Narrative of Ngrams

An N-gram is a sequence of n consecutive tokens. All possible N-grams are often calculated for a given corpus or text.

N-grams can be used for text messaging. For example, a user can skip typing out the entire word and save time if the application can accurately guess what word the user is typing next. Additionally, n-grams can be used for pattern finding. For example in spelling correctors. Another use case can be speech recognition similar to how it was used in this program.

There are multiple ways a probability can be calculated from unigrams and bigrams. Both can be calculated using Laplace smoothing. For each bigram in the testing string, the occurrence of the bigram(b) and the occurrence of the first unigram of said bigram(u) are calculated. (b + 1) / (u + number of unique tokens). The result is multiplied for each bigram in the test dataset. The resulting value is the probability.

The source text can radically change the resulting model. For instance, imagine using Romeo and Juliet as the source text from which the model is built. Unigrams in this model would include "`Tis" and "capulet". But this model is being used to predict text messages. Both models are English, but the trained model would be completely irrelevant to the final purpose.

Smoothing is very important because no source text is perfect. Smoothing essentially fills in gaps where the value of zero exists in the visualized list of possible n-grams. One way to do this is to add 1 to the number of times a bigram occurs. Even if a particular bigram doesn't occur in the source text, the Laplace smoothing calculation will consider that the bigram occurred once.

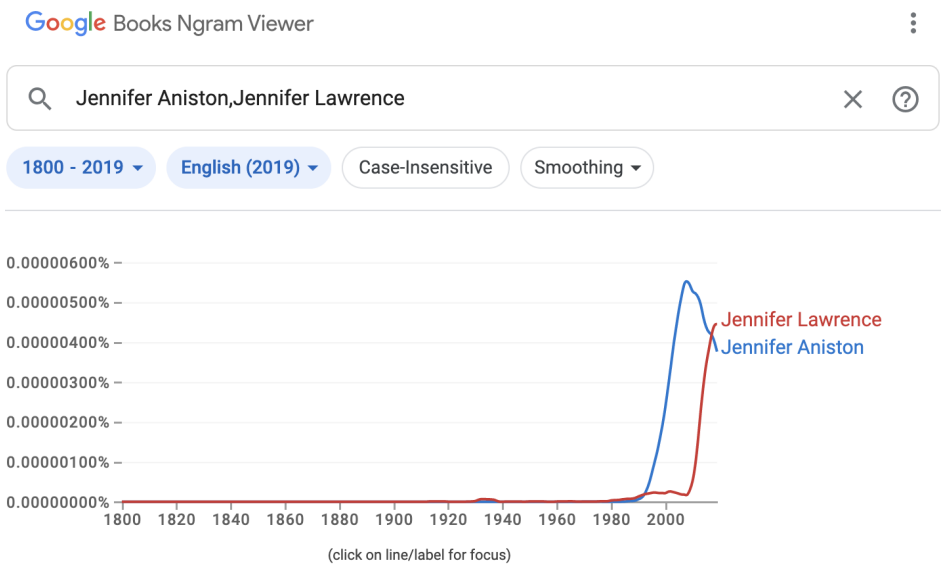
Essentially, language models using ngrams can help estimate the probability that a particular word will occur next given the previous words. Ngrams language models provide insight into following and preceding words for a particular word. For instance, in the English

language, the word ‘water’ can be followed by ‘bottle’, by ‘fountain’, or can be found alone. N-grams, specifically for the bigrams in the example above, help us find the word that water is followed by according to the probability they occurred in the corpus. Some limitations include bias of corpus. As stated previously, the chosen corpus is very important and can influence the model a great deal. Additionally, the size of the language model is also a limitation. Currently, the time it takes to create bigrams and produce probabilities.

Language models can be evaluated with a test dataset. A dataset that is pre-classified or predetermined can be tested against the predicted responses using the model. Then a percentage of correctness can be calculated and used as an evaluation of the model.

The google Ngram viewer is a chart showing the frequency of selected ngrams that uses google’s yearly count of n-grams found in printed sources of that year.

Example:



Frequencies of Jennifer Lawrence versus Jennifer Aniston.