Daniel Temkin

Project Report

Objective:

The issue is how to balance public trust with original/polarizing content. Major news organizations and personal bloggers alike struggle with this dichotomy. After all, there are two primary ways to generate interest in your news as opposed to another, either you must say something or find a story that no one has said or found before. Or, as is usually the case take an existing story and say something about it that others have not said yet. The resulting perturbations can be unsightly to say the least especially as late-comers increase the severity of their extremism to regenerate interest. However, that is not to say that some organizations do not try and minimize the degree of editorializing in their published materials especially when it does not add value to the story. Moreover, I would venture to say that there is not a single news organization, save for Associated Press maybe, that does not have some editorializing in even the most “informative” pieces. The question that remains is, whether it is possible to identify any random internet post pulled from an RSS feed as a blog or not. Of course, we also only want to explore methods that do not require reviewing the full article, because what would be the point. In some instances it is easy to discern from a news headline, short summary, or even the tags, whether a story contains a sufficient degree of editorializing to be deemed an editorial or blog. However, more often than not, there are no unified set of variables included with each RSS post. As a result, we must also add feature discovery and generation to our list of objectives. In summary, it is the primary objective of this project to discover a unifying set of quantifiable features that can be derived from the small set of shared information fields; and apply that feature set in an algorithm to determine its efficacy in predicting the probability that a random document is, in fact, from a biased news source.

Methodology:

First I compiled a list of feeds from various sources that were either of personal interest or were part of a third-party aggregate list. While adding feeds, I used a subjective labeling of feeds, denoting them “blog” or “not blog”. Then using feedparser iteratively over the web addresses associated with each feed I was able to collect a master set of entries. In each of the entries there was a summary field which contained a brief excerpt from the article. At which point, I cleaned the summary text, used spaCy to tokenize each summary into individual words, removed stop words and artifacts left over from original cleaning, and finally applied a word stemmer to each of the tokens. The stemmed tokens for each individual entry were converted into bags-of-words (bows) and used as input in an LDA model which was parametrized to look for a total of 10 topics. The topic probabilities then output for each document were all normalized and compared against the boolean value ‘is\_blog’ which each entry inherited from its respective feed, using both a logistic regression (LR) and a decision tree classifier (DT) for comparative purposes.

Results:

Logistic Regression

*Validation Report:*

precision recall f1-score support

Not Blog 0.8243 1.0000 0.9037 1009

Blog 0.0000 0.0000 0.0000 215

avg / total 0.6795 0.8243 0.7450 1224

*Test Report:*

precision recall f1-score support

Not Blog 0.8121 1.0000 0.8963 994

Blog 0.0000 0.0000 0.0000 230

avg / total 0.6595 0.8121 0.7279 1224

Decision Tree Classification

*Validation Report:*

precision recall f1-score support

Not Blog 0.8348 0.8513 0.8430 1009

Blog 0.2308 0.2093 0.2195 215

avg / total 0.7287 0.7386 0.7335 1224

*Test Report:*

precision recall f1-score support

Not Blog 0.8281 0.8431 0.8355 994

Blog 0.2642 0.2435 0.2534 230

avg / total 0.7221 0.7304 0.7261 1224

Interpretation:

This disparity in precision and recall between classes could be explained by a variety of factors. First, the difference between blogs and news might not be so significant, which would result in a difficult to identify class. Although, this could also be the result of the biased ‘is\_blog’ attribute assignment at the onset. In other words, I may have missed what should have been considered a blog or not. Second, the number of total blog entries was around 1000 while the number of total news entries was closer to 4000. This disparity is evident in the support values above. This introduces significant bias in the model training which obviously carries over into the testing and validation sets. One potential method for combating this issue is providing the model with a set expected class weights equal to the proportion of total entries by class (.25 for blogs). The data set could have also been truncated at the onset so that they are more equivalently weight.

Ultimately, we are left with the conclusion that we should use the decision tree classification model (because it provides some explanation for the blog class) but we can be confident that the model can find items that are not blogs strictly from the probabilities determined using an LDA topic model and the summary text.