Text Mining to Find Interactions between Political Contributions and Political Speech

Lincoln Swaine-Moore (advised by Dragomir Radev)
Yale University
New Haven, CT 06520
lincoln.swaine-moore@yale.edu

Abstract

This project aims to demonstrate an approach for identifying conflicts of interest in political speech—that is, instances where politicians voice opinions (e.g. via Twitter, speech on the congressional floor, or simply voting on a bill) that are related to industries from which they have received donations. This involves training models to recognize text as related to a particular industry, then running those models on political speech, and highlighting instances where politicians speak about the top industries to have funded them. The primary goal of this project in this semester is to investigate whether it is possible to train these models to be effective, and to document the methodology of this investigation.

A large portion of the time spent on this project was dedicated to the data collection process. Several types of data were necessary: donations data, political speech on which to test the models, and—crucially—labeled training data. The last source proved the most difficult to gather.

Preliminary results from the model-training process suggest that it is possible to train models to categorize the training data fairly accurately. The trained models demonstrate some basic success identifying political speech (in the form of tweets) that is related to industries, though they are far from perfect. Filtering results using the collected donations data then permits identification of instances where politicians discuss subjects related to top donor industries.

These results suggest that with further refinement, it should be possible to reliably recognize political speech as related to particular industries, and, using these classifications alongside donation data, to identify potential conflicts of interest.

1 Introduction

Politicians frequently accept donations (directly or indirectly) from individuals or entities associated with particular industries. This allows donors access (again, directly or indirectly) to politicians' ears, and potentially gives priority to their agendas.

Projects like OpenSecrets (run by the Center for Responsive Politics) have collected information about these donations, and do an excellent job of making it readily available. Unfortunately, though, many individuals remain unaware of the ways in which donations affect the policies and speech of their representatives. This is likely due to the burden on the individual to seek out the information—that is, to learn about a politician's donors, one must first exhibit an impetus to learn this information. Because of this, many potential voters who might be interested in knowing relevant aspects of the donation profiles of their representatives do not ultimately acquire this information, though it is available.

Applications have attempted to resolve to resolve this issue by making donation information more accessible to users in the course of their regular browsing of the internet. Greenhouse, for instance, is an extension for web browsers that allows users to hover over the names of politicians present in text they are reading to see a listing of the industries from which they've received the most money. Greenhouse does an excellent job of making information about campaign finance more easily available and apparent.

However, this approach introduces a new problem: information overload. Because the names of politicians are highlighted in every instance in which they appear, the information the extension provides is often irrelevant to the context. Users reading about, say, Mitch McConnell's "tribute to the Louisville Metro Police officer who died in the line of duty this week", do not need to know that the industry from which he receives the most donations is that of Securities & Investment. Presenting this information every time his name shows up dilutes the resonance of the information, making users less likely to be paying attention in situations where data about donations would actually be useful.

A second drawback of the Greenhouse-style approach is that it focuses on third party mentions of politicians—in news articles and the like. However, the nature of discourse between politicians and the American people is evolving. Now more than ever, representatives have a direct line of communication to their constituents through services like Twitter and Facebook. In light of this dynamic, it is useful to understand the donations politicians receive in the context of their own speech and actions.

This project proposes a potential solution to both these problems—that is, an approach to providing information about political donations in a context-dependent manner that focuses on the direct action and speech of politicians.

More specifically, we undertake an analysis of the feasibility of designing a system that utilizes machine learning and natural language processing techniques to automatically classify political speech as related to various industries, and seeks instances in which speech about an industry coincides with significant donations from that industry. By way of clarification, political speech in this context means both literal speeches—oration in front of Congress, for example—and text related to the office of the politician (official Tweets, Facebook posts, press releases, etc.). Successfully creating such a system would be quite valuable as a means of automatically identifying political conflicts of interest.

2 Data Collection

Collecting data was a crucial component of this project, and proved the most time consumptive. There were three main types of data that I gathered.

2.1 Political Speech

The most obvious sort of information necessary for this project is the political speech itself. Several potential sources were investigated.

The most attractive source was an API formerly run by the Sunlight Foundation, and now formally under the aegis of ProPublica, known as the Capitol Words API. This API supposedly made access to the Congressional Record relatively easy, and likely could have been used to gather a corpus of political speech organized by politician. Unfortunately, it was down for the duration of the project as part of its organizational transfer.

Lacking that API, I investigated several other sources of data, and ultimately settled on Twitter. Without too much difficulty I found a list of the current Congress's Twitter handles. Originally, I undertook to use the Twitter Streaming API to gather a corpus of tweets, but a script located online (along with Twitter's relatively lenient rate limits) made it more productive to use the Twitter Search API to gather up to several thousand tweets made by each Twitter handle. Tweepy was an invaluable resource for this stage.

2.2 Donations

The second type of data necessary for the project was information about what sort of organizations donated the most money to each member of Congress. This data was the simplest to gather.

OpenSecrets provides API endpoints that can be used to query—for a given Member of Congress—what industries and what sectors (sectors being more general umbrella categories for industries) donated the most. The dataset containing Twitter handles also contained OpenSecrets IDs, so it was straightforward to query these endpoints repeatedly and gather the information. The only wrinkle was that the API was rate limited, so requests had to be staggered over several days.

2.3 Labeled Text

In order undertake the modeling stages of this project—which aimed to create classifiers to identify text as related to industries—it was necessary to acquire text documents labeled with related industries. This third type of data is the most subtle and was by far the most challenging to gather.

I considered several types of labeled data, but much of it was out of the question, because (for instance, in the case of the Twitter data I was considering) it would've required manual labeling of data, an undertaking that couldn't be justified in a project with this short a timespan.

Ultimately, at the suggestion of Professor Radev, I decided to pursue text data associated with companies. This solution was elegant because companies are already organized by industry (using Standard Industrial Codes, or SICs), so labels could be produced for a given document provided it was associated with a known company.

However, actually acquiring a list of a set of companies associated with SIC numbers, along with text data, proved more challenging than anticipated. I looked into a wide variety of data sources, before ultimately stumbling on a solution that pieced together several different websites' information. A list of publicly traded companies traded on the Nasdaq or the New York Stock Exchange was obtainable from Nasdaq itself, and companies were therein associated with their stock tickers. These stock tickers could be substituted into a url on the U.S. Securities and Exchange Commission's EDGAR search system to find their associated SIC number. And, the tickers could also be used similarly to find pages on Bloomberg, Google Finance, and Reuters that contained company descriptions. Combining this informational allowed for the collection of a corpus that contained text about companies and a corresponding SIC code. This SIC code could be mapped to OpenSecrets's own particular industrial classification using a mapping found online. Ultimately, after removing companies that did not successfully match to both an OpenSecrets industry code and at least one text document, there were about 3,600 companies with labeled text data. Ultimately, only Reuters descriptions were used for training, bringing the size of potential training data down to about 3,400 documents.

3 Language Modeling Work

The next step after gathering the relevant data as above was to begin the process of training natural language models.

3.1 Approach

Because the only labeled text data was not the same data as that to which the models would ultimately be applied, a particular approach was required.

More specifically, the goal is to produce a classifier that labels political speech with related industries, but the only labeled data (out of necessity) is company descriptions. Therefore, we need to first train models that predict the industry of company description. These models should have their accuracy evaluated also on the labeled company description data set (though, of course, only on examples not seen during training). These models can then be applied to political speech (here, tweets) and evaluated on how well they select tweets that are relevant to industries.

Another nuance to this task should be noted: while all the companies in the training data set have industries associated with them, not every tweet made by a politician should be understood to be affiliated with an industry. This means that models trained on the labeled data will attempt to classify any text they encounter into one of the known industry labels. To account for this change, the models used should allow for probabilistic predictions—that is, for a given piece of text, the model should output probabilities of its association with each label (that is, industry). When evaluating the model on the testing dataset (company descriptions not used for training), it is fine to understand the label with the highest probability to be the correct label. But when applying the model to the out of domain text, there should be a higher standard for labeling tweets as related to an industry. In particular, there should be a threshold, which if exceeded by the probability of a particular label, that label is assigned to the text. Note that depending on the choice of threshold (that is, as long as the threshold does not exceed 0.50), it is possible for multiple labels to be assigned to an out of domain text. This is acceptable because text outside of the training domain truly may concern multiple industries.

Another difficulty arises from the distribution of the labels of the training data. While OpenSecrets has 13 sector labels, and the training data has companies that fit into 9 of them, the distribution is not very even. The problem is more acute when looking at the industry labels. By my count there are 103 possibly industries in the OpenSecrets classification scheme, but the training data contains only 50. Worse, certain industries make up the bulk of the training data. Out of almost 3,400 samples, about 500 have the industry label "Pharmaceuticals/Health Products". This artificially inflates classification rates, and prevents successful learning of features that describe the less popular labels. Resampling techniques can perhaps address this, but only to some degree.

3.2 Pre-processing and Feature Extraction

Text was put through a pipeline in the process of training and testing.

Text was tokenized and tokens counted, with stopwords removed. Bigrams and lemmatization were tried, but actually detracted from the classification rates (and increased time to run models), so they were removed.

Term frequency-inverse document frequency (tv-idf) was then utilized to weight the features more appropriately. This improved the model for all model types except Multinomial Naïve Bayes.

3.3 Model Choice and Selection

The four primary models used were Multinomial Naïve Bayes (MNB), Bagging Classifier (using the default setting of an base classifier of a decision tree) (BDT), Linear Support Vector Machine (SVM), and Logistic Regression—all models from scikit-learn, and the latter two falling under the umbrella of stochastic gradient descent classifiers.

MNB, SVM, and Logistic Regression were selected as fairly standard choices of easy to train models. BDT classifier was chosen in an attempt to combat the issues arising from imbalanced labels, because the Bagging Classifier by default involves bootstrap resampling.

Recurrent Neural Nets (RNNs) were considered, but dismissed partially because of time constraints. However, it is unlikely that RNNs would've provided much lift over the other models given that their primary utility is in learning features from the sequential nature of input text. Since even simple sequential features such as bigrams didn't prove useful (likely because, as mentioned earlier, the primary classification value is in the vocabulary itself of a document), it is unlikely that RNNs would've been worth the time they would take to set up and train.

3.4 In-Domain Results

Task	*MNB*	*BDT*	*SVM*	*Logistic*
Sector	0.861	0.813	0.893	0.892
Industry	0.708	0.703	0.814	0.795
isPharma	0.985	0.961	0.984	0.980

Table 1: Classification rates for several models on several different tasks.

The table above reports the classification rates for the models mentioned above, as tested on labeled company descriptions, for several different sets of labels (that is, several different tasks). The tasks are: classifying company descriptions into OpenSecrets Sectors (the broadest categorization of companies), classifying company descriptions into OpenSecrets Industries (the more granular version of Sectors), and classifying company descriptions as either belong to the OpenSecrets Industry label "Pharmaceuticals/Health Products" or not. Several points are worth remarking upon.

First, the models perform better on the binary isPharma classification, than they do on the Sector classification, and better on the Sector classification than they do on the Industry classification. This is unsurprising because the number of possible classes increases from isPharma to Sector to Industry.

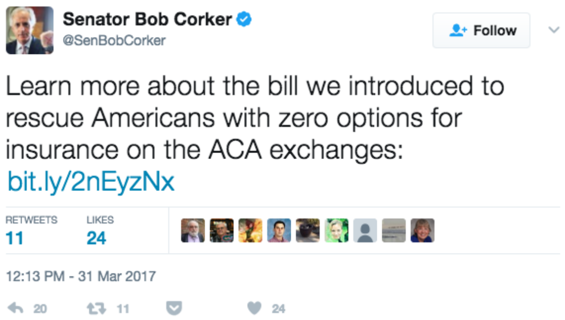
Second, SVM is the best performing model by a hair—though unfortunately, it cannot easily be used for tweet categorization because it does not provide probabilities of categorizations. BDT is the worst performing model (though this was not as clearly true every time I ran this experiment). However, this does not mean BDT is not a worthwhile use of time—we shall see below that it may have some advantages in the domain adaptation task.

Third, the models are capable of achieving high classification rates, even on the Industry task, which is impressive given the number of potential labels. However, it is worth recalling that fewer than half of all OpenSecrets Industry labels were represented in the training data, and some labels contained as few as one documents.

3.5 Domain Adaptation Results

Evaluation of the models trained on company descriptions during testing on tweets (out of domain inputs) is difficult (because of a lack of labels. For lack of time, in lieu of manually labeling data, I've settled for running the models on tweets, and selecting tweets to highlight where the probabilities of a category exceed a certain threshold (the value of which can be tinkered with). This method is easily extensible to checking for conflicts of interest: the same process can be undertaken, simply further filtering tweets that have labels within the top donors of the politician from which they originated. At this point, I have not established a manner of systematically evaluating these results, but they have been somewhat promising, particularly for some models.

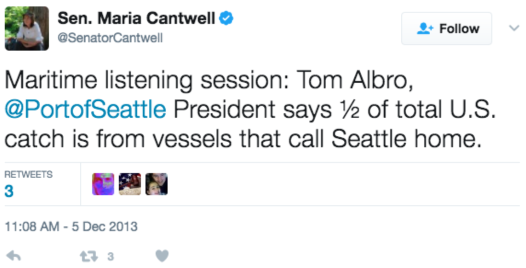
Below are two tweets identified with the BDT model as likely related to an industry:


Senator Bob Corker
@SenBobCorker

Learn more about the bill we introduced to rescue Americans with zero options for insurance on the ACA exchanges: bit.ly/2nEyzNx

12:13 PM - 31 Mar 2017

Figure 1. Tweet labeled industry "Insurance" with probability 1.0. "Insurance" one of top ten industry donors: potential conflict of interest. Model used: bagging with decision trees.


Sen. Maria Cantwell
@SenatorCantwell

Maritime listening session: Tom Albro, @PortofSeattle President says ½ of total U.S. catch is from vessels that call Seattle home.

11:08 AM - 5 Dec 2013

Figure 2. Tweet labeled industry "Sea Transport" with probability 1.0. The "Sea Transport" Label has only 39 samples. Note: not a suspected conflict of interest. Model used: bagging with decision trees.

A couple remarks follow.

First, the tweet from Senator Corker is actually a tweet that has been filtered to suggest a possible conflict of interest; the "Insurance" industry is one of his top industry donations. That his tweet is anti-Affordable Care Act may not be surprising in light of that. However, it's worth noting that there were plenty of examples of tweets categorized as "Insurance"-related that took the opposite position, even among politicians for whom the Insurance industry was a top donor. This suggests that the nature of having a conflict of interest is more complicated than it might seem.

Second, that the tweet from Senator Cantwell was correctly identified as "Sea Transport"-related with such high probability (1.0) seems impressive, given that there are only 39 documents in the training dataset labeled as "Sea Transport" (though, unfortunately, this number places it in the top half of Industry labels collected). Interestingly, Logistic Regression and SVM also categorize this tweet as "Sea Transport", but Logistic Probability assigns it a much lower probability (SVM does not allow extraction of probabilities). MNB predicts that it is "Telecom Services", somewhat inexplicably.

More generally: the results are promising. Though there are certainly false negatives (especially when the threshold value is set lower), it is certainly possible to identify tweets that are related to industries—though the data currently used produces models far more likely to identify certain types of industries. From identifying industries, it is trivial to proceed to identifying potential conflicts of interest, though some care should be taken to ensure that the phrase "potential" is always used, given the nature of the models. For instance, certain topics frequently covered in tweets, such as disease awareness and fundraising, should not be mistaken as conflicts of interest when coming from people who take many donations from "Health Professionals."

4 Future Directions

Because of the time spent collecting the data for this project, there remain many avenues for future progress.

4.1 Data Collection

The most obvious improvement to this project could be achieved by continuing the data collection process. This could have several potential improvements. First, more training text would necessarily improve the accuracy of the models by allowing more effective evaluation of features. Second, gathering companies beyond those traded on the Nasdaq and NYSE could allow a greater distribution of industries to be represented. This could be achieved by scraping the OpenSecrets website directly for names of companies that are affiliated with particular industries (removing the need to match to SIC numbers). This produces a subsequent challenge: how to gather text about these countries, which is made more difficult without the ability to search websites by stock ticker. Several resources could be used: Wikipedia articles about the companies (though these may be difficult to correctly identify), or Twitter itself (searching company names could turn up text related to them)—the latter of which might help bridge the gap between the training and test domains.

4.2 Modeling

Given more time, it would be advantageous to invest more effort in the model selection process. Different models could be more systematically investigated, and optimal parameters for each could be identified using gridsearch techniques. This would permit more definitive statements about the efficacy of different models.

In the same vein, it would be useful to investigate further techniques for domain adaptation. Such techniques have been documented extensively, even particularly in the context of natural language processing—for a thorough discussion, see Li (2012). Incorporating techniques of this sort could only improve the results on the out-of-domain data.

4.3 Expansion of Domains

Successful models could potentially (with some tweaking, of course) be adapted to target domains other than Twitter. Potential target domains could involve other social media, press releases, or the Congressional record—or, branching out from strict understanding of political speech, perhaps Congressional bills (combined with cosponsorship and vote information) or even news articles that mention politicians in the context of issues that involve industries that donate.

4.4 Applications of Results

Upon settling on a set of modeling techniques that produce satisfactory results, it would be ideal to produce a tool that allows users to access results, though such a tool could take several forms.

The tool could be a Twitter bot that assesses new tweets by politicians for potential conflicts of interest, and replies to those tweets with a message notifying users. It could also be a website that displays potential conflicts of interest (in, e.g., new bills, congressional speeches, tweets, etc.) that have arisen in a live way. A final form might be a (free) subscription service; users could sign up to “follow” particular politicians, and would be notified when potential conflicts of interest have occurred.

5 Conclusion

This project has demonstrated the feasibility of collecting data and training basic models that can help identify instances in which political speech and political donations may interact. Though not as much modeling progress was made as would have been ideal, groundwork is laid for a tool that can help enable greater public participation in the political process by informing citizens of potential conflicts of interest in a context-dependent manner.

6 Acknowledgements

I’d like to thank Professor Dragomir Radev for his support and insights while advising this project.

Greenhouse deserves credit for inspiring this project.

I’d also like to thank the Center for Responsive Politics’ excellent OpenSecrets.org for the data regarding donations, and Twitter, Google Finance, Bloomberg, Reuters, the U.S. Securities and Exchange Commision’s EDGAR, and Nasdaq for all hosting data relevant to this project.

Tweepy, BeautifulSoup, Natural Language Toolkit, pandas, Requests, NumPy, tqdm, scikit-learn, and of course, Jupyter Notebook were all invaluable resources.

7 Relevant Academic Work

Two very helpful sources for natural language processing and machine learning:

Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O’Reilly Media Inc.

Pedregosa, Fabian, et al. "Scikit-learn: Machine learning in Python." *Journal of Machine Learning Research* 12.Oct (2011): 2825-2830.

Related work (both in political science, and in statistics/computer science) includes:

Potters, Jan, and Randolph Sloof. "Interest groups: A survey of empirical models that try to assess their influence." *European Journal of Political Economy* 12.3 (1996): 403-442.

Grier, Kevin B., Michael C. Munger, and Brian E. Roberts. "The determinants of industry political activity, 1978–1986." *American Political Science Review* 88.04 (1994): 911-926.

Claessens, Stijn, Erik Feijen, and Luc Laeven. "Political connections and preferential access to finance: The role of campaign contributions." *Journal of Financial Economics* 88.3 (2008): 554-580.

Saloojee, Yussuf, and Elif Dagli. "Tobacco industry tactics for resisting public policy on health." *Bulletin of the World Health Organization* 78.7 (2000): 902-910.

Luke, Douglas A., and Melissa Krauss. "Where there's smoke there's money: Tobacco industry campaign contributions and US Congressional voting." *American Journal of Preventive Medicine* 27.5 (2004): 363-372.

Ramos, Juan. "Using tf-idf to determine word relevance in document queries." *Proceedings of the first instructional conference on machine learning*. 2003.

Tokunaga, Takenobu, and Iwayama Makoto. "Text categorization based on weighted inverse document frequency." *Special Interest Groups and Information Process Society of Japan (SIG-IPSJ*. 1994.

Robertson, Stephen. "Understanding inverse document frequency: on theoretical arguments for IDF." *Journal of Documentation* 60.5 (2004): 503-520.

Zhang, Wen, Taketoshi Yoshida, and Xijin Tang. "A comparative study of TF* IDF, LSI and multi-words for text classification." *Expert Systems with Applications* 38.3 (2011): 2758-2765.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3.Jan (2003): 993-1022.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Vol. 1. No. 1. Cambridge: Cambridge University Press, 2008.

Li, Qi. "Literature survey: domain adaptation algorithms for natural language processing." *Department of Computer Science The Graduate Center, The City University of New York* (2012): 8-10.