Ngrams are groups of n words. An unigram is a group of one word. A bigram is a group of two words. They can be used to build language models because probabilities can be calculated based on how many times bigrams occur together. When bigrams and trigrams and so on are found occurring with a high probability, you could say that the phrase is common in the language and finding numerous of these ngrams would allow someone to build a language model. Ngrams can be used in machine learning, where they can be used in the feature space for supervised machine learning algorithms such as SVM and Naive Bayes. They can also be used in text mining, where they can help break down a large dataset and provide valuable information to the user.

Smoothing is created for unigrams and bigrams through a formula called Laplace smoothing. Smoothing involves filling in zero values with a little bit of the probability mass. The idea of Laplace smoothing is to add one to the zero count so that it is not zero, and add the total vocabulary count to the denominator to balance everything out. It is easy to implement but does not perform well since it aggressively adjusts all probabilities.

The source text is very important in a language model. The source text needs to be fully representative of the language and should contain meaningful information that can be extracted using natural language processing. It should have relevant information pertaining to the type of processing that is occurring on it and be large to have a better chance of getting more useful information.

Probabilities are created for unigrams and bigrams using rules of independence and statistical formulas. The probability of a bigram would be the probability of the first word multiplied by the probability of the second word knowing that the first word occurs. This corresponds with the rules for calculating independent events. If the probability shows that the two words are not independent, there is a chance that they could make a bigram.

Language models can be used for text generation since probabilities of words that occur after a certain word can be stored by a model and the model can always generate text by appending the most common word that occurs after the previous word generated by the model. THis approach is limited because often the text could seem robotic and rigid as the model does not have the vast knowledge and emotion that humans possess so the text generated by the model would not be on the same level as a human.

Language models can be evaluated since the probabilities they can generate can be looked at and used to understand the text. The training data they assign probabilities to can be used to analyze test data and create more complex models.

Google’s ngram viewer sorts through the entire Google Books library for the entered phrases and returns their probabilities from a selected year range. For example, if Albert Einstein, Frankenstein, and Sherlock Holmes are entered, the ngram viewer says that frankenstein has the highest probability right now.