Project 4 - Narrative

What are N-Grams and how are they used to build a language model

Ngrams focus on N elements on a text. For example a unigram is an N-gram with N=1 and would just be each word in a given text, and a bigram would be an N-gram with N=2 and would be each sequential pair of words in a given text. These are useful for building language models because we can calculate the probability and likelihood of words being followed by other words and phrases.

List a few applications where N-grams could be used

Since Ngrams focus on the probability of a sequence of words it would be good for a number of NLP applications. One application could be classification based on training data to determine language like we did in the project. Ngrams are good at computing subsequent words and phrases so another good application would be autocorrect and in that case it could be trained on the user's most frequent word choice and phrasing.

How are probabilities calculated for Unigrams and Bigrams

For Unigrams since it's just the probability of a single word occurring in text it would be:

$$P(Word)/N$$

Where P(Word) is the number of times the word occurs in the text, and N is the number of words in the text.

For Bigrams it's similar, but calculates the probability of the two words in sequence and mathematically would look like:

$$P(W1W2) = P(W2)P(W2|W1) = \frac{C(W1)}{\sum_n C(Wn)} \frac{C(W1W2)}{C(W1)}$$

Where the first word(W1) is followed by a second word (W2). This follows the probability of the second word occurring and the probability of word 2 occurring after word 1.

What is the importance of the source text in building a language model

The source text is extremely important in building a language model, since it's the source of training the model to work effectively. If the source text is poor in quality or quantity of items the resulting language model may be inaccurate and produce incorrect results. For N-grams, if the source text has bigrams that are only occur in the source model but don't make sense that would be reflective when it's utilized.

Why is smoothing important and a simple approach

Smoothing is important because when we're building N-grams it's impossible for every combination of words to exist in our models. Smoothing fills in the blanks with weight so it's not completely lost in the probability calculations. A simple approach to this would be LaPlace smoothing which follows:

$$P(w_i) = \frac{C(w_i)}{N} \implies P(w_i) = \frac{C(w_i) + 1}{N + V}$$

Where the former equation is before smoothing and latter is after. By adding 1 and dividing by the vocabulary size it ensures missing combinations aren't completely left out from probability calculations.

Language Models for Text Generations & Limitations

Because N-grams calculate the probability of certain phrases or words followed by other certain phrases and words they can be used for text generation. This works by generating words and following them with words and phrases that are most likely to follow. Because this is a probabilistic approach, it works better with higher N-grams and needs a large corpus of data to be effective.

How can Language Models be Evaluated

Language models can be evaluated by an external source such as human annotators or testing. They can also be measured by numeric metrics like accuracy of classification on certain test data, or their perplexity score which is the inverse probability of encountering the words normalized by the number of words.

Google's N-Gram viewer

Google's N-gram viewer displays the occurrence of certain phrases or words in a corpus of books over the years. Here's an example of a few phrases and their frequency in literary works over the years.