump

We segment, clean, and format a Turkish wiki dump consisting of *330k articles*.

After thoroughly processing, it is converted to *1.7M sentences*, *1.3M words* formed by only the letters of the Turkish alphabet, the space and the point.

2) Dataset Transcriptions

We also apply the same process to the Mozilla and METUbet transcriptions. There are 2k+5k sentences combined.

3) Merged Set

We combine the Wikipedia dump and the dataset transcriptions in a single set.

Using the Corpora above, we train character level ngrams, word level bigrams and form a lexicon.

Character Level LM

We train unigram, bigram, trigram and 4gram LMs with no smoothing, to get an approximation.

The LMs are integrated in the Beam Search decoding process.

We encountered overflow issues when we tried to implement the 4gram. The trigram and prior models worked fine.

The METUbet trained NN outputs are decoded with LMs from the 3 different corpora described above. In each case we search for the best LM weighting factor and Beam Width parameter and report below.

Corpus	**CER**	**WER**
Wikipedia	0.214	0.84
Datasets	0.212	0.82
METUbet* (Itself)	0.210	0.79
No Language Model (Vanilla BS)	0.225	0.88

Table 1: Character Level LM Beam Search Performances

Recep Oğuz Araz
Prof. Engin Erzin
ELEC390
26.01.2021

Although limited, we see a slight improvement with the character level LMs. The best improvement for the WER, around %10, was achieved with using the dataset as the corpus.

To further decrease the WER, we implement Word Level LMs during the decoding process.

Word Level LM

In the literature, there are various ways to incorporate a word level LM during CTC decoding. We integrate some of the existing implementations to our system.

- **Prefix Search**: Bigram Word Level LM is imposed whenever we decode with a space character. Requires low WER to be effective.

- **Lexicon Search**: We approximate a decoding for the complete sentence, and for each chunk separated by spaces, we look for low edit distance close words and order them by finding their probabilities from the CTC output matrix.

    - The approximation can be done by the character level Beam Search or Argmax decoding.

In order to deal with the underflow problems, we convert given implementations to log probabilities using the above equation. While doing so, to deal with 0 probabilities that result from the LM, we approximate ln0 with -100.

$$\ln(a+b) = lna + \ln\left(1 + \exp\left(lnb - lna\right)\right)$$

Because both algorithms require sorting through the vocabulary, they take much more time and we only use the Mozilla and METUbet transcription corpus for testing.

	Prefix Search	Lexicon Search
CER	0.33	0.32
WER	0.96	0.89

Table 2: Word Level LM Beam Search Performances for Model I

The result of the Lexicon Search provides insight into the high error rates. The probabilities of the true words in the CTC matrix are too low to be detected by the Beam Search algorithm.

Also, the merging type spelling errors make it impossible to use a word level language model, which are quite common. Without having a low CER, the word level LMs have no positive effect.

Therefore, we look for a better NN. As a first step, we train a more complex model with 4 times the parameters and 2 times the depth and report the performance.

Recep Oğuz Araz
Prof. Engin Erzin
ELEC390
26.01.2021

We train the most complex model that fits the validation set the best, and decode it with the same procedures again.

	Beam Search + LM	**Lexicon Search**
CER	0.21	0.30
WER	0.78	0.87

Table 3: Word Level LM Beam Search Performances for Model II

It is apparent that increasing the model complexity did not yield improved results.

Therefore, we conclude that we obtained the best achievable results from *this dataset* with *DS2* and *the LM BeamSearch Decoder*.

Challenges:

In order to achieve the results of the original paper, we need 10.00 hours of speech

We have 8 hours of speech in METUbet and 14 hours in Mozilla.

To increase the dataset size, we need a larger memory to store all the spectrograms. (Colab 40 GB limit)