Alejo Vinluan

Abv210001

CS 4395

N-Grams Narrative

N-grams are sets of words over text. They can be any size in length, with the most common being unigrams and bigrams. Unigrams are all the individual words in text whereas bigrams are sets of two words in text. N-grams can make a probabilistic model of language and find statistical properties within language. The probability that a sequence of words appear within text can be used to find how often these sequences occur. For example, the probability that the sequence “Jack jump over” can significantly help design the language model.

N-grams are commonly used for spelling correction, machine translation, speech recognition, and typed out suggestion when messaging and searching. For example, the suggestions when completing a search are useful since humans will generally make similar searches according to their language. The search “how to make macaroni and cheese” is extremely common and can be used to auto complete the search.

Probabilities for unigrams and bigrams are created by creating a dictionary from a corpus. In the program, a unigram dictionary was created by tokenizing the text and creating a dictionary from the tokenized text. The key being the unigram and the value being the count. The value of the dictionary would serve as a probability score since the higher the count, the higher the probability. Finding probabilities for bigrams is the same process. A dictionary was made for each bigram, with the key being the bigram and the value being the count of the bigram.

The source text when building the language model is extremely important as it will contain different sets of phrases, thus creating a different model for the N-grams. A Shakespeare play would lean towards sets of phrases in Old English and look for a language that resembles that English. In another example, a science textbook would lean towards writing that only state facts and no emotion. These extreme cases would cause the language model to heavily favor certain phrases and vocabulary.

Smoothing is important since the chosen text for training cannot contain every set of phrases, words, or ways to say certain statements. When looking for a probability in the dictionary, a pure zero in the probability would heavily skew the results. Smoothing would fill the zero results with probabilities that are slightly above zero, in order to avoid zero probability results. A simple example is the add-one smoothing, where every zero probability has 1 added into it.

Language models can be used for text generation by finding common N-grams associated with the beginning word and adding onto the beginning word. The N-gram can be created until the final statement has been created. In one instance, when statements start with “How are”, there is a high probability that the statement can end in “you?”. The text generated can complete the statement.

Language models can be evaluated by having human annotators evaluate the results of the language model with certain predefined metrics. Text generated by a language model can be graded on accuracy for text classification or machine translation.

Google has an n-gram viewer available on their website - https://books.google.com/ngrams/. Users can type their n-grams in the search bar on top of the website. Then, the user can view the frequency of the phrase across different languages and the usage of the phrase over time. In the following screenshot, the unigram “Alejo” is used to see the usage of the word “Alejo” from 1800 to 2019.