HW1: N-gram Language models

Rahul R Huilgol (rrh2226)

February 17, 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(history|</s >), p(made|history), p(you|made), p(<s >|You). To calculate *Word Perplexity* we ignore the case where we predict a sentence marker, i.e. last term p(<s >|you).

Interpolation of unigram probabilities and bigram probabilites was done similar to the Forward Bigram Model and with same parameters. Smoothing for Backward Bigram Model was performed in way similar to the Forward Bigram Model, except that it now works in the reverse direction. Now the right most occurence of a word is the first time it is seen. So it was treated as <UNK >for this occurence. This decision was made because a Backward Bigram Model should be seen as modelling a sentence from right to left.

The Word perplexity measure is a fair way to compare the Forward Bigram and Backward Bigram models because it does not consider prediction of 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. Datasets used were ATIS, WSJ and Brown.

		Word Perplexity	
Data	**Set**	*Forward*	*Backward*
ATIS	train	10.59	11.68
	test	24.05	27.12
WSJ	train	88.89	86.65
	test	275.12	266.58
Brown	train	113.36	110.78
	test	310.67	299.83

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

The Backward model performed very slightly better than Forward model on Brown and WSJ datasets, for both training and test data. But it performed slightly worse on ATIS dataset. We note here that Brown is a carefully crafted dataset comprising text from different domains. WSJ being a news dataset also comprises many domains. ATIS on the other hand is different. It only comprises of travel requests, and it is spoken text unlike the others. For domains like ATIS, Forward seems to perform slightly better. It is hard

to say for sure, because ATIS is a pretty small dataset. Because the Brown and WSJ are more general, I think the peformance on those datasets, can be thought of as the average case.

So in general, it seems like using the context of the next word, only makes the Bigram Model slightly better. Results are in line with our intuitions. If the Backward Bigram Model performed vastly better, it would be very surprising. This is because, at least in English, the words which follow a word do not have a lot stronger association than the words which precede a word. Sometimes right context might help more, sometimes left context help better.

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 computes the 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. We observed this in the results of above section. Please note that interpolation is done at a word level, because that's how we can exploit information of both models to predict a word.

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.

Word perplexities of all three kinds of models for ATIS, WSJ and Brown datasets are shown in Table 2.

		Word Perplexity		
Data	**Set**	*Forward*	*Backward*	*Bidirectional*
ATIS	train	10.59	11.68	7.26
	test	24.05	27.12	12.66
WSJ	train	88.89	86.65	46.55
	test	275.12	266.58	126.16
Brown	train	113.36	110.78	61.54
	test	310.67	299.83	167.55

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

The results confirmed my hypothesis 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. Even though Backward Bigram model did not perform greatly better than Forward Bigram model, combining it with Forward Bigram model in Bidirectional Bigram model gave drastically better perplexities. I believe this is because by conditioning on both the previous word and the next word, it gains more information than either model and better understands the context. The context information captured by Backward Bigram model is different from the context information captured by the Forward Bigram model. We also see that both these models performed similarly using this different context information. So it makes sense that the Bidirectional Bigram model can make a significantly better prediction of the word by exploiting both sources of context information.

3 Conclusion

The Backward Bigram Model performed almost the same as the Forward Bigram Model. The Bidirectional Bigram Model however performs significantly better than either of those models. It uses both previous and next words as context while predicting a word and models the data better. The Bidirectional approach seems valuable and is worth using for many applications.