a. N-grams are a way to represent sequences of words in chunks of “n” words. For example, 2-grams could include “New York”, “Ariana Grande”, etc. N-grams are helpful for building language models because they can be used to predict words based on occurrences of a previous word. For example, if we have a sentence like “Raise your”, an n-gram model could predict that the next word would be “hands”. A model is trained on large corpuses, and modeling the probabilities of words being associated together can create a language model that is able to generate associated words.
b. Applications of n-grams in NLP include text generation, summarization, translation, language detection, and other text analysis tasks.
c. For bigram models, the probability can be calculated
d. The quality and quantity of the corpus that the model is trained on can directly affect the effectiveness of the model. Having more data will help lower the influence of outliers in the language affecting the model, and having well annotated and representative data for one’s task are important steps before training.
e. When we train models on test data, it may or may not contain the words or data we intend to actually evaluate the model on. Smoothing algorithms can help increase accuracy by smoothing out the effects of zero probability n grams on the model so that lower probability words are assigned reasonable probabilities of actually occurring.
f. Text generation can be done using language models by predicting what words may come next based on the association of previous words or mapping relationships between texts. A text generation model can generate code, complete a story or requirement given a few sentences, and translation tasks. One big limitation of text generation models is the data that it is being trained on. For example, ChatGPT is limited to data before September 2021, and it is not able to complete text generation tasks on data after 2021.
g. There are extrinsic and intrinsic ways of evaluating a language model. Extrinsic ways include comparing model performance to human annotated data. An intrinsic way would be using the perplexity probability, which calculates how complicated/perplex a given text is using the inverse probability of the test text and normalized by the word count of the test text. Lower perplexity means the text is more similar to how a human would write a text.
h. The google n-gram viewer shows the frequencies of any sequences of words from printed sources. For example, searching “coronavirus” from 1800-2019 shows a near 0% occurrence from 1800-1999, and starts peaking upwards from 2000 onwards, most likely because of the discovery of the first strains of the coronavirus.