N-grams are basically sequences of n words occurring consecutively. Utilizing sets of n-grams, one can build models which can predict and generate languages. Some examples of the applications of n-grams include language translation, auto-complete sentences and automatic spelling checks.

The probability of a unigram (a 1-gram) is calculated as the ratio of the count of the unigram to the total count of the unigrams. The probability of a bigram is calculated as the probability of the first word as a unigram times the count of the bigram divided by the count of the unigram. Since all the unigrams and the bigrams are taken from the source text (training set), it is very important for the source text to be big enough to not skew results of the language model. It is important that there are a good enough number of n-grams present in the model so that the computation of probabilities is more robust.

Sometimes a particular n-gram may not exist in the test, and this means that the probability function would be equal to zero, as the count of the n-gram will always stay at zero. A good way to combat this is smoothing. A smoothing function that would help with calculating probabilities better, is the laplace smoothing method. In this smoothing approach, we add 1 to the probability in the numerator(count of the n-gram), so in the case that the n-gram does not exist, then the minimum value would still be 1. We also add the length of our vocabulary to the denominator in order to account for the smoothing.

Language models can be used for text generation, and this can be an extremely useful feature. However, a n-gram language model has its limitations in terms of the complexity and semantic accuracy of the sentences it is able to build. The way sentences can be generated using a n-gram model, is that the most probable bigrams are taken consecutively to form a sentence. This can be conceived as a naive way of forming a sentence, as this is completely dependent on the robustness of the training data and might not always imply any meaning in terms of the actual sentence. There are many ways to test how a language model performs. One way would be to test how accurate it is on some test data that models the real world. Another way would be to see how well it can perform a certain task such as language generation.

The Google N-gram viewer is essentially a search engine that utilizes n-grams to return the frequencies of search queries. It was created by Jon Orwant and released in 2010. This engine also supports searches for part of speech. The engine is mainly used for textual analysis and research. For example, you can type in a word, and see how often it has been used through the years. The phrase, “Albert Einstein” has only been heavily used from the 1940s, which makes sense considering that he made his monumental discoveries during this era.