CS 6140: Data Collection Report

Brian Kimmig, Jesus Zarate

We obtained the base of our data from a dataset published on Kaggle. The dataset contained information on ~ 5000 movies. We essentially used this dataset for the movie list and the IMDB IDs. From the IMDB IDs we were able to perform requests to the OMDB API where we compiled the title, plot summary, and genres for all ~ 5000 movies. From there we stored them in a JSON file, with each entry containing the fields ['title', 'plot', 'genres']. We may need/want to add more movies to this data set as our analysis progresses.

The general JSON data file is about 2.3 MB. This is the raw data, we will be processing the data to create features that can then be used for classification. Creating the features is a main chunk of the project, but with as many film synopses as we have (~ 5000) we expect our feature matrices to be large. The size of the data will be largely dependent on our k-gram methods - obviously some will produce more rows than others. As a first pass we used the TFIDF vectorizor in sklearn and broke it on words, giving us a matrix that is about (~ 25000) rows by (~ 5000) columns, in sparse form, where rows represent the word and the column represents the movie. For the genres of the movies, we will use a one-hot encoding.

We are currently storing our data in a JSON file. It contains an entry for each movie and within each entry we have 3 descriptors of the movie – title, plot and genres – essentially giving the file 3 columns. As processing continues we will store the feature matrices we create in separate files, assuming they are not quick to compute. We will create matrices from both the plot (k-gram matrices), and the genres. The genre matrix will be a one-hot encoding of the genres and can be constant for every k-gram matrix unless we decide to limit and/or change the genres.

We aim to test a number of processing methods within the k-gram universe, from single words to combinations of words. Most of our matrices will be represented using one-hot encoding. The processing of this data will come as part of our 'intermediate report'. We will also be looking into the TFIDF vectorizor in sklearn for comparison.

One way to do this would be with Markov Chains. We build probability distributions over all of our words and subsequently the words that follow them. This would allow us to create a new document by selecting a word and sampling from the distribution of words that follow. However, in the case of this type of data it seems that this would be slightly unnecessary as there is an abundance of text we could just use for documents.