Homework 1: N-gram Language Models

Ankit Goyal
ankit@cs.utexas.edu
Natural Language Processing

February 17, 2015

1 Introduction

N gram Model assumed that the probability of observing the i^i $word_i$ in the context history of the preceding $i-1$ words can be approximated by the probability of observing it in the shortened context history of the preceding $n-1$ words (nth order Markov property).

In this assignment, the forward bigram model was modified to get the backward bigram model. Backward model aims at parsing sentences from right to left as compared to left to right for forward model. The performance of both the models were tested on three different corpus based on Word Perplexity Measures. In the next part of the assignment the two models were combined to a bidirectional model which combined both forward and backward models with a pre-decided weight given to each model.

Table 1

Atis	Forward	Backward	Bidirectional (F = 0.5, B = 0.5)	Bidirectional (F = 0.4, B = 0.6)	Bidirectional (F = 0.6, B = 0.4)
$Perplexity_{Train}$	9.04	9.012	NA	NA	NA
$WordPerplexity_{Train}$	10.59	11.63	7.23	7.32	7.25
$Perplexity_{Test}$	19.34	19.36	NA	NA	NA
$WordPerplexity_{Test}$	24.05	27.16	12.70	12.94	12.69
Brown	Forward	Backward	Bidirectional (F = 0.5, B = 0.5)	Bidirectional (F = 0.4, B = 0.6)	Bidirectional (F = 0.6, B = 0.4)
$Perplexity_{Train}$	93.52	93.51	NA	NA	NA
$WordPerplexity_{Train}$	113.36	110.78	61.47	61.85	62.22
$Perplexity_{Test}$	231.30	231.20	NA	NA	NA
$WordPerplexity_{Test}$	310.67	299.68	167.48	168.31	169.86
WSJ	Forward	Backward	Bidirectional (F = 0.5, B = 0.5)	Bidirectional (F = 0.4, B = 0.6)	Bidirectional (F = 0.6, B = 0.4)
$Perplexity_{Train}$	74.27	74.27	NA	NA	NA
$WordPerplexity_{Train}$	88.89	86.66	46.51	46.83	47.11
$Perplexity_{Test}$	219.71	219.52	NA	NA	NA
$WordPerplexity_{Test}$	275.12	266.35	126.113	127.07	127.81

Table 1. summarizes the results of each case. In the following sections I discuss the results.

2 Backward Bigram Model

In this we train the corpus from top to bottom as done in forward model, however for each sentence instead of training from left to right, we train from right to left.

Analysis: As can be seen from results in table 1, Forward model performs better in case of a smaller data set (Atil), whereas backward performs better in larger datasets (Brown, WSJ).

It seems counterintuitive that backward would perform better than forward model since we speak english from left tor right. I believe in case of Atis, forward performs better because the data consists mainly of the sentences like "Show me the cheapest fair", "Show me the meal", etc. making the start of the sentences very predictable. However since the endings are much more diverse than starting, backward model performs worse than forward model in this case.

Brown, WSJ datasets, consists of natural language and backward model seems to provide more information than forward model. For example, one of the data points in Brown corpus is

```
[ Surveys/NNS ]
show/VBP that/IN
[ one/CD ]
out/IN of/IN
[ three/CD Americans/NNPS ]
has/VBZ
[ vital/JJ contact/NN ]
with/IN
[ the/DT church/NN ]
```

Codeblock 1: Sentence from Brown data

In a forward model , "Surveys show that one out of three Americans has vital contact with the church", the probability of $church$ after the is less than the probability of the before the $church$. Since there are few determiners and lot of nouns. Therefore Backward model performs slightly better in this case.

3 Bidirectional Bigram Model

In this model forward bigram model and backward bigram model were combined. While testing the probability from each model was used to calculate the final bigram probability.

Analysis: From table 1, we can see that Bidirectional model significantly outperforms both Forward and Bidirectional models. Equally weights from both Forward and Bidirectional resulted in the best results.

The reason that it outperforms both the models is because it has more context than either of the models. It is equivalent to saying the $ngram$ model performs better that $(n-1)gram$ model. The probability of word is now dependent on two different models each good at predicting certain type of data. Natural language is complex and due to more context we are able to see better performance.

4 Conclusion

In this homework, we studied the basic implementation that goes in making $ngram$ models and how more context is always useful to predict better results. Using the bidirectional bigram model is a much less expensive approach to providing more context than increasing the N in $ngram$.