Classifying Debate Speech for Political Prediction

Aksheetha Sridhar, Cody Stancil, Yizhe Ge, Seth Green

Executive Summary

Text mining is both a fascinating and complex problem due to the unique challenges that deconstructing natural language pose. To accurately make predictions and mine textual data is a powerful tool that has many applications whether it be in the realms of social media, journalism or academia to name a few. Text classification is the specific task of interest for the problem to follow and has practical uses in news aggregation (RSS readers) which utilize algorithmic methods for the purposes of topic clustering. This study involves classifying the transcripts of the first and second 2016 presidential debate speeches to determine if portions were either spoken by Clinton, Trump or a moderator. The relevance such classification is that it provides a deeper perspective into the psychology of the candidates at a sociological level; it is also important for the purposes of integrating this classification system with social media such as Twitter in order to better understand demographics and political affiliations of specific areas.

Transcripts of the first and second presidential debate speeches were utilized in order to transform and construct a Term Frequency-Inverse Document Frequency (TF-IDF) matrix of predictors. A total of five different models were evaluated at a sparsity threshold of 0.05 to reduce the feature space to 60 predictors. The following models were tested, Multinomial Logistic Regression, Naive Bayes, Partial Least Squares (PLS), Support Vector Machines (SVM), and Random Forest, by 5-fold cross validation. The models were evaluated primarily on the raw accuracy and the balanced error rate (BER) metric, with a low BER score being more optimal. The model with the lowest BER was the Random Forest at a value of 0.30 and was then evaluated further by changing the lower bound of the sparsity threshold. A threshold of 0.03 with 120 predictors was found to improve (lower) the BER and was concluded to be a better sparsity lower bound setting.

In conclusion, the Random Forest model was determined to be toptimal based on the evaluation metrics. In order to better improve this model and overall prediction accuracy, feature engineering and incorporating ‘bigrams’ would be utilized. This could result in a more robust model that also takes into account the high-dimensionality of the feature space and tries to find a balance between managing the number of predictors and optimizing model performance.

Problem Statement

Text classification is a persistent problem which has applications in a wide variety of domains. News aggregation services classify articles to bucket them into topic clusters and deliver the most relevant information to each user. Research journals and libraries use similar systems to organize the ever more vast resources which they are expected to catalog. Email programs are constantly trying to classify incoming email, initially as “spam” or not, but also into useful categories like “Social” and “Promotions” to make navigating your inbox easier. Automated customer support services attempt to categorize user queries in order to better route them to relevant and helpful information.

An emerging field of study which has gained a lot of attention in recent years involves doing similar text classification on social media data. Twitter in particular is a fertile ground for this sort of analysis because it is easy to access large numbers of bite size raw text fragments which are often tagged with useful attributes like timestamps and sometimes geo-locations. Researchers have begun mining this data to attempt to predict a wide array of things (Gerber 2014; Gayo-Avello 2012; Moy and Murphy 2016). The scope of potential for this data set is evident from the fact that the Library of Congress recently announced that it will be archiving all public tweets since 2005 for future research (Parry 2010).

The 2016 Presidential Election has likewise gained a lot attention recently, as many aspects of it have been historic, though time will tell which will be filed under “famous” and which under “infamous.” The televised debates have been notable in much the same way, and one of the most notable aspects of these debates has been the drastic difference in the way the two candidates speak.

Our plan is to model these two different styles of speech, using a transcript of the first two debates. Our hypothesis is that this model could then be applied to Twitter data to predict that holy grail of political prediction: election results. The intuition is that, once we have a model of how each candidate speaks, we will be able to classify tweets as “Trump-like” or “Hillary-like” in their ways of speaking. By looking at the distributions of these tweets over geographic areas (for instance cities, or perhaps congressional districts), we will be able to predict which candidate that area will favor in the election.

Objectives

Our high-level objective is to build a model that can be used to predict election results in specific areas based on analysis of geotagged tweets from those areas. There are obvious stakeholders for a successful model like this. Between journalists, bloggers, pundits, and politicians themselves, predicting elections is a thriving and lucrative industry.

The final model could be validated by comparing the predictions of the model to the actual election results. While actual results are only made public on a state-by-state level, many surrogate measures like exit polls are available at a very granular local level. However, none of these data will be available until after the election on November 8th. Therefore, this project focuses on building the underlying model that will be used to classify those tweets.

We built this model by applying a variety of text mining methods to a data set of transcriptions of the first and second US Presidential Debates of 2016. The model takes as input passages of uninterrupted speech (as raw text) and classifies whether this passage was spoken by Clinton, Trump or a moderator. This data set is ideal for this because it features 701 such passages which are clean and labeled with the speaker. This allows us to rigorously train and test models in a relatively controlled environment and therefore to isolate the effects of specific features on our overall prediction quality. The data set was downloaded from the website Kaggle on October 16, 2016.

Metrics

We measure this prediction quality with two metrics. Our model is a classifier which distinguishes between the speech of Republican nominee Donald Trump and Democratic nominee Hillary Clinton and a “neutral” or “independent” moderator. We use 5-fold cross-validation to compare several models on the basis of raw cross-validated test set Prediction Accuracy and Balanced Error Rate (BER).

Raw Prediction Accuracy is simple to calculate, even for multiple classes. It is the number of correctly classified observations divided by the total number of observations. This is useful and easy to interpret, but in cases where there are an unbalanced number of observations in the different classes, this will be an unreliable and misleading metric.

Our data set featured roughly balanced numbers of observations for each class (see table 1 below), although they were by no means equal. For this reason, we decided to include the raw Accuracy scores, but to augment them with another metric.

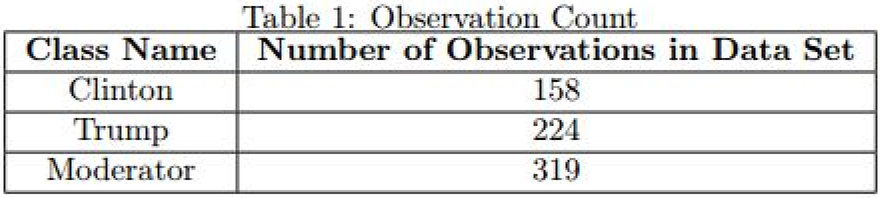


Table 1: Observation Count

Class Name	Number of Observations in Data Set
Clinton	158
Trump	224
Moderator	319

There was some dispute in the literature as to the best metric to evaluate accuracy in a multi-class classification problem. We ended up deciding on Balanced Error Rate. BER is promoted by Read and Cox, among others, in their 2007 paper. They claim, “the BER gives a more sensitive indication of an algorithm’s performance as it gives equal weighting to each of the classes...regardless of the number of examples in each class.” (Read and Cox 2007)

BER is calculated by taking the mean Recall (True Positives divided by the sum of True Positives and False Negatives) of all classes and then subtracting this mean from 1. Opposite to accuracy, a low score (approaching zero) is desirable. It is also a more robust measure because it is less susceptible to unbalanced numbers of observations in the different classes.

The optimal model could then be used to classify tweets, as described above. These classifications would in turn be used to predict election results, which would involve a second phase of training and testing, obviously after November 9th when these results become available. For this reason, our analysis in this paper will be limited to the building, training, and testing of the initial debate speech classifier. The application of this classifier to Twitter data and the subsequent use of that to predict election results will be reserved for a later study (potentially in lieu of the final exam for this course…).

Previous Research

Methods for text mining have been extensively applied to document classification since the 1980s. Since then, these methods have been applied to different forms of text, both structured and unstructured, such as Twitter data or transcripts of speeches. A survey of the literature indicates that commonly utilized supervised techniques have been Naive Bayes classification, nearest neighbors classification, decision trees, support vector machines and neural networks.

Naive Bayes classifiers have been observed to have high accuracy and speed as well as similar execution time in comparison to decision trees and neural networks which are more complex methods. The following is an implementation of Naive Bayes as implemented by Kamruzzaman et al that tries to maximize the probability of observing particular words found in the training set, subject to the assumption of independence where for a given attribute, its effect on the class is independent of other attribute values (Kamruzzaman et al., 2010). Due to its simplicity and comparable classification success rate compared to more advanced methods, Naive Bayes is a viable option for the purposes of this analysis. Decision Trees are also another method that will be applied to the text. Li et al performed a comparison between several different classifiers and chose decision trees for their robustness against noisy data and learning disjunctive expressions in text. Another option is to combine several classifiers to improve accuracy.

The aforementioned classification algorithms have been extensively applied and proven to be successful when it comes to binary classification but can also be extended to a multi-class classification problem. There are several strategies to accommodate for multiclass classifications and are transformation to a binary classifier, extension from a binary classifier and hierarchical classification (Aly, 2005). Transformation to a binary classifier, is the decomposition of the multi-class classification problem into individual binary classification tasks. One particular method is known as error-correcting output coding (ECOC) in which N number of binary classifiers are trained to be able to distinguish between K classes. Many of the binary classification methods can be extended to a multi-class situation via adaptation of the algorithm. Lastly, hierarchical classification is the partitioning of the output space so that the classes are arranged as a tree and subsequently divided into clusters and at each node a binary classifier is applied until a single class is achieved.

In his paper, Daniel Velez provided a method for generating K-Means seeds for doing unsupervised text mining (Daniel Velez, 2015). His deterministic model is based upon the tight relationship between dimension reduction methods and unsupervised classification methods. Daniel acquired the data from mixed sources and treated it by tokenizing the data, eliminating the stop words, stemming the tokens, generating TF-IDF values for each token, and selecting the most important words based on the transition point threshold. He then used eigenvectors obtained from principal component analysis (PCA) to extract the initial seeds that explained most of the variabilities. Daniel tested the goodness of his method using a sample of labeled e-mails (NG20), which represented a gold standard within the field of text mining. The final model was tested to have significant higher possibility to get the best results than other comparative models, and it performed even better when more than two classes were involved.

In an effort to classify documents into one or more categories, Juho Rousu et al. created a "kernel-based algorithm for hierarchical text classification…[which is] a variant of the Maximum Margin Markov Network framework" (2005). This model is meant to be able to classify documents into one or more categories; but to test the modelling structure and accuracy, Rousu et al. used two sets of public corpuses: Reuters Corpus Volume 1 and the WIPO-alpha patent dataset.

Rousu et al. used zero-one error rates, level-wise precision and recall values to evaluate the model performance on these two sets. They found, in the WIPO data set that using the hierarchical classification model 0/1 loss in the test set was reduced over other methods. This benefit was not observed in the Reuters Corpus. Moreover, they found "recall of all methods... diminishes when going farther from the root" (Rousu et al., 2005). Overall, they observed that "using the hierarchical structure of multicategory labelings leads to improved performance over the more traditional approach of combining individual binary classifiers" (Rousu et al., 2005)

Thorsten Joachims conducted another study that used learning techniques for classification models. However, the focus of this study was around support vector machines as a method for classification. To test these models Joachims used the following collections: Reuters-21578, WebKB, and Ohsumed(Joachims, 2002).

Joachims compared expected and predicted error rate as a measure of accuracy of his model. Reuters-21578, WebKB, and Ohsumed are ordered from easiest to predict to hardest to predict based on the expected error of the model. When Joachims evaluated his model on the three data sets he found that in fact Reuters-21578, WebKB had both low error and a relatively small difference between expected error and test set error; on the other hand the error and the difference between expected error and test set error was much larger for Ohsumed (Joachims, 2002). However, this paper intended to "model only upper bounds on the error rate" but the lower bound could tell us that "text-classification cannot be learned with SVMs" (Joachims, 2002). So there is still more to understand about using learning methods for text classification.

To conclude, text classification is a challenging problem that still remains unsolved due to the complexities of natural language, due to difficulty in identifying semantics and abstraction, and the high-dimensionality of features. But as refinement of current text mining methods, experimentation with new techniques and a more nuanced understanding of linguistics and natural language processing occur, continued progress is being made.

Models

We tried five different models, and then evaluated the best model at several different sparsity thresholds. The initial models were all evaluated on a Term Frequency-Inverse Document Frequency matrix of predictors.

In order to compute the TF-IDF matrix, we went through a series of data cleaning stages. We first tokenized the debating text we got from Kaggle. In this stage, unstructured text was transformed into individual variables, which were composed of words with uncleaned structures. We further processed the variables by trimming the white spaces, removing the punctuation, and transforming all words into lower case. In addition, unuseful variables like numbers and stopwords were removed from our model. We stemmed all variables and used the unique stemmed words as our final predictors. Finally, we quantified the importance of a term in a document using the TF-IDF metric, which was defined as the relative presence of each term in the document to its presence within the corpus.

Once this was done, we set sparsity thresholds for the terms. Any term that appeared in more than 99% or less than 5% of the documents was removed. By doing this, we removed both overly common words that would not be useful, and also very rare words. This left us with 60 predictors on which to build our models.

Multinomial Logistic Regression

Multinomial Logistic Regression allows us to perform a logistic model when there are more than two classifications. This model will deliver the probability an observation belongs to each one of the categories. In our problem, we were given three probabilities for each observation: the probability the speaker was Clinton, Trump, and the Moderator. The class with the highest probability is then chosen as the prediction. To assess this model we cross validated our results and received a cross-validated accuracy of 0.66 and a balanced error rate of 0.36.

Naive Bayes

Although Naive Bayes is typically implemented in a binary classification setting, extending it in the context of a multi-classification problem is straightforward as the probability of each class is calculated normally, with no modifications to the classifier. The class with the largest probability is then chosen as the predicted value. This model was assessed via cross-validation and achieved a cross-validated accuracy of 0.58 and a balanced error rate of 0.40.

Partial Least Squares (PLS)

Because of relatively high observations to predictors ratio, we decided to investigate whether using principle components as predictors could boost the performance of our predictive model. We used Partial Least Squares (PLS), a supervised alternative to Principal Component Regression (PCR), for modeling. The new set of features (components) were linear combinations of the original features, and they were ranked by their abilities to explain the variabilities in the data. The optimal number of components was chosen based on the cross-validated error rates, and the metrics suggested us to use the top 7 principal components as regressors for our model. The final model achieved a cross-validated accuracy of 0.61 and a balanced error rate (BER) of 0.40.

Support Vector Machine (SVM)

We investigated the performance of Support Vector Machine (SVM), which have been shown to perform well in a variety of supervised classification settings. We tested the performances of different combinations of costs and gammas on different kernels (linear, radial, polynomial, and sigmoid). The cross-validated error rates showed that the linear kernel model with cost 10 and gamma 0.5 could reach the highest accuracy. The final model achieved a cross-validated accuracy of 0.66 and a balanced error rate (BER) of 0.37.

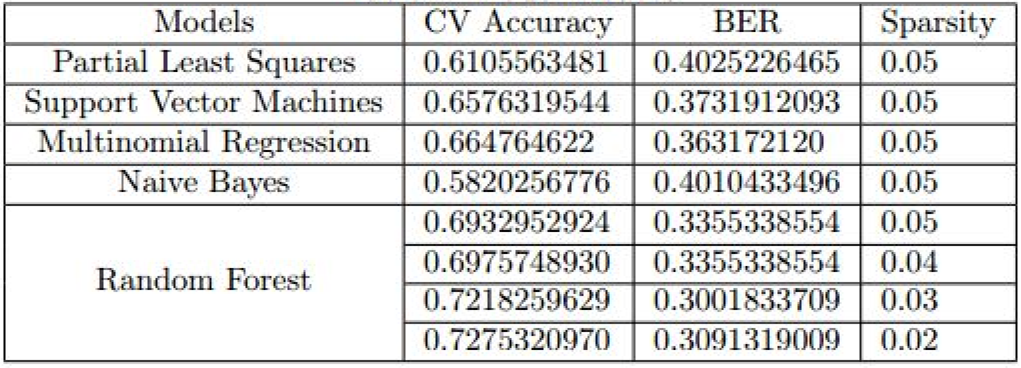
Random Forest

Finally, we tested a Random Forest model. This proved to be the best model by both metrics. Including all of the predictors, the model achieved cross-validated accuracy of 0.69 and a BER of 0.33.

Sparsity Thresholds

As mentioned above, we restricted the terms in our TF-IDF matrix to those terms which appeared in *at least* 5% of documents and *at most* in 99% of documents. Once we had baseline scores for each of our models, we tuned the lower bound sparsity threshold to see if we could get an improvement in our models' accuracy from including more rare terms. The changes we made affected each model roughly equally; Random Forest was still the best model at every threshold we tested. Therefore we only report the results from the adjusted thresholds for the Random Forest Model in Table 2 below.

Table 2: Model Metrics



Models	CV Accuracy	BER	Sparsity
Partial Least Squares	0.6105563481	0.4025226465	0.05
Support Vector Machines	0.6576319544	0.3731912093	0.05
Multinomial Regression	0.664764622	0.363172120	0.05
Naive Bayes	0.5820256776	0.4010433496	0.05
Random Forest	0.6932952924	0.3355338554	0.05
	0.6975748930	0.3355338554	0.04
	0.7218259629	0.3001833709	0.03
	0.7275320970	0.3091319009	0.02

Both the cross-validated accuracy and BER improved in the Random Forest model as the threshold was decreased (and thus included more predictor terms). The improvement from 0.05 to 0.04 was only marginal, but a notable improvement was made when the threshold was lowered to 0.03. Lowering the threshold to 0.02 again showed a very small improvement, but at the cost of adding an additional 75 predictor terms, from 120 in the 0.03 model to 195 in the 0.02 model. Therefore, we decided 0.03 was likely a better setting for the lower bound. This was also partially because of a concern that the higher number of predictors could become problematic on anticipated future testing cases which might be shorter in length (tweets, for instance).

Conclusions and Future Directions

The Random Forest model with 120 predictors, generated by setting the sparsity bounds to 3% and 99%, was our optimal model. In cross-validation it achieved a raw accuracy of 0.72 and Balanced Error Rate of 0.30.

While this will be a useful model, there is certainly room for improvement. The incorporation of bigrams could yield some promising predictors, as many “trademark” phrases of one candidate or the other are more than one word. Some feature engineering could also prove fruitful. For instance, a “average sentence length” predictor was proposed. Our model was based purely on using values from the TF-IDF as predictors, so it is certainly possible that some creatively engineered features could provide additional predictive power. Of course, only time will tell how this model will fare on the Twitter data on which we plan to use it.

Summary

There continues to be much research on the use of Twitter to make predictions about social phenomena. These have had widely varying degrees of success. In their 2016 survey, Moy and Murphy write:

> Twitter has been used to measure reactions to breaking news, the outbreak of diseases, and other public phenomena (Bandari, Asur, & Huberman, 2012; Hu et al., 2012;

> Lanagan & Smeaton, 2011; Petrovic et al., 2013). These studies generally demonstrate an ability of Twitter to conform to other traditional data sources like surveys or predict outcomes in elections. However, other researchers have found inconsistent results, especially in the political arena (Gayo-Avello, 2011, 2013; Jungherr, Jürgens, & Schoen, 2012).

While previous studies have often used sentiment analysis or other dubiously engineered features (Gayo-Avello, 2012), we will bring a novel approach to this problem by initially building a model on the candidates' own speech and then classifying tweets by how similar they are to the candidates' way of speaking. We will also be positioned to make actual predictions in advance of the election, and then verify those predictions with election results. This methodology addresses many of the critiques that both Gayo-Avello and Moy and Murphy had of previous studies in this field and positions us to make some interesting and quantifiably testable observations.

References

Aly, Mohamed (2005). Survey on multiclass classification methods. *Neural Netw*: 1-9.

Chelba, Ciprian, Milind Mahajan, and Alex Acero (2003). "Speech utterance classification." *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*. Vol. 1. IEEE.

Gayo-Avello, Daniel (2012). A Balanced Survey on Election Prediction using Twitter Data. Cornell University Library. https://arxiv.org/abs/1204.6441

Gerber, Matthew S. (2014). Predicting crime using Twitter and kernel density estimation. *Decision Support Systems*, Volume 61, May 2014: 115–125.

Joachims, T. (2002). A Statistical Learning Model of Text Classification for SVMs. *Learning to Classify Text Using Support Vector Machines*, 45-74. doi:10.1007/978-1-4615-0907-3_4

Kamruzzaman, S. M., Farhana Haider, and Ahmed Ryadh Hasan (2010). "Text classification using data mining." *arXiv preprint arXiv:1009.4987*.

Li, Yong H., and Anil K. Jain (1998). Classification of text documents. *The Computer Journal* 41.8 : 537-546.

Moy, Patricia and Joe Murphy (2016). Problems and Prospects in Survey Research. *Journalism & Mass Communication Quarterly*. Vol. 93(1) 16–37.

Parry, Mark (2010, May 7). Library of Congress, Facing Privacy Concerns, Clarifies Twitter Archive Plan. Chronicle of Higher Education.

Read, Ian, and Cox, Stephen J. (2007). Automatic Pitch Accent Prediction for Text-To-Speech Synthesis. In 8th Annual Conference of the International Speech Communication Association (INTERSPEECH-2007) (pp. 482-485).

Rousu, J., Saunders, C., Szedmak, S., & Shawe-Taylor, J. (2005). Learning hierarchical multi-category text classification models. *Proceedings of the 22nd International Conference on Machine Learning - ICML '05.* doi:10.1145/1102351.1102445

Velez, D., Sueiras, J., Ortega, A. et al. (2016). Stat Methods Appl. 25: 477. doi:10.1007/s10260-015-0345-4