Statistical Natural Language Processing
Assignment 1

Luisa E. Quispe Ortiz

lqo202@nyu.edu

1

1. Data Description

The assignment training data consists of Wall Street Journal words split in training, test and validation sets. For evaluation there are also some HUB documents.

2. Experiments

Some of the algorithms described in **Jurafsky and Manning Chp4** are already constructed. In general terms, I believe the programs work this way:

- There are two auxiliar functions in the Util folder: Counter and Countermap which are in charge of giving the counts of the specified word (s) as input arguments.
- All of the codes, have a generate word and sentence part, which I believe Use the former Counters for the N-grams to generate randomly words and sentences, according to the distribution gotten in the same code.
- The empirical N-gram model given follow the algorithms of linear interpolation. There are also two Katz-backoff implementations for bigrams and trigrams. Finally, there is also a SRI algorithm.

The evaluation metrics are already coded and detailed in the assignment, WER for the HUB is the general metric but also perplexity should be taken into account.

First running the empirical N-gram and Katz models, I got the following results in WER and perplexity:

Table 1. Word Error Rates and Perplexities

Model	WER	Perplexity
Unigram	*9.1%*	*1519.12*
Bigram	*7.3%*	*457.29*
Trigram	*6.9%*	*487.88*
Katz-Bigram	*7.5%*	*422.61*
Katz-Trigram	*6.9%*	*487.88*

Perplexity is an intrinsic indicator of how good the model is, as the value decreases the better the model is. It is expected to have lower perplexity values as N grows, this does not mean that it is a good model.

It is clear as the perplexity for bigrams is less than in the trigram, despite the WER being higher. To dig a bit into why this happens, the default word probablities sorted are from prepositions, such as *the* (5.1%), *</S >* (4.6%), *of* (2.4%) among others. After

seeing the sentences generated by the models, I noticed that some contractions appeared a lot, more with 3 than with 2, this may be playing against the trigram model, as they contain more than one word.

In the bigram model, if the $\lambda ischanged, resultsgetworse, sotheoptimumvalueisaround0.6asres$

Regarding the trigram Katz model, the script is the same as the one for the empirical trigram one. The bigram model had some bugs as the sentences generator did not displayed one.

I tried to implement the Kneser-Ney algorithm for bigrams, I based the logic in the following formula, in Jurasky book:

$$P_{KN}(w_i|w_{i-1}) = \frac{max(c(w_{i-1}w_i) - d, 0)}{c(w_{i-1})} + \lambda_{w_i-1} P_{CONTINUATION}(w_i)$$

Where :

$$\lambda_{w_i-1} = \frac{d}{c(w_{i-1})}|w : c(w_{i-1}, w) > 0|$$

$$P_{CONTINUATION}(w_i) = \frac{c(w_{i-1})|w_{i-1} : c(w_{i-1}, w_i) > 0|}{c(w_{j-1})|w_{i-1} : c(w_{j-1}, w_j) > 0|}$$

I considered the denominator of the first term and the λ term as count of unigrams, and to be the same. While in the second part of the λ calculation, I considered it to be the total number of bigrams that started with the word w_{i-1}. In the continuation probability, the numerator was considerd to be the total number of bigrams for by w_{i-1} and w_i , while the denominator was the total number of bigrams.

However, I found that even when we are running the model in the HUB data, some counts result in zero, which is not supossed to happen, and that actually led to infinity or zero intermediate results in the code.

As a result I notice that some of the sequence of words even though appearing in HUB, were not present in the training data. Therefore as the algorithm mentions that those counts have to be at least one (as they ARE existing in the document) I settled the counts to 1 by default. The other thing I tried was to group them in the UNK token, but this gave me infinity results too.

With this things settled in the code, I decided to run it, generating a WER around 9% and 11% (whetheher I corrected unigram, bigram or both counts) with a Perplexity around 150. Something interesting I noticed is that I got a 5.3% WER if no corrections were made, but at the same time I was getting a infinite perplexity (due to the zero division) and some parts of the predictions with symbols.

I believe that as in the Course page, the results (with corrections) were at the baseline, my implementations is not correct or is missing something. Theoretically, this method is supposed to give better results than the bigrams of trigrams, as it uses the discount factor and uses previous knowledge.