N-grams Project Narrative

By: Harshith Nanda and Charith Muppidi

An N-gram is a contiguous sequence of n words from a text, where the meaning of the n-gram changes with the addition of new words or descriptors. N-grams are useful for building language models primarily due to their statistical properties, in that in an N-gram model, the probability of next word based on given previous words follows a distribution. As a result, an N-gram model can be used to predict the next given word and glean more meaning from a given text, which greatly simplifies text processing. N-grams are fairly widely used, such as in applications like speech processing (where phonemes are modeled using an N-gram distribution) and language identification (like our current assignment, where N-grams are modeled for each language).

The probability of a bigram is calculated by dividing the number of times a bigram appears in a text by the number of times the first word in a bigram appears in a text plus the total vocabulary of the text. The source text is important in building a language model because it provides the data the model will be trained on. If the source text isn’t sufficiently large or representative of the language as a whole, it may skew the language model, leading to inaccurate results when applying the model to the real world.

Smoothing is necessary in N-gram modelling because of an imbalance in the weightage of N-grams that differ in frequency, especially to N-grams not seen in training data, which are otherwise automatically assigned a value of zero. Smoothing the probability distribution by assigning non-zero values to the weights of N-grams not seen in training data helps make the model more accurate by reducing bias, since one of the reasons why an N-gram has gone un-seen could be due to the bias of the training corpus. Some models for smoothing, like laplace, simply assign unseen N-grams a value of 1, whereas more sophisticated models like GT discounting accomplish this via a more complicated statistical technique to predict the probability of unseen N-grams based on already seen N-grams. N-gram language models can even be used to generate new text using the statistical properties of N-grams (the probability of the next word can be modeled via a distribution), but this is limited in scope by the accuracy of the model. The model also can’t generate any sentences with words it has not seen before, which is an inherent limiting factor, so a text generation model requires a massive corpus with as many words as possible.

Language models can be evaluated via several different metrics, like cross entropy, perplexity, word error rate, etc. Perplexity is a numerical value computed for each word based on the underlying probability distribution of words in the language as a whole. The word error rate is a general measure of accuracy of a given outputs from the language model. Cross entropy looks at the amount of information conveyed per word in output, and compares it to the average amount of information conveyed by the language as a whole, which measures the effectiveness of a model relative to the language.

Google’s N-gram viewer is an application that displays the relative frequency of a given N-gram and how it changes over time. One example would be: 