**Assignment 5 – Ngrams**

N-grams are a tool to help build a language model by recognizing patterns over sequences of n-words. A body of text called a corpus is used to create a probabilistic model of a language. The choice of corpus can make a massive difference. There are several use cases for N-grams, they can be used for autocompleting of sentences, auto spell check, and semantic analysis. N-grams can also be used for DNA sequencing and other computational linguistic applications.

Probabilities for unigrams and bigrams are calculated in the following way. You calculate the number of a particular unigram or bigram, and then divide that number by the total number of unigrams/bigrams.

The source text that is chosen is very important. Depending on which source is used could affect the probabilities and what will be contained in the N-grams used. Smoothing is important and is used to make sure there are no unigrams that have a 0 value. If a unigram value had a value of 0 then that would cause an error when a probability is run. There are many smoothing techniques. An easy one is to add small probability mass, or 1 to every value so that every value will be greater than 0.

Language models can be used to generate text in a few ways. There are few strategies such as sampling, choosing highest probability and others. This approach isn’t amazing though as it is likely going to generate some not very useful text. Language models can be evaluated in two main ways. One way is to measure how well the model captures what it’s supposed to with probabilities. Another way is to measure how good the model does a particular task.

Google’s n-gram viewer allows you to select words and see how often those words appear in some/all Google Books.