Noah Gonzales
CS4395 - Human Language Technologies

n-grams

An n-gram is a section of text n tokens long. This section can apply to any part of a text, and can overlap other n-grams. n-grams cover the whole text and encompass every combination of a series of n tokens, so a probabilistic model of a language can be made that describes the likelihood of certain words appearing with other words in a certain order. n-grams can be used in many situations since they describe a text well. One of which is what this assignment is about: probabilistic modeling of a language. Another use case is the classification of text. n-grams, because of their flexibility and grouping, are useful for identifying features in text. n-grams also can be use to achieve more accurate sentiment analysis than what can be done with just tokens, since a combination of words can take different meaning than just the words by themselves.

Probabilities for unigrams and bigrams are loosely defined as the likelihood of a unigram or bigram to appear in a language. That is, it is the probability that a token occurs in a language for unigrams. For bigrams, it is the probability that the first token occurs multiplied by the probability that the second token occurs given that the first token occurs (conditional probability).

A model is only as good the text it is trained on. It is the same for the usefulness of n-grams in building a language model. The source text for building a training model affects the probabilities of certain n-grams in the model. One source text could have more occurrences of an n-gram than a different source text, and thus the model would have different probabilities for different n-grams. A source text that accurately represents a language is important to build an accurate model for a language. Even if a source text is good, it is almost impossible for it to represent all aspects of a language fairly. Smoothing evens the odds for n-grams that appear too often in the source text and n-grams that don't appear enough or at all. n-grams that don't appear at all are given some probability. A way to do this is to simply consider the minimum number of occurrences to be 1.

Language models can be used for text generation. Large n-grams can be used to describe possible combinations of tokens. The larger the n-gram, the better the generation is, since larger n-grams more accurately represent the likelihood that tokens appear together. However, generation with language models is limited by the corpus. If a corpus doesn't describe something, it won't be able to be generated. No language model is perfect. Models can be evaluated by looking at the model with actual humans. Actual humans can evaluate how accurate a model is. The model can also evaluate itself using perplexity. Perplexity is the inverse probability of seeing the words we observe, normalized by the number of words, with low perplexity being better.

Google's n-gram viewer is a good resource for understanding n-grams and also gaining insight on a language over time. By searching several n-grams, you can compare their likelihood over time. For example, the likelihood of Batman, Superman, and Spiderman occurring in the human language over time can be seen.

Google Books Ngram Viewer

Batman,Superman,Spiderman

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