HW1: N-gram Language models

Rahul R Huilgol (rrh2226)

February 14, 2016

Abstract

In this homework, I implemented Backward Bigram Model and a Bidirectional Bigram Model (which interpolates both normal Bigram and Backward Bigram) and discuss the results.

1 Backward Bigram Model

A simple Bigram Model (also referred to below as Forward Bigram Model for the sake of clarity) works by predicting the probability of a word given the context of the previous word. It interpolates both unigram probability and bigram probability. In this work, the interpolation weights are 0.1 for unigram and 0.9 for bigram. A Backward Bigram Model instead predicts the probability of a word given the context of the next word.

Consider the sentence "*You made history*". After adding markers for start and end sentence, the list of tokens are [<s >, You, made, history, </s>]. The probability of each word with the context of its next word has to be calculated. For the above sentence to calculate *Perplexity*, we need to use p(<s >|You), p(you|made), p(made|history), p(history|</s >). To calculate *Word Perplexity* we ignore the case where we predict a sentence marker, i.e. first term p(<s >|you).

Interpolation was done similar to the Forward Bigram Model and with same parameters. Smoothing for Backward Bigram Model was performed in the same way as in Forward Bigram Model. The left most occurence of a word was treated as the first time it is seen, so it was treated as <UNK >. I believe the left most occurence should still be treated as the first occurence because English is read left to right. A Backward Bigram Model uses the context on the right, but it is still attempting to model the English language.

The Word perplexity measure is a fair way to compare the Forward Bigram and Backward Bigram models because it does not predict sentence markers, which both models treat differently. The results listed in Table 1 use the same split of 10% data for testing, and same default interpolation parameters. The Backward model

		Word Perplexity	
Data	**Set**	*Forward*	*Backward*
ATIS	train	10.59	9.91
	test	24.05	24.79
WSJ	train	88.89	53.82
	test	275.12	180.39
Brown	train	113.36	61.43
	test	310.67	189.16

Table 1: Comparing Word perplexities of Forward and Backward Bigram Models

performed significantly better than Forward model on both Brown and WSJ datasets, for both training and test data. This suggests that the future context is more valuable than backward context for data similar to Brown and WSJ structure. Brown dataset has been carefully crafted to include text from different domains. WSJ being a news dataset also captures many domains. So we could say that in general for most domains, future context is more helpful. Backward Bigram Model did not perform better on the ATIS dataset however. The perplexity of Backward was almost similar to the perplexity of Forward. It performed slightly better on training and performed slightly worse on the more important criterion of test set. This suggests that for Airline reservations domain, the future context of a word is a worse predictor than historical context.

2 Bidirectional Bigram Model

A Bidirectional Bigram Model uses both future and historical context, i.e. both previous and next word to predict the probability of the current word. It computest he probability of a word by interpolating Forward and Bigram Models. Since both Forward and Bigram Models internally interpolate unigram and bigram probabilites, the Bidirectional Bigram Model interpolates using unigram probabilty, forward bigram probability and backward bigram probabilty. The weights used to interpolate are 0.1 for unigram and 0.45 for each of the bigram models. Both styles of the Bigram Models are treated similarly by using same weights because there is no reason for us to claim that a particular kind of context is objectively better.

As an example consider the word "*you*" in the sentence "*You made history*" again. Its probability is computed using both p(you | <s >) both p(you | made), in other words using contexts of previous word and next word.

The Word perplexities of all three kinds of models are shown in Table 2.

		Word Perplexity		
Data	**Set**	*Forward*	*Backward*	*Bidirectional*
ATIS	train	10.59	9.91	7.34
	test	24.05	24.79	11.95
WSJ	train	88.89	53.82	46.41
	test	275.12	180.39	121.14
Brown	train	113.36	61.43	61.13
	test	310.67	189.16	160.22

Table 2: Comparing Word perplexities of Forward, Backward and Bidirectional Bigram Models

The results confirmed my intuition that a Bidirectional Bigram Model would perform the best. It greatly improved on the perplexities that the Forward or Backward Bigram Model could get. It produced perplexities almost half that of the Forward Bigram Model, for both training and test data. We can also observe here that even though Backward Bigram model performed worse than Forward on ATIS dataset, combining the context informations in Bidirectional Model still gave drastically better perplexities. I believe this is because by conditioning on both the previous word and the next word, it gains more information and better understands the context. So using more context information, it can make a better prediction of the word, and thus better fits the data.

3 Conclusion

The Bidirectional Bigram Model performs much better than the standard (Forward) Bigram Model in my experiments. It effectively uses both previous and next words as context to better model the data. This approach seems valuable and is worth using for many applications.