Controversial Topic Discovery on Members of Congress with Twitter

Aleksey Panasyuk, Edmund Szu-Li Yu, Kishan G. Mehrotra*

Dept. of Electrical Engineering & Computer Science
Syracuse University, New York, USA

Abstract

This paper addresses how Twitter can be used for identifying conflict between communities of users. We aggregate documents by topic and by community and perform sentiment analysis, which allows us to analyze the overall opinion of each community about each topic. We rank the topics with opposing views (negative for one community and positive for the other). For illustration of the proposed methodology we chose a problem whose results can be evaluated using news articles. We look at tweets for republican and democrat congress members for the 112th House of Representatives from September to December 2013 and demonstrate that our approach is successful by comparing against articles in the news media.

Keywords: Twitter; Latent Dirichlet Allocation; Topic Modeling; Polarizing Topics; Semantic Extraction; Social Media Mining

1. Introduction

Twitter has become an important social media site since its inception in 2006. It is a micro blogging service, which allows users to post messages up to 140 characters known as tweets. Twitter users are followed and are themselves following others, thus creating a social network. This social network can be used to identify communities. Are there communities in this network with opposing views? How do we identify such communities? How do we aggregate sufficient information from micro blogs to assess if the communities have opposite views?

Twitter users typically tune in to listen to popular, smart, informative members of society. In this paper we don’t analyze the Big Data problem that is associated with listening to all of Twitter; we simply focus on a small set of informative Twitter accounts that belong to members of 112th House of Representatives. Congressmen are powerful individuals that must choose their words carefully. There are a great number of news media articles, but which topics are the most important as viewed by these high government officials? This paper illustrates how, in an automated fashion, important news media topics that worry our congressmen can be extracted. Our goal was to investigate not just popular topics but those topics with high levels of disagreement between republicans and democrats. From Sep to Dec 2013 there were a large number of public disagreements between the two parties; in particular the government’s shutdown and Obamacare’s catastrophic website rollout. Our study demonstrates how

* Corresponding authors:
E-mail address: apanasyu@syr.edu, esyu@syr.edu, mehrotra@syr.edu
Twitter data may be used for identifying, in an unsupervised fashion, specific topics of disagreement between the two parties. Our findings are evaluated by comparing extracted topics against articles in the news media.

The paper is organized as follows. In Section 2 we discuss related research and how it shaped our direction. In section 3 we present our approach for data collection, preprocessing, aggregating documents for topic modeling, aggregating documents by Latent Dirichlet Allocation (LDA) topics, performing sentiment analysis, and finally determining LDA topics with most disagreement between groups of users. Section 4 presents results that are specifically related to 112th House of Representatives. Conclusions and future research are summarized in Section 5.

2. Background

Twitter’s use by congressmen has been studied by [1, 2, 3]; however these papers describe basic statistical facts such as rates of Twitter adoption and why Twitter is becoming popular among congressmen. They find that Twitter’s use is accelerated if followers increase as the congressman posts more tweets. In [2] the authors attempt to find the factors that contribute to congressmen using Twitter and who have the largest following. Authors in [3] analyze variables for predicting how likely a congressman is to use Twitter.

In [4] researchers analyze 6000 tweets by hand; they group tweets into a number of categories such as fundraising, personal messages, information, and others and provide basic information. For example, they argue that congressmen use Twitter as a one-way broadcasting medium to advertise themselves and their agendas. No recent work had been found that analyzes, collectively, views of members of the republican and democratic parties. Congressmen have become active Twitter users with tens of thousands of tweets on a monthly basis and therefore automated approaches are necessary to make sense of their messages.

The statistical semantic hypothesis [5] argues that to understand the meaning of words it is enough to consider statistical word usage. The approach taken by most search engines is to organize documents and their terms in a Vector Space Model (VSM) [6]; generated by calculating term frequency versus inverse document frequency (TF-IDF). Documents matching a search engine query are those that have the closest distance between the vector of keywords in the query and the TF-IDF values inside the VSM model for those keywords [7]. VSM stores TF-IDF values for all terms and thus tends to be very large. At first dimensionality reduction techniques like latent semantic indexing (LSI) [8] have been utilized by researchers for identifying key terms within the VSM matrix. This was followed by algorithms to identify key terms within the VSM matrix without having to compute the VSM model [9]. And finally as demonstrated in the algorithms were extended to the level of documents [10] in a probabilistic approach called Latent Dirichlet Allocation (LDA). LDA is an unsupervised topic-modeling algorithm that generates a list of topics; each topic is a list of weighted keywords that are necessary for understanding similarities and differences among documents.

Extracting keywords that can be used to effectively differentiate between a set of documents requires documents that are large enough to carry this type of information. Messages on Twitter are short and do not carry enough information to be properly differentiated. A typical approach for Twitter data is to aggregate all Twitter messages into documents by user [12]. LDA can then be used to find keywords that effectively differentiate all of the authors considered. In [13] the authors further illustrate the importance of properly forming documents from Twitter data by comparing documents consisting of single tweets, documents consisting of all messages for a particular user, and documents that are formed via custom sampling approach they call Twitter LDA. Human judges evaluated the top 10 keywords of resulting topics for each approach and found that Twitter LDA produces the most relevant results. For our problem we wanted to use LDA in order to find topics that exist within two specific communities. The type of documents to consider therefore had to be community specific. To the best of our knowledge there does not exist any work that aggregates tweets based on community information. We aggregated tweets around Twitter topics and Twitter user mentions (words that begin with # and @ respectively) into separate documents for republicans and democrats. LDA was used to extract a set of keywords that define the topics present in these documents. Sentiment analysis (overall positive/negative tone of all tweets) is performed to compute each community’s sentiment for each LDA topic. We utilized SentiWordNet 3.0 [14], a popular lexicon with large word coverage, when performing sentiment analysis.
3. Our Approach

3.1. Data Collection

Python was used for getting the data required by the project because all application programming interfaces (APIs), listed below, return results as JavaScript Object Notation (JSON) dictionaries and JSON is the main data type in Python. Each API is free, but requires a private access token for making requests. APIs return a JSON dictionary in response to a properly formed web address. MySQL was used for storing and querying data.

- Klout API (http://klout.com/s/developers/v2)
- Twitter API (accessed via Tweepy: Twitter library for Python (https://pypi.python.org/pypi/tweepy)

The New York Times Congress API maintains a list of all congress members for both house and senate. We used the API to query for members of the 112th House of Representatives. The results contained the twitter id as well as the party affiliation, district served, percent of votes with party, and other information. We collected data for a total of 399 out of 435 members (187 democrats and 212 republicans) who have active Twitter ids.

The next step was to focus on those members of Congress that had a strong influence with Twitter. As in [3], we used Klout to measure the influence of a congressman on Twitter. Klout assigns a score from 0 to 100 to every Twitter user. For further analysis we kept congressmen with a Klout score greater or equal to 50, which reduced the number of members to 377 (174 democrats and 203 republicans).

Twitter maintains the most recent 3200 tweets for each user that can be accessed via the Twitter API. We used this API for collecting a timeline of messages for all 377 congressmen for a total of 300,034 republican tweets and 204,203 democrat tweets. These tweets were stored in a MySQL database for ease of querying. For the time period of 09/01/2013 to 12/01/2013 there were 34,624 republican and 27,724 democrat tweets.

3.2. Data Selection and Preprocessing

All tweets were tokenized on whitespace. Custom routines were written to help get rid of all punctuations except for # and @ in the beginning of words since these have a special meaning in Twitter. All letters were converted to lowercase and non-ASCII characters were disregarded. A sample of original versus so processed tweets is shown in Table 1. Table 2 summarizes the number of tweets with @-mentions and #-topics.

Table 1: Original vs. Processed Text

<table>
<thead>
<tr>
<th>original text</th>
<th>processed text</th>
</tr>
</thead>
<tbody>
<tr>
<td>In case you missed it, great article from @thedayct on @SenChrisDodd's final floor speech. <a href="http://bit.ly/eNOLZO">http://bit.ly/eNOLZO</a></td>
<td>case missed great article @thedayct @senchrisdodds final floor speech. <a href="http://bit.ly/eNOLZO">http://bit.ly/eNOLZO</a></td>
</tr>
<tr>
<td>RT @austinenergy: Austin 3-1-1 is the place to report potholes, streetlights out, traffic snarls &amp; other nonemergencies <a href="http://bit.ly/gRTnlm">http://bit.ly/gRTnlm</a></td>
<td>@austinenergy austin 311 place report potholes streetlights traffic snarls nonemergencies <a href="http://bit.ly/gRTnlm">http://bit.ly/gRTnlm</a></td>
</tr>
<tr>
<td>Ending extended unemployment benefits will slow economic growth, Kill a million jobs, increase poverty &amp; homelessness: <a href="http://yhoo.it/gTJb63">http://yhoo.it/gTJb63</a></td>
<td>ending extended unemployment benefits slow economic growth kill million jobs increase poverty homelessness <a href="http://yhoo.it/gTJb63">http://yhoo.it/gTJb63</a></td>
</tr>
</tbody>
</table>

1: It seems that all congressmen should have a big impact with the words they say, but some congressmen don’t use Twitter much while others let their staff manage their account resulting in too many irrelevant messages.
2: @-mention refers to existing Twitter user ‘@username’, #-topic associates a tweet with a user defined topic: ‘#topicname’
We hypothesized that tweets that do not mention a specific #-topic or @-mention might be more closely aligned to personal messages and were therefore not important. Furthermore for topic modelling we aggregated tweets by #-topic or @-mention into documents for each community and used those as input for LDA. From Table 2 we see that #-topics are much more frequent then @-mentions. #-topics are probably more relevant because #-topics are typically used to associate a tweet with a descriptive word while @-mention used simply for giving credit to person.

Table 3 shows number of tweets for varying levels of users that use a specific #-topic or @-mention for time period of September to December 2013. For instance Table 3 shows 70 specific #-topic or @-mention that appeared over 32 times and mentioned by over 8 republican users. There was 561 unique #-topic or @-mention that appeared over 4 times and was mentioned by over 2 republican users. And there were 3878 unique #-topic or @-mention that appeared at least once by at least one republican user. While there are a lot of #-topic or @-mention features, we were primarily interested in those that are used by many congressmen. Instead of simply getting rid of tweets that do not contain frequently mentioned #-topic and @-mention we chose to analyse four different scenarios shown in Table 4. For each scenario listed in Table 4 we form a set of documents, perform LDA, sentiment analysis, and determine polarizing topics, topics where the two parties differ most.

### Table 2: All Tweets by Republicans and Democrats vs. tweets containing # and @, # or @

<table>
<thead>
<tr>
<th></th>
<th>Number of Republican Tweets</th>
<th>Number of Democrat Tweets</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Tweets</td>
<td>34622</td>
<td>27723</td>
</tr>
<tr>
<td>Tweets containing #-topic and @-mention</td>
<td>25903</td>
<td>21710</td>
</tr>
<tr>
<td>Tweets containing #-topic</td>
<td>17940</td>
<td>13938</td>
</tr>
<tr>
<td>Tweets containing @-mention</td>
<td>7963</td>
<td>7772</td>
</tr>
</tbody>
</table>

3.3. Extracting Topics and their Corresponding Sentiments

LDA is an unsupervised topic-modelling algorithm. Given a set of documents LDA creates N topics, pre-specified integer. Associated with each topic is a list of weighted keywords that are useful for describing the set of documents. As discussed earlier, use of a single tweet as input to LDA does not give good results. We believe that the congressmen belonging to the same party will typically be in agreement with each other; therefore we utilized community information to form our set of documents. That is, we aggregate tweets by #-topic and @-mentions into separate documents for each community. Hence there are as many documents as unique #-topics and unique @-mentions for each party. From Table 3 we see that if we are to consider #-topics that appear over 32 times and by
over 8 users then there will be 70 documents for republicans and 61 documents for democrats. An example of a
single document would be aggregation of all tweets by republicans which contain “#jobs”. Below is the pseudo code
for creating documents for the republican community (same process is repeated for democrat tweets):

```python
for feature in { all #-topics, all @-mentions}:
    for tweet in { Republican tweets }:
        if tweet contains feature:
            republicanDocuments[feature] = republicanDocuments[feature].append(tweet)
```

The documents so generated are run through the LDA algorithm. We used University of Massachusetts (uMASS)
LDA implementation called Mallet (http://mallet.cs.umass.edu/). Mallet package uses an iterative procedure to
optimize the LDA-hyper-parameters (all LDA implementations do not use hyper-parameters). This is important
because authors in [11] demonstrate that, if LDA-hyper-parameters are optimized, then the implementation is
relatively insensitive to number of topics N as long as N is sufficiently large. We have tried LDA with 50, 100, and
200 topics and have found 50 to 100 topics are high enough for good results.

Table 5 shows LDA output for two topics whence LDA was run with 100 topics and 1000 keywords per topic.
Feature weights specify how relevant the keywords are. It is possible for same keyword to appear under several
different topics, but it will have different weight for each.

<table>
<thead>
<tr>
<th>Topic</th>
<th>F0</th>
<th>W0</th>
<th>F1</th>
<th>W1</th>
<th>F2</th>
<th>W2</th>
<th>F3</th>
<th>W3</th>
<th>F4</th>
<th>W4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>#keystonexl 0.070261</td>
<td>#timetobuild 0.066284</td>
<td>years 0.039549</td>
<td>pipeline 0.03513</td>
<td>delays 0.022757</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>#wrrda 0.096784</td>
<td>@transport 0.032749</td>
<td>house 0.024708</td>
<td>bill 0.024269</td>
<td>water 0.022661</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The two topics in Table 5 are related to a delay in the Keystone pipeline and a Water Resources Reform and
Development Act (WRRDA). WRRDA contains additional keywords that are not shown in table 5 due to space
constraints. In particular two features are “feature 7” and “feature 17” corresponding to #4jobs and #jobs
respectively. #4jobs is typically used by republicans and #jobs is used by democrats. When either party is using
the keyword they are typically talking about job creation. For the keystone pipeline, in contrast to WRRDA topic, the
democrats don’t use #jobs but republicans do use #4jobs. This implies that, unlike republicans, the democrats don’t
want to talk about job creation when talking about the pipeline. These examples illustrate that LDA analysis allows
us to group #-topics, @-mentions, and other keywords into more general topics that are open to interpretation.

Once we have a set of LDA topics we can use the weighted features to calculate for each document the topic it is
most closely associated with. Here is the pseudo code for determining which LDA topic each republican document
belong to (same process repeated for democrat documents):

```python
for feature in republicanDocuments:
    document = republicanDocuments[feature]
    bestTopic = -1
    maxScore = -1
    for topic in topicsToFeaturesFromLDA:
        totalTopicScore = 0
        for tweet in document:
            for word in tweet:
                if featureToWeight.has_key(word):
                    totalTopicScore += featureToWeight[word]
        if maxScore < totalTopicScore:
            maxScore = totalTopicScore
            bestTopic = topic
    republicanDocumentTopics[feature] = bestTopic
```
In this way for each community we have a set of documents associated with an LDA topic. For each topic all of the words in the documents associated with that topic are aggregated and used in sentiment analysis. The overall sentiment score is computed by adding the sentiment values for all of the words associated with the LDA topic, separately for republicans and for democrats. Sentiment values for each word come from the value specified by the SentiWordNet lexicon. Input to SentiWordNet is not just the word but also whether the word is a noun, adjective, noun, or adverb. For the Twitter data we have not performed any part of speech (POS) tagging and so for each word SentiWordNet simply returns the maximum sentiment that is possible by trying all possible part of speech tags. SentiWordNet produces a score between -1 and 1. Neutral words have scores around 0.0. To calculate the sentiment score we removed neutral words and focused only on words with strong sentiment. We experimented with four word sentiment values -- absolute value above 0.25, above 0.5, above 0.6, and above 0.7 and found that words with sentiment values below -0.5 and above 0.5 worked the best for LDA with 100 topics and sentiment values below -0.25 and above 0.25 worked the best for LDA with 50 topics.

Pseudo code for calculating sentiment of republican for each LDA topic is presented below (same process repeated for democrat documents):

```python
for feature in republicanDocuments:
    document = republicanDocuments[feature]
    for tweet in document:
        for word in tweet:
            if abs(sentiwordnetvalue(word)) >= 0.25 (also tried 0.5, 0.6, and 0.7):
                republicanDocumentSentiments[feature] += sentiwordnetvalue(word)
for topic in topicsToFeaturesFromLDA:
    count = 0
    for feature in republicanDocuments:
        if republicanDocumentTopics[feature] == topic:
            republicanTopicSentiments[topic] += republicanDocumentSentiments[feature]
            count += 1
    republicanTopicSentiments[topic] = republicanTopicSentiments[topic]/count
```

Using the sentiment scores for all LDA topics, for each party, we identified the topics with strong disagreement as described in the following pseudo code:

```python
for topic in topicsToFeaturesFromLDA:
    if (((republicanTopicSentiments[topic] > 0) and (democratTopicSentiments[topic] < 0))
    OR (((republicanTopicSentiments[topic] < 0) and (democratTopicSentiments[topic] > 0)))):
        polarizingScore=abs(republicanTopicSentiments[topic]-democratTopicSentiments[topic])
        controversialTopics[topic] = polarizingScore
```

As the pseudo code illustrates, the polarizing score is the absolute value of the difference between republican and democrat sentiment scores. Topics can be arranged by sorting the polarization scores in decreasing order; the first entry in the sorted list represents a topic where the two parties have maximum disagreement.

4. Results

Ranked list of those topics where democrats and republicans are in disagreement (topics where one party is overly positive and the other is overly negative) are generated using the sentiment scores. Table 6 shows the top five controversial topics and the corresponding top 5 keywords.

Using the keywords and large negative sentiment score with the first topic one can easily conclude that the republicans are complaining about the failed rollout of the Obamacare website. Likewise, using the large positive democrat sentiment score and associated keywords with the second topic one can easily conclude that the Democrats are advocating people to sign up for Obamacare (this is before the website became available). Similar conclusions can be drawn from the other topics, namely in topic 3 democrats are blaming republicans for the shutdown; in topic 4 democrats express their negative view about the $40 billion republican sponsored cut to food stamps; and in topic
5 republicans, unlike the democrats, are emphasizing the positive job creation aspect associated with keystone pipeline.

Table 6: Top 5 controversial topics using scenario 1 from Table 4

<table>
<thead>
<tr>
<th>Topic</th>
<th>Polarity</th>
<th>Republican Sentiment</th>
<th>Democrat Sentiment</th>
<th>Top 5 Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32760.57</td>
<td>-29612.1</td>
<td>3148.508</td>
<td>#obamacare, health, insurance, #trainwreck, keep</td>
</tr>
<tr>
<td>2</td>
<td>14426.36</td>
<td>-55.0724</td>
<td>14371.29</td>
<td>#aca, health, #getcovered, care, #obamacare</td>
</tr>
<tr>
<td>3</td>
<td>3371.414</td>
<td>3.006345</td>
<td>-3368.41</td>
<td>#gopshutdown, #enoughalready, end, house, #demandvate</td>
</tr>
<tr>
<td>4</td>
<td>1345.3</td>
<td>1.070715</td>
<td>-1344.23</td>
<td>#snap, cuts, #endhungernow, food, cut</td>
</tr>
<tr>
<td>5</td>
<td>345.1585</td>
<td>332.9644</td>
<td>-12.1941</td>
<td>@repjustinamash, #keystonexl, #timetobuild, @politico, years</td>
</tr>
</tbody>
</table>

We used the New York Times Articles Search API V2 for automatically finding relevant news media articles based on the top keywords associated with each topic. We evaluated the top three articles returned by the API. In most cases the API generated relevant articles, but because it is sensitive to punctuation and misspelled words we found that it is best if an analyst is involved for making a query from the top keywords. Topics for the overall time period produced expected results i.e. strong disagreements were found over Obamacare and the government shutdown. The keywords were clear enough to understand the overall topic implied even if only the top 5 keywords are considered. For each topic relevant news media articles were found to exist that confirmed that republicans and democrats did in fact have disagreements over these topics.

Topics for the overall time period produced expected results, but we also wanted to analyze topics over smaller time periods. This allows us to form a timeline of how the topics evolve. Day to day conversations may not represent the overall topics that define the congressmen; we have chosen to analyse two-week periods. Fewer messages were present in two weeks so we performed LDA analyses with 25 topics and 50 keywords for each topic (words with absolute sentiment above 0.5 used). Top polarization score fluctuations over time are shown in Fig 1.
All of the peaks with polarization scores above 1000 were labeled using headlines from new media articles; for example first peak is labelled “Testimony on Benghazi” which corresponds to news coverage of testimony on Libya by an American diplomat given on May 8, 2013. News articles corresponding to a specific two week period were found by using the top 5 keywords from the top topic in that time period. We were able to successfully find news articles that demonstrated tensions between republicans and democrats for all of the peaks shown in Figure 1. In Figure 1, the timeframe starts with topics mainly related to proposed budgets by republicans and democrats. The first peak (#benghazi) is a sudden outburst of emotion over negative testimony over Libya (republicans blamed the administration and praised the diplomat who gave the testimony). This is followed by growing agitation between republicans and democrats (first red curve) over Obamacare leading to second peak that is associated with July 2, 2013 (when a portion of Obamacare was delayed by a year). Following this victory republican required Obamacare to be delayed in entirety in order to pass the budget (peak 3). Peak 4 is because the democrats refuse to agree with repeal of Obamacare leading to a government shutdown (democrats blame republicans). Peak 5 is due to the reason that republicans are angry because Obamacare hasn’t been repealed and criticize Obamacare’s botched website rollout. Ultimately the conversation dies down and republicans accept defeat. We found that this timeline of events is well documented by the news media stories we found.

5. Conclusions

The research goal of this project was to identify areas of conflict among large communities of users using tweets only. Using data generated by congressmen belonging to republicans and democrats parties of the 112th House of Representatives we have successfully demonstrated that this objective is achievable. An immediate implication of this study is that analysis of tweets, collected over a reasonable period of time, for known communities, can be used to identify the topics of conflict and agreement. In our future work we wish to investigate how to predict resulting consequences. Finally, our long-term goal is to replicate these objectives for communities that have yet to be detected and how the levels of disagreement can be used as an alert for potential violence and unrest.

References