May 27th, 2018

Lab 4 Report

Devon Martin and Shiv Sulkar

Experiments

The experiment you ran for each of the two analytical methods for determining authorship: what datasets were explored, which values of hyperparameters were investigated.

All algorithms were implemented in Python 3.5.2 using built-in modules, as well as one external package, numpy.

K-Nearest Neighbors

To determine authorship using the KKN algorithm, we used the standard cosine similarity metric when comparing queries to documents. We experimented with using different values of k, as well with smaller subsets of the larger dataset.

Clustering

We used single linkage clustering to determine when to merge clusters. This algorithm was helpful in understand which leaves were closely related. In addition we limited the number of merges until there were 50 clusters. This left leaves that had large amounts of data.

Results

The results of your best runs for each of the two analytical methods. You should present in tabular (or easy to read text) form the accuracy measures for determining each author, as well as the overall accuracy.

Unfortunately, we were unable to complete the full implementation of the KNN algorithm, and have only completed basic querying. There are no metrics available for accuracy, recall, etc. See "Personal Reflection" for an explanation.

What we have completed is as follows:

- Text Vectorization: ability to generate tf-idf table of all files passed in through command line. Caches all intermediate tables for reuse in KNN and Clustering
- KNN: ability to handle single queries with any size k. Compares query to documents using cosine similarity.

- Clustering: Our clustering algorithm is able to undo the last 50 merges and output them into an xml file.

Personal Reflection

The vectorization of the text files was a real challenge, and unfortunately we made a design decision early on that later cost us this lab. We focused on building the table of tf-idf values for the documents, but designed it in a way that made it impossible to handle queries. We didn't realize that queries are handled exactly the same way as an individual document would be, and so when it came time to implement KNN, we had to rework the text vectorization first. After 6+ hours of refactoring, by the time we had finished rewriting all of that code, the lab was due, and we still had a lot left to implement for KNN.

At this point, we have put 20+ hours total into the lab, and have no results to show for it. We are at a point where we don't know if it is worth it to finish this lab, and have decided to focus our remaining efforts on the team project. We should have had a stronger understanding of the algorithm before starting to code, and are disappointed that we didn't finish. Going forward, we will have a more detailed implementation plan before we start coding to save hours of refactoring later.