Frontier Culture and Modern Politics in the U.S.

Jean P. Vazquez, Kevin Chen, Zhiwei Tang

jpva@bu.edu kchen005@bu.edu zwtang@bu.edu

Project Task

This project seeks to understand the defining features of the American Frontier by identifying "frontier language" and its persistence over time through natural language processing and machine learning. The main difficulties with this project lie primarily with 2 tasks: Firstly, having to handle a dataset of such a large size (in excess of 1,200 lengthy text documents); Secondly, the possibility of not finding any natural language processing algorithm which, out of the box, can analyze the data and instead having to heavily modify either the algorithm or our data in order to obtain our analyses.

Dataset

Our data consisted of two historical datasets: Frederick Jackson Turner's speeches from http://xroads.virginia.edu/~hyper/turner/, a folder containing the histories of several hundred U.S. counties organized by state from www.dropbox.com/county-histories as well as two modern datasets to compare with the historical datasets consisting of presidential nominees nomination acceptance speeches and political party platforms from https://www.presidency.ucsb.edu/documents. In order to use the dataset, however, several steps had to be taken. First, all the files were converted into .txt format, which the county histories files already were in, but all others required conversion into this. Following this, several preprocessing steps were taken to streamline the reading process; all punctuation marks and special characters were filtered out. Lastly, each individual text was compacted such that there was no distinction between different lines or sentences, only separate words.

Approach

The project was divided into two separate phases. The original first phase consisted of finding as many "frontier words" as possible within the historical datasets provided; which are words that are much more frequently spoken by "frontier-esque" persons and, as such, would serve as the basis for the second phase of this project, wherein the change of "frontier-esque" behavior over time would be measured. However, several attempts were made using unsupervised machine learning algorithms in order to find these words to extremely limited success. Additionally, consulting with Dr. Lapets about the attempts made to find words given a theme such as "frontier" only confirmed the information inferred from the previous results, that no machine learning or natural language processing algorithm would yield any results unless the method was supervised and very heavily modified. As such, instead of attempting to find words, the project focus was shifted immediately towards the second phase thanks to a list of “frontier words” given by the project head, Dr. Martin Fitzsbein. The second phase consists of then using the “frontier words” found in the first phase of the project in order to place candidates on a continuum based on how “frontier-esque” they are determined to be. In order to determine the “frontierness” of a candidate, three separate methods were applied across both historical datasets: term frequency analysis, a tf-idf score analysis, and what was dubbed “near word association”.

As for what exactly these three metrics mean, they can be summarized as follows: term frequency simply finds the amount of times each frontier word appears in a document; tf-idf shows the relative importance that each frontier word carries within a certain text by assigning a relative weight which increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word across all documents in the dataset[1]; and "near word association" consists of scanning through the dataset and grabbing the 10 words closest to every frontier word across all texts and creating a new list of "pseudo-frontier words" from a certain subset of the 10 most common words within this “pseudo-frontier words” list. This last metric was suggested by Dr. Lapets as he found that, in his years of experience with thematic analysis and natural language processing, it is often the case that certain words are used in conjunction with certain other words more often than not. For example, the words clouds and rain are almost exclusively used within the same context. In other words, it tends to be the case that not only are there certain words which fall under a certain theme useful for classifying documents as belonging to that theme, but nearby words which might not by themselves be classified as frontier words could also be used to classify a certain piece of writing under that same theme. To verify a positive correlation between frontier words and its nearest words, we compared our tf-idf analysis of frontier words to additional tf-idf analysis of frontier words’ nearest words. We ran a final tf-idf analysis of frontier words plus the 10 most common “pseuo-frontier words”.

Methodology

The entire project was done within Jupyter Notebook (version 5.6.0) along with several different python packages to handle both the dataset preprocessing and the natural language processing algorithms mentioned above. The python packages used were: sklearn for the term frequency analysis and tf-idf; matplotlib for creating the initial graphs to display the data; os and codecs for reading in all the files across the separate folders that they were stored in and decoding text files, when necessary; numpy for the various useful functions which can be applied across entire lists generated using numpy.array; regex for the symbol

[1] Description taken from http://www.tfidf.com/

and special character filtering of the texts in the dataset; pandas for creating dataframes to simplify sorting and grouping the data.

For the term frequency analysis, all the historical texts were run through sklearn's CountVectorizer package with the vocabulary set to the list of frontier words given. The final result of this being a list of lists wherein each list corresponds to a single text and contains the amount of times each specific frontier word appeared in that text. Then, numpy was used to find the average amount of times each frontier word was used and then all these averages were summed together to obtain the average amount of frontier words are used within the historical documents confirmed to be exemplary versions of "frontier-esque" writing. The only issue that resulted from this was that within the modern datasets, even the ones which contained the largest amount of frontier words still had far less than the historical text average. In order to account for this, the same process of summing the averages was performed across all the modern texts and the texts were then order in descending order by the difference between the historical sum of averages and the sum of averages within that text. These values were then run through sklearn's min-max scaling package in order to create a more useful range of values which was dubbed "frontierness".

For the Tf-Idf analysis, we built a matrix of Tf-Idf scores of all frontier words with respect to each party platform document of Democratic and Republican party in history as well as Frederick Jackson Turner's frontier related literatures. We used the Tf-Idf transformer in python's sklearn module to implement it. This process gave us a feature vector with different frontier words which were used across all the documents analyzed. To simplify the complexity of analysis, we reduced the dimension of the features by getting a score of the "frontierness" of each document. To do this, we did average value of the Tf-Idf score for all the frontier words in a document. Then by comparing the trend of the scores from party platform documents in time-period basis, we can get a brief result for the popularity trend of frontier words over time among politicians. We also tried to use the minimum and maximum Tf-Idf score of frontier words in each document to find the best way to process the data. Besides, we also made averages of the "frontierness score" we got among each document corpus (i.e. Jackson Turner's speeches, Democratic and Republican party platforms) to see the general difference in "frontierness" between frontier related documents and the party platforms. According to the result of the analyses above, a weight term was added to improve the performance and make a comparison.

For the "near word association", the 10 nearest words of each frontier word were, by default, the 5 words before each frontier word and the 5 words after each frontier word. If there were less than 5 non-frontier words before or after a frontier word, then the nearest words on the opposite side would be added until there were 10 words nearest to the frontier word. If there was any overlap amongst nearest words, then those nearest words would only be counted once. If frontier words were too close to each other, then there would be less than 10 nearest words. For example, if a document began with a frontier word, followed by 8 non-frontier words, and then another frontier word, then there would be no nearest words to the left of the first frontier word, only 8 nearest words on the right of the first frontier word associated with the first frontier word, and 10 nearest words on the right of the second frontier word associated with the second frontier word. We used the same methods as those used in the term frequency and tf-idf analyses, but we compared the tf-idf scores of the nearest words rather than term frequencies or tf-idf scores of frontier words.

Results and Conclusions

Figure 1 contains the calculated frontierness of each nominee, which are sorted by the year of their nomination, with a red line showing the general trend of frontierness across all candidates over time. Figure 2 describes the frontierness of party platforms submitted by every party as far back as 1840 as well as two trendlines for the two most prominent current parties, the Republican and Democratic parties. These trendlines were generated using moving averages where each following point was generated by averaging the frontierness of the past 5 platforms of each party which splits each line into 9 segments consisting of the average frontierness of each party across the previous 20 years. As for final results, based on the term frequency analysis of the data, it seems that the general frontierness across nominees is increasing slowly over time while the general frontierness of a party has increased much more rapidly over time, becoming an almost exponential curve. As for what these results imply with respect to this current generation, meaning the results in both graphs that span the past 80 years, it seems that while nominees are likely to continue becoming more "frontier-esque", the future nature of the parties is not so clear. That being said, the trendlines in these graphs seem to point towards the Democratic Party either maintaining a constant "frontier-esque" nature or possibly becoming more "frontier-esque" as time passes. The trendlines seem to point toward Republican Party becoming more "frontier-esque" as time passes.

Figure 1

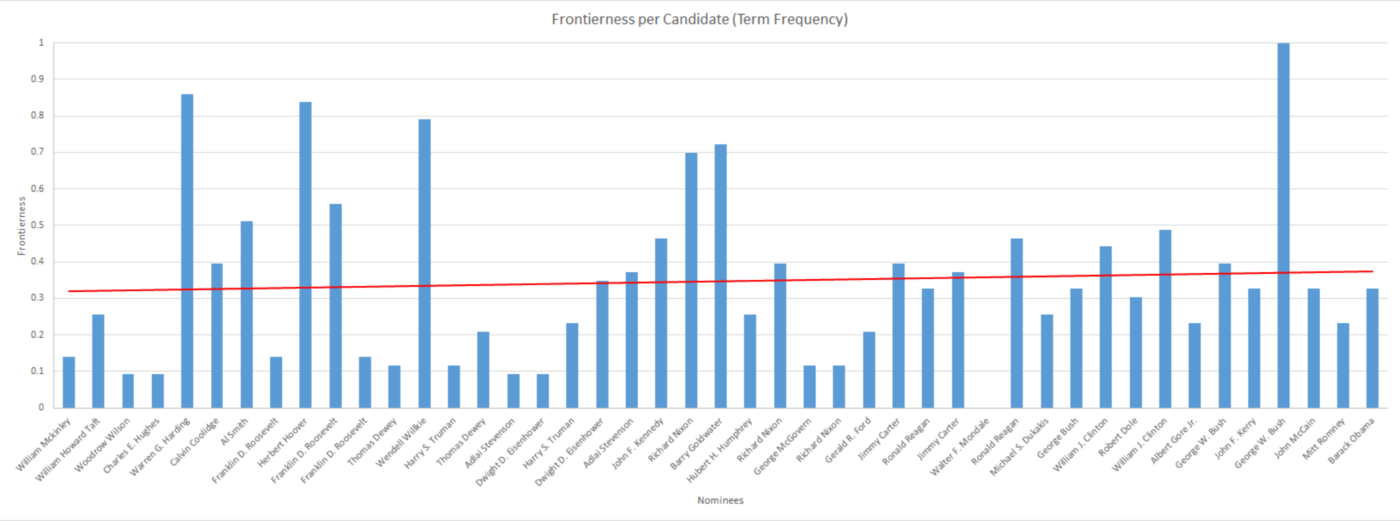


Figure 2

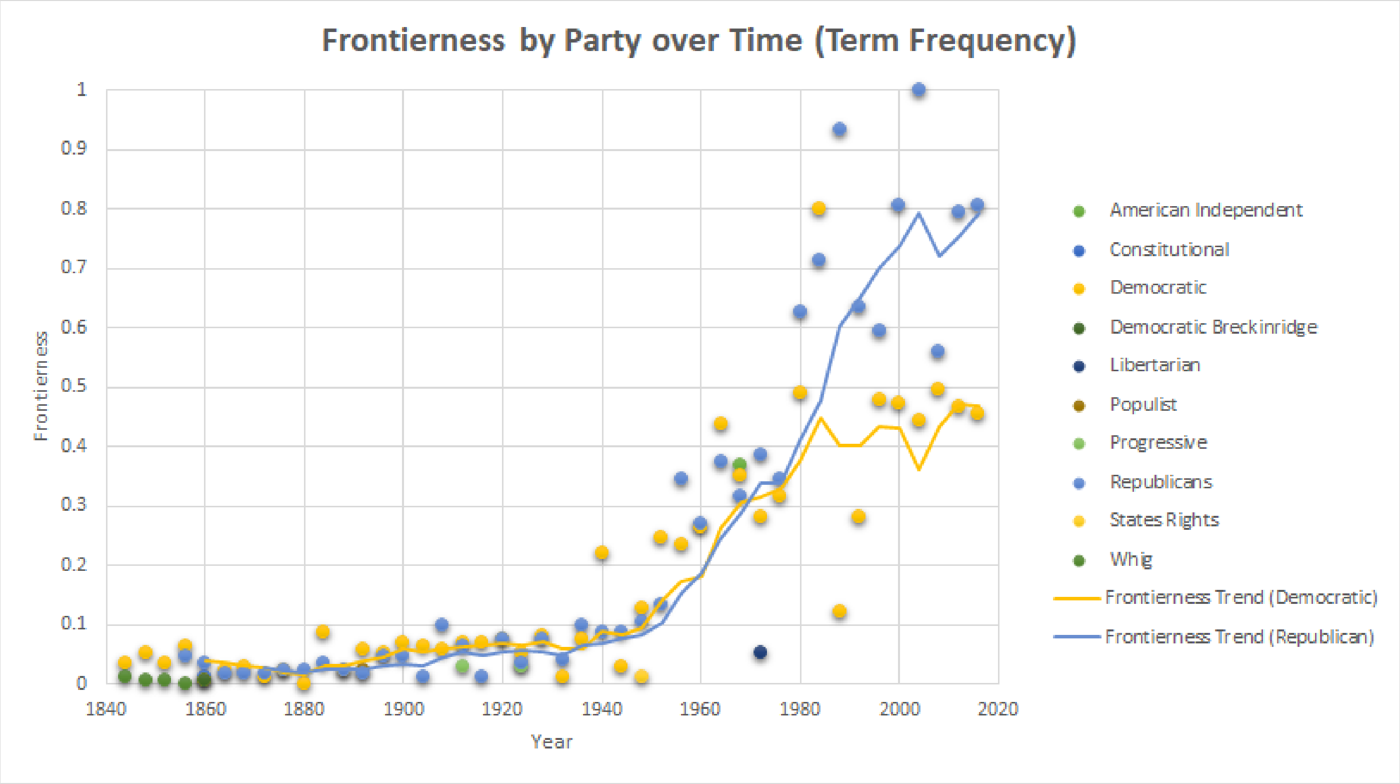


Figure 3

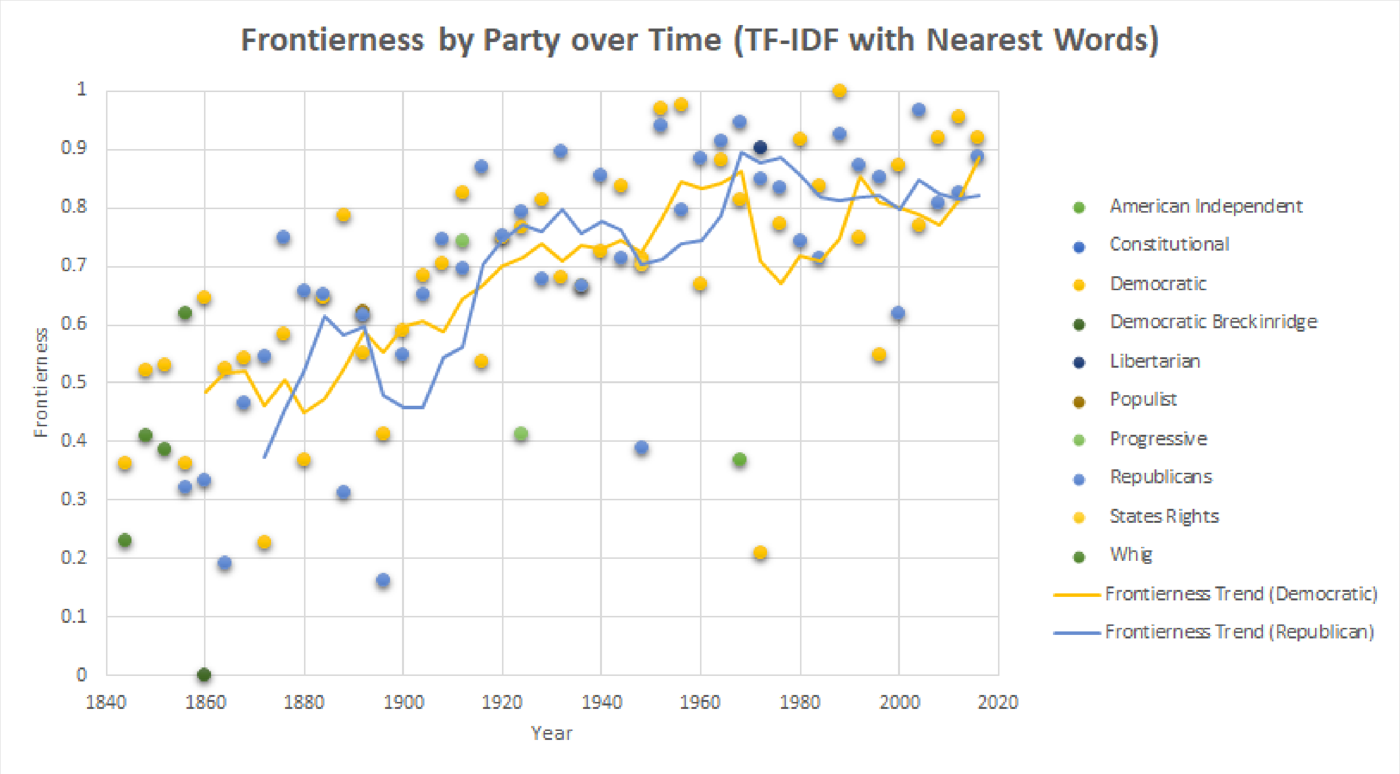
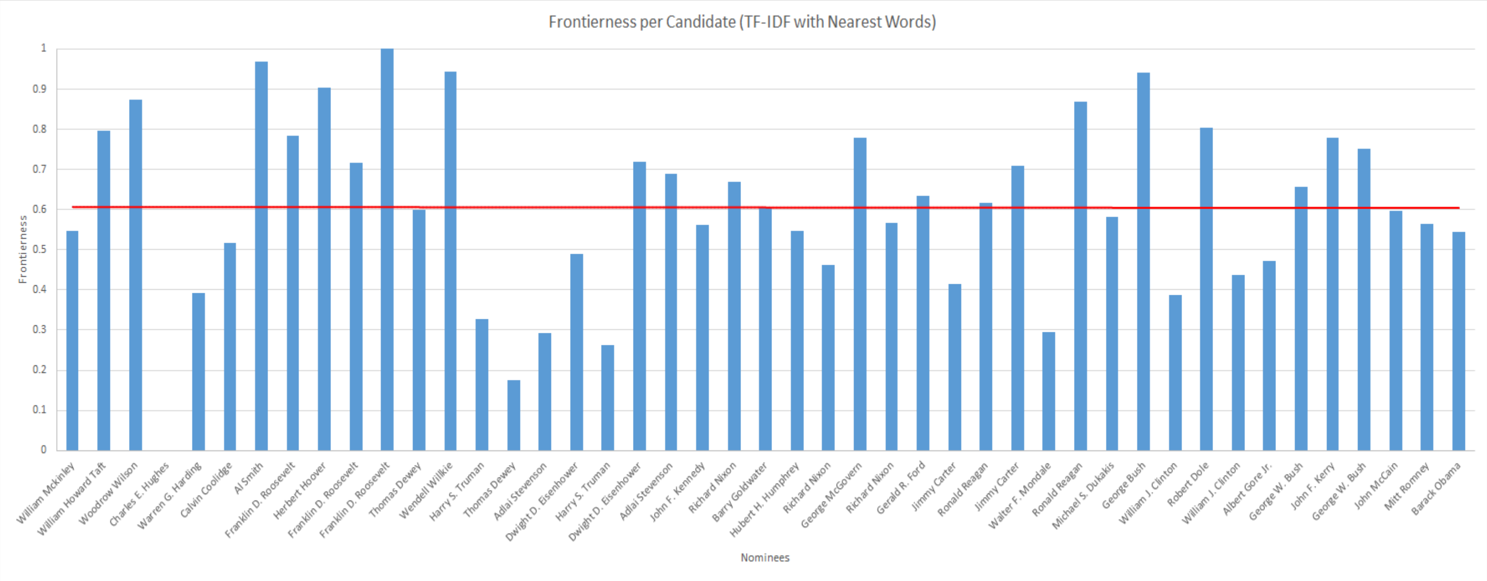


Figure 4



The results for Tf-Idf analysis are shown below:

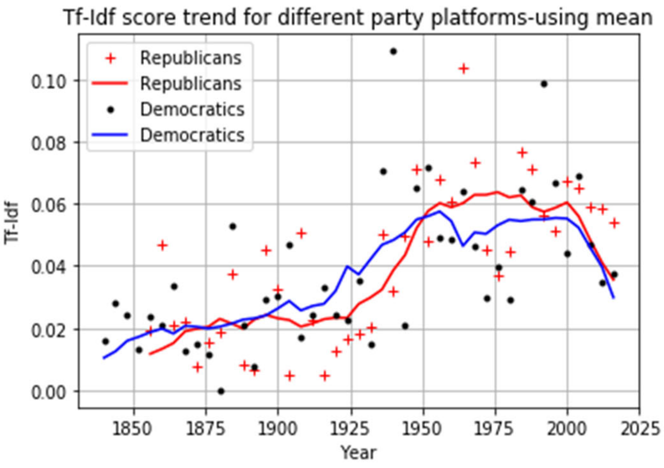
1. Comparison of general “frontierness” between frontier related literatures and party platforms

	Frontier Literatures	**Democrats**	**Republicans**

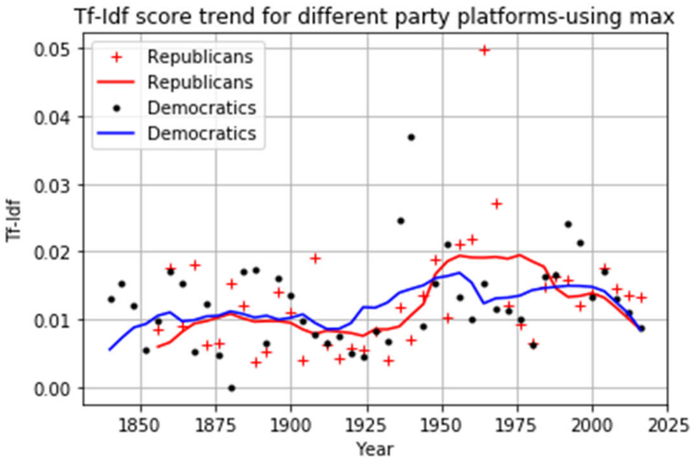
Whether with weight term	Yes	No	Yes	No	Yes	No
Mean Tf-Idf Score	13.51	6.87E-04	6.11	4.81E-04	6.68	5.12E-04
Maximum Tf-Idf Score	372.08	1.90E-02	287.57	1.25E-02	247.11	1.28E-02
Minimum Tf-Idf Score	0	0	0	0	0	0
Summation Tf-Idf Score	892.14	4.54E-02	402.98	3.17E-02	440.65	3.38E-02

Table 1. Comparison of general "frontierness" between frontier related literatures and party platforms

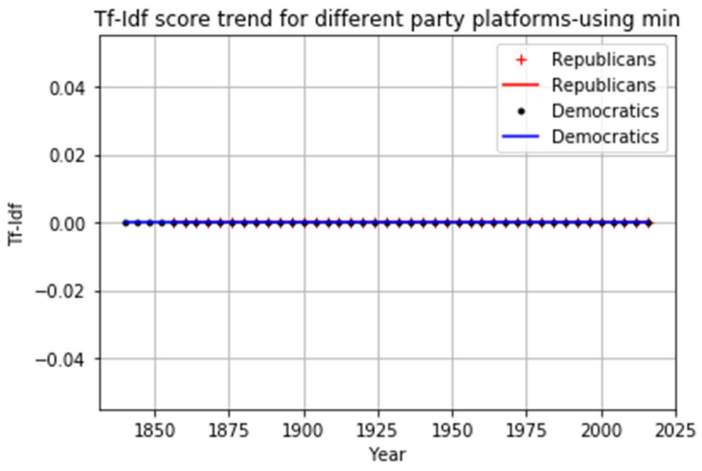
2. Plots



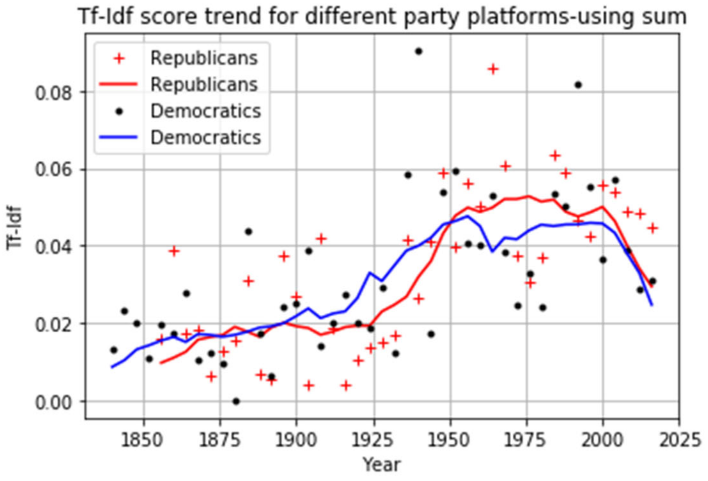
Plot 1. Tf-Idf Score trend for different party platforms – using mean



Plot 2. Tf-Idf Score trend for different party platforms – using maximum



Plot 3. Tf-Idf Score trend for different party platforms – using minimum



Plot 4. Tf-Idf Score trend for different party platforms – using summation

(it's zero because there're always frontier words that doesn't appear in the document)

Table 1 indicates that there is not much difference between the frontier literatures and the test literatures. After thinking about the reason, we found that there is a chance by which the importance of a word with respect to a particular document may not be accurately reflected by the Tf-Idf score. Here is an example with extreme conditions to show this fact more clearly.

Assume we have a corpus of documents, there is a word which have the same number of appearing time in several documents in that corpus and those documents have the same total number of words. If the number of distinct words increases in one document, the average appearing time of the other words in that document will decrease, so the importance rank of that word will probably rise.

However, as the appearing time of the word is the same among those documents, the total number of word in the documents are the same, the Term Frequency (Tf) are the same. As the appearing time of that word in each document within that group of documents in the corpus are the same and doesn't change, so the inverse document frequency (Idf) will keep constant as well. So the Tf-Idf score can't reflect the importance change of a word if we compare Tf-Idf score of that same word across different documents.

As a result, we added a term with the number of distinct words in the document to multiply with the Tf-Idf score. To more precisely measure the impact of this fact (because there is still probability that the importance doesn't increase when distinct word number increases), we also added the actual importance rank of that word to the denominator which is inversely proportional to the importance. (i.e. the larger the rank number, the lower the importance)

So the "frontierness" score is now set to be:

$$\text{□□} - \text{□□□} \times \frac{\text{□□□□□□ □□ □□□□□□□□□ □□□□□ □□ □}h\text{□□ □□□□□□□□□}}{\text{□□□□□□□□□□ □□□□ □□ □}h\text{□□ □□□□ □□□□□ □□□□□□□□□ □□□□□ □□ □}h\text{□□ □□□□□□□□□}}$$

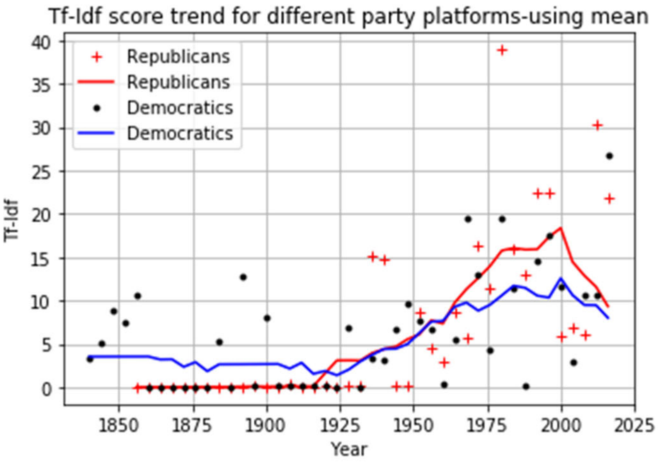
Although the rank is from the comparison of Tf-Idf score, it's not only related to the Tf-Idf score of the single word that we are looking at currently. So the rank won't cancel with the Tf-Idf score which is the first term. The reason for not solely use the importance rank in the Tf-Idf analysis is because this new score can also be seen as comparing the importance rank and use the number of unique words to balance the fact mentioned above and used Tf-Idf score to balance the scale.

The results are shown below:

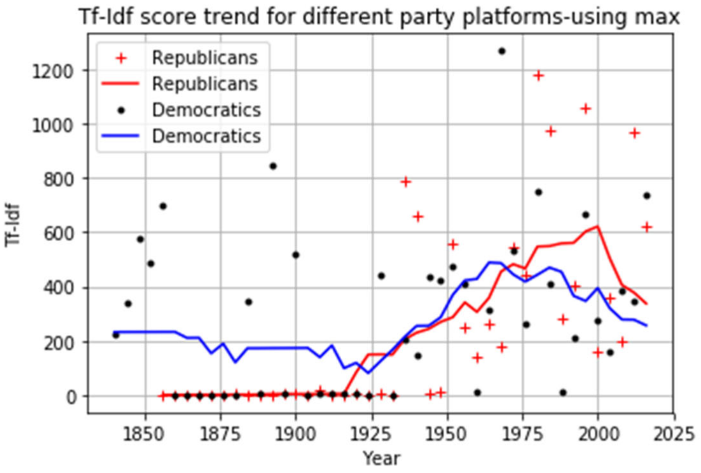
1. Comparison of general "frontierness" between frontier related literatures and party platforms

(See table above)

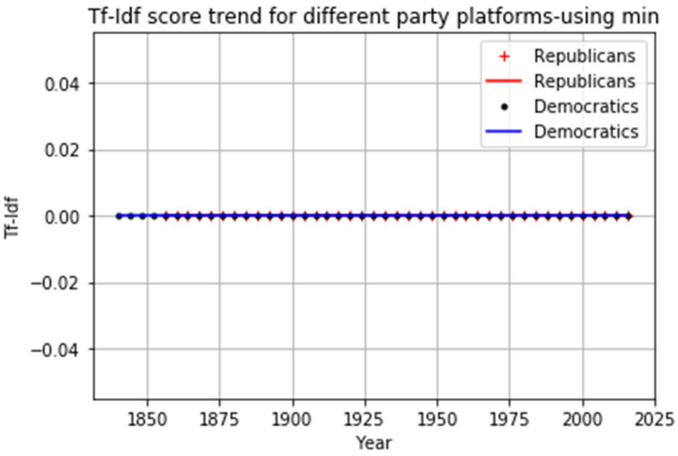
2. Plots



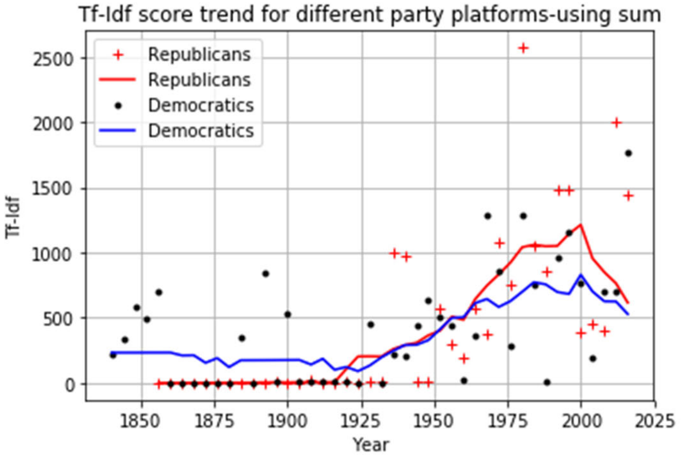
Plot 5. Tf-Idf Score trend for different party platforms – using mean



Plot 6. Tf-Idf Score trend for different party platforms – using maximum



Plot 7. Tf-Idf Score trend for different party platforms – using minimum



Plot 8. Tf-Idf Score trend for different party platforms – using summation

From the data we can see that the new term induced enlarges the difference of the score value between known frontier literatures and the test literatures. So it means that it has reduced the negative impact of the unique word number to the result. The trend changes and it has a more accurate reflect to the trend. So we used Plot 8 to see the popularity trend.

From the trend lines in plot 8 we can see that the popularity of Frontier words in party platforms remained in a low level generally before 1930s and rising after that until around 2000. This may because the frontier words were mainly used by farmers and cowboys in the west and not widely used by politicians at that time in formal documents. As time passes, some of the frontier words may be used gradually more often in formal situations and politicians nowadays also tend to use more less formal words to get votes from people with different backgrounds.

Future works

As we got feature vectors for each document, we could try to use machine learning classification algorithms in the future works to see the probability of each test document being related to frontier topic or have a frontier writing style. Then by comparing the performance of machine learning algorithms to the method we used, we will know if our dimension reduction is proper and we will probably get the best way of investigating this problem. Also in order to get a clearer trend, more complicated trend extraction algorithms (e.g. Fractal Adaptive Moving Average and Hull Moving Average) can be implemented.