Portfolio 4: Ngrams Narrative
Shreya Valaboju, Soham Mukherjee
CS 4395.001

A n-gram is a sliding window over some text. The "n" represents the number of words included at a time. For example, a bigram is 2 words (ie, "the cat") while a trigram is 3 words (ie, "the cat in"). N-grams can be used to construct a language model through counting how often a sequence of words occurs in a corpus and then estimating probabilities for those sequences. More specifically, say we wanted to compute the probability that a trigram will occur in a corpus. We would solve this by making a markov assumption and compute $P(w_1,w_2,w_3) = P(w_1)P(w_2|w_1)P(w_3|w_2)$ *,* where P is the probability and w1, w2, w3 are words 1, 2, and 3 respectively.

A few examples of where n-grams can be used are in speech recognition, spell checking, auto-complete, machine translation, or as features in a machine learning model. By analyzing the frequency of ngrams in each of these applications, patterns and reappearing sequences can be derived from these calculations.

For instance, say we have some text, "Jack be nimble. Jack be quick. Jack jump over the candlestick." The probability for the unigram "Jack" would be the count of the number of times Jack occurs divided by the total number of tokens in the text. So, P("Jack") = 3/14. In order to find the probability of a bigram, we would do P(w1, w2) = P(w1)P(w2|w1). For example, the probability of the bigram "Jack be" would equal P("Jack") * P("be"|"Jack"). This would solve to 3/14 * ⅔ = 0.14.

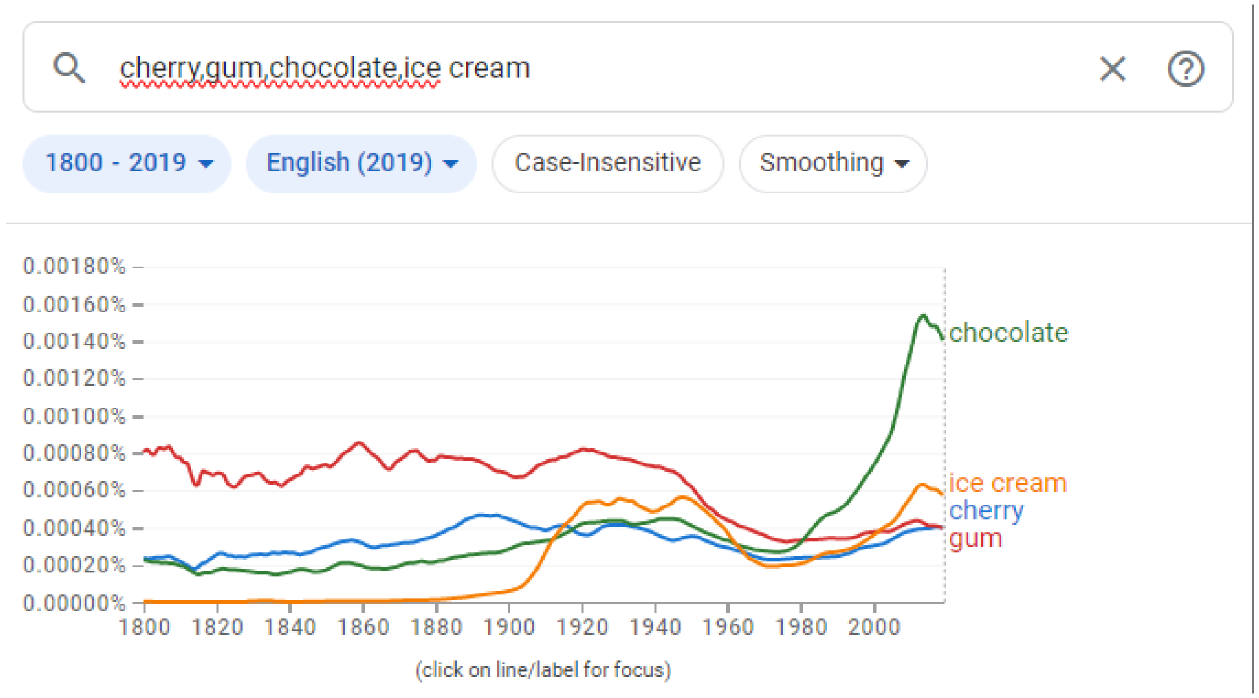
Source text is important in building a language model because the model needs a basis from where to make its predictions. The source text allows the model to analyze how text would normally be outputted when it comes to actually generating output from the given model. As a result, it's important to have a good source text that the language model can make appropriate assumptions off of.

Data smoothing removes noise from a given dataset. It does so by ignoring one-time outliers, allowing more significant patterns to stand out. This allows us to predict trends more accurately, such as securities prices in economic analysis. One example of smoothing is "moving average", which filters out volatility from random movements within a given data point. If plotted on a graph, a moving average smoothing technique generates a line that is similar to the overall structure of the data points.

Language models are a probability distribution of various words, and they can be used for text generation because sentences formed with the model can have a similar distribution of words. It can be used for text generation because syntactically, they allow for the construction of sentences that together, make sense. Given text data as input, language models can analyze them to find a basis for word predictions that can then be used to formulate sentences. Limitations happen when it comes to words that are significant but can be outliers in terms of probability. An example would be an impactful word like "however", which only appears a few times relative to a word like "of" or "the", but adds a different meaning to the sentence. Moreover, these models are strictly limited by their input data.

There are two ways they can be evaluated: intrinsic and extrinsic evaluation. Intrinsic evaluation captures how well the model captures probabilities, independent of any other application. This is used to evaluate possible improvements in a language model, without any external factors. Extrinsic evaluation evaluates the performance of a language by applying it to an application. It's similar to a production level deployment, where it can be analyzed whether a particular improvement will be pertinent to the task at hand.

The n-gram viewer charts the frequencies of a set of search strings using an annual n-gram count. The program can search for specifics of a given word or phrase, including misspellings and parts of speech.



cherry,gum,chocolate,ice cream

1800 - 2019 · English (2019) · Case-Insensitive · Smoothing