Introduction

The purpose of this project was to utilise data involving Trump's tweets and to determine the popularity of his tweets across countries. Then we also wanted to see if similar reactions occurred if his tweets were mentioning a specific topic. To do this, we scraped Twitter to obtain replies to several of Trump's tweets and computed sentiment values on these to see if a person from a specific country talked about Trump in a positive, negative or neutral manner.

The reason why investigating these reactions are of importance is because it can give people an idea of where in the world Trump is most popular and what specific topics are causing the positive reactions. For example, a specific economic policy by Trump's administration may be beneficial to one nation but may hurt another and using the information ….

Related work

A lot of work has been done with classifiers that do a texanalysis directly on Trump's tweets.

Probably the most relevant work on Trump's tweets is David Robinsons' text analysis, which shows how the tweets can be categorized into "written by Trump" or not by classical manual methods (such as on the basis of which device they were posted with). Robinson's work confirms our findings on the two Trump's personalities: benign versus angry. Our contribution is however the world reaction to the two different tones, regardless of the author of the tweet.

Throughout this work we applied a specific machine learning technique for topic detection, namely Latent Dirichlet Allocation (LDA in short). This approach combined with tweets sentiment analysis is not new and it has been noted by many authors how it is remarkably less performant than in other contexts. Indeed, the limited size of a document (a tweet) implies a limited amount of observations that the model can make to learn its parameters. As noted by Alvarez-Melis & Saveski (2014), LDA is meant to perform well on lengthy documents and can be easily hard to interpret on tweets.

Data collection

While a lot of the data was already collected and processed for us, we still had to collect some data on our own using the Twitter website. Trump tweets were already available in a condensed format and contained all of the features we required. To obtain the replies, we would look at the id of every Trump tweet and create a specific url using that id. When the url is visited, it shows the tweet on the twitter website as well as several replies to the tweet. We then scrape that page to obtain several replies to each tweet.

Methods

The way that the textblob library calculates sentiment values (polarity) of sentences using the NLTK package's Naive Bayes classfier.

Clustering of tweets was implemented via topic modeling, more specifically via Latent Dirichlet Allocation (in short LDA). In LDA the documents of a corpus are generated according to a mixture of (dirichlet) word distributions over a vocabulary of fixed size. Guiding intuition, such word distributions are named "topics". Each document is characterized by a per-document (dirichlet) topic distribution and a total number of words N. Each word of the document is generated by first sampling a topic assignment out of the per-document distribution and then by sampling a word out of the word distribution. The machine learning task is to backtrack this process to "infer" the latent topic distribution of the document. Given the model and the per-document distribution, a corpus can be fully characterized (up to words order, since documents are bag-of-words). This entails that if the number of topics K is s.t. K << D, where D is the length of the dictionary, then under LDA every document has a more succinct representation.

Pre-processing has been performed iteratively: at each step, the output of simple functions and tokenization was analyzed seeking for the most occurring incorrect tokens. After determining the cause of the incorrect tokenization, a rule was written to fix it. This process was repeated until, after a thorough inspection, most of the tokens were meaningful. Particular care was devoted to this phase since the performance of clustering algorithms generally depends on the underlying data. The pre-processing included shifting to lower case, stripping links, dates and time. Punctuation was ignored except for the dot and the apostrophe since many proper names include these symbols (single or multiple periods are still ignored however). Other accepted symbols are "@" and "#" at the begin of a word. Stopwords were stripped out. Finally, stemming was applied, even though we found out some cases of ambiguity and reduced readability. We decided to deploy stemming anyway since the various verb conjugations such as "said" and "say" were a far bigger issue than some small inconvenience. Moreover, readability was maintained by keeping track of both processed and unprocessed tweets.

Right after feature pre-processing, a crucial step in clustering is determining the number of clusters. We started with a non-small number of topics (k = 100) attempting to first overfit the model to the training set (but still keeping a reasonable amount of topics for manual inspection) and then to progressively reduce the number of topics to increase performance. The goodness of fit was measured by first splitting in training and testing samples, inferring a test tweet's most probable topic and then assigning the tweet to that topic (hard clustering). However, this method seemed to be slow since every model seemed to derive small topics of approximately the same size (10 ~ 80 tweets each, where the biggest topic was 4% of the whole dataset). While selecting the number of clusters just so that the dataset would be nicely partitioned in bigger chunks was tempting, we were not sure whether a small cluster could still have a great impact in clustering. To this aim, many of the 100 topics were inspected and we found out that some topics were fairly similar (for example all topics addressing Ted Cruz were divided in several topics).

At this point, our aim became minimizing the "uncertainty" of the per-tweet topic distributions (i.e. by having higher maxima). In order to do so, we considered that in LDA a test sample document of D features is reduced to a K-vector in the K-topic space. We had the idea of computing the silhouette function in this space to improve goodness of fit. We decided to implement that using centroids (the average point of each cluster) and euclidean distance (which is equivalent to cosine distance since the K-vector can be interpreted as a probability distribution). Another test that would be interesting to make is to use the total variation distance since every point in the topic space is a probability distribution over K topics.. The maximum silhouette value was achieved for k = 2. This behavior is certainly suspicious. It might be that as K increases the clusters naturally become less and less distinguishable, i.e. less and less sparse. This behavior is unexpected since the sparsity hyperparameter alpha of the per-document dirichlet distribution scales according to the number of topics (1 / K according to the documentation). Nonetheless, the resulting 2 clusters seems to be of some use.

Results

Indeed, the two derived topics seems to have an interesting side-effect: the classifier acts as Trump's sentiment detector. It infers 1 when Trump attacks a controversial issue in his tweet, generally defending himself or attacking his political adversaries (Clinton, Obama, the Democrats, the FBI CEO...) in a manner that might resemble an angry rant. It is not unusual to read keywords such as "FAKE NEWS" or "rigged system".

Examples:

1. "FAKE NEWS media knowingly doesn't tell the truth. A great danger to our country. The failing @nytimes has become a joke. Likewise @CNN. Sad!"
2. "What is our country coming to when a judge can halt a Homeland Security travel ban and anyone, even with bad intentions, can come into U.S.?"
3. "James Comey will be replaced by someone who will do a far better job, bringing back the spirit and prestige of the FBI."
4. "Karen Handle's opponent in #GA06 can't even vote in the district he wants to represent...."

5. "Everybody is asking why the Justice Department (and FBI) isn't looking into all of the dishonesty going on with Crooked Hillary & the Dems.. "

In contrast, we observe a completely different pattern in the other cluster. Trump glorifies and thanks either his supporters, his government or even his family. He celebrates having met "great people" or festivities. In this cluster we observe a Trump that inspires people hope. Frequent keywords are "honor", "hope", "pray", "God", "kind", "jobs"...

Examples:

1. "My warmest condolences and sympathies to the victims and families of the terrible Las Vegas shooting. God bless you"
2. "Getting ready to celebrate the 4th of July with a big crowd at the White House. Happy 4th to everyone. Our country will grow and prosper! "
3. "Today on #NationalAgDay, we honor our great American farmers & ranchers. Their hard work & dedication are ingrained..."
4. "Thank you for such a wonderful and unforgettable visit, Prime Minister @Netanyahu and @PresidentRuvi."
5. "Nick Adams, "Retaking America" "Best things of this presidency aren't reported about. Convinced this will be perhaps best presidency ever." "

Although we did not detect specific topics concerning Trump’s presidency (relationship with North Korea, tax cuts, building the wall...) we decided to exploit this model to study how Trump’s tweets tone affects the reaction of people all over the world. In our model, Trump’s tone can either be provocative, sarcastic, aggressive or sympathetic, enthusiastic, respectful, passionate...

References

http://ai.stanford.edu/~ang/papers/nips01-lda.pdf LDA

http://socialmachines.media.mit.edu/wp-content/uploads/sites/27/2014/08/topic-modeling-twitter.pdf LDA on tweets

http://varianceexplained.org/r/trump-tweets/ Trump benign vs angry