Brendan Lim

CS4395.001

March 4, 2023

Assignment 4: N-grams

N-grams are a continuous sequence of n number of words from a given text. For example, if you were to convert this sentence into a list of bigrams, the first bigram would be “For example”. They can be used to form a probabilistic model by multiplying the probability of the first token multiplied by the probability of the second token showing up after the first multiplied by the third token showing up after the first two tokens, etc. The calculated probability is can then be used to generate the next word in a sentence like word predictions on your phone or determine how similar a body of text is to your training corpus to detect plagiarism.

The training corpus is the most important part of building the model because the probabilities of all the n-grams are determined by the text. For example, if you wanted your model to generate formal language, then you might want to have a training corpus of public addresses or speeches.

Earlier I outlined calculating the probability of each n-gram. However a corpus cannot have every single n-gram ever, even if the n-gram shows up occasionally in normal speech. Because of this, any n-gram that does not show up in the training corpus will just not exist to the n-gram model. To avoid this, we use different equation which ensures that any possible n-gram will have a tiny probability of existing. This process is called smoothing.

Another limitation of using n-gram models is that the generated text doesn’t make sense and doesn’t have any overarching ideas in it because it is just pure probability.

There are two ways of determining how good a given language model is. There is extrinsic evaluation, where how good the output is determined by humans, and intrinsic output, which is determined by internal metrics. A commonly used internal metric is perplexity, which is the inverse of the probability of seeing words from the text corpus. The ideal perplexity is as low as possible, since we want a high probability of generated text being from the training corpus.

The Google Books Ngram Viewer is an online tool where someone can search any given sequence of words and it will show the n-gram’s probability/frequency over time in Google’s entire corpus of books from 1800 to 2019. The below example shows how smoothing can be wrong. Clearly “google” wasn’t a word in the 1800s (“googol” does not count), but it is shown in the graph as existing ever so minutely.

Chart

Description automatically generated with medium confidence

Example use of Google Books Ngram Viewer