# CS294 Project Proposal: Politwitics

Erin Summers and Kristin Stephens

March 16, 2012

## 1 Introduction

This is a presidential election year, which means many people will send tweets of a political nature. Tweet topics could range from political candidates to particular political issues. However with so many political tweets how do we get a sense of the overall view of a particular political topic?

Out goal is to summarize twitter's political landscape. We want to look at all the political tweets and cluster them based on their topic. Once we have these clusters we will provide summary text of all the tweets in that cluster. This will provide a much more simple and succinct way for users to get a general view of the political Twitter conversations.

We chose the political twitter landscape as our dataset because it is inherently a bimodal opinion space.

## 2 Background

Twitter is a unique collection of short statement no longer than 140 characters. Because of the concise document size, and the casual audience, the use of several types of slang abounds. In addition to misspellings of words and colloquial slang, it is typical for users to abbreviate words and Internet slang such as "LOL" for "laugh out loud" or less common slang such as "IMO" for "in my opinion." This creates a challenge since it increases the vocabulary.

Multi document summarization (MDS) is a type of news aggregation and is used to reduce a large corpus of documents into a concise summary, which expresses the key topics within the entire corpus. The advantage of employing MsDS is that a person can quickly ascertain the key points within a long list of documents, rather than having to sift through the whole corpus.

The MDS algorithms work by first discovering the top hidden topics within the corpus. Next, the distance of each sentence from the top topics is measured using various heuristics. A summary is constructed by piecing together the most relevant whole sentences for each topic. It is important to note that the corpus is treated as a "bag of words". Hence, the order of the words and the part of speech is not accounted for in the model.

MDS is typically applied to a corpus of long documents, rather than short tweets. Additionally, the typical input documents are assumed to be standard, coherent texts. Because the twitter data is so noisy and irregular, we foresee that some of the standard techniques may give us erroneous output.

## 3 Data Collection and Preprocessing

We plan to first set up a filter on twitter to receive all the tweets that include a political keyword. The political keywords are from a list of about 40-50 words we generated ourselves and include things such as: obama, romney, ricksantorum, cain. Notice "ricksantorum" stands for Rick Santorum, however since it is common to concatenate words together for a hash tag we will need to filter on many possible variations.

After looking at the provided tweet sample we determined that about 0.05% of the tweets are political. We expect that this will provide us with a manageable data size.

The tools we plan to use to process the data have not yet been decided. We are considering Twitter4J. Whatever we do decide to use will have to provide us with preprocess capabilities. As we work more on this project we will need to add heuristics to clean the noisy data. For example remove all tweets that are not actually relevant, such as any tweet that has the text "cain" but is actually about "cocaine."

# 4 Proposed Model

Initially, we will start with a simple Naïve Bayes model for the data. Our latent variable will be the topic clusters $c$. Representing the documents as $d$, we would like to maximize the likelihood that document $d$ lies within cluster $c$,

$$p(c|d) = \frac{p(d|c)p(c)}{p(d)}.$$

Since the clusters are hidden variables, we will employ the Expectation Maximization (EM) algorithm [1], which is an iterative 2 step process where the cluster values and model parameters are adjusted every other iteration to converge to the maximum likelihood estimate. We will also use a simple $k-$means clustering algorithm, to try a frequentists approach.

Based on these results, we will adapt our model and perhaps employ some more sophisticated generative model such as Latent Dirichlet Allocation [8] or the Gamma Poission [9] model. At this point, it would be foolish to impose a sophisticated model on highly irregular data such as tweets.

Several other papers [2, 3, 4, 5, 6, 7] propose measures for finding the "top sentences" for multi document summarization by using word counts or graph structures. We will utilize several of these methods and improve based on those results.

# 5 Proposed Analysis

Measuring the accuracy of our results is a difficult task, since the data does not come pre labeled. Given a set of topics, the tweets must be hand labeled in order to judge whether or not the tweet actually belongs in a cluster. If time permits, we will do a rough estimate by using volunteers or Mechanical Turk to label the tweets.

Another method for extracting value is by counting the number of retweets. If we have selected the most relevant tweet for a topic, the relevance is perpetuated if more people retweet that sentence.

# References


[1] Dempster, A.P. and Laird, N.M. and Rubin, D.B. "Maximum likelihood from incomplete data via the EM algorithm". Journal of the Royal Statistical Society. 1–38, 1977.

[2] Inouye, D. and Kalita, J. "Comparing Twitter Summarization Algorithms for Multiple Posts",IEEE Conf on Privacy, Security, Risk, and Trust. pp. 298–306. 2011.

[3] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, Beyond SumBasic: Task-focused summarization with sentence simplification and lexical expansion, Information Processing & Management, vol. 43, no. 6, pp. 16061618, 2007.

[4] D. Radev, S. Blair-Goldensohn, and Z. Zhang, Experiments in single and multi-document summarization using mead, DUC-01, vol. 1001, p.48109, 2001.

[5] G. Erkan and D. Radev, Lexrank: graph-based centrality as salience in text summarization, Journal of Artificial Intelligence Research, vol. 22, pp. 457480, 2004.

[6] R. Mihalcea and P. Tarau, TextRank: Bringing order into texts, inEMNLP. Barcelona: ACL, 2004, pp. 404411.

[7] S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine* 1, Computer networks and ISDN systems, vol. 30, no. 1-7, pp. 107117, 1998.

[8] Blei, D.M. and Ng, A.Y. and Jordan, M.I. "Latent Dirichlet Allocation". Journal of Machine Learning Research. pp 993–1022. 2003

[9] Canny, J. "GaP: A Factor Model for Discrete Data". Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. 122-129. 2004.