Using Naive Bayes to Determine Political Bias in News Articles

Colin Schimmelfing
Department of Computer Science
Swarthmore College
Swarthmore, PA 19081
cschimm1@swarthmore.edu

Matthew Baldwin
Department of Computer Science
Swarthmore College
Swarthmore, PA 19081
mbaldwi1@swarthmore.edu

Abstract

This report details an approach to automated bias determination of news sources. Given a politically conservative source and a politically liberal source, this approach can determine the bias of any new source relative to these sources. The process involves querying Google News for articles on a particular subject and then applying the Naive Bayes algorithm to each article from a source to determine overall source bias. We find this problem to be much more difficult than previously estimated, due to the small sample size of sources and the difficulty to access all or even part of many articles.

1 Introduction

An important application of Natural Language Processing is the determination of bias for a source or document. Using a computer to determine bias can give humans valuable information without the cost of reading documents. When there are many differing opinions on a topic or many documents from a source, the cost of reading can be prohibitively high. This occurs in the context of product reviews (Dave et al., 2003), movie reviews (Pang et al., 2002), and published works. It is especially crucial to determine the bias, if there is one, when readers assume that a source is unbiased. This pertains to our topic, which is determining the bias of news sources based on a large selection of articles from the source. Using our program, a human wishing to read news from a truly neutral source does not have to review large samples of documents from all sources, and can be confident that the coverage is fair and balanced. Additionally, if a human wishes to read articles from sources that conform to their own political leanings, this program can aide them as well, but the creators of this program would advise such a human to broaden their horizons.

Unfortunately, we cannot put to rest the age-old debate about the presence of a liberal bias in American media. There is no objective measure of bias-everyone has a different definition of what the "liberal" and "conservative" mean. Fox News, which even many conservatives consider biased, calls itself "Fair and Balanced". Similarly, many who read the New York Times believe that what they read is objective, when again even many liberals concede a bias. Thus, our measure only can distinguish relative bias. Using a map of political bias to news source from a recently published analysis of media bias (Groseclose and Milyo, 1974), we see the spectrum of media bias. While these results are not exactly correct (the paper has serious issues), the general ordering probably would be found by anyone to be the same. The Wall Street Journal is the most liberal, with the CBS Evening News and the New York Times following. The Washington Times was found to be the most conservative, with Fox News close behind. We can use these two most biased sources as bases on which to measure the biases of other news sources, and at least give some relative measure of news source bias.

2 Literature Review

There is a large body of literature on bias classification. There are several different methods to determine bias, including SVM, Naive Bayes, and Maximum Entropy Classification. Since our project was time-constrained, we chose to look primarily at Naive Bayes treatments, although most papers used more than one approach.

The paper that most interested us was "Which Side are You on? Identifying Perspectives at the Document and Sentence Levels" (Lin et al., 2006). These authors chose to look at a corpus of editorials on the Israeli-Palestinian conflict, in which articles were written from the perspective of Israelis or Palestinians. There is an additional complication that some articles were written by editors and some by guests, and depending on which set was training and which was testing there were different results. Since this is a very polarizing topic, it is not surprising that the worst their algorithm achieved was 81% accuracy in determining whether a particular article had been written by an Israeli or a Palestinian. This worst performance was from SVM, as compared to 84% and 85% for the two different Naive Bayes models, a factor that influenced this project's use of the Naive Bayes algorithm. Their research focused on first determining if any given sentence was biased, then determining that bias, then finally adding up all biased sentences in a document to determine the perspective of the document. This differs from our research in that we tally up the total number of biased documents to determine the bias of the source, instead of looking as low-level as sentence bias.

Mining the Peanut Gallery: Opinion Extraction and Semantic Classication of Product Reviews (Dave et al., 2003) is a paper that shows a direction that we would like to expand our research on in the future. This research consisted of reading online product reviews and attempting to automatically determine if the review was positive or negative. While this group used the Naive Bayes algorithm as well, they also use WSD (using WordNet) to equate tokens such as "not good" to"bad" in order to improve the accuracy of the NB. They also looked into SVM, but found the method less accurate, more evidence pushing this research towards using NB, in agreement with (Lin et al., 2006) but against (Joachims, 1997) and (Pang et al., 2002). What would be interesting for us is to also use WordNet to try to figure out if certain terms (such as "rights" or "force") have slightly different meanings in conservative documents as compared to liberal ones. It is unlikely that there is enough precision in WordNet to implement this, however.

Similar to (Dave et al., 2003), Thumbs up? Sentiment Classication using Machine Learning Techniques (Pang et al., 2002) looks at movie reviews to determine a positive or negative review. These authors used Naive Bayes, maximum entropy classication, and SVM, and found SVM performed the best, unlike (Lin et al., 2006) and (Dave et al., 2003) As a baseline, they used features of words chosen by graduate students to see if the words students chose could determine the reviewer's rating of a movie. Interestingly, the students achieved very poor accuracy. The cause of the inaccuracy is closely related to our problem regarding quotations, and while we took no steps to alleviate the possibility of error, it was informative to analyze a similar situation.

Bias classification has been done using non-machine techniques as well. There are many ways to try to define what makes a conservative and what makes a liberal, and all are incomplete, due to the spectrum-like distribution of bias. Most come to the same distribution (Groseclose and Milyo, 1974), as showed in table 1. Unfortunately, most seem to come to this distribution by either citing this "working paper" which has serious flaws or by sheer invention. It is incredibly difficult to find good research on this topic; most sources are merely "spin" on the topic. This may cause error in the calculations (if there is no bias, it would be very difficult to detect it with a machine!), but there is no alternative. The paper we looked at rated newspapers based on favorable citations of "think tanks". If a source rated liberal think tanks more than conservative, it must be liberal. This side-steps the issue of the bias of the think tanks themselves, but since it is the only decent source, and by human inspection we can see that it - at least coarsely - holds (the New York Times is easily more liberal than CNN, which is easily more liberal than Fox), we accept their distribution.

3 Algorithms and methods

Our code begins with two bash scripts, `make_trainers` and `find_bias`, which access sequentially the python code and `Lynx`. The `make_trainers` program creates the data for the Naive Bayes algorithm to train on, while the `find_bias` determines the bias for a given source based on the training data. The important difference between them is that `make_trainers` never runs the Naive Bayes program.

The execution of `find_bias` is as follows: In the first stage, the script sends the HTML code from a Google News search to `CutHtml.py`. `CutHtml.py` extracts the relevant web addresses from the page and outputs them into a text document. This file is then sent through lynx to capture the text from each page and appends all of this into a new text document. `CutArticles.py` then takes this document and removes all or most on the irrelevant data that is also extracted and puts the relevant data in a file of articles from that source. The most conservative and liberal sources have been saved into text files as training data, and are input along with the file of articles from the testing source into bayes.py. `Bayes.py` runs a Naive Bayes comparison between the two sets of training data and the testing data to determine which bias the source favors. The implementation of Naive Bayes is of the utmost concern, so the other steps will only be discussed in passing.

$$\hat{s} = \text{argmax}_{s \in S} \frac{P(\overrightarrow{f}|s)P(s)}{P(\overrightarrow{f})} \tag{1}$$

`Bayes.py` begins with the implementation of equation 1, on the document level. The $\hat{s}$ in this initial equation is the determined bias in the document. The second half of the Bayes equation claims that the $\hat{s}$, the final bias, is the the sense that maximizes the probability that a specific feature vector occurs given the sense times the probability that that sense occurs divided by the probability that that feature vector occurs. In our implementation, the probability that a feature occurs is going to be 1 since they should occur with equal probability in a given article.

$$P(\overrightarrow{f}|s) \simeq \prod_{j=1}^{n} P(f_j|s) \tag{2}$$

The probability that a feature vector occurs given a sense, is approximated in equation 2. Here, we claim that the overall probability is approximately all of the individual feature's probabilities given the senses multiplied together. Since we are only using two features, our two-dimensional feature vector is fairly simple. The basis for each of these individual probabilities is displayed below in equation 3. Our first feature looked at the training data and determined the overall probability that any word occurred in the corpus and compared this with the probability that a given word occurred in the testing article. By looking at the common probabilities, the degree which the article favored one bias or another could be determined. Our second feature looked for the word distribution around different forms of "America" for the training and testing groups. This second feature worked less reliably than the first since many articles had not used "America".

$$P(f_j|s) = \frac{count(f_j, s)}{count(s)} \tag{3}$$

Finally, plugging in the above equations and setting the probability for a given vector equal to 1, we get equation 4.

$$\hat{s} = \text{argmax}_{s \in S} P(s) \prod_{j=1}^{n} P(f_j|s) \tag{4}$$

We decided to take P(s) to be .5 since equating the number articles in the training data was more work than it was worth when we were assuming that the probabilities were equal to begin with. One possibility for taking the project to the next level in mentioned in section 5, where we determine the overall web bias to get an accurate value of P(s).

When it comes to scoring, we tried to follow the the "UCLA score" format that (Groseclose and Milyo, 1974) uses. A score of 0 indicates a source is as conservative as the training source, while a score of 100 indicates a source is as liberal as the liberal training source. Of course, a score of 50 would then indicate that the source is halfway in between, in

respect to bias. The UCLA rating was normalized to allow comparison with our data, which is relative to the most conservative and liberal sources, whereas the original UCLA ratings were relative to the most liberal and conservative congressional representatives.

Originally, we had intended to use most of the largest domestic news sources, but several were inadequate. Newsweek only had half of a page of an article available to subscribers, if at all, while Swarthmore College does not have a subscription to the Wall Street Journal. NBC's online presence is considerable, but the fact that any NBC links older than six months are unlikely to work resulted in removal of NBC from our sources. In the future, we would like to increase the scope to include more sources. In the data, only the Washington Post, Time Magazine, ABC, and CNN are truly independent of the training. We found that editorial or regional differences were causing erratic results, and while this decreases our true testing size, it was necessary to train on both the New York Times and CBS, as well as on both Fox News and the Washington Times.

4 Results and analysis

We ran the programs for five different topics: "Iraq War", "Iran", "Global Warming", "Abortion", and "Amber Alert", the results of which are in table 1. The liberal training corpus consisted of data from the New York Times and CBS, while the conservative training corpus consisted of data from the Washington Times and Fox News.

While the relative order is generally the same, oddly enough, only the "Global Warming" and "Amber Alert" queries resulted in ratings throughout the whole range between liberal and conservative. "Iraq War" and "Abortion" resulted in all sources being labeled liberal, while "Iran" caused most sources to be labeled conservative. It is possible that these shifts are caused by one base source using similar terminology as the two on the other end of the political spectrum. For example, if the Washington Times called the Iraq War "The Second Gulf War", while Fox called it the "Iraq War" along with the rest of the news sources, the moderate sources would all be reported as more liberal than is the case. If then the same phenomenon occurred in the same query with a term that only Fox used "terrorists" instead of "insurgents", for example, the Washington Times would also be shifted towards the liberal end of the spectrum.

Even without a consistent center, we found that there is a general trend in the distribution of bias, in accordance with the distribution widely accepted.

5 Future work

Although we were able to identify bias in many of the articles, the accuracy that we were able to attain was significantly lower than that in several of the studies that we looked at. Our study only implements two feature vectors that neither rely on nltk nor WordNet. Several attempts were made to establish more complex feature vectors but the difficulties of implementing reliable code using the databases stalled development. Our existing "America" feature detector framework could yield additional features if we applied it to other words that might be surrounded by bias. Often times biased verbs are not included or concealed in either of the existing features, by utilizing verb identification we could approach analyzing bias on the sentence level rather than relying on article averages.

The current implementation only works for domestic news sources- foreign news sources have too many word choice and syntax differences to be accurately modeled by our program. We would like to implement a check to determine if a source is foreign, and then compare these sources to different international sources. An offshoot of this project would be to include the domestic sources in this international bias calculator, to create a truly general system. This is especially pertinent because the most liberal domestic news sources are not the most liberal views available globally , as much as the most conservative domestic views are not the most conservative views available.

An overall survey of the news articles that Google News returns could reveal the general web bias and hence determine an accurate value for P(s) that is assumed to be .5 in bayes implementation, section 3. We would have to filter these results to acquire a basis for reputable news sources (Google News picks up everything, even the Daily Gazette!)

Source	UCLA rating	Iraq War	Iran	Global Warming	Abortion	Amber Alert
New York Times [1]	100	88	37	96	93	67
CBS [1]	100	65	32	82	91	50
Washington Post	81	62	8	79	86	54
Time	78	79	36	76	94	15
ABC	54	57	2	76	76	9
CNN	54	36	2	71	71	24
Fox [1]	10	35	0	60	68	12
Washington Times [1]	0	40	9	27	80	25

Table 1: Relative Bias Distribution

it should help the accuracy of our Naive Bayes algorithm.

Another means by which accuracy could improve is by looking only at sentences which have a good likelihood of being biased. This is related to the work of (Lin et al., 2006), and (Yu and Hatzivassiloglou, 2006) points out that 56% of non-editorial news articles contain "mostly factual" (i.e. non-biased) sentences. This implies that most of the sentences we reviewed were not actually biased, and contributed to the problem of style influencing the ratings. In the future, we would like to classify sentences based on whether we believe they contain bias, and obtain a baseline for how many neutral sentences there are in news documents. This will aide in classifying the centrist news sources.

6 Conclusions

The problem of political bias categorization is a difficult one. Few news sources admit to a bias, and there is even debate in America whether there really is a bias or not. This makes trying to classify news source bias much harder than other problems. In addition, political bias is not binary- there is a spectrum of classifications that are available.

Given the difficulty of the project, we are pleased to have re-created the trend of political bias that is generally accepted, albeit for a small testing size. To achieve more meaningful results, we would need more data. We could accomplish this by buying subscriptions or even expanding the scope of the project internationally, although this causes an increase in complexity.

While the problem of objective political bias is a difficult one for both humans and machines, classifying relative bias is possible with Naive Bayes. We expect to see more research done on this subject in the future, especially as there are many different interesting inquires yet to be done.

[1] These sources were part of the training set

References

Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: Opinion extraction and semantic classication of product reviews.

Tim Groseclose and Jeff Milyo. 1974. A measure of media bias. *UCLA-working paper*, 10:371–385.

Thorsten Joachims. 1997. Texy categorization with support vector machines: Learning with many relevant features. *Technical Report*, 8.

Wei-Hao Lin, Theresa Wilson, Janyce Wiebe, and Alexander Hauptmann. 2006. Which side are you on? identifying perspectives at the document and sentence levels. In *Proceedings of the Tenth Conference on Natural Language Learning*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Which side are you on? identifying perspectives at the document and sentence levels. In *Proceedings of the EMNLP*.

Hong Yu and Vasileios Hatzivassiloglou. 2006. Which side are you on? identifying perspectives at the document and sentence levels. In *Proceedings of the EMNLP*.