CS 388: Natural Language Processing
Homework 1: N-gram Language Models

Manu Agarwal

February 16, 2016

Introduction

An N-gram is a sequence of N tokens in some text or speech. Interestingly, it has been observed that we can predict what the next word in a sequence of text or speech will be based on what the previous N words were. Such probabilistic models of language are called N-gram models. In this assignment, we explore a couple of variations of the conventional bigram (2-gram) model - Backward Bigram Model and Bidirectional Bigram Model. In a conventional N-gram model, we model the generation of text from left to right. However, it has been observed that modeling generation of text from right to left gives better results for certain languages, for instance Bangla [1]. In this assignment, we try out both the Forward Bigram approach and the Backward Bigram approach on three datasets - atis (airline booking query), wsj(Wall Street Journal Text) and brown(Brown corpus of mixed-genre text). We also try combining the probabilities from both the forward and backward bigram models in a Bidirectional Bigram model.

Metrics

In order to evaluate the performance of our models, we use two metrics - Perplexity and Word Perplexity. The perplexity of a sentence $W = w_1 w_2 \dots w_N$ is defined as

$$W = \sqrt[N]{\frac{1}{P(w_1, w_2, \dots, w_N)}}$$

In this experiment, we also predict the end/start of the sentence when we talk of perplexity and ignore the prediction of the sentence boundaries when we talk of word perplexity.

Backward Bigram Model

We implement the Backward Bigram model with a slight change in the forward bigram method - model each token based on the probability of the following token rather than the previous token. Like the forward bigram model, we interpolate the unigram and the bigram probabilities in the ratio 1:9. We replace the first occurrence of every word with <UNK> as in the forward bigram model for smoothing. Thus, every word in the unigram model has some probability of occurrence (in the worst case, the current token is an <UNK>). Hence, we need not apply smoothing explicitly.

We observe that there is almost no change in the perplexity for both the training and the test data as compared to the forward bigram model. This comes as no surprise since we could argue that forward and backward contexts are more or less the same in such huge corpora. However, we observe that Word perplexities are better in the backward bigram model as compared to the forward bigram model on both the training and test data for WSJ and Brown while opposite is the case for ATIS. A possible reason for this better performance on diverse corpora such as WSJ is that English is primarily a right branching language [2] and hence it is easier to predict the parents of the parse tree given its children. Diverse sets such as WSJ or Brown contain a wide spectrum of sentences and hence, the right branching nature of English becomes apparent when calculating metrics such as word perplexity.

We observe better performance of the forward bigram model on ATIS may be because it is a conversational(spoken) sequence of words and spoken English might behave differently as compared to written English. People might use interjections in spoken English more often that might help condition the coming words in a better fashion.

	Forward model	Backward model
Atis		
train data	9.04	9.01
test data	19.34	19.36
WSJ		
train data	74.27	74.27
test data	219.72	219.52
Brown		
train data	93.52	93.51
test data	231.30	231.21

Table 1: Comparison of 'perplexities' of Forward and Backward Bigram models

Bidirectional Bigram Model

Bidirectional model is similar to a trigram model where we try to condition the current word based on the previous two words. In a bidirectional model, we condition the current word based on the previous word and the following word and then combine estimates from both weighting them equally. This helps us surpass the problem of not encountering a specific set of tokens <a-b-c> together in the training data as we might condition b on just a if c never follows b in the training data.

We implement the bidirectional model using linear interpolation of the forward and backward estimates from our previously implemented Bigram and Backward Bigram classes. We do not account for the start-of-sentence token or the end-of-sentence token in this model as modeling it is non-trivial. As expected, the word perplexity output by the bidirectional model for both the training and test data is quite less as compared to both the Bigram Model and the Backward Bigram Model particularly on large corpora such as WSJ and Brown. Combining probabilities from both the contexts helps especially when the prediction of the occurrence of a particular word is unclear using one direction. This might occur for a large number of words in the test corpora and hence the bidirectional model performs better. One problem with the bidirectional model though may be that it overfits the training data. This problem is not quite apparent in our experiments but might be visible when training on one corpus and testing on the other.

	Forward model	Backward model	Bidirectional model
Atis			
train data	10.59	11.64	7.24
test data	24.05	27.16	12.70
WSJ			
train data	88.89	86.66	46.51
test data	275.12	266.35	126.11
Brown			
train data	113.36	110.78	61.47
test data	310.67	299.69	167.49

Table 2: Comparison of 'Word perplexities' of Forward, Backward and Bidirectional Bigram models

References

[1] Khan Naira, et al "History (Forward N-Gram) or Future (Backward N-Gram)? Which Model to Consider for N-Gram analysis in Bangla?" 2006

[2] Wikipedia search : Branching(linguistics)