November 11, 2025
Lab 6: Menu Review Text Classifier

Program Usage:

Start the program by running it directly from your terminal. From there, it will prompt you to give your own review for a restaurant to test with the algorithm. When you have finished typing your review, press enter. Finally, the program will give its predicted tone for your review using two different models, either positive, negative, or neutral.

Testing and Training:

For our dataset, we decided to manually hardcode in 10 sentences for the positive, negative, and neutral tones. The total size of our dataset used was 30 sentences. After this, our extract_features function converts all of these sentences into a bag of words. Therefore, to change the dataset, all that is needed is changing these example sentences and running extract_features again. The training section of our code uses an 80/20 split, with the training set using the first 8 sentences from each type of tone, and the testing set using the last 2. This makes sure that the model is tested on new data all the time, not just memorizing already seen sentences.

Situational Differences:

Our classifier seems to work best when noticing keywords or vocabulary it is directly familiar with, i.e. the ones within our example sentences. For example, words like “love”, “perfect”, and “exceptional” are all strongly linked with positivity, since they all appear in positive example sentences, so using any one of these within your own review will help the algorithm determine that it is positive. The same is true for negative and neutral sentences. Since this classifier is so good with keywords, the size of the review doesn’t seem to have much difference as well, with short reviews including keywords still being recognized well.

However, these strengths also go the other way with its weaknesses. If a given sentence has new vocabulary, the classifier has no data to go off of from its data set, and therefore struggles. Other easy mistakes include compound sentences, where both negative and positive words are used, as well as making positive words negative like in “not delicious”. Since these tests are hard for our classifier, it often defaults to neutral for these examples, but still shows which sections it is sensing more during the results section.