Lisa Chen

Homework 8 Narrative

N-grams is a connected string that acts as a sliding window over text, that takes in a number of words at a time, where the number of words taken in determine what the n-gram is called. For example, if at least 3 words are taken in at a time, then n-grams are called n-grams, where n is specified as the number of words taken in or if one word is taken at a time then it’s called a unigram. How n-grams are used to build a language model consist of the n-gram taking a corpus, as in a body of text, and then using it to predict upcoming terms or words by predicting the probability of a given n-gram within the given corpus through probability calculations. Usually, start and stop symbols are added to the beginning and end of each sentence in the corpus. The language model you end up with will depend on what kind of corpus the n-gram is used, as n-grams are a probabilistic model.

A few applications where n-grams could be used include being used as a probabilistic model of language, language generation, correcting spelling, as a feature in a program that automatically suggests what word(s) to use next, such as for messages, auto completion of searches, speech recognition, and machine translation of texts.

For unigrams their probabilities is calculated by taking the count of the specified unigram divided by the total amount of unigrams. For bigram, their probabilities are calculated by how many times the specified bigram happens divided by how many bigrams there are.

When it comes to the source text in a building a language model, it’s importance includes how the choice of source text influences the language model. The kind of model you get will be change depending on what kind of source text is used. Models based on different source texts will usually look different from other models based on other types of source texts. This in turn can effect the accuracy and precision of the model.

Smoothing is important because is that it fills in the 0 values with some probability of the overall mass to make the distribution smoother and ensure the probability doesn't end up being zeroed out and deals with the sparsity problem. This is the case because when you're working with data, it's not possible for it to have every sequence of every type of possible words. One simple approach to smoothing is called laplace smoothing, also known as add-one smoothing. The approach involves adding 1 to the 0 count for it to not be 0 and that ahead of time, 1 is added to all counts, where the total vocabulary count is added to the denominator to balance it out. However, this tends to adjust the probabilities a little too aggressively.

Language models can be used for text generation by using them to take a start word and then look through the prepared bigram probabilities etc to find the bigram etc that has the start word in the 1st position that has the highest probability where this is then concatenated to the phrase it generates. All of this is continued until the last token added is a period. However, the limitations of this approach is of its small size and simple approach that vary from the type of n-gram used or if nltk's generate function is used. In addition, there is the amount of time it takes for the language to happen.

Language models can be evaluated by doing a extrinsic evaluation through the usage of human annotators doing the evaluation of the results using a predefined metric. Another way of evaluating language models is intrinsic evaluation that uses a metric to compare the different models where sometimes the metric used is perplexity.

Google’s n-gram viewer is takes the user enters and displays a graph that shows how the entered phrases have occurred in a corpus of books over the selected duration of time, as in years. When you hover over the graph using the mouse it highlights the selected line plot and shows the percentaage for each line plot present on the graph. What data is displayed changes how you select the graph in order to look at a specific n-gram. In addition, you can change the type of language you want to look for, the case-insensitivity of the phrase, and the smoothing of it. One example of this can be seen below:

Graphical user interface, chart, application

Description automatically generated