Detecting Opinions in a Temporally Evolving Conversation on Twitter

Kasturi Bhattacharjee and Linda Petzold

Department of Computer Science,
University of California Santa Barbara.
kbhattacharjee@cs.ucsb.edu, petzold@cs.ucsb.edu

Abstract. The immense growth of online social networks from simply being a medium of connecting people to assuming a variety of roles has led to a massive increase in their use and popularity. Today, networks like Facebook and Twitter act as news sources, mediums of advertising and facilitators of socio-political revolutions. In such a scenario, it is of vital importance to be able to detect the opinions of social network users in order to study the opinion flow processes that unfold in these networks. For many topics, the focus of the conversation evolves over time based on the occurrence of real-world events, which makes opinion detection challenging. Since it is not practical to label samples from every point in time, a general supervised learning approach is infeasible. In this work we propose a temporal machine-learning model that has its underpinnings in social network research conducted by sociologists over the years, to detect user opinions in evolving conversations. It uses a combination of hashtags and n-grams as features to identify the opinions of Twitter users on a topic, from their publicly available tweets. We use it to detect temporal opinions on Obamacare and U.S. Immigration Reform, for which it is able to identify user opinions with a very high degree of accuracy for a randomly chosen set of users over time.

1 Introduction

Online social networks were initially developed as a means of connecting people from different parts of the world, by facilitating communication between them. However, in today's day and age, they have grown to assume various other roles, including news sources, platforms for users to voice their opinions on current events, mediums for viral marketing, and facilitators of socio-political revolutions [7, 5]. This variety of roles has led to the tremendous increase in the use and popularity of these networks in general [3, 8], and makes them an interesting subject of study.

To understand the opinion flow processes that unfold in these networks, the detection of opinions and sentiments of users is of great importance. Moreover, owing to the volume of posts being generated daily, it is important to be able to perform this task in an automated fashion. In this paper we present a method for detecting the opinions of Twitter users on a given topic *over time*, using data mining and machine learning techniques.

Recently proposed methods in the field of opinion detection in general are based either on machine learning, or lexicons of words. There is no temporal aspect to these approaches. They are trained on labeled data and/or use a predetermined lexicon of words. However, in the case of *temporal opinion detection*, which is the problem we address here, the focus of the conversation shifts from one sub-topic to another, thus new textual features emerge at every time point. The lack of training data at every timestep renders general supervised approaches infeasible.

The method we propose in this paper for *temporal opinion detection* borrows from social network research conducted by sociologists over the years [16, 19]. A key observation from social network research is that temporal evolution of user opinions is a *slow* process. People are inherently resistant to changing their opinions. We propose a regularized supervised approach that requires training only at the initial time, and enables us to use opinions detected in a previous timestep when performing predictions for the future. Additionally, the method can capture relevant textual features over time, thus highlighting the conversational sub-topics that emerge at every timestep.

We select Twitter as the source of data for our experiments, and Obamacare as the primary topic of interest. *Obamacare* is a popular term coined to represent the Affordable Care Act (ACA) which was signed into law by President Barack Obama on March 23, 2010 [1]. Since its inception, it has garnered much political and social attention in the US, and has emerged as one of the most popular topics of discussion in social media platforms [6]. The Act also underwent several reforms over time, each addressing a different issue. This led to an *evolving* online conversation on the topic, since the focus of the discussions would *shift* from one sub-topic to another over time. The above characteristic makes this topic interesting and challenging for opinion detection, as we shall illustrate in the later sections.

In order to demonstrate the generality of our method, we selected another topic for our experiments, namely, the U.S. Immigration Reform bill (the Border Security, Economic Opportunity, and Immigration Modernization Act of 2013) that was introduced in the US Senate in April, 2013. The bill would allow for many undocumented immigrants to gain legal status and become U.S. citizens. Additionally, it would make the border more secure by adding up to 40,000 border patrol agents [4]. This topic was also extensively discussed on Twitter. The details about the data collection process for both topics are elaborated in Section 4.1.

Contributions of the paper

1. This work proposes a machine-learning model to accurately detect opinions of Twitter users over time using their tweets, even when the topic of conversation is evolving in nature. Training is required only at the initial time.
2. The proposed method also showcases the textual features that are most effective at identifying the opinions at different time points. These features aid in identifying the most popular sub-topics that emerge at every time point.

The remainder of the paper is organized as follows. Section 2 discusses the existing literature in this field. Section 3 describes in detail our proposed method for solving the problem at hand. Section 4 contains details on the data we collected and the techniques we used for pre-processing the data. Section 5 discusses the implementation of the method. Section 6 elaborates on the experiments conducted to validate the method and the results obtained.

2 Related Work

The prior research on opinion detection or sentiment analysis can be broadly classified into two groups: lexicon-based methods and machine learning-based methods. The lexicon-based methods work by using a predefined collection (lexicon) of words, where each word is annotated with a sentiment. Various publicly available lexicons are used for this purpose, each differing according to the context in which they were constructed. Examples include the Linguistic Inquiry and Word Count (LIWC) lexicon [31, 30] and the Multiple Perspective Question Answering (MPQA) lexicon [25, 37, 38]. The LIWC lexicon contains words that have been assigned into categories, and matches the input text with the words in each category [24]. The MPQA lexicon is a publicly available corpus of news articles that have been manually annotated for opinions, emotions, etc. These lexicons have been widely used for sentiment analysis across various domains, not just specifically for social networks [21, 13, 9]. Other popular sentiment lexicons that have been designed for short texts are SentiStrength [35] and SentiWordNet [10, 17]. These lexicons have been extensively used for sentiment analysis of social network data, online posts, movie reviews, etc. [23, 20, 34, 32]. However, as seen in [12], they do not perform well for opinion detection on Twitter users.

Machine learning techniques for sentiment analysis include classification techniques such as Maximum Entropy, Naive Bayes, SVM [22], k-NN based strategies [15], and label propagation [36]. These require labeling of data for training, which is accomplished either by manually labeling posts [36], or through the use of features specific to social networks such as emoticons and hashtags [15, 22]. Some of the existing research combines lexicon-based methods and machine-learning methods [33]. None of the above methods address the problem of temporal opinion detection that is the topic of this paper.

In prior work [12], we addressed the problem of opinion detection on Twitter users over a fixed period of time. There was no temporal aspect to the problem. We developed a supervised learning approach using a regularized logistic regression model. We used textual features, namely hashtags and n-grams, to detect user opinions on two topics: U.S. Politics and Obamacare, with a high accuracy. The Obamacare dataset used in that work contained tweets over a short time period and hence did not capture the evolving nature of the conversation. However, when we applied the same method to the current dataset that spans a larger timeline, it failed to detect user opinions accurately (details in Section 6), thus leading us to the development of the proposed model for temporal opinion detection.

3 Temporal Opinion Detection over an Evolving Conversation

In this section we describe the problem at hand and discuss the social network research that our proposed model is based on. Thereafter, we delve into the details of the proposed model.

3.1 Opinion Change Processes over Time

The key point of our proposed opinion detection model is that users tend to change their opinions very slowly. This forms a basis of the seminal opinion change models from sociology [19, 16]. We present three factors owing to which transition to a different opinion takes place gradually. First, people vary in their readiness to be influenced by their neighbors. Every person has some amount of stubbornness and attachment to their own opinions and beliefs. This is a factor that most models of opinion change consider. For example, a widely-used opinion change model arises from the Social Influence Network Theory of Friedkin and Johnson [19], and is given by

$$\mathbf{y}^{(t)} = \mathbf{A}\mathbf{W}\mathbf{y}^{(t-1)} + (\mathbf{I} - \mathbf{A})\mathbf{y}^{(1)}, \tag{1}$$

where $\mathbf{y}^{(t)}$ is a vector of the users' opinions at time t, $\mathbf{W} = [w_{ij}]$ is the matrix of interpersonal influences, which stores the amount of influence user j has on user i. $\mathbf{A}$ is a diagonal matrix of the users' susceptibilities to interpersonal influence. As is evident from (1), $\mathbf{A}$ determines how anchored the users remain to their initial opinions $\mathbf{y}^{(1)}$, which regulates how much they are influenced by their network neighbors to change their opinions.

Second, we treat the responses of all users as homogeneous from the point of view of opinion change. Thus the opinion of any user, as well as the opinions of all the users she is influenced by, evolve over time. The influenced user slowly changes her opinion in response to the changing opinions of her influencers.

Third, multiple neighbors influence each user. Most opinion models, including Social Influence Network Theory (1) and the DeGroot model [16], assume that a user's opinion is the average of the opinions of her neighbors and her own opinions. This averaging effect tends to dampen dramatic changes [19], making opinion change a slow process. This key observation leads to the main assumption in our proposed model. *For a sufficiently large set of users, most users are not likely to change their opinions drastically over a short period of time.*

3.2 Opinion Detection Models

In this section we discuss our previous model on opinion detection for Twitter users (with no temporal aspect) [12]. Thereafter, we present our proposed model for temporal opinion detection.

Static Opinion Detection Model. In previous work [12], we assumed user opinion to be a distribution over positive and negative types, and used textual features derived from the tweets to learn a weighted combination of the features that would best classify the opinions.

We begin with training data $(\mathbf{x}_i, y_i)$, $i = 1, ...n$, where n is the number of users, $\mathbf{x}_i$ is the i^{th} data vector of size $k \times 1$, with k number of features, and y_i is the i^{th} user's discrete opinion value in $\{-1, 1\}$. For the i^{th} user, the probability that she has a positive opinion is given by:

$$P(y_i = 1 | \mathbf{x}_i, \beta) = \frac{1}{1 + \exp\left(-\beta^T \mathbf{x}_i\right)}, \tag{2}$$

where β is a $k \times 1$ feature weight vector. Note that there is no concept of time in this model.

We minimized an l_2-regularized logistic loss function to learn β:

$$\begin{aligned} L(\beta) &= -\log\left(\prod_{i=1}^{n} P(y_i | \mathbf{x}_i, \beta)\right) + \lambda \|\beta\|_2^2 \\ &= \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i(\beta^T \mathbf{x}_i)\right)\right) + \lambda \|\beta\|_2^2, \end{aligned}$$

where λ is the regularization parameter. Thus, given a set of features $\mathbf{x}$ and a set of known outputs y in the training data, the logistic regression model learns the parameter β that determines the relationship between $\mathbf{x}$ and y. Once the model has been learned, it can then be used to predict the outcomes of the test data, given their features $\mathbf{x}$.

Temporal Opinion Detection Model. In this work, we extend the above regularized logistic regression model, with an added element of time. As in the previous work, user opinions are classified as positive and negative types. Here, we have data samples $\mathbf{x}_i^{(t)}$, $i = 1, ...n$ and $t = 1, 2, ...$ Further, we have labels only for the first timestep, i.e., $y_i^{(1)}$, $i = 1, ...n$. Labeled samples are required for the first timestep, but not for the subsequent timesteps.

Now, extending (2) for any t^{th} timestep for user i, we obtain

$$P(y_i^{(t)} = 1 | \mathbf{x}_i^{(t)}, \beta^{(t)}) = \frac{1}{1 + \exp(-{\beta^{(t)}}^T \mathbf{x}_i^{(t)})} \tag{3}$$

where $y_i^{(t)}$ is the discrete opinion value in $\{-1, 1\}$ in timestep t, $\mathbf{x}_i^{(t)}$ is a $k \times 1$ data vector and $\beta^{(t)}$ is a $k \times 1$ feature weight vector for timestep t.

We do not have labels on the samples for timestep $t+1$, as previously stated. Hence, to predict the opinions for timestep $t+1$, we apply the key observation from Section 3.1 that *most* users do not change their opinions drastically in a single timestep. Thus, we assume that *most* users hold the same opinion as in the previous timestep. Most of the opinions in the previous timestep will therefore

Table 1. Examples of hashtags and n-grams over time on Obamacare

Feature type	**Timestep 1**	**Timestep 5**	**Timestep 8**
Hashtags	#obamacare, #koch, #getcovered, #cvs, #gop	#obamacare, #fullrepeal, #dontfundit, #aca, #trainwreck	#obamacare, #irs, #koch, #debtceiling, #gop
Unigrams	obamacare, gop, health, republicans, healthcare	obamacare, website, insurance, fix, coverage	obamacare, enrollment, work, hhs, job
Bigrams	obamacare will, the gop, benefits to, howard dean, fund obamacare	obamacare enrollment, signed up, fix obamacare, website failed, obamacare promises	3.3 million, signed up, million jobs, the koch, the irs

be the same as those in the next timestep, i.e. $y_i^{(t)}$ is the same as $y_i^{(t+1)}$ for most users. Following this assumption, we use $y_i^{(t)}$ from the previous timestep, and new textual features $\mathbf{x}_i^{(t+1)}$ from the current timestep to learn $\beta^{(t+1)}$.

Thus, we minimize the following l_2-regularized logistic loss function over consecutive timesteps t and $t+1$:

$$L(\beta^{(t+1)}) = -\log\left(\prod_{i=1}^{n} P\left(y_i^{(t)}|\mathbf{x}_i^{(t+1)}, \beta^{(t+1)}\right)\right) + \lambda\|\beta^{(t+1)}\|_2^2 \tag{4}$$

$$= \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i^{(t)}({\beta^{(t+1)}}^T \mathbf{x}_i^{(t+1)})\right)\right) + \lambda\|\beta^{(t+1)}\|_2^2 \tag{5}$$

The regularization helps to avoid overfitting [26] and to take care of the fact that this is an underdetermined system since $n << k$. Thus, by minimizing (4), we learn $\beta^{(t+1)}$ even in the absence of labeled samples at time $t+1$. We use the open-source machine learning tool scikit-learn [28] to implement logistic regression with l_2 regularization.

4 Data Collection and Preprocessing

In this section we describe the method used to collect the dataset for this work, and the data pre-processing steps involved.

4.1 Data Collection

To crawl tweets on a topic of interest, we randomly selected users and collected their tweets over a period of time using the Twitter Streaming API. For

Table 2. Examples of hashtags and n-grams over time on Immigration

Feature type	Timestep 1	Timestep 3
Hashtags	#immigration, #takeit-tothehouse, #weall-shallovercome, #movefor-ward, #immigrationen-forcement	#immigration, #immigra-tionnews, #protests, #de-port
Unigrams	immigrants, taxes, system, reform, drafted	gop, population, reforms, senator
Bigrams	million people, to diver-sity, immigration reform, require immigration	gop is, for immigration, need jobs, domestic issue, immigration reform

Obamacare, tweets were crawled over a period of 8 months from July 2013 to February 2014. We have 757,960 users and 4,203,900 tweets in our dataset. For the topic of Immigration, tweets were crawled over the months of July, August and September, 2013, yielding a total of 15,001 users and 44,626 tweets. We consider each month to be 1 timestep for the sake of our experiments. On the topic of Obamacare, we selected 936 users that have tweets every month on which to test our model, and for the topic of Immigration, we picked 111 users.

4.2 Data Cleaning and Preprocessing

Twitter data is inherently noisy and filled with abbreviations and informal words. We clean and pre-process the dataset in the following manner to enable a better extraction of features.

- **URL removal:** In our method, URLs would not contribute to the feature extraction and were therefore removed.
- **Stopword removal:** Stopwords such as "a", "the", "who", "that", "of", "has" , etc. were removed from the tweets before extracting n-grams, which is a common practice.
- **Punctuation marks and special character removal:** Punctuation marks such as ":", ";" etc. and special characters such as "[]", ",", "", etc. were removed before extracting n-grams.
- **Additional whitespace removal:** Multiple white spaces were replaced with a single whitespace.
- **Conversion to lowercase:** Tweets are not generally case-sensitive owing to the informal language used. For instance, for our method, the word "Obama" should be considered the same as "obama" when parsing through a tweet. We converted the tweets to lowercase to preserve uniformity in feature extraction.
- **Tokenization:** The tweets were tokenized into words to extract n-grams from them.

5 Implementation Details

In this section we describe the features we chose to use in the model, and also explain the steps taken to implement the model.

5.1 Feature Engineering

As mentioned in the Introduction, we used textual features extracted from the tweets for opinion detection in [12]. The features used were hashtags and n-grams. Apart from highlighting the topic of a tweet, hashtags have been found to carry some additional information regarding the bias of the tweet itself [15, 36]. For example, on the topic of Obamacare, *#defundobamacare, #getcovered, #fullrepeal* are examples of hashtags that clearly portray the opinion of the person that uses them.

However, at times, hashtags by themselves are not sufficient to capture the opinion, for instance,

"Let's abolish the IRS before it enforces #obamacare! please sign and rt this petition if you agree"

In the above tweet, the hashtag *#obamacare* is not sufficient to capture the opinion of the tweet. The entire tweet needs to be considered to get the actual opinion. For this purpose, we use the n-gram model which is considered a powerful tool for sentiment extraction [11, 27, 14]. We extract n-grams out of the tweets to capture the bias from the tweet itself.

At every timestep, we order the features according to the number of users that use them. We use the 1000 most popularly used hashtags, 2000 most popularly used unigrams and 2000 most popularly used bigrams from each timestep for our experiments. The choice of the number of features was governed by the usage of the features. For instance, after the first 1000 hashtags, the usage of the hashtags drops significantly, thus motivating us to use the most popular 1000 tags as our features. Similar reasons led to the use of the top 2000 unigrams and bigrams. Thus we had 5000 features at every timestep.

For every user i at time t in (4), $\mathbf{x}_i$ contains the number of times user i uses each of the 5000 features at that timestep. Owing to the evolving nature of the conversation, this set of features changes over time. However, using our model described in Section 3.2, we can *automatically* learn a new β at every timestep for a new set of features by minimizing (4). Tables 1 and 2 show a few examples of features found on several timesteps.

5.2 Implementation

In our experiments, we consider each month to be a timestep, and study the same set of n users across all timesteps. The following provides a detailed description of the steps taken at every timestep.

- At timestep 1:
  - We begin by labeling a subset of the users such that those with a positive opinion on the given topic are assigned a label +1 and those with a negative opinion are assigned a label -1. Let d be the number of users that are labeled at timestep 1. The data matrix is built using the 5000 textual features (as described in Section 5.1), thereby leading to a $d \times 5000$ matrix, $X^{(1)}$. We use this data to train the model (4) to learn $\beta^{(1)}$. For Obamacare, $d = 201$ (89 positive, 112 negative), and for the Immigration dataset, $d = 30$(24 positive, 6 negative).
  - We then assign opinion labels to the larger unlabeled set of $n - d$ users using the learned $\beta^{(1)}$. This step is performed to get the opinion labels for all n users at this timestep. We now proceed with the entire set of n users for the subsequent steps.
- For each subsequent timestep, $t + 1$:
  - We minimize the regularized logistic loss function (4) between the opinions of users at t and $t + 1$ to learn $\beta^{(t+1)}$.
  - We then use the learned $\beta^{(t+1)}$ to predict opinions at time $t + 1$. This forms $y_i^{(t+1)}$.

6 Experimental results

In this section, we outline in detail the experiments we conducted on the dataset, and the metrics we used to evaluate it. Further, we report the insights that the method provided with respect to the sub-topics that were being discussed at every timestep.

6.1 Temporal opinion detection results

To evaluate the model on our primary topic of interest, *Obamacare*, we label the opinions of a random group of users on some of the key timesteps to test whether our model captures their opinions correctly. We were particularly interested in determining whether the model detects the opinions correctly after the occurrence of a significant event with respect to Obamacare. One such event occurred on October 27, 2013, when the main website for the Affordable Care Act, *Healthcare.gov* crashed. This created a great deal of chatter on Twitter (see Figure 3 for a plot of the number of users that mentioned the website crash over time. As is evident, the number of users goes up significantly towards the end of October which was when the website crash occurred, and continues to be a focus of conversation during November as well.) To determine whether our model captures the opinions being echoed right after this occurrence, we focus on Timestep 5 which contains tweets from the beginning of November 2013, and throughout the rest of the month. We select 88 users at random from that timestep, for testing our model.

The other timestep that we pick for these tests was timestep 5, which was the month of February 2014. In that month, the Department of Health and Human

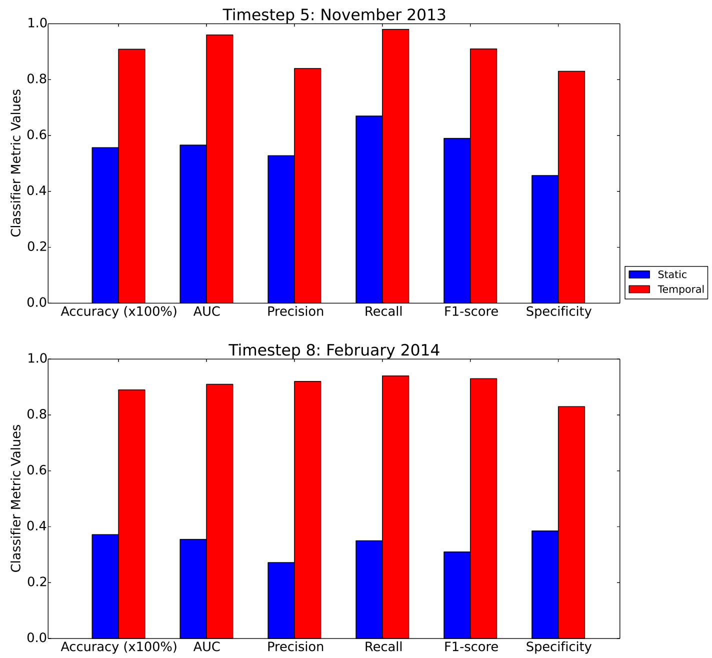

Fig. 1. Comparison of the static and temporal opinion detection methods on Obamacare. The methods are compared on two timesteps of interest across all classifier metrics.

Services (HHS) announced the signing up of 3.3 million people for Obamacare, which was a significant event in the Obamacare timeline. Another event that generated a large volume of tweets at that time was that some firms were firing employees to avoid Obamacare costs, but were certifying to the IRS that the firings were not on the grounds of Obamacare, to avoid penalty of perjury. We label 43 randomly selected users from this timestep.

To validate the usefulness and the need for our method, we first present the results obtained by simply using the *Static Opinion Detection Model* described in Section 3.2 for temporal opinion detection. Thus we used the β learned from the training samples at timestep 1 to predict opinions for later timesteps. As seen in Figure 1, the accuracies achieved using the static method on timesteps 5 and 8 are 55.68% and 37.2% respectively, while our proposed temporal method yields accuracies of 90.9% and 89.0% respectively for the two timesteps. Moreover, the temporal method outperforms the static method across all popularly-used classifier metrics [18] such as AUC, F1-score, etc.

To demonstrate the generality of our method, we also conducted experiments on the topic of *U.S. Immigration Reform bill*. Since we only have 3 months' data on the topic, we evaluated the classifier metrics on the last month. The results are reported in Figure 2. The temporal method yields better performance than the static method in this case as well. As is evident, the static method yields about 50% accuracy, which can simply be obtained by random guessing. However, using the temporal method yields a significantly higher accuracy of 85%. The temporal method also performs much better in comparison to the static method across all classifier metrics as well.

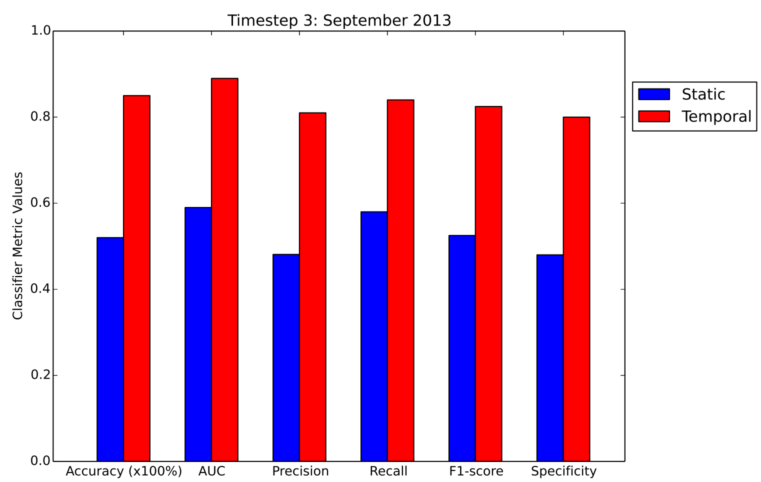

Fig. 2. Comparison of the static and temporal opinion detection methods on Immigration. The static method is no better than random guessing after 2 timesteps, but the temporal method shows high predictive power.

6.2 Significant feature detection and emergence of temporal sub-topics

Out of the 5000 features used at every timestep, some of the textual features are more informative in detecting opinions than others. To determine this set of informative features over time, we evaluate the statistical significance of each feature of the Obamacare dataset for predicting user opinion. We follow the technique described in Section 5, Algorithm 3 of [29] for significance testing, which we describe here for the sake of completeness. For the timestep of interest, we run our l_2 regularized temporal model on the data, and store the weights that each of the features are assigned by the model. Then we randomize the

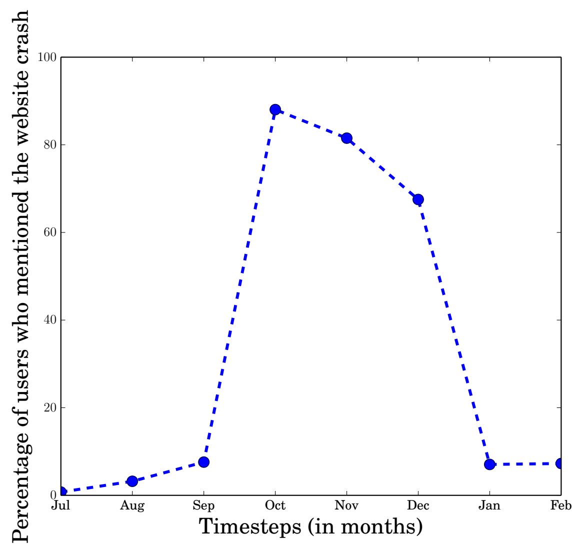

Fig. 3. Mentions of Obamacare website crash over time

labels on the samples and run our model on the randomized data. Let $\hat{\theta}$ be the coefficient obtained from this set. For each randomized run m, let $\tilde{\theta}$ be the random coefficient vector obtained from the fixed feature vector $\mathbf{x}$ and the randomized response $\tilde{y}$. For ν randomized runs, we obtain ν coefficient vectors $\tilde{\theta}$. For each dimension, the coefficient value in each $\tilde{\theta}$ represents a random statistical relationship between the feature and the response. Then the p-value of the l^{th} dimension is computed as

$$\frac{\text{Count}(|\tilde{\theta}_l| > |\hat{\theta}_l|)}{\nu + 1} \tag{6}$$

where "Count" represents the number of times the absolute value of the random coefficient for the l^{th} dimension exceeded the absolute value of the same coefficient obtained from the training set. This is a commonly used permutation test for statistical hypothesis testing [2]. Features that had a p-value less than 0.05 are selected as the most significant features with a confidence of at least 95%.

Using the significant features obtained at each timestep, we examine the dataset for tweets carrying these features. This led to the discovery of the various sub-topics of conversation (related to the main topic of Obamacare), that users participated in over time. Most of the sub-topics can be tied to real-world events that aligned with the timestep under consideration. This further reflects the evolving nature of the topics of conversation. Table 3 illustrates the sub-topics of interest that were detected over the various timesteps. For example, in July 2013, we find that the IRS emerged as an important sub-topic of discussion. Similarly, Obama's apology and a count of how many people were enrolling in Obamacare were popular sub-topics in November 2013. In February 2014, the 3.3 million

Table 3. Significant features (95% statistical significance) on Obamacare at three different time steps. Significant features capture the temporally evolving sub-topics.

Time step	**Significant Features**	**Temporal sub-topics inferred from tweets**
Jul 2013	*braveheart, gifs*	The Washington Examiner publishes funny series of gifs from movie Braveheart depicting Republicans' failed attempts at defunding Obamacare.
	employees	News sources report that Obamacare call center employees were not being offered healthcare benefits.
	kyle	News report by reporter Kyle Cheney on Politico.com stating that CVS was going to publicize Obamacare.
	irs	IRS employees unwilling to sign up for Obamacare, although IRS was heavily involved in enforcing Obamacare.
	howard	Howard Dean, former Democratic National Committee Chairman, comments that Independent Payment Advisory Board will be unable to keep costs down.
	premiums	Obamacare premiums are lowered even further in eleven states.
	empire	Cited article discussing civil lawsuits, environmental damage caused by the output from industries, etc. of the Koch brothers empire and related controversies.
Nov 2013	*warning*	Republicans "warning" people of Obamacare, and that the website crash is a "warning" in itself.
	case	Blog by Peter Suderman ("Time To Start Considering Obamacares Worst-Case Scenarios") discussing failure of online enrollment system negatively affecting Obamacre.
	apology	– Obama apologizing to people whose insurance plans were being canceled, even though he said that people could keep their existing coverage if they liked. – Ed Schultz demands that Republicans, rather than the President, should apologize "for not having any plan".
	scorecard	Obamacare scorecard: how many actually enrolled, and how a larger number of people lost their insurance.
Feb 2014	*@megynkelly*	Megyn Kelly, a Fox news anchor who covered (negative) news related to Obamacare.
	wednesday	Dept. of Health and Human Services announces on a Wednesday (Feb 12, 2014) that 3.3 million people signed up for Obamacare, but it includes hundreds of thousands of individuals defaulting their first premium payment.
	firings	Firms required to certify to the IRS that Obamacare was not a factor in their firing their employees (although it was).
	tgdn	New hashtag (Twitter Gulag Defense Network) started in January 2013 to counter Twitter Gulag, a way to trick Twitter systems into thinking that live profiles are actually spambot profiles. Apparently, many conservative profiles were being shut down by leftists employing this policy.

enrollment mark and Megyn Kelly (a Fox News anchor who covered a great deal of negative news related to Obamacare) were sub-topics that emerged as being popular. Thus our method is able to detect evolving sub-topics of conversation among users over time.

7 Conclusion

In this work we have proposed a novel temporal opinion detection method that can successfully detect the opinions of Twitter users engaging in an evolving conversation. Our primary topic of interest is Obamacare, for which the focus of conversation shifted from one sub-topic to another due to the various events associated with the event that occurred over time. We also selected the topic of U.S. Immigration Reform to demonstrate the generality of our method. Our proposed temporal machine-learning method performs well across all classifier metrics of importance. Additionally, it leads to automatic detection of informative features that point to important, and changing sub-topics.

8 Acknowledgment

This work has been supported by the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office. The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The authors would like to thank Prof. Noah Friedkin of the Sociology Department at the University of California Santa Barbara for his insightful comments during the research.

References

1. Obamacare facts, `http://obamacarefacts.com/obamacare-facts/`
2. p-value, `http://en.wikipedia.org/wiki/P-value`
3. Twitter usage statistics, `http://www.internetlivestats.com/twitter-statistics/`
4. U.s. immigration reform, `https://en.wikipedia.org/wiki/Border_Security,_Economic_Opportunity,_and_Immigration_Modernization_Act_of_2013`
5. Revolutionizing revolutions: Virtual collective consciousness and the arab spring (2012), `http://www.huffingtonpost.com/yousri-marzouki/revolutionizing-revolutio_b_1679181.html`
6. Obamacare sees social media surge ahead of deadline (March 2014), `http://www.nextgov.com/health/2014/03/obamacare-sees-social-media-surge-ahead-deadline/81625/`
7. Social networks and social media in ukrainian "euromaidan" protests (2014), `http://www.washingtonpost.com/blogs/monkey-cage/wp/2014/01/02/social-networks-and-social-media-in-ukrainian-euromaidan-protests-2/`
8. The top 20 valuable facebook statistics – updated february 2015 (2015), `https://zephoria.com/social-media/top-15-valuable-facebook-statistics/`

9. Akkaya, C., Wiebe, J., Mihalcea, R.: Subjectivity word sense disambiguation. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1. pp. 190–199. Association for Computational Linguistics (2009)
10. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: LREC. vol. 10, pp. 2200–2204 (2010)
11. Bespalov, D., Bai, B., Qi, Y., Shokoufandeh, A.: Sentiment classification based on supervised latent n-gram analysis. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. pp. 375–382. ACM (2011)
12. Bhattacharjee, K., Petzold, L.: Probabilistic user-level opinion detection on online social networks. In: Social Informatics. pp. 309–325. Springer (2014)
13. Bono, J.E., Ilies, R.: Charisma, positive emotions and mood contagion. The Leadership Quarterly 17(4), 317–334 (2006)
14. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th international conference on World Wide Web. pp. 519–528. ACM (2003)
15. Davidov, D., Tsur, O., Rappoport, A.: Enhanced sentiment learning using twitter hashtags and smileys. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters (2010)
16. DeGroot, M.H.: Reaching a consensus. Journal of the American Statistical Association 69(345), 118–121 (1974)
17. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC. vol. 6, pp. 417–422 (2006)
18. Fawcett, T.: An introduction to ROC analysis. Pattern recognition letters 27(8), 861–874 (2006)
19. Friedkin, N.E., Johnsen, E.C.: Social influence networks and opinion change. Advances in Group Processes 16(1), 1–29 (1999)
20. Garas, A., Garcia, D., Skowron, M., Schweitzer, F.: Emotional persistence in online chatting communities. Scientific Reports 2 (2012)
21. Gilbert, E., Karahalios, K.: Predicting tie strength with social media. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2009)
22. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford pp. 1–12 (2009(a))
23. Kucuktunc, O., Cambazoglu, B.B., Weber, I., Ferhatosmanoglu, H.: A large-scale sentiment analysis for yahoo! answers. In: Proceedings of the fifth ACM International Conference on Web Search and Data Mining (2012)
24. LIWC: LIWC software. `http://www.liwc.net/index.php` (2001)
25. MPQA: MPQA. `http://mpqa.cs.pitt.edu/lexicons/` (2005)
26. Ng, A.Y.: Feature election, l 1 vs. l 2 regularization, and rotational invariance. In: Proceedings of the Twenty-First International Conference on Machine Learning (2004)
27. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREC (2010)
28. Pedregosa, F., et. al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
29. Pendse, S.V., Tetteh, I.K., Semazzi, F.H., Kumar, V., Samatova, N.F.: Toward data-driven, semi-automatic inference of phenomenological physical models: Application to eastern sahel rainfall. In: SIAM International Conference on Data Mining (2012)

30. Pennebaker, J.W., Chung, C.K., Ireland, M., Gonzales, A., Booth, R.J.: The development and psychometric properties of LIWC2007. Austin, TX, LIWC. Net (2007)
31. Pennebaker, J.W., Francis, M.E., Booth, R.J.: Linguistic inquiry and word count: LIWC 2001 (2001)
32. Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., Li, P.: User-level sentiment analysis incorporating social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1397–1405. ACM (2011)
33. Tan, S., Li, Y., Sun, H., Guan, Z., Yan, X., Bu, J., Chen, C., He, X.: Interpreting the public sentiment variations on twitter. IEEE Transactions on Knowledge and Data Engineering 26(5), 1158–1170 (2014)
34. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment in twitter events. Journal of the American Society for Information Science and Technology 62(2), 406–418 (2011)
35. Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A.: Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology 61(12), 2544–2558 (2010)
36. Wang, X., Wei, F., Liu, X., Zhou, M., Zhang, M.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management (2011)
37. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language resources and evaluation 39(2-3), 165–210 (2005)
38. Wilson, T.: Fine-grained subjectivity analysis. Ph.D. thesis, Doctoral Dissertation, University of Pittsburgh (2008)