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

Lihang Liu
EID: ll32632

1 Introduction

Standard N-gram Language Models assume the text are generated in the order of from left to right. In this report, we are going to present 3 variants of N-gram model: BigramModel, BackwardBigramModel and BidirectionalBigramModel.

2 Implementation

2.1 BigramModel

The source code for the BigramModel is already given along with the instructions and we assume the readers already know how to implement it.

2.2 BackwardBigramModel

A simple and straight-forward way of implementing a BackwardBigramModel is to reverse the order of the input sentences. More concretely, during training, we reverse the order of tokens in every training sentences and then feed into the BigramModel for training. While during testing, we reverse the order of tokens in every test sentences and use the trained BigramModel to evaluate its Perplexity and WordPerplexity.

One may argue that in the original notation of the BackwardBigramModel, for a given sentence " $< S > ABC < \backslash S >$ ", the reversion of it should be " $< \backslash S > CBA < S >$ ". However, in our implementation, the reversion becomes " $< S > CBA < \backslash S >$ ". Actually, the $< S >$ and $< \backslash S >$ here are both symbols. In the scenario of BigramModel, $< \backslash S >$ stands for "the end of the sentence". While in the scenario of the BackwardBigramModel, we can think of $< \backslash S >$ as "the start of the sentence".

2.3 BidirectionalBigramModel

To implement the BidirectionalBigramModel, we have to combine both the BigramModel and the BackwardBigramModel. In this case, for a given sentence, instead of returning only the probability of the whole sentence, we enforce the model to return a list of probabilities for each gram. Then we can average the corresponding probabilities for the BigramModel and the BackwardBigramModel.

3 Experimental Results

3.1 Results

We show the WordPerplexity and Perplexity of 3 different Bigram Models on 3 datasets: atis, wsj and brown, as shown in Tab. 1 and Tab. 2.

	BigramModel		BackwardBigramModel		BidirectionalBigramModel	
	train	test	train	test	train	test
atis	10.59	24.05	11.63	27.16	7.23	12.70
wsj	88.89	275.11	86.66	266.35	46.51	126.11
brown	113.35	310.66	110.78	299.68	61.46	167.48

Table 1: WordPerplexity

	BigramModel		BackwardBigramModel	
	train	test	train	test
atis	9.04	19.34	9.01	19.36
wsj	74.26	219.71	74.26	219.51
brown	93.51	231.30	93.50	231.20

Table 2: Perplexity

3.2 Analysis of the Results

3.2.1 Perplexity

The perplexity of the BigramModel and the BackwardBigramModel differ within 1% on both train and test set of all 3 corpus.

To analyze the result, we assume the perplexity is calculated from bigram only:

$$Perplexity_{forward} = \prod_i \frac{P(s_i, s_{i-1})}{P(s_{i-1})} \tag{1}$$

$$Perplexity_{backward} = \prod_i \frac{P(s_i, s_{i+1})}{P(s_{i+1})} \tag{2}$$

From the above equations, we can see that $Perplexity_{forward}$ equals to $Perplexity_{backward}$ under the assumption of bigram only.

If we add the smoothness of the unigram and the difference in handling the $< UNK >$ words, the result will be slight different. But the result shows that the influence from the unigram and $< UNK >$ is almost negligible.

3.2.2 Word Perplexity

Word Perplexity is different from Perplexity on whether to predict the sentence boundaries.

Comparing the Word Perplexity between the BigramModel and the BackwardBigramModel, the most significant difference is the term $P(< \backslash S > |s_l)$ versus $P(< S > |s_1)$. Approximately and intuitively, if $P(< \backslash S > |s_l) > P(< S > |s_1)$, then the Word Perplexity of BigramModel will be higher.

- In the atis corpus, sentences aren't ended with specific symbols but sentences always starts with a capitalized word, which makes $P(< \backslash S > |s_l) < P(< S > |s_1)$, so the Word Perplexity of BigramModel is slow.
- In the wsj and the brown corpus, sentences are ended with "." or "!", etc, which makes $P(< \backslash S > |s_l) > P(< S > |s_1)$, so the Word Perplexity of BigramModel is higher.

Now we compare the Word Perplexity of BidirectionalBigramModel to BigramModel and BackwardBigramModel. The result shows the BidirectionalBigramModel has a far more lower Word Perplexity. This is not surprising since the BidirectionalBigramModel combines information from both directions and average the probability from both directions. So extreme low probability will possibly be neutralized. For example, we have $m^2 > (m + v)(m - v)$.