Homework 4: Narrative

N-grams are groups of words commonly found adjacent to each other in a text of N words. N-grams can very easily be used to build language models. We can analyze which words are commonly found together in an input text and this training data can be used to train a model with the aim of predicting future occurrences.

One type of application that n-grams are probably used in is spelling and grammar checkers. For example, Grammarly likely uses n-grams in some form to inform its grammar-checking process. Another type of application could be some form of language detection software. The application would compare already input n-grams to a text that the user inputs to detect the language.

Calculating probabilities for unigrams and bigrams is pretty straightforward. You first need to read in a large input file to be used for training. For unigrams, you'll count the number of individual tokens while for bigrams you'll also be looking at adjacent tokens. To calculate probability for unigrams, divide the number of appearances by the total number of tokens in the file. For bigrams, divide the number of bigrams by the total number of tokens.

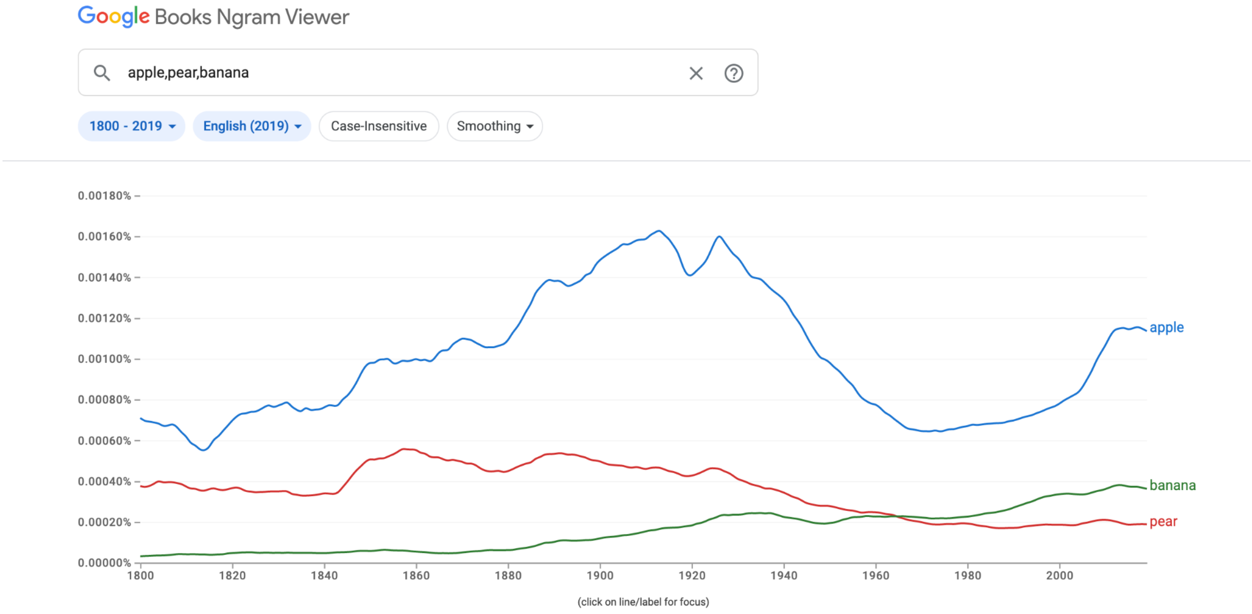
The source text is a very important part of the process. The source text will be how the language model will be trained because it provides the training data. Having a bigger and more complex source text will strengthen the model and make it more useful.

Smoothing is important because it prevents certain calculation errors from happening. It helps to keep an n-gram with a probability of zero from making the total probability equal to zero during calculations. One approach to smoothing is to add a certain number to bigram and unigram counts so that the zero won’t affect calculations.

Language models can be used for text generation by using the probabilities calculated to suggest or autofill certain words. Higher probability n-grams would be suggested based on what is originally typed. There can be some limitations though. For example, this process can result in long sentences with no clear meaning or subject. Another drawback would be grammatically incorrect sentences due to words having different meanings and uses.

Language models can be evaluated by seeing how well they evaluate real languages, predict text based on an input, and form grammatically correct sentences. This depends a lot on the training data used to train the model. Language models trained on a large amount of data containing many different words will tend to perform better.

Google has a tool that allows users to find out the rate of various n-grams over a certain period of time. The number of parameters or n-grams can be changed by the user and the same is true for the period of time being looked at. With Google’s n-gram viewer, the x-axis is the year and the y-axis is the probability. The following example was used to see the prevalence of various fruit names such as apple, banana, and pear. Apple seems to be the most common across time periods while banana has recently taken over pear.



Google Books Ngram Viewer

apple,pear,banana

1800 - 2019

English (2019)

Case-Insensitive

Smoothing

(click on line/label for focus)