Progress Report on Debate Stance Clustering

Daniel Fagerlie
Sam Shinn
Nicolas Salas

Completed:

We have written the overall framework to support different feature vector types so exploration of feature buildup can be easily done. The program currently reads the data, builds basic feature vectors, clusters them with kmeans, and gives a raw evaluation.

The corpus is read into memory with minimal change to the text. Each post is broken down into a data structure containing the stance (to be used later for evaluation) and message. The message is merely a string with all newline and multiple whitespace characters flattened down into a single space. This raw storage of the text was chosen so we could easily preprocess the data however is appropriate for the type of features selected (ie. trying out word_tokenize vs. tweet_tokenize).

The feature vectors can currently be based on a very simple Bag-Of-Words (BOW) model or a TF-IDF model. The BOW model was designed mostly just to get something working to test our system. It does remove punctuations, but does nothing about words like “the”, “and”, “is”, etc.. But, the feature vector is composed like so: <word1, word2, … word_topN>. No normalization was done. The results are pretty gruesome, but this isn’t our end goal.

As an additional preliminary exploration, we implemented a TF-IDF version. This does eliminate overly common words like “the”, “and”, and “is”, but surprisingly yielded even poorer-looking clustering results. We aren’t sure why this is, but some intuition points to perhaps oversights in the clustering algorithm (discussed below), or it could be that clustering based on TF-IDF highlights differences in the discussion which we are not actually interested in (such as on-topic vs. off-topic discussion).

ToDo:

We need to finish up the evaluation portion. The current idea is to make some sort of polarity score that represents how well clustered the data is. If a table is created with the raw counts some attributes can be determined as to whether the outcome is well grouped.

After all that, the most interesting portion is the feature exploration. We are looking at sentiment, POS (for identifying keywords), emotion, and word similarity identifications. We will try out some of these using pre existing implementations and report their effectiveness.

Also, after playing around a little bit and seeing how poorly the kmeans worked on tf-idf and bag-of-words, we might try other clustering methods, just to see what happens. We’ll probably stay within the sklearn library since they all seem to have common usage patterns, so switching them out shouldn’t be much more than a change in one line of code.