N-Grams

Joshua Durana

CS 4395.001

N-grams are words or sets of words from a corpus. A unigram is a set of one word, a bigram is a set of two words, etc. They're used to create a probabilistic model of a language. One application is suggesting the next word while a user is typing. Another use is correcting spelling.

The probability of a certain n-gram to be in a text is calculated by first checking the model if it has the n-gram. If not, then the probability is 0. If the n-gram is in the model, we divide the number of occurrences of the n-gram to the total number of n-grams.

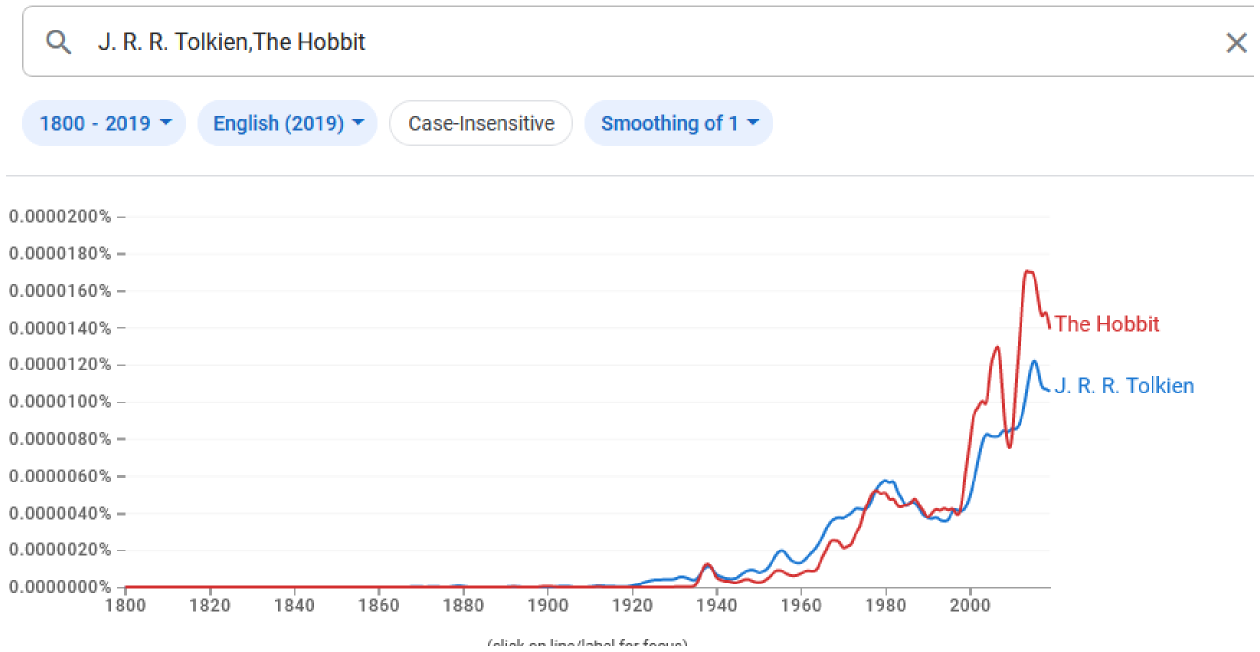
Source text for the n-gram model is important because it will influence which words are more common than others. A word from two different authors can have different probabilities for certain words which can influence what kind of text the model can be used for.

Smoothing is important to prevent the sparsity problem. The sparsity problem is the model being unable to contain every word and their possible sequence and when there's a new word that's not in the model, then it's probability is 0. One simple smoothing method is Laplace smoothing. Whenever you encounter a new word we add 1 to the probability and add the size of the vocabulary to the denominator.

Text generation with an n-gram model starts with a start word. Then it finds the n-gram model with the highest probability and appends it to the phrase. This repeats until the period token is added to the phrase. It's mainly limited due the corpus that the model is trained with. It cannot use words not in the corpus, so we need a bigger corpus to have better results.

Language models can be evaluated in 2 ways. One way is extrinsic evaluations with human annotators that evaluate the model with a predefined metric. They're costly and take a long time, so they're mainly used at the end of a project. Intrinsic evaluation is using some metric to compare different models. Perplexity is a common metric used, it measures how well the model predicts the text in the test data. A good model should have low perplexity to show that it has less chaos in the training data.

The Google Ngram Viewer shows the frequency of an n-gram within Google Books over time. You can input multiple n-grams and compare their frequency over time. One example is J.R.R. Tolkein and the Hobbit and it will show how often that n-gram appears within Google Books.


J. R. R. Tolkien,The Hobbit

1800 - 2019 | English (2019) | Case-Insensitive | Smoothing of 1