COM6513 Lab 3: Homophone Disambiguation
Registration Number: 160122440

1. INTRODUCTION

The aim of this lab exercise is to create simple language models to estimate the probability of each candidate words being the correct one to fit in the blank of 10 sentences.

2. MODELS

Three models were created in this lab, which are: **Unigram model, Bigram model, Bigram & Smoothing model**.
For the creation of the models the frequency of the words was taken into account, by importing a large corpus (Brown corpus) from NLTK.
The models were modified to read the sentences from a file called *questions.txt*, and then output the selected word to fit in the blank.
The output of each model is initialised to **None**, and if the model fails to select any of the available words, then the output is **None**.

2.1 Unigrams

The unigram program uses the unigram model to find the most frequent word in the corpus (using unigram frequencies) and assign that word to fit in the blank. The unigram program is not very accurate as it only takes into account the occurrence frequency of the words in the large corpus, without taking into consideration any other information in the sentence; such as syntax or semantics.

2.2 Bigrams

The bigram program uses the bigram model to find the most frequent bigrams in the corpus (using bigram frequencies). This model is more accurate than the unigram model because it takes into consideration the word that comes before the blank space and also the word immediately after it. As an example, if the sentence is: 'University___ Sheffield' and the words to choose from are: 'of / off', then the bigrams considered in the selection process are: 'University-of, of-Sheffield, University-off, off-Sheffield. Therefore, this model takes syntax into consideration, meaning that is more accurate than the unigram model. This is also proved in section **3: ACCURACY & RESULTS**, of this report.
As also explained at the beginning of section **2: MODELS**, the output is initialised to **None** if the model fails to select any of the optional words to fit in the blank. Some output results of the bigram model show this **None** output (see ection **3: ACCURACY & RESULTS**). This occurs because the probabilities are found by dividing the bigram probability with the unigram probability, and if the divisor probability (unigram) is zero, then the division is not possible. Therefore, the program is created to only calculate divisions which are possible, and output the initial state (**None**) if the calculation is not possible.

2.3 Bigrams & Smoothing

The Bigrams & Smoothing program words in the same way as the bigram model, with some minor modifications, which are mainly done in order to avoid the problem created when having to divide with a zero value (zero unigram probability). The modifications are to add 1 in the numerator and add the summation of the all the unigram counts (*python code: 'len(unigram_count)'*) to the denominator (divisor). This method is known as *Add-One Bigram Smoothing (Laplace Smoothing)* and it is mainly used to allow the assignment of non-zero probabilities to words which do not occur in the sample, or in this case it removes the problem of zero probabilities that do not allow the division to happen.

3. ACCURACY & RESULTS

In the first column of the table below, the available words that the algorithms had to choose from are shown. The correct word is in **bold**.

OPTIONS:	UNIGRAMS:	BIGRAMS:	BIGRAMS & SMOOTHING:
weather/**whether**	whether✓	whether✓	whether✓
site/sight	sight**X**	site✓	site✓
through/threw	through✓	through✓	through✓
knew/**new**	new✓	new✓	new✓
piece/peace	peace**X**	piece✓	piece✓
caught/**court**	court✓	court✓	court✓
aloud/**allowed**	allowed✓	None**X**	allowed✓
hire/higher	higher**X**	hire✓	hire✓
paw/poor/**pour**	poor**X**	None **X**	poor**X**
cheque/**check**	check✓	None **X**	check✓

Table 1: Outputs of the three programs (Unigrams, Bigrams, Bigrams & Smoothing) .

As it can be seen in table 1, when using the Bigram model, for some sentences none of the words is chosen. This is because some words do not appear in the corpus and therefore their probability is zero. Division by zero is not possible, therefore the model is adjusted to only perform the division when the probability of the divisor is not zero.

When using the Unigram model, 6/10 words are selected correctly. Therefore the **Unigram Accuracy is 0.6**.

When using the Bigram model, 7/10 words are selected correctly. Therefore the **Bigram Accuracy is 0.7**.

When using the Bigram Smoothing model, 9/10 words are selected correctly. Therefore the **Bigram Smoothing Accuracy is 0.9**.

4. DISCUSSION

As shown in the previous section, the most accurate model is the Bigrams & Smoothing model, with accuracy of 0.9. This is not a surprising result as this model takes into consideration more language techniques such as syntax and smoothing, than the other two models. The unigram model just counts the occurrence frequency of the words, and the bigram model only the bigram occurrence frequency. On the other hand, the Bigram & Smoothing model takes into consideration what the previous two models do, but also uses the Laplace Smoothing method to account for zero probabilities, therefore this leads to the Bigrams & Smoothing model to be the most accurate of the three.