Assignment 1 : N-gram Language Models

Kartik Sathyanarayanan (EID: ks46373)

February 21, 2018

1 Introduction

Language models compute the probability distribution of a sequence of words or the probability of the next word in a given sentence. To estimate the probability of the next word given the previous sequence of words (or tokens), we can use chain rule. Its very difficult to use chain rule because we might never have enough training data. To ease this computation, we have to use Markov assumption which states that the probability distribution of a current state is dependent only on its parent state. In unigram model, the probability of a word occurring in a sentence depends only on itself. In Bigram model, the probability of a word depends on its previous word or also the next word. Conventional Bigram models model this probability only using the previous word. In this assignment, we also wish to see if the successor of the word (Backward Bigram model) in a sentence has any effect on the probability. We also wish to check if a linear combination of Forward and Backward Bigram models help in getting a better estimate of sentence token probabilities.

2 Methodology

To train the forward bigram model, we accumulate the unigram and bigram counts of all sentences in the training data. If a token is seen for the first time, we count it as an unknown token (<UNK>) to handle out-of-vocabulary words. The model also takes into account the beginning and end of sentence probabilities. To test the forward bigram model, we have a test set from the same corpus. It takes a log of the token probabilities to avoid underflow. The total sentence token probability is calculated as a sum of individual token log probabilities instead of multiplying the probabilities.

The implementation of the backward bigram model wasn't done from scratch, but rather by reversing the sentences before passing it on to the relevant methods. The startToken and endToken weren't interchanged because the sentence now starts with the previous final word and ends with the previous starting word.

To implement Bidirectional Bigram model, I initialized two instances of forward and bigram models. We train both the models on the training set and calculate the probability of the tokens in the bidirectional model by interpolating the probability values of sentence tokens calculated by forward and backward bigram model. The ratio of split in my experiment is 1:1.

3 Experiments

We run our models on LDC tagged data - atis (Airline Booking Query), wsj (Wall Street Journal), brown (Brown corpus) corpora using 90% of sentences for training and the remaining for testing. We calculate perplexity and Word perplexity (excludes start and end sentence tokens) for forward and backward bigram models. For Bidirectional model, we calculate only word perplexity.

3.1 Perplexity measure

Table 1 and Table 2 show perplexity values for forward and backward bigram models. The values seem to be almost the same because they try approximate the actual probabilities. Also, the actual probabilities are calculated using Chain rule instead of Markov substitution.

Dataset	Forward	Backward
atis	9.043	9.013
wsj	74.268	74.268
brown	93.519	93.509

Table 1: Perplexity of training data

Dataset	Forward	Backward
atis	19.341	19.364
wsj	219.715	219.520
brown	231.302	231.205

Table 2: Perplexity of test data

3.2 Word Perplexity measure

For this measure, forward bigram model is slightly better than backward bigram model for atis dataset, which is small in size. Backward bigram model seems to have a better word perplexity measure for larger datasets wsj and brown. This happens because they try approximate the actual probabilities. Also, the actual probabilities are calculated using Chain rule instead of Markov substitution.

Dataset	Forward	Backward	Bidirectional
atis	10.592	11.636	7.235
wsj	88.890	86.660	46.514
brown	113.359	110.783	61.469

Table 3: Word Perplexity of training data

Dataset	Forward	Backward	Bidirectional
atis	24.053	27.161	12.700
wsj	275.117	266.351	126.113
brown	310.667	299.686	167.487

Table 4: Word Perplexity of test data

Forward and backward models have similar values of word perplexity but the bidirectional model's measures are almost 50% lesser. This is so, because it has more information on both the models (Forward and backward models are complementary) and bidirectional model averages out both the token probabilities.

We can also see that for a particular model (forward or backward bigram), perplexity measure is less than the word perplexity measure. This might be because its easier to predict the start and end of sentence in comparison to the next word in a sentence.

4 Conclusion

In this assignment, we explored the bigram language model. We extended the conventional bigram model (forward model) to generate text in reverse (backward bigram model). We also implemented the bidirectional bigram model using both forward and backward models and calculating word perplexity by giving equal weights to both models (1:1 split). The results of the experiments show that the backward bigram model doesn't show much promise with respect to the forward bigram model. The word perplexity measure for bidirectional measure seems to be much better because it is an ensemble of both forward and backward bigram models.