Building a Popularity Index of Politicians through Twitter Tweets Analysis

Introduction and problem description:

Over the past few years, opinion polls have been an ever-increasing field of politicians who want to know public opinion about them through receiving feedback, which can help them improve their functioning, because of their desire to please the public at large and their constituency in particular. For this reason, such surveys have become extremely popular in general and especially during election periods. Such opinion polls are usually based on traditional methods such as telephone interviews, face-to-face interviews, street surveys, self-filling questionnaires in the mail, etc. The advancement of technology over the last decade has created new sources from which to obtain particularly large amounts of opinions on political and current issues, from sources such as social networks, news sites and blogs, the amount of data that can be extracted is infinitely greater than traditional methods.
Entering the Big Data era has brought with it a variety of methods for dealing with and analyzing such large amounts of information, from which insights and conclusions can be drawn, such as public opinion trends and public opinion on political issues. These include reviews of news and current events involving politicians on a daily basis, which may vary according to their functioning and influence the popularity they gain in the public.
In this project, the problem we will be addressing is building a personal popularity index for four key politicians, which varies over time based on user reactions to political events in which they are reviewed. These metrics represented positive, negative, and neutral opinions about them. This will be done based on the gathering and analysis of numerous responses to political / current affairs articles published in the New York Times in the United States from 2017-2018. To build the popularity metrics of every politician, we will analyze the responses of these articles to determine whether they are positive, negative or neutral. By analyzing a large number of responses, far greater than any traditional survey, we can formulate a public opinion weighted on the politician in question and build for him a measure of popularity that varies over time according to the publications he is involved in, expressing public opinion about his functioning.

The data we used:

We used three different types of data in this project, with each type being designed for different use. We'll explain what types we used with an explanation of each:
1. All articles and articles in the New York Times during January-May 2017 as well as January-April 2018. These are nine CSV files that describe all the articles and articles published during these months in the newspaper. Each file contains entries that each represent an article that was published that month, and each article details such as its unique identifier number, site publication date, title, web address, and associated list of indications. Each such file has an average of 1000 entries representing 1000 articles. A total of 9 such months total about 9000 articles published in the newspaper during this time period. Weight of each file averages 509KB, total data volume is 4,581KB.

2. All comments posted by users to articles and articles published in the New York Times during January-May 2017 and January-April 2018. These are nine CSV files describing all comments posted by users to all articles and articles published in the newspaper in the months mentioned above. Each entry represents a comment that a user wrote to the article, with

the exception of the comment language itself, the date of publication of the comment, a unique ID of the article for which the comment was posted, a unique ID of the comment, and the geographic location (U.S. city and state) of the author who wrote the comment. Average records totaling 2,250,000 responses to articles published in the months listed above, the responses to these files are the source through which we build the politicians' popularity metrics. Weight of each file averages MB160, total data volume in this section 1.43GB.

The above data is at https://www.kaggle.com/aashita/nyt-comments.

3. A tagged data of Twitter comments. These are 1.6 million (1,600,000) tweets of comment on political / current / news topics gathered through the Twitter API. Each entry in this repository contains a tweet that is a response to some political / topical issue as well as tagging a number that indicates whether it is a positive or negative response. The responses in these files are the source of the classified training we will use and their accuracy test. Total data volume in this part MB 233. This repository is at https://www.kaggle.com/kazanova/sentiment140. Total data volume used is 1.66GB.

Our Solution:

As mentioned in the introduction, in this project we will analyze user responses to public opinion and to formulate general opinions about elected politicians, from which we would like to build popularity metrics that change over time. To that end, we have taken two major approaches to analyzing responses: the first is lexicon-based and the second is machine-learning-based. Before describing each, first, I describe how the relevant responses we want to analyze are collected from the Datasets mentioned above.
First of all, we extracted from Dataset # 1 all the articles related to the politicians we would like to analyze, which deal with the events they are involved in which have political influence. Each entry representing an article has a field containing indicative tags associated with the article. If the article deals with or is related to a particular politician or the events he is involved in, the name will be included in the article's tagline. We will build such a tag list for every politician and scan all records in the data files for articles containing tags related to it. If an article contains one of the tags, we will save the identifying number for that article. In this way, we will keep all the identifying numbers of articles published in a particular month related to the politician of our choice.
Then, we will extract the user's responses to relevant articles for the politician, using the identifying numbers we have saved, and then scan the comment files in Dataset # 2. Each record representing a comment has a field with a unique identifier representing its article. By listing the IDs we have saved, we will collect all user responses to articles whose ID was in the ID list. In this way, we will gather all of the user comments to articles related to the elected politician.
In order to create a time-varying popularity index, for each politician, the following metrics for each method (lexicon or ML) were calculated separately:
1. General Monthly Popularity Index by analyzing the responses collected on a particular politician over a full month period (months are January-May 2017 as well as January-April 2018). This is done through a serial pass on the reactions related to the elected politician who was received from the process described earlier. For each such response, we will

classify the emotion expressed therein according to one of the classification methods we will elaborate on later. In the lexicon-based method the classification moves on a scale of values of very positive emotion, slightly positive, neutral, slightly negative or very negative. In the ML based classification, the emotion moves on a positive, neutral and negative value scale. At the end of the calculations, the distribution of opinions for that month is calculated and displayed as a pie chart.

2. Monthly Popularity Index of a particular politician by country affiliation. Here, too, the classification is done separately for each response for the two approaches mentioned. For each relevant response from the response list, the emotion expressed in it will be categorized similar to the general monthly index, but the state from which the response writer comes will be taken into account, based on the position field in the response file record. Thus, the distribution of emotions expressed in responses according to affiliation to the United States is calculated and the results are presented in a table describing the distribution of opinions in each country as a percentage.

3. Index of sub-monthly popularity by analyzing the responses collected on a particular politician over a one-month period, ie 10 days. In this index, each month will be divided into first, second, and third trimester and in each such third, the distribution of opinions for each classification method will be calculated separately. Each entry in the comment file has a field that states the date of publication of the response, which allows the responses to be sorted in chronological order of writing, thus effectively dividing each month into three parts and allowing the distribution of emotions in responses to be shorter in more than one month, for only 10 days.
We will now explain how the classification methods we use will be explained:
Lexicon-based method: This method classifies text according to predefined words that appear in it. This method relies on lexicon indicative words to express English language emotions, with each word appearing a certain score. Each realization of the method computes differently the sentiment expressed in a particular text given as input. In order to perform lexicon-based classification, we have chosen to use the textblob directory which holds a TextBlob object that allows a variety of actions to be performed on text. We will use the polarity function under the Sentiment class of the TextBlob object. This function receives a string of text and returns a tupple whose left limb is a real number from 1 to 1 expressing the emotion expressed in the text, with 1- expressing a highly negative emotion, 1 expressing a very positive emotion, and 0 expressing a neutral emotion. This method uses a module called pattern.en which holds a lexicon of adjectives (such as good, bad, amazing, irritating) that appear frequently in the text. For each such word, a predefined score is attached, which varies by being a word that expresses a positive / negative emotion, with the score given to a particular text based on the adjectives it contains. Text classification that represents a response using this method will be as follows: In receiving some text, a TextBlob object consisting of that object is created. At this point the polarity function of the sentiment class will be run, which will return a value between 1- and 1. For the returned value, five levels of emotion will be defined: extremely positive (between 0.3 and 1), slightly positive (between 0 and 0.3), neutral (0), slightly negative (between 0.3 and 0), and most negative (between 1- To 0.3). 5 counters will be defined for each of the values, and this function will be run on each of the

responses received as relevant to each of the politicians. Thus, for each set of responses in each category listed above (full-month response emotions, by country affiliation, and by sub-month breakdown), the percentage distribution of opinions will be calculated according to the scale set out above and the results presented in pie chart form.

Machine learning-based method: In this classification method, no lexicon is based on predefined indicative words but on machine learning-based methods, using classifiers to classify text that represents some response as expressing a positive emotion or negative emotion. This method involves two steps before we use it to run the New York Times comment file from which we will build the final index.

The training phase: The training phase is classified. To do this, we must use a pre-tagged data that we will trust the classifiers to. The data we will use will be Dataset # 3 detailed above in the form of 1,600,000 tweets for political / news events collected from Twitter. This tagged repository is suitable for use for our specific purpose, since these are responses to political / news events, which are similar in terms of vocabulary and style of writing and are in the same domain-text, which will help in training the classifieds in terms of selecting features used. We will do the training in three separate steps:
1. Noise Removal - The preprocessing phase of training. We'll take 15,000 positive and 15,000 negative, total of 30,000 responses (this number is determined by the RAM usage limit during training, more than GB16 consumes on your computer, which is not present on most PCs)). We will define the part of speech of the words that will be for us pitchers, which will only be based on adjectives, nouns (including proper nouns) and verbs. Now we will build two lists of responses we took: a list consisting of all the words of the selected responses that are part of the POS we set (The rest of the words and punctuation will be thrown) when the letters in them are converted to lowercase, and another list of tupples, each tupple containing a comment and tagging Yes positive or negative We will take the 10,000 most common words from the list of words, which will form the basis for the next step, and save the two lists we created as reusable pickle files if necessary (saving the time needed to produce them each time).

2. Feature Selection - The feature creation stage. We will compile a list of tupples using the tupples list of comments and tags we created in the previous step. The right-hand organ in each tupple will label the response as positive or negative and the left-hand organ is a dictionary of 10,000 keys and values, with each key being one of the 10,000 most commonly found words and its value being a boolean variable representing whether that word is in the current response. This dictionary contains the features as boolean keys and variables as the value for each key that represents each word in the 10,000 most common words whether it is part of the current response, combined with the original tagging of the response. We'll now mix this list randomly, then compile 80 percent of it (24,000 responses) as Training Data and 20 percent (6,000 responses) as Testing data.

3. Sentiment Extraction - Define 5 different classifiers that will use supervised learning for the Training Data we created. We will use numeric scikit-learn classifiers and are: Naive Bayes, Multinomial Naive Bayes, Bernoulli Naive Bayes, Logistic Regression and Linear Support Vector. Trust each of them to the Training Data and then save them as quick reusable pickle files without the need for repetition each time, a fact

that will save a lot of time when we run them for comment from the New York Times.
Testing Step: Now we will apply each of the five classifiers we built on the Testing Data to calculate the accuracy of each classifier. In this metric, calculate the percentages that Justice classified from all the ground truths of the responses in the Testing Set:
Testing Data Accuracy (%) vs Classifier
78.45 Naive Bayes
79.11 Multinomial Naive Bayes
79.23 Bernoulli Naïve Bayes
81.95 Logistic Regression
84.68 Linear Support Vector

Reminded that these are the results for the Testing Data tagged from the Twitter tweet (used for training classified and tagged accordingly) and not the NYT comment files from which we will build the politicians' popularity metrics (which do not contain any tags).
Once we are done with the training and initial examination of the tagged files, we will use the classifiers we prepared for the relevant NYT responses. The ML responses will be categorized into the categories on which the lexicon classification and distribution spread over a full month, based on state affiliation and subdivision. Because in the Twitter repository, the comments were labeled as positive or negative only (there are no comments labeled as neutral), any classifier that will trigger a comment will be whether it has only positive or negative emotion. Therefore, in receiving any text that represents a specific response, we shall state that the classification algorithm will be run in the voting system model configuration: Instead of receiving 5 different final answers from each of the classifiers for a response, we will define a voting system where each classifier gives an answer whether positive or negative is expressed. If there is at least 80% agreement on the emotion in the response, ie 4 out of 5 classifiers return the same result, the response will be classified accordingly as having a positive or negative emotion. If there is less agreement, that is, 3 classifieds give one result and 2 another, the reaction will be classified as neutral. For this reason, the value scale in this method for each response will include 3 levels: a response that expresses a positive, neutral, or negative emotion. To build a representative sample of randomly sampled independent responses that will be used to build the popularity metric, in each category we will randomly mix the list of responses we would like to analyze and also limit the number of samples to a maximum of 6,000, as the Testing Set size we used for accuracy. Tagged from Twitter. Similar to the lexicon method, for each category, the distribution of emotions expressed in percent responses will be calculated according to the value scale indicated above and the results presented in pie chart form.

Experiments:

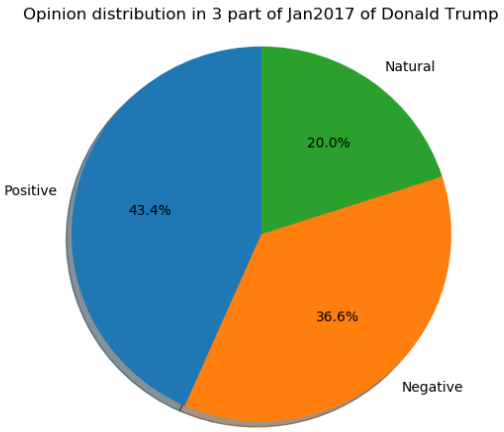
Evaluation Criteria:
In general, the evaluation criteria for models that include text categorization according to emotion can include metrics such as Accuracy, Precision, Recall and F-score with which the test model performance can be assessed. All of these require tagged data with the help of measuring each one.
Because we build long-term popularity metrics for politicians based on untagged NYT repository, our evaluation criterion will be based on comparing US and national research and political / news / current trends and in-depth public opinion studies Reviewers are politicians, and the

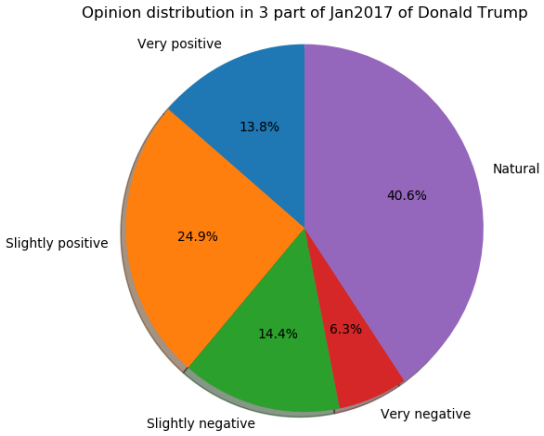
main bodies we will compare to the results we have received will be, for example, new companies, newspaper systems, strategic and business consulting companies, markets and research institutes and universities. Since these bodies are skilled in conducting in-depth political research and trends in society, we will compare the results we get to the results published by them in order to evaluate the results we received and make sure they are not the case.
It should be noted that at the time we examined the Testing Data classifier from Twitter that was tagged (but not used to build the final popularity index), our evaluation criterion did include the Accuracy metric and the results are listed in the table above.

Setup:
First, we will run the classifier training on the file with the examples tagged as above and save them for quick reuse in the folder we are working with the pickle files, a fact that will save the need to retrain every time you run.
We will perform the experiments themselves chronologically according to the 3 categories listed above in our Solution section. The order of the politicians will be Donald Trump, Hillary Clinton, Barack Obama and Benjamin Netanyahu in both classification methods. Each politician in each of the month order classification methods will be January-May 2017 and then January-April 2018. The order of running within each month will be the collection of articles and articles related to the same politician that month, gathering user comments for the same articles, calculating the distribution of emotions in response to the whole month, calculating the distribution The emotions in the reactions that month according to state affiliation and finally the distribution of the emotions in the reactions according to each month divided into three parts and the distribution of the first, second and third parts of each month separately. At the end of the process, each politician, in each classification method each month, will be created into the three categories of graphs describing the distribution of reaction emotions in the form of a pie chart that we will present in the following section.

Results:
From the process described in the previous section, we will generate a large number of graphs that are short of containing all of this report. All graphs will be uploaded to Google Drive under the link posted in the project forum along with all code and data. Therefore, in this section, we will present selected results of all types along with the data with which The comparison and at the top of each result is a brief description of what it presents, followed by an explanation of the difference between lexicon-based and ML-based results.

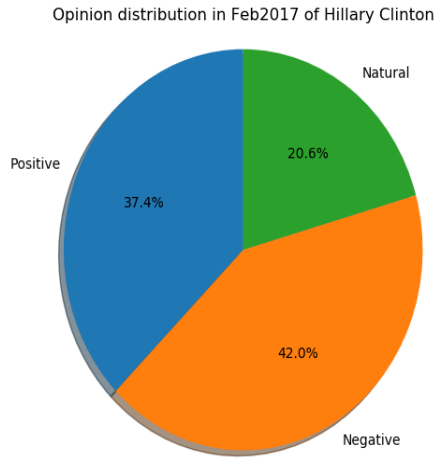
Distribution of emotions in the last trimester of January 2017 for Donald Trump: right lexicon classification and left ML classification. Below are results from Wikipedia of several US research institutes this month:

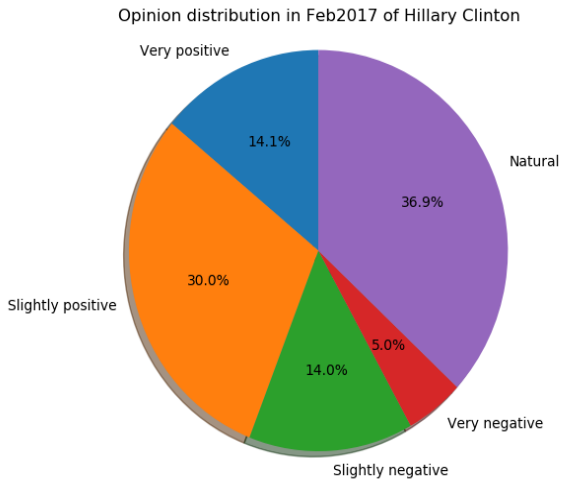


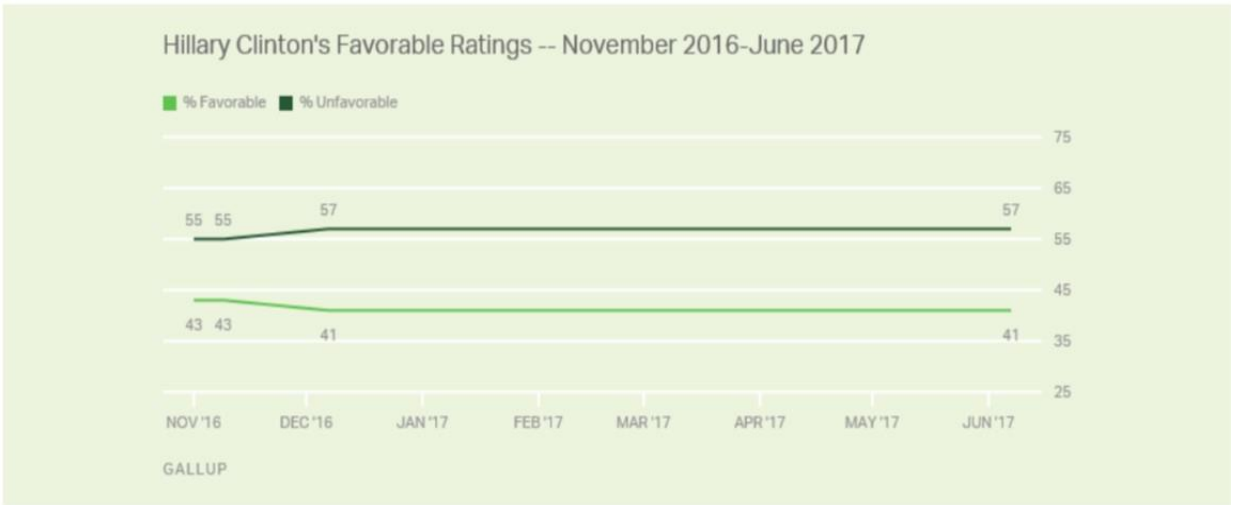


Area polled	Segment polled	Polling group	Date	Approve	Disapprove	Unsure	Sample size	Polling method	Source
United States	All adults	Quinnipiac University	January 20–25, 2017	36%	44%	19%	1,190	telephone	[447]
United States	Registered voters	Public Policy Polling	January 23–24, 2017	44%	44%	12%	1,043	telephone and online	[448]
United States	All adults	Gallup	January 22–25, 2017	46%	45%	9%	1,500	telephone	[449]
United States	All adults	Ipsos (for Reuters)	January 20–24, 2017	43%	45%	12%	1,282	online	[450]
United States	All adults	Gallup	January 20–22, 2017	45%	45%	10%	1,525	telephone	[451]

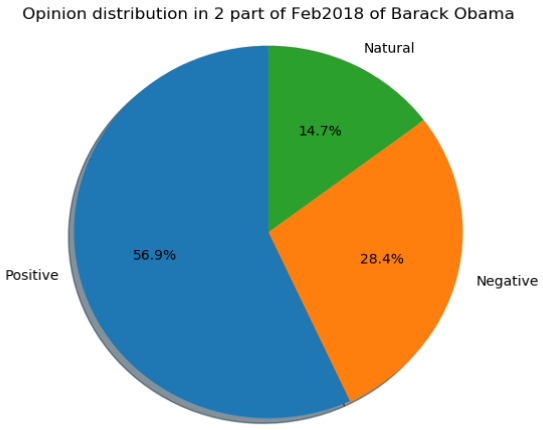
The Distribution of Emotions in February 2017 Responses for Hillary Clinton: Right to Lexicon Classification and Left to ML Classification. Below are results for comparison from the American GALLUP Research Institute this month:

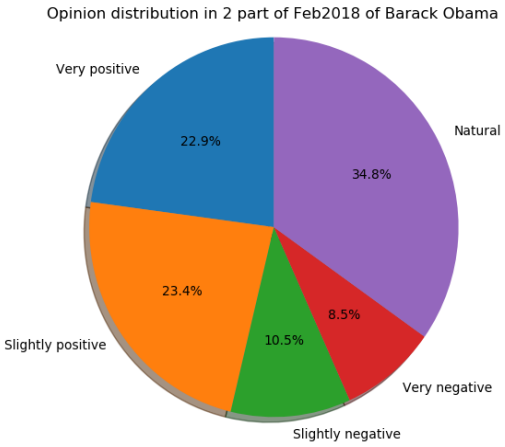


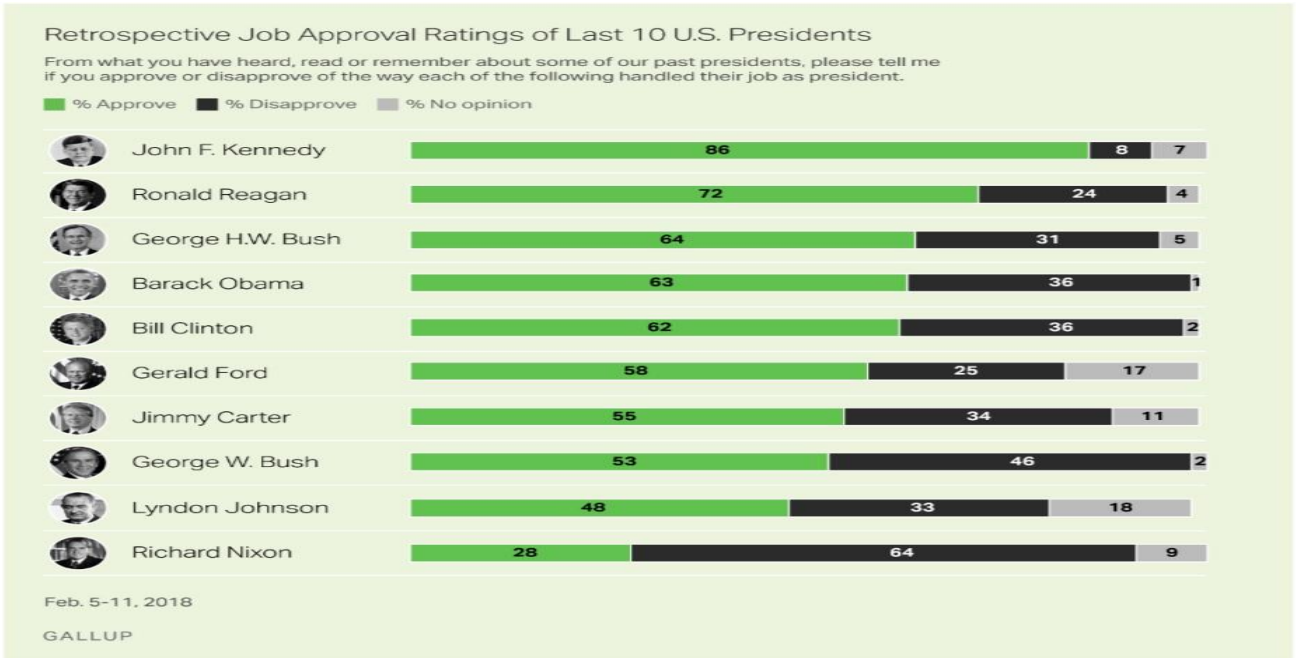




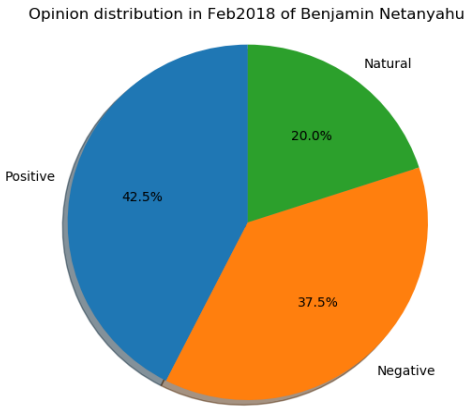
Distribution of emotions in the second trimester of February 2018 for Barack Obama: right lexicon classification and left ML classification. Below are results for comparison from the American GALLUP Research Institute this month (Barack Obama in fourth row):

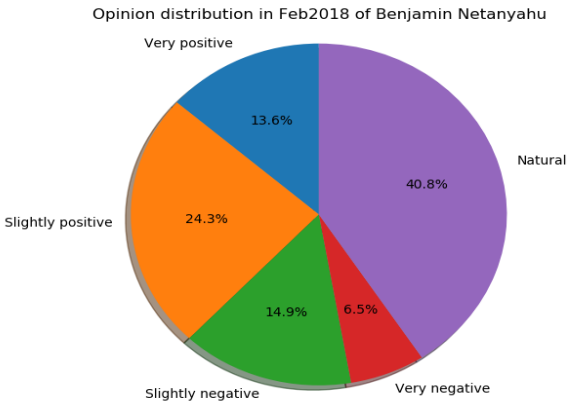


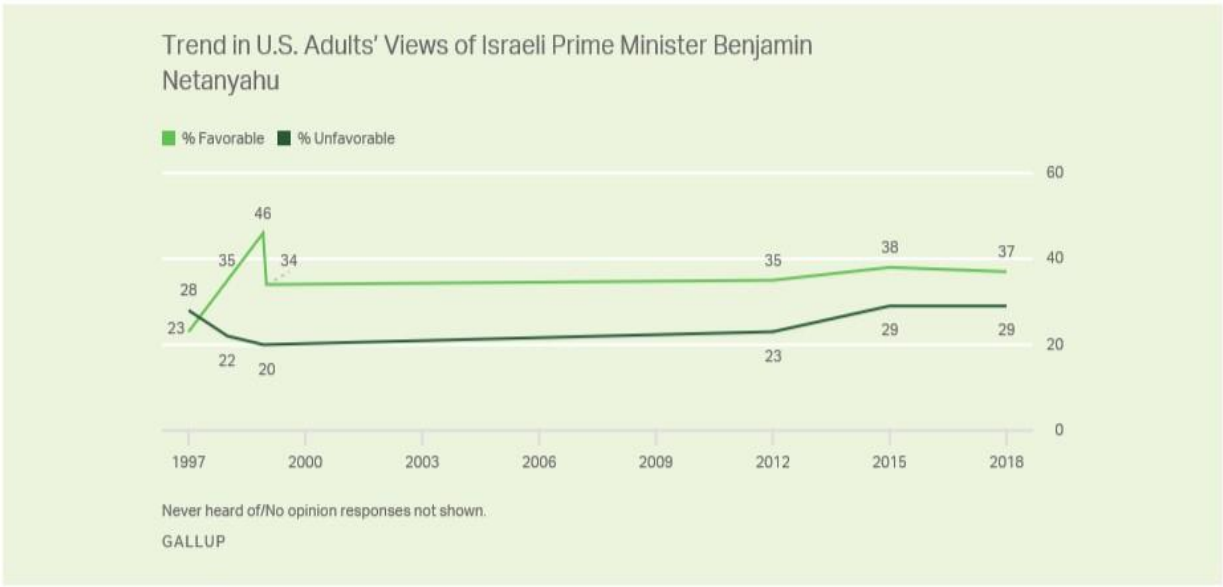




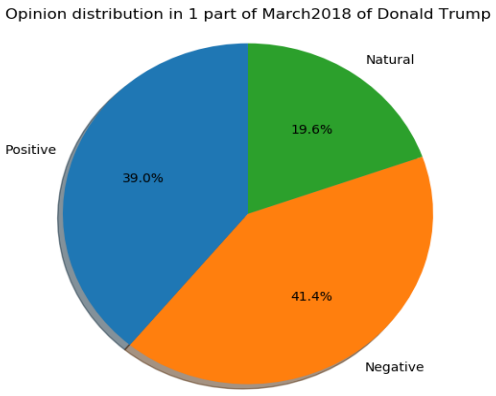
Distribution of Emotions in February 2018 Responses for Benjamin Netanyahu: Right Classification of Lexicon and Left ML Classification. Below are results for comparison from the US GALLUP Research Institute this month (February 2018 on the far right):

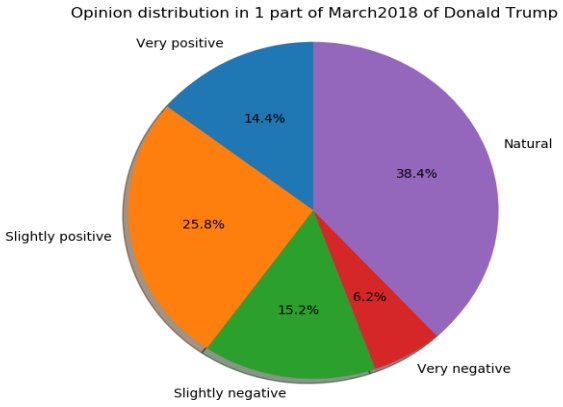






Distribution of emotions in the first trimester of March 2018 for Donald Trump: right lexicon classification and left ML classification. Below are the results from Wikipedia of two US research institutes this month:





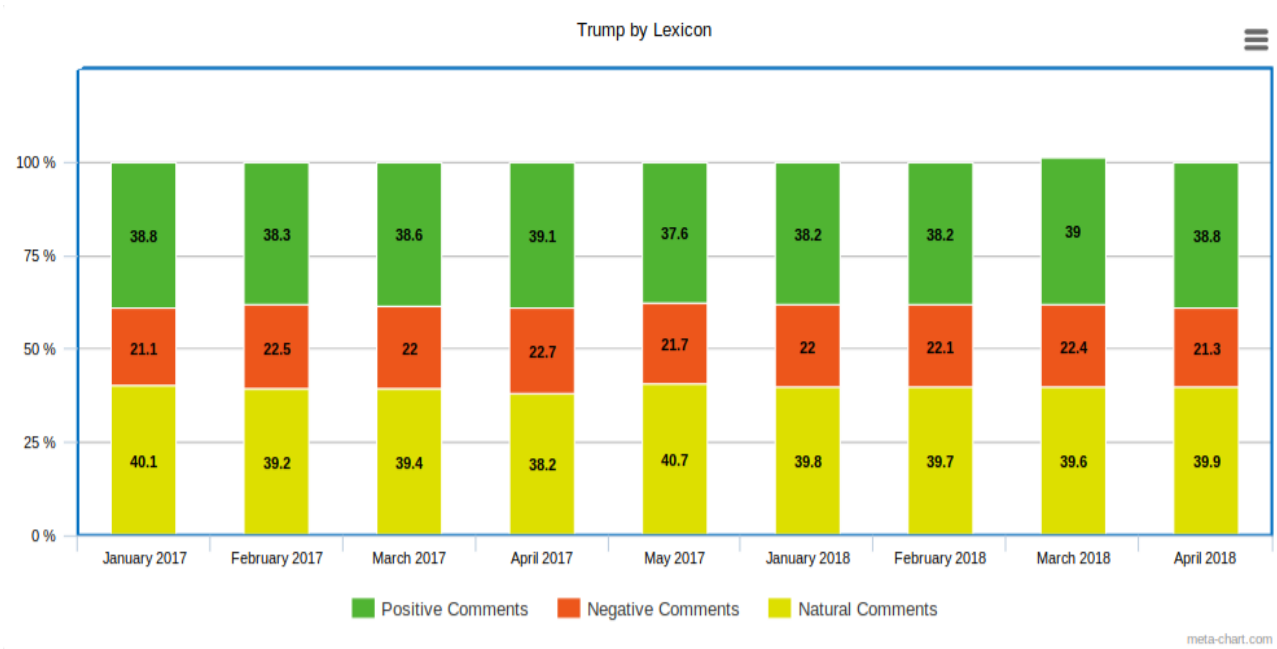
Area polled	Segment polled	Polling group	Date	Approve	Disapprove	Unsure	Sample size	Polling method	Source
United States	All adults	Marist	March 5–6, 2018	42%	50%	8%	1,050	telephone	[320]
United States	All adults	YouGov (for *The Economist*)	March 4–6, 2018	40%	50%	10%	1,500	online	[321]

As can be seen, compared to results obtained from institutional research institute studies, the results of emotion analysis based on ML data are closer to reality than the results of the lexicon-based analysis. This is reflected in the gap of individual percentages (up to a maximum of 10 percent in each measure of positive, neutral, or negative emotion) in ML-based analysis and up to several tens of gaps in lexicon-based analysis. We also note that both methods better classify positive responses than negative responses, in most cases the index expressing positive emotion closer than the index expressing negative emotion to the results of the research institutes. We will try to explain the meaning of these results:
1. Lexicon-based classification gives a direct result from syntactic analysis of the sentence. He does not try to determine whether the sentence contains any positive or negative meaning since he works with adjectives and not with his general structure. This means that it checks whether the adjective alone is in a negative or positive context. This may be a problem in cases where there is negation in the trial. For example, the statement "not the best" will have a polarity of about 1 and opposite the "the not best" will contain a polarity value of 0.5.
2. A textblob-based lexicon classification that we used uses a general English-language dictionary that is not tailored to politics-specific text domain. The classifieds' training was conducted on a data-based response to political events taken from Twitter.
3. The amount of features in ML that we used (10,000) is greater than the amount of English textblob lexicon adjectives estimated at slightly less than 3,000. Therefore, after training, the classified "recognized" a greater number of words (including adjectives) that do not appear in the lexicon. The English textblob lexicon can be found at: https://github.com/sloria/TextBlob/blob/eb08c120d364e908646731d60b4e4c6c1712ff63/textblob/en/en-sentiment.xml
4. Unlike the lexicon, the various ML classifiers do indeed try to test whether the sentence contains a positive or negative meaning as a whole and does not work against adjectives alone.
5. The use of a five-class voting system model that we used in ML-based classification is an improvement to the general classification algorithm over using only one classifier.

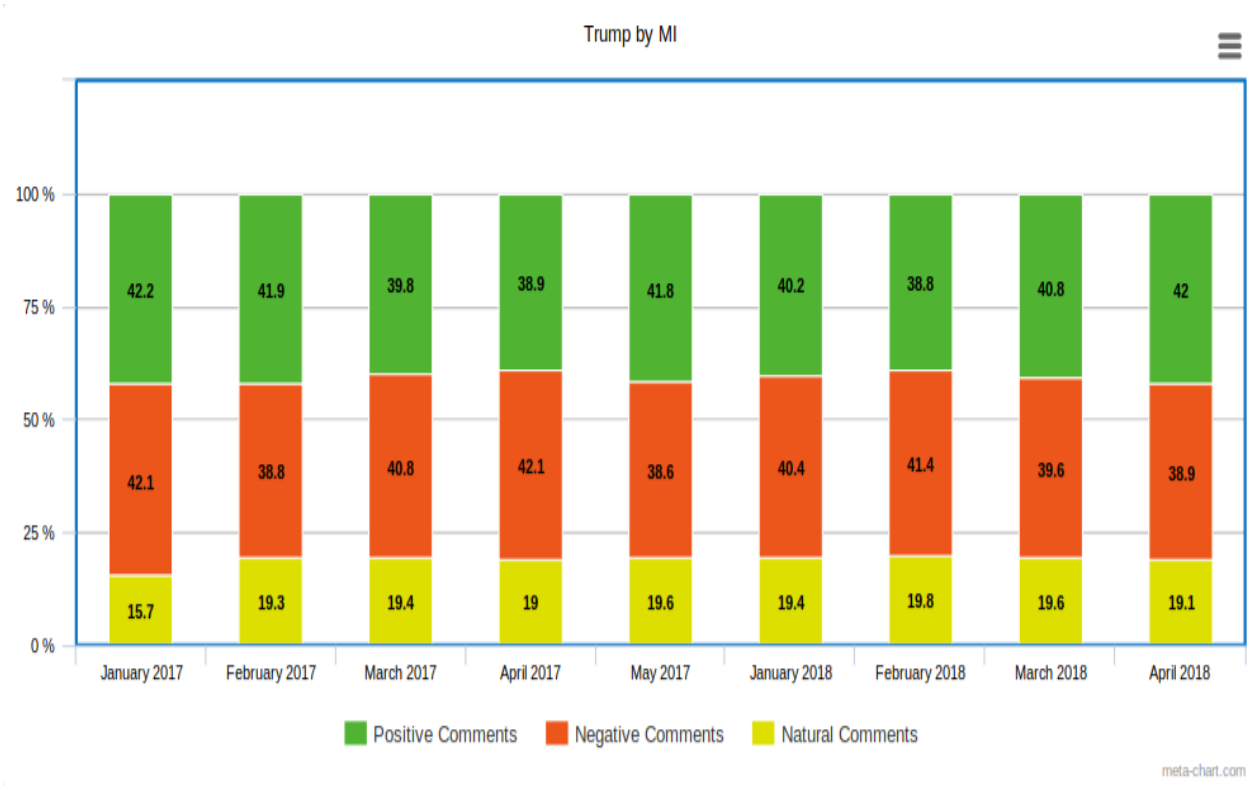
Visualization:

Now we visualize the results we got, show it, and then explain our selection:

Results of the time-varying popularity index for Donald Trump using lexicon-based analysis:

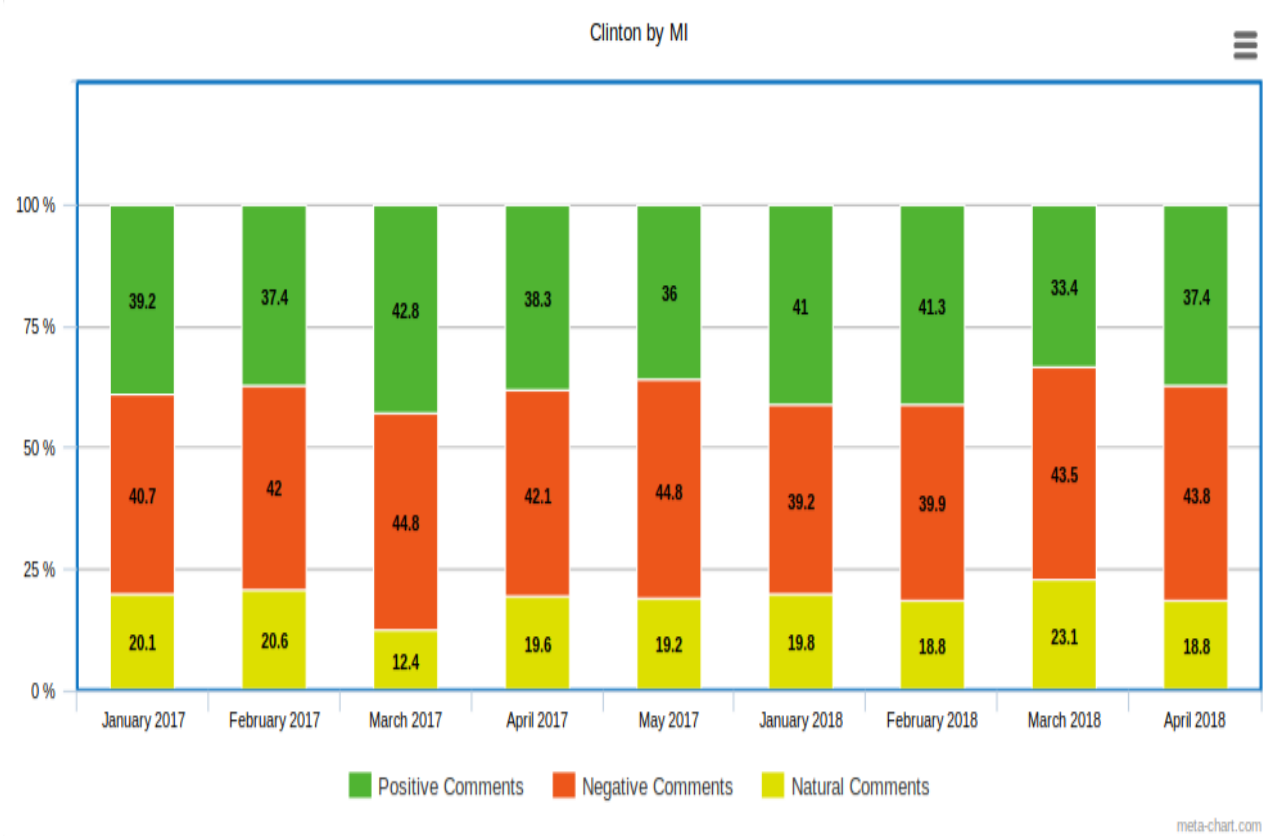


Results of the time-varying popularity index for Donald Trump using ML-based analysis:



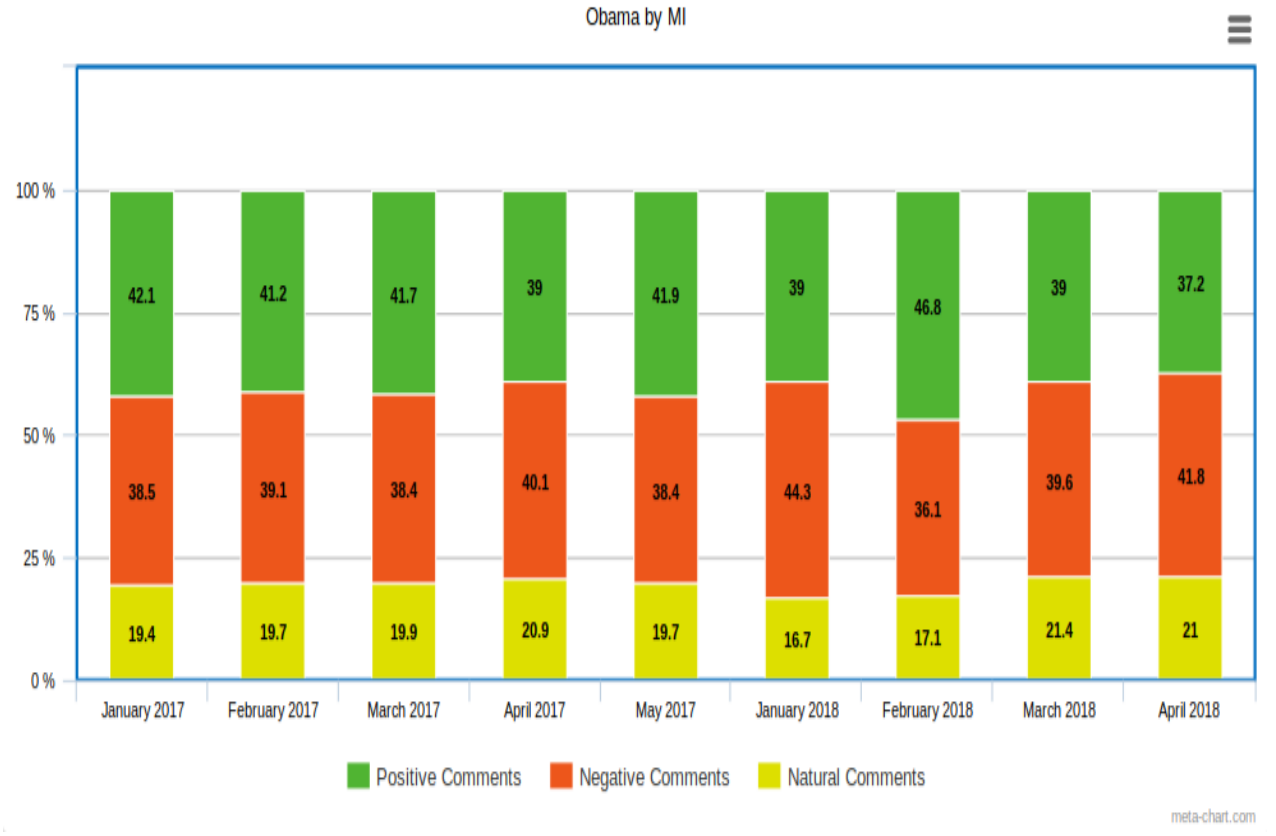
Results of the time-varying popularity index for Hillary Clinton using lexicon-based analysis:

Results of the time-varying popularity index for Hillary Clinton using ML-based analysis:

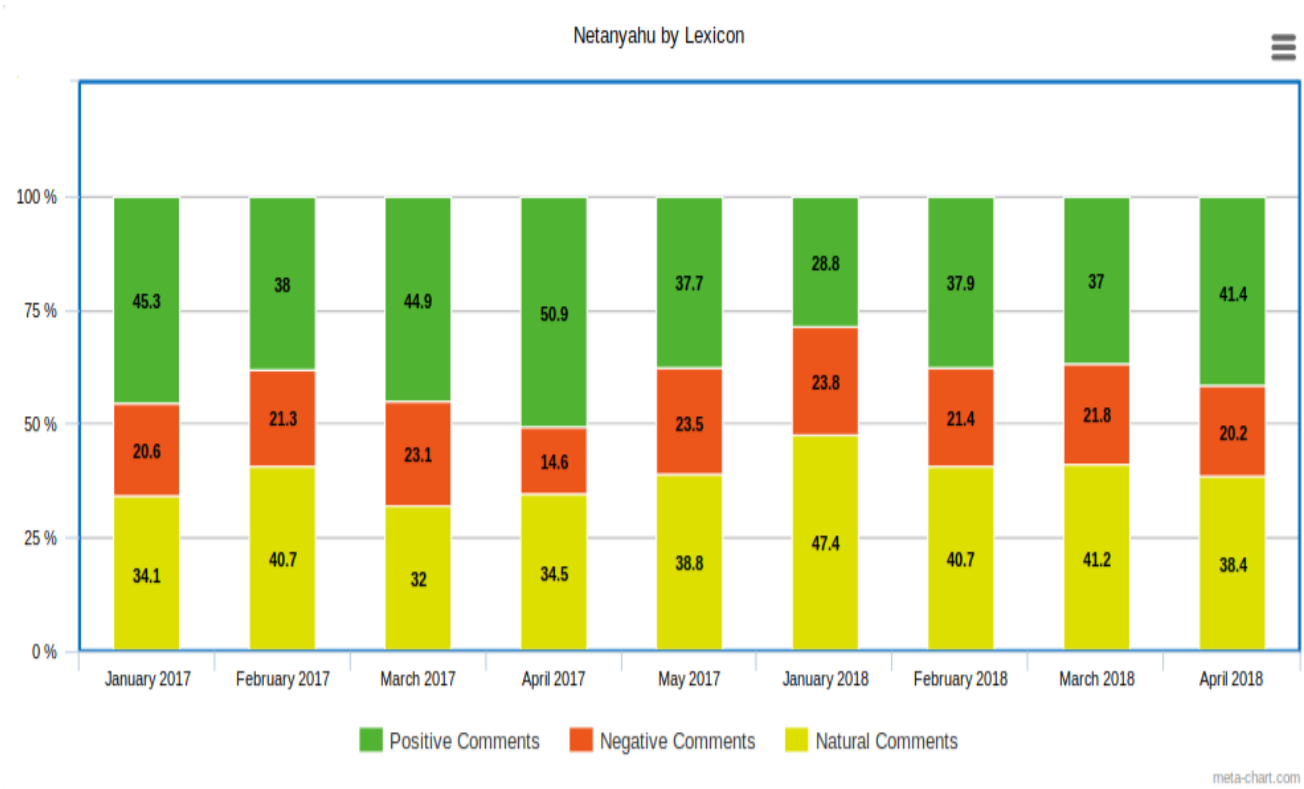


Results of the time-varying popularity index for Barack Obama using lexicon-based analysis:

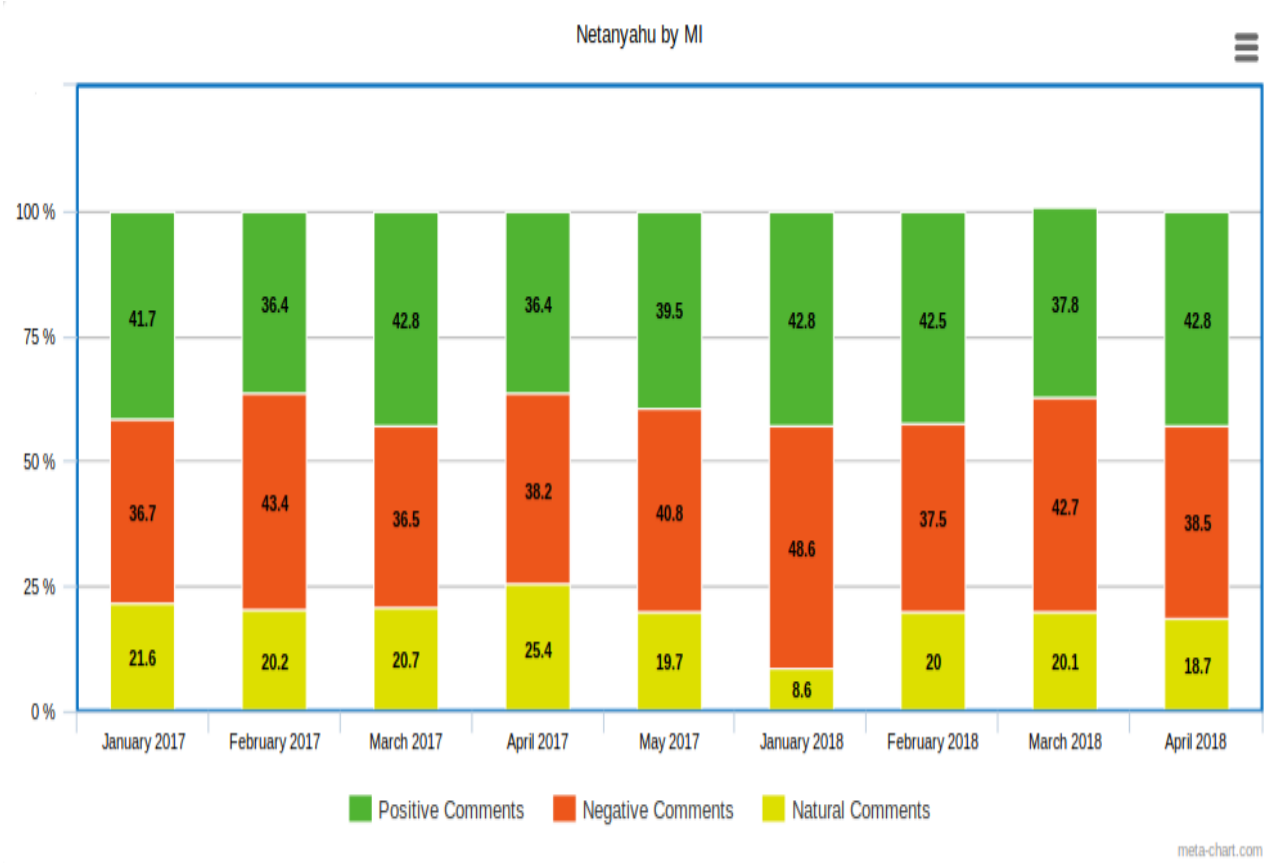
Results of the time-varying popularity index for Barack Obama using ML-based analysis:



Results of the time-varying popularity index for Benjamin Netanyahu using lexicon-based analysis:



Results of the time-varying popularity index for Benjamin Netanyahu using ML-based analysis:



We chose to use the subdivided bar graphs of visualization since it is a good idea to describe the ever-changing popularity index we wanted to present.
By presenting this, you can clearly see the time-varying index for the selected nine-month period for each of the politicians, for each of the classification methods. It shows the distribution of positive, neutral and negative opinions in each of its columns for all these months.
In addition, this visualization can be attempted to relate interesting connections between events that happened in reality related to each of the politicians, and how they affected their public popularity and how it was interpreted in each of the classification languages.
For example, in mid-February 2018, a number of extensive articles were published about the criminal cases against Netanyahu in the New York Times. Looking at the graphs, it can be seen that the publication of the article on the one hand increased Netanyahu's percentage of support by almost ten percent from January according to the lexicon, and on the other, reduced the percentage of negative feelings toward him by a similar number from January according to the ML analysis.
These changes stem from taking the position of the surfing public exposed to the affair. What is interesting about these results is that the surfers are not in a hurry to believe media publications regarding the investigations that are being published, in which negative publications even increase the positive sentiment towards Netanyahu, or alternatively reduce the negative sentiment towards him.

Impediments:

During the Experiments, we faced the following issues:
1. Classifier training takes a very large amount of runtime and RAM.
2. To deal with the time problem, we used a Python pickle object that lets you save the post-workout classifiers as pickle files. The files must be placed in the folder where the code is run and can be run repeatedly without having to retrain them every time.
3. To deal with the place problem, we limited the Training and Testing data size to a maximum of 30,000 responses. In addition, there is a requirement for more than GB16 of RAM on the computer running the classified training code.
4. The use of a voting system model to classify the responses in the NYT (post-training) should be conducted in an environment similar to that in which the classifiers were tested on the tagged Testing data. Therefore, to obtain a representative sample of independent responses that we will analyze in each category, and to obtain results in the same environment as we had when running Accuracy testing, we randomized 6,000 responses (as the size of the Testing data) in each category we analyzed.
5. Classifying a very large amount of comment text is a matter that takes a great deal of running time (tens of hours). How did we deal? Lots and lots of patience :)

Future Work:

As a sequel to future work, we can take into account the following fun points:
1. Attempt to further improve the accuracy of the results obtained through the implementation of hybrid classification methods - combining ML-based and lexicon methods together that work together to achieve even more accurate results.

2. Attempt to make prediction about candidate by comments. That is, given a response, try to predict which politician she wrote (without her explicit name of course).
3. Attempt to build a public influence measure on a politician - how constituent events of various kinds affect the popularity index of politicians and how much they rise and fall as a result.
4. Continuing Section 3, Building Recommendation Systems for Politicians on How to Act in the Public Arena to Maximize Their Public Sympathy Based on Analysis of the Effect of Past Constituent Events and Their Current Index of Popularity.
5. Analysis and attempt to predict what the public will see as fake news, such as negative publications about politicians who actually reinforce their positive image and raise their sympathy in the public.

Brief conclusion:

In this project, we analyzed user comments in the New York Times and built time-varying popularity metrics based on analyzing a very large amount of user responses to articles and articles published in the paper. We used two major methods to accomplish the task and both lexicon-based and ML-based methods. We explained how we defined, created, and applied each of these methods to the relevant NYT pool of responses to build the Politics Popularity Index. We presented the results obtained together with a comparison with the data collected from research conducted by organizations and bodies engaged in the field and described selected cases, together with the reasons for the differences between the results of the two methods. We then presented a visualization of the results, explained why it fits with the politician's popularity index and what can be deduced from it with an example. To conclude, we had to deal with a number of difficulties during the work on the project, as well as interesting future development options that could be implemented later.