Natural Language Processing Assignment 1

Huihuang Zheng, huihuang@utexas.edu

hz4674 Spring 2016

1 Using Code

To run me code, put your data folders of *atis, wsj, brown* in the folder **PartOfSpeechTaggedData/** then you just need to run **run.sh** (using *bash* **run.sh** in Linux terminal). It will show trace file analogous to the BigramModel trace file on course page in the terminal. You can also run **run_and_output_to_file.sh** (using *bash* **run_and_output_to_file.sh** in Linux terminal). That will show trace file in the terminal and generate a file **./traceFile/traceFile.txt**. You can see trace file for all three models in the **traceFile.txt**.

2 How did I implement

2.1 BackwardBigramModel

For **BackwardBigramModel**, the easiest way to implement it is to reverse the sentences when train and predict, and then do same thing as **BigramModel**. So I reverse the input sentences and then call **BigramModel**'s method *trainSentence*, *sentenceLogProb*, *sentenceLogProb2*, *sentenceTokenProbs*.

2.2 BidirectionalBigramModel

To implement **BidirectionBigramModel**, I train two model **BigramModel** and **BackwardBigramModel** for sentences. Then combine their results. In this model, we are trying to get prediction of token T from (left, T, right) by combine of unigram (T), forward bigram model (left, T) and backward bigram model (T, right). I set same weight for bigram and backward bigram. That is, my prediction for token T is

$$p(T) = \frac{1}{2}p_{forward} + \frac{1}{2}p_{backward}$$

.

Notice that the

$$p_{forward} = \frac{1}{2}p(T) + \frac{1}{2}p(T|left)$$

$$p_{backward} = \frac{1}{2}p(T) + \frac{1}{2}p(T|right)$$

So my prediction for token T is

$$p(T) = \frac{1}{2}p(T) + \frac{1}{4}p(T|left) + \frac{1}{2}p(T|right)$$

In addition, the prediction for whole sentence is that we multiple prediction of all tokens together.

3 Experiment

I used 90% data from atis, wsj and brown for training and remaining 10% for testing. My result is:

			atis	wsj	brown
Train	Preplexity	BigramModel	9.04	74.27	93.52
		Backward BigramModel	9.01	74.27	93.51
	Word Preplesity	BigramModel	10.59	88.89	113.36
		Backward BigramModel	19.36	86.66	110.78
		BidirectionalBigram Model	7.24	46.51	61.47
Test	Preplexity	BigramModel	19.34	219.72	231.30
		Backward BigramModel	11.64	219.52	231.21
	Word Preplesity	BigramModel	24.05	275.12	310.67
		Backward BigramModel	27.16	266.35	299.69
		BidirectionalBigram Model	12.70	126.11	167.49

4 Result Analysis

From the table, we can see if we use **BigramModel** and **BackwardBigramModel** each, the preplexity doesn't change a lot. But if we combine them together, the word preplexity of bidirectiondecreases a lot. The reason should be the backward bigram model does something similar to forward bigram model. So the result is similar. But the **BidirectionBigramModel** have both information from forward and backward. It can get lower word preplexity.