AJ Broderick
UIN - 01244170

CS 733 Module 1 Assignment: N-Grams

Throughout the analysis, the Financial News headlines were utilized to train different n-gram models to determine how each would handle unseen sequences. These models were rooted in n-grams, to break up the training data into unigrams, bigrams and trigrams and calculate the conditional probability of each to then train the models. Three models were created, trained and tested on the unseen sequences and finally the complexity was calculated to evaluate and compare each model.

The table below shows the final complexity that each calculated for each of the models when applied to the unigrams, bigrams, and trigrams.

	N-Gram	Laplace Smoothed	Linear Interpolated
Unigram	526,645	7,442	950,414
Bigram	924,584	60,644	
Trigram	950,412	72,798	

The first was a basic n-gram model. This model did not transform the training data at all, and utilized the conditional probabilities to handle the unseen sequences. From the results of this model, there were some very high perplexities, and the unigrams handled the sequences the best. A hypothesis of why the unigrams worked the best could be the nature of the training and testing data. Given that the data was financial data, which an emphasis on retail investment, there is the chance that unique words exist in the vocabulary (such as company names). From this, the chance of the bigrams and trigram existing in the training data are low and sometimes zero, resulting in higher perplexity.

To address these are and missing n-grams, the second model was developed, this model used an add-one smoothing technique (Laplace smoothing). For each of the n-grams that could be present in the data, each count had one added to it. From this, the previously missing n-grams had at least one instance and therefore a conditional probability. In calculating the probability, the denominator of the frequency was adjusted by the size of the vocabulary to address the addition the missing n-grams. Looking at the table, the overall perplexity of each n-gram decreased by a significant amount (upwards of 90+%), which means that the model has a better probability of handling the unseen n-grams. Again the unigram model handles the data the best.

From here, the last model was created, and this model utilized an alternative was to solve the problem of the missing n-grams in the training set. This model used linear interpolation the iterate down from the trigram to the bigram and the unigram, and applies a weight to each n-gram. By doing this, it uses less context to generalize more for the unseen data that model doesn't know about. In the instructions, the highest weight was assigned to the unigram at 70%, and then bigram at 20% and 10% for trigrams. When applied to the testing data, the missing n-grams at the bigram and trigram level had a big impact on the perplexity, resulting in a result like what was seen in for basic n-gram model.

At the end of the analysis, the Laplace Smoothed unigram model performed the best on the unseen sequences of the financial data. As mentioned earlier, this could be explained by the training and testing data itself, and steps could be taken to address the data to create better frequencies and probabilities. These steps include stemming the data and ignoring company names, which would create a smaller dictionary and increase the frequency of the n-grams and decrease the end perplexity.