**Report**

**Probability Calculations:**

Below are few examples on how the probability is calculated in this assignment for -uni, -bi and -tri gram models.

1. Unigram model: Consider the following sentence.

“I want to”.

P(I)=Number of occurrences of ‘I’ / Total number of words = 1 / 3

P(want)=Number of occurrences of ‘want’ / Total number of words = 1 / 3

P(to)= Number of occurrences of ‘to’ / Total number of words = 1 / 3

1. Bi-gram model: In case of bi-gram model, below is the calculation of probabilities.

P(I|want) = Frequency of bi-gram (I, want) / frequency of ’I’

P(want|to) = Frequency of bi-gram (want,to) / frequency of ‘want’

1. Tri-gram model: Similar to bi-gram, tri-gram probabilities are calculated as below.

P(I|want to) = Frequency of tri-gram (I,want,to) / frequency of (I,want)

**Challenges faced:**

* Probability of a certain character being 0, caused the probability of whole word to be 0. To fix this, I took the logarithmic probabilities of individual characters, summed them up and took the Anti-log of the result as prescribed in the text book. Based on this result, I am classifying the word to be either ‘English’ or ‘French’.
* Apart from this, I encountered a few trivial coding errors, which were easily handled.

**Performance:**

Below is the accuracy percentile for language models:

1. English v/s French language models on English test words.

* **Uni-gram model:** 64.7 %.
* **Bi-gram model:** 40.3 %.
* **Tri-gram model:** 24.4 %.

1. Spanish v/s Italian language models on Spanish test words.
   * **Uni-gram model:** 46.6 %
   * **Bi-gram model:** 49.8 %
   * **Tri-gram model:** 32.8

Based on the above results, I believe Spanish v/s Italian language pairs were harder to detect.