**[N-Grams for two Corpora](https://uncc.instructure.com/courses/60562/modules/items/999586" \o "N-Grams for two Corpora)**

**Calculations of probabilities:**

1. Unigram Model:

For unigram model, I counted the total number of characters in the corpora(N) and the occurrences of character in the corpora(M).

Probability(P) = M/N

1. Bigram Model:

For bigram model, I counted the occurrences of two adjacent characters of a word and divided the occurrence of first character to create the model.I usef FreqDist() function to calculate the count of words.

1. Trigram Model:

For trigram model, I counted the occurrences of three adjacent characters of word and divided it by the occurrence of first two adjacent characters to create the model.

To calculate the probability using these models, I extracted first 100 words from raw dataset and applied each word to the model individually. For computing probabilities for English v/s French I wrote a function :

**getProbabilityUnigram(testSet, unigramModel1, unigramModel2, predictionLanguage, secondLanguage)**

If the probability computed by first model is greater than that of second model assign the language as prediction i.e. actual language or assign it the other language. I created the **tuple of (“Word”, Probability with Actual Language Model, Probability with Expected Language Model, Assigned Language)**.

I have written the tuples to .csv files for each model.

Computing the accuracy, I checked if the predicted language is what was expected then increase the correct prediction count. Finally divide the count by number of word in test set i.e. 1000.

**Challenges Faced:**

The first challenge I faced was to figure out which model is to be created that is a word model or a character model. Once I understood what exactly is to be predicted then I decided upon using character model.

**Accuracies of all the models:**

1. English v/s French Language Models:

|  |  |
| --- | --- |
| Name of the Model | Accuracy (%) |
| English Unigram Model | 42.9 |
| French Unigram Model | 59.5 |
| English Bigram Model | 67.8 |
| French Bigram Model | 76.9 |
| English Trigram Model | 77.5 |
| French Trigram Model | 71.3 |

1. Italian v/s Spanish Language Models:

|  |  |
| --- | --- |
| Name of the Model | Accuracy (%) |
| Italian Unigram Model | 48.8 |
| Spanish Unigram Model | 61.4 |
| Italian Bigram Model | 53.7 |
| Spanish Bigram Model | 73.6 |
| Italian Trigram Model | 71.8 |
| Spanish Trigram Model | 77.7 |

**Observations and Conclusions:**

* Comparing the two models English and French, French models gives better accuracy over English.
* The test set consists of same context of data but in different languages. English Bigram Model perfectly identifies words such as ‘of’, ‘freedom’, ‘every’.
* It is observed that the probability of word being in a actual language increases as we change from bigram to trigram. In trigram model lot of English words have zero probability in French model. We get a probability of 1 for word like ‘world’, ‘in’, ‘enjoy’, ‘friendly’.
* In case of Italian and Spanish language, Spanish model gave better accuracy.
* One more observation made is that French Bigram model shows higher accuracy than French Trigram Model. The reason to this can be insufficient dataset i.e. small dataset and the model fitting the data. If model overfits or underfits the data then it tends to predict inaccurately.