Automatic Summarization of Events From Social Media

Freddy Chong Tat Chua*
Living Analytics Research Centre
Singapore Management University
80 Stamford Road, Singapore
freddy.chua.2009@smu.edu.sg

Sitaram Asur
Social Computing Lab
Hewlett Packard Research Labs
Palo Alto, California, USA
sitaram.asur@hp.com

Abstract

Social media services such as Twitter generate phenomenal volume of content for most real-world events on a daily basis. Digging through the noise and redundancy to understand the important aspects of the content is a very challenging task. We propose a search and summarization framework to extract relevant representative tweets from a time-ordered sample of tweets to generate a coherent and concise summary of an event. We introduce two topic models that take advantage of temporal correlation in the data to extract relevant tweets for summarization. The summarization framework has been evaluated using Twitter data on four real-world events. Evaluations are performed using Wikipedia articles on the events as well as using Amazon Mechanical Turk (MTurk) with human readers (MTurkers). Both experiments show that the proposed models outperform traditional LDA and lead to informative summaries.

Introduction

When a news-worthy event occurs in the real world, Twitter users instantaneously post numerous tweets detailing all aspects of the event. Due to these real-time updates, there is a flood of information propagated through these networks. By closely monitoring these streams of information, prior research have shown that it is possible to detect real world events from Twitter (Popescu and Pennacchiotti 2010; Sakaki, Okazaki, and Matsuo 2010; Sayyadi, Hurst, and Maykov 2009; Watanabe et al. 2011; Weng and Lee 2011). An event refers to any concept of interest that gains the attention of the populace. Examples of real-world events range from global catastrophes such as earthquakes (Sakaki, Okazaki, and Matsuo 2010), political protests or unrest (Weng and Lee 2011), to launches of new consumer products.

The easiest way to extract tweets related to an event is through a search query. However, for popular events, this typically results in a significantly large stream of tweets, which makes the task of understanding the aspects of the event and the opinion of people, a difficult and mostly futile task. It has been observed that, despite the high frequency, the actual information content in the tweet stream is fairly limited (Chakrabarti and Punera 2011; Sharifi, Hutton, and Kalita 2010). This is due to the fact that several of the tweets contain redundant information. Also, many of the tweets that are returned by a search query are not relevant to the event. This is due to ambiguity in the search keywords and the noise prevalent in social media. In this paper, we address this problem of summarizing a **targeted event of interest** for a human reader by extracting the **most representative tweets** from the time-ordered sample of tweets for the event.

Problem definition. Formally, we define our problem as follows. Given a time-ordered sample of tweets D representing an event of interest e, the task is to extract K number of tweets from D to form a summary S_e, such that each of these tweets $d \in S_e$ adequately covers different $aspects$ of the event e, where K is a choice of parameter that the human reader may choose, with larger values of K yielding longer summaries. Note that, the event of interest can refer also to search terms used to query for tweets. It is common, in industry and politics, for people to query what is being said in social media about a particular brand, a political candidate or a campaign. In these situations, it becomes paramount to gain an understanding of the key aspects of discussion quickly, particularly to detect changes in opinions and sentiment over time.

Challenges. The above problem definition leads to the question of how we can measure different $aspects$ of the event e from D. By $aspects$, we mean the features of the event that serve as the main discussion points on social media. For example, in the case of a product launch, aspects might include the date of the launch, the features of the product, the price and initial reviews of its performance. Several challenges arise when we attempt to perform basic text analysis on tweets. 1) Words are often misspelled in tweets which means that we cannot use a dictionary or knowledgebase (Freebase, Wikipedia, etc.) to find words that are relevant for e. 2) Many tweets in D are the result of noise and are irrelevant to e, causing unnecessary computation on majority of the tweets. 3) Tweets, by their very nature, are short and that causes poor performance when we apply unsupervised learning techniques that have been developed for traditional text analysis.

A naive solution for the problem of extracting relevant tweets would be to apply a standard topic model on the sam-

*The work was completed when the author visited HPLabs in the summer of 2012.

ple of tweets D to obtain a set of topics Z that can be inspected by the human reader. The event of interest e may emerge as one of the topics $z_e \in Z$ found after topic modeling. Then using the words ranked highly in the topic z_e, we can obtain a set of words related to the event which addresses the first challenge. The tweets which are found to have low probability in the discovered topic can be discarded as irrelevant to e which will address the second challenge. However, the problem of such an approach is that there is no guarantee we can find topic z_e for event e regardless of how many topics we use.

To overcome the problems of this naive approach and address the aforementioned challenges efficiently, we propose a framework that performs search and summarization in a bootstrapping manner.

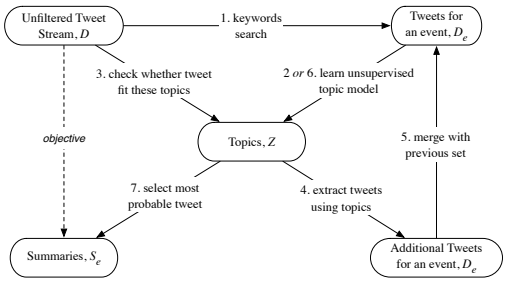


Figure 1: Framework for Search and Summarize

Our Search and Summarize framework (shown in Figure 1) proceeds by first applying a simple keyword-based query Q on the time-ordered sample of tweets D to obtain an initial subset of relevant tweets D_e^1 for the event e. We propose two topic models that can then be applied on D_e^1 to obtain topics that are highly relevant to the event. Using the discovered topics, we subsequently uncover other tweets D_e^2 from D that are relevant to the event e but may not contain the keywords given by the query Q. The combined set of tweets $D_e := D_e^1 \cup D_e^2$ is then used to refine the model and find more aspects of the event e.

The main contributions of this paper are as follows:

1. Our analysis of Twitter data revealed that the content of tweets for an event e is typically strongly related to other tweets about the same event written around the same time. That is, given three tweets $d_1, d_2, d_3 \in D_e$, that are written respectively at t_1, t_2, t_3, suppose $t_1 << t_2 << t_3$, then the similarity between d_1 and d_2 will be higher than the similarity between d_1 and d_3.
2. Based on this observation, we proposed a Decay Topic Model (**DTM**) which learns improved posterior knowledge about tweets $d \in D_e$ written at a later time, given the prior knowledge of earlier tweets. The importance of this prior knowledge with respect to each topic z decays as an exponential decay function with two parameters - the actual difference in time between the tweets and a δ_z parameter for each topic $z \in Z$.
3. By assuming that the time associated with each topic z is distributed with a Gaussian distribution $\mathcal{G}_z$, we infer the decay parameters δ_z using the variance of the Gaussian distributions. Thus, if a topic z has a large time variance, it implies that the topic "sticks" around longer and should have a smaller decay, while topics with a smaller time variance lose their novelty quickly and should have a larger decay. By adding Gaussian components to the topic distribution, we obtain the Gaussian Decay Topic Model (**GDTM**).
4. Based on these two models, we propose a framework for solving the extraction and summarization problem for events from a social media stream.
5. We perform a qualitative and quantitative evaluation of these models on the summarization of four real-world events and demonstrate that the use of temporal correlation facilitates the generation of concise and relevant summaries. Both our methods were found to outperform traditional LDA for this purpose with GDTM providing the best overall performance.

Related Work

Text Summarization : (Nenkova and McKeown 2012) have reviewed an extensive survey of text summarization techniques. According to them, summarization systems perform three successive independent steps to summarize a given target text: 1) Using an intermediate representation for the target text which captures its key features, 2) Using the intermediate representation to assign scores for individual sentences within the text and 3) Selecting a set of sentences which maximizes the total score as the summary for the targeted text.

(Ganesan, Zhai, and Viegas 2012) have proposed the generation of abstract short text summaries from text. They first obtain lists of n-grams (minimum of n is 2) from the raw text and generate a score for each n-gram based on its representativeness and readability. Subsequently, optimal n-grams are chosen for summarization. In our event summarization, we follow the traditional approach of finding an intermediate representation using topics and modeling n-grams using noun phrases in tweets. The distinctive feature in our work is the use of the temporal correlation between tweets which has not been considered in traditional text summarization.

Micro-Blog Event Summarization: (Lu, Zhai, and Sundaresan 2009) have proposed a variant of Hidden Markov Models to obtain an intermediate representation for a sequence of tweets relevant for an event. Their approach does not use the continuous time stamps present in tweets and does not address the problem of obtaining the minimal set of tweets relevant to an event.(Meng et al. 2012) have summarized opinions for entities in Twitter by mining hash-tags to infer the presence of entities and inferring sentiments from tweets. However, not all tweets contain hash-tags which makes it difficult to gain sufficient coverage for an event this way. (Sharifi, Hutton, and Kalita 2010) have proposed the Phrase Reinforcement Algorithm to find the best tweet that matches a given phrase, such as trending keywords. They produce one tweet as a summary for one

phrase while we propose to provide a set of tweets to summarize an event. (Yang et al. 2012) have also proposed a framework for summarizing a stream of tweets. Their main focus is on creating a scalable approach by compressing the tweet stream to fit in limited memory, followed by the use of Nonnegative Matrix Factorization (NNMF) to find topics in the tweet stream. Since they do not filter the tweets for a specific event of interest, the topics discovered using their framework will only contain globally major events. Our proposed framework finds a summary for a targeted event of interest. (Metzler, Cai, and Hovy 2012) proposed a structured retrieval approach for obtaining a set of tweets that are most relevant for an event. It uses a query expansion technique and also exploits the temporal correlation of related event words. The added benefit of our topic model approach is that using the time-variance of each topic for each event, we can gauge how fast each facet of the event decays.

Dynamic Topic Models for Social Media: (Ahmed et al. 2011) have used an exponential decay function to model the dynamic user behavior in search logs. But they have assumed that the parameters of the decay function remain constant for all topics. We have taken a different approach by assuming that there is a decay parameter for each topic and we infer the parameters of the decay function using the variance of Gaussian distribution on the time of the written words. (Saha and Sindhwani 2012) have improved upon existing non-negative matrix factorization to provide an online version for finding emerging topics in social media. But unlike our work, they do not address the problem of short sentences in social media. (Wang and McCallum 2006) have proposed a non-markovian approach to model the trend of topics evolution in text. Their approach assumes that the time stamp on each word is generated by a Beta distribution because of the different shapes a Beta distribution can take. We have used a Gaussian distribution instead because the symmetric shape of the Gaussian curve allows us to use the variance for inferring the decay parameters of our Gaussian Decay Topic Model (GDTM). (Wang, Agichtein, and Benzi 2012) have proposed a temporal topic model called (TM-LDA) that exploits the temporal correlation between the posts for each specific author. They assume that a tweet topic distribution is related to the next tweet via a square matrix with dimensions equal to number of topics. But the algorithm solves for the matrix by minimizing the transition error in Euclidean space. Our approach describes the model as a generative process to preserve the probabilistic foundations of LDA. We have also explicitly used the time for each tweet to describe the amount of temporal correlation between consecutive tweets.

Search and Summarize Framework

Figure 1 provides an overview of the framework we propose in this paper. To summarize for the event of interest e, we first begin by assuming that we have access to a time-ordered sample of tweets. This can be obtained via a set of search queries Q, where each query $q \in Q$ is defined by a set of keywords. For example, the set of queries for the event "Facebook IPO" can be { { facebook, ipo }, { fb, ipo }, { facebook, initial, public, offer }, { fb, initial, public, offer }, { facebook, initial, public, offering }, { fb, initial, public, offering } }.

1. From D, we extract all tweets that match at least one of the queries $q \in Q$. A tweet matches a query q if it contains all of the keywords in q. The set of tweets obtained is denoted by D_e^1.
2. Next, we apply a topic model on D_e^1, to find keywords that describe the main aspects of the event that are being discussed. We have developed two topic models DTM and GDTM that are designed to extract relevant tweets.
3. Once we have obtained the set of topics Z from the topic models, the top ranked words in each topic $z \in Z$ are the keywords that describe various aspects of the event e. We may obtain the additional set of tweets D_e^2 by finding tweets $d \in D$ that are not present in D_e^1 by selecting those with high perplexity score with respect to the topics.
4. D_e^1 and D_e^2 can be merged to refine the model and improve upon the topics for the event e.
5. Using the final set of topics $z \in Z$, we can summarize the event e by selecting the tweets d from each topic z that give the best (lowest) perplexity.

The whole process can be performed for several iterations to improve the quality of the summary.

NP+LDA

Due to the noisy nature of tweets, it is typical to find that many of the words in a tweet contribute little or no information to the aspects of the target event. In order to avoid processing the unnecessary words in tweets, we remove stopwords and only consider noun phrases by applying a Part-of-Speech Tagger to extract noun phrases using the following regular expressions

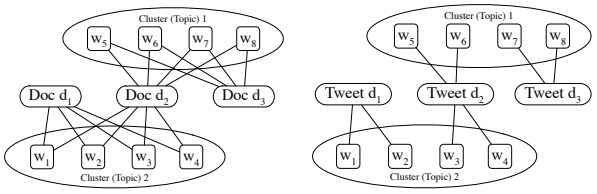
$$Base\_NP := determiner?\ \ adjectives*\ \ \ nouns+$$
$$Conj\_NP := Base\_NP(of\ \ Base\_NP)*$$

We then model the noun phrases in tweets using the NP+LDA model as described in Chua et al. (Chua et al. 2012). Instead of generating a topic for every word, we only generate a topic for each noun phrase which may consist of several words (Chua et al. 2012). The subsequent topic models we propose in the rest of the paper extends from NP+LDA.

The Problem of Short Documents

As we mentioned earlier, one of the inherent difficulties of modeling tweets is the short length of the tweets, most of which consist of typically 20 to 30 words. In order to understand why this is difficult, let us examine how topic modeling works on traditional documents using Figure 2.

Figure 2(a) shows three documents, d_1, d_2 and d_3 containing certain words $w_{1:8}$. The document d contains the word w if there is an edge connecting d and w. Topic models exploit the co-occurrences of words between documents to find relations between words. Given that d_1 and d_2 share the common words $\{w_1, w_2, w_3, w_4\}$, we can infer that this densely



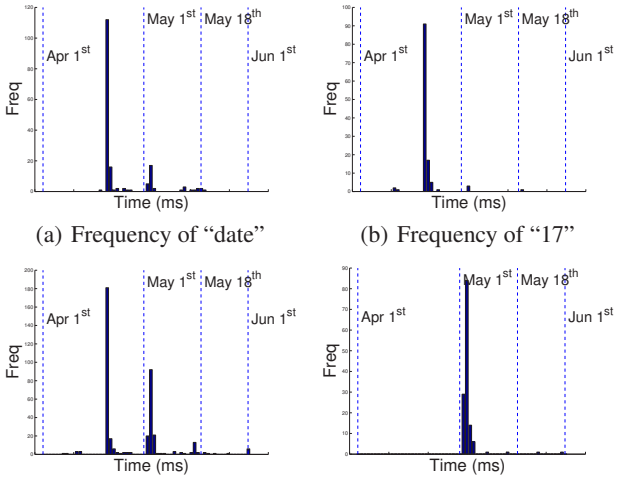
(a) Topic Modeling in Traditional Documents (b) Topic Modeling in Tweets

Figure 2: Topic Modeling on Documents vs Tweets

connected set of words forms a topic and has semantically similar meanings. Similarly for the set of common words between d_2 and d_3, we can infer that $\{w_5, w_6, w_7, w_8\}$ are semantically related and belong to the same topic.

However, in the case of tweets as shown in Figure 2(b), due to the smaller number of words, there is less likelihood for words to co-occur with one another across different documents. The words which could be inferred as belonging to the same topics as before now have a weaker co-occurrence relationship with other words.

Temporal Correlation of Twitter Content

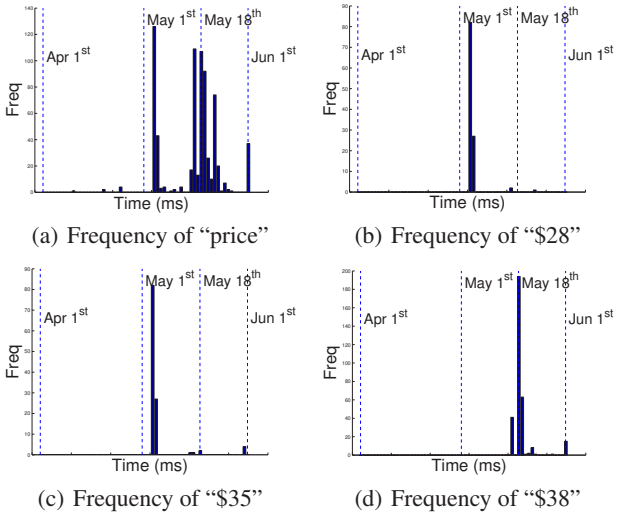


(a) Frequency of "date" (b) Frequency of "17"

(c) Frequency of "may" (d) Frequency of "18"

Figure 3: Facebook IPO Launch Date

Given this problem, the question is whether we can exploit other form of features to make up for the weakness and sparsity of Twitter. An additional feature that Twitter provides is the timestamp on each of the tweets, showing when the tweet was published. Figures 3 and 4 shows the trend of words written by Twitter users for the event "Facebook IPO". In these figures, the x-axis represents the timestamps with each vertical bin representing a day while the y-axis represents the frequency of the words written for the respective day (bin). In Figure 3, the words { "date", "17", "may", "18" } represent the topic of Twitter users discussing the launch date of "Facebook IPO". Figure 3(a)



(a) Frequency of "price" (b) Frequency of "$28"

(c) Frequency of "$35" (d) Frequency of "$38"

Figure 4: Facebook IPO Launch Price

for "date" and 3(c) for "may" show two spikes around the same period of time. Figure 3(b) shows that "17" has temporal co-occurrence with "date" and "may" in the first spike while Figure 3(d) shows that "18" has temporal co-occurrence with the second spike. Based on such temporal co-occurrence relationships, it leads us to infer that these words { "date", "17", "may", "18" } possibly belong to the same topic.

In Figure 4, the words represent the topic of Twitter users discussing the launch price of "Facebook IPO". Similar to the previous analysis of the launch date, the first spike in Figure 4(a) shows that the word "price" co-occurs with "28" in Figure 4(b) and "35" in Figure 4(c). Figure 4(d) shows that the word "38" co-occurs with the word "price" in the second spike. Using such temporal co-occurrences, we can infer that these words { "price", "28", "35", "38" } are likely to belong to the same topic.

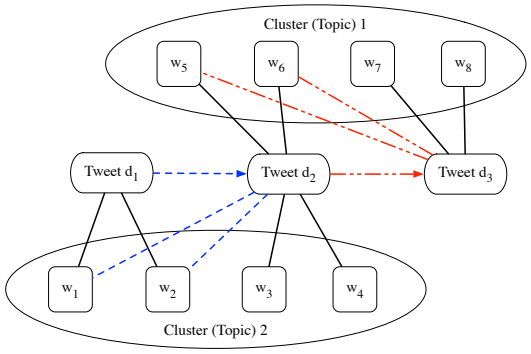


Figure 5: Topic Modeling in Temporal Correlated Tweets

By assuming that tweets written around the same time for the same event are similar in content, we could arrange the set of tweets in a sorted order such that tweets written around the same time can "share" words from other tweets to com-

pensate for their short length. Figure 5 shows an illustrated example of this idea. Assuming that tweet d_2 is written after tweet d_1, we could imagine d_2 as inheriting some of the words in d_1 as shown by the blue - - lines. Similarly, d_3 could also inherit some of the words written by d_2 as shown by the red -.. lines. The inheritance need not be strictly binary, instead it can be weighted according to the time difference between the two consecutive tweets. The next section will explain how we model the inheritance using an exponential decay function. As a result of this inheritance between tweets, the sparse Twitter data becomes denser and will improve the inference of topics from tweets.

Decay Topic Model (DTM)

Given that we want to allow tweets to inherit the content of previous ones, we need to define a model such that each tweet inherits not only the words of the immediately preceding tweet but also earlier tweets, subjected to an increasing decay as we increase the time difference between tweets. However there are several computational issues that we have to cope with. 1) Suppose we duplicate the existence of these words in later tweets for their inheritance, the size of the corpus will be inflated due to the duplication. The inflated corpus causes unnecessary repeated computation for inference of the duplicated words. 2) Suppose the duplication proceeds for every subsequent tweet, this accumulation of words will result in a snowball effect that eventually causes tweets with newer timestamps to inherit the content of all previous tweets. The snowballed tweets of later timestamps will have less diverse variations in their topic because of the baggage incurred from the inheritance.

We need to define our model such that 1) it avoids repetitive computation and 2) It decays the inheritance of the words such that the content in newer tweets do not get overwhelmed by the content of previous tweets. We address the first issue by the use of the topic distribution for each tweet. Since topic models summarize the content of tweets in latent space using a K (number of topics) dimensional probability distribution, we can allow the newer tweets to inherit this probability distribution instead of raw words. We address the second issue by the use of an exponential decay function for each dimension of the probability distribution. Here, we show the generative process of the Decay Topic Model (DTM).

1. For each topic z, sample the prior word distribution from a symmetric Dirichlet distribution,
$$\phi_z \sim Dir(\beta)$$
2. For the first tweet $d_1 \in D_e$, sample the prior topic distribution from a symmetric Dirichlet distribution,
$$\theta_{d_1} \sim Dir(\alpha)$$
3. For all other tweets $d_n \in D_e$, sample the prior topic distribution from an asymmetric Dirichlet distribution,
$$\theta_{d_n} \sim Dir\left(\left\{\alpha + \sum_{i=1}^{n-1} p_{i,z} \cdot exp\left[-\delta_z(t_n - t_i)\right]\right\}_{z \in Z}\right)$$
where $p_{i,z}$ is the number of words in tweet d_i that belong to topic z and δ_z is the decay factor associated with topic z. The larger the value of δ_z, the faster the topic z loses its novelty. t_i is the time that tweet d_i was written. The summation is over all the tweets $[1, n-1]$ that were written before tweet d_n. Each $p_{i,z}$ is decayed according to the time difference between tweet d_n and tweet d_i. Although the summation seems to involve an $O(n)$ operation, the task can be made $O(1)$ via memoization, a programming technique.
4. For each noun phrase np in tweet d, sample a topic variable $z_{d,np}$ from a multinomial distribution using θ_d as parameters.
$$z_{d,np} \sim Mult(\theta_d)$$
5. For each noun phrase np, sample words $w_{np,v}$ for the tweet d using topic variable $z_{d,np}$ and the topic word distribution ϕ_z.
$$P(w_{d,np}|z_{d,np} = k, \phi) = \prod_{v \in np} P(w_{d,np,v}|z_{d,np} = k, \phi_k) = \prod_{v \in np} \phi_{k,v}$$

The inference algorithm for DTM is given by,

$$P(z_{d,np} = k|w_{d,np}, \alpha, \beta, \delta_k) \propto \left[\prod_{v \in np} \frac{\Gamma(\beta + q_{k,v} + |v \in np|)}{\Gamma(\beta + q_{k,v})}\right] \frac{\Gamma(V\beta + q_k)}{\Gamma(V\beta + |np| + q_k)} \left[\alpha + \sum_{i=1}^{n-1} p_{i,k} \cdot exp\left(-\delta_k(t_n - t_i)\right)\right]$$

where V is the total size of vocabulary, $|np|$ is the number of words in the noun phrase, $|v \in np|$ is the number of times v appear in np, $q_{k,v}$ is the number of times v is inferred as topic k and q_k is the number of words that are in topic k.

After performing inference on the set of tweets $D_{facebook\_ipo}$ for the "Facebook IPO" event, we observe the trend of the topics by plotting how important the topics are at different time points. Figure 6 shows the intensity (y-axis)

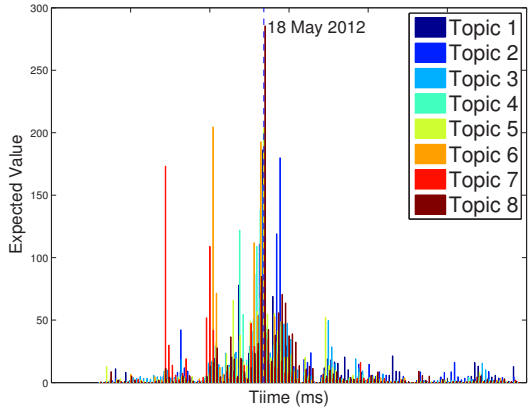


Figure 6: Chronological Intensity of Topics

of every topic differentiated by different colors with respect

to time (x-axis). Each vertical bar for a topic (color) represents the expected number of tweets in the day (bin). The expected value $E_{day}(z)$ of topic z for a day (bin) is given by

$$E_{day}(z) = \sum_{d \in D_{day}} \theta_{d,z}$$

where D_{day} represent the set of tweets in a given day.

However, we are not able to observe a smooth transition of topics between different times. Based on the original definition of the model, we assumed that tweets written around the same time should share high similarity in their content and hence topic distributions as well. But Figure 6 does not show that the resulting topics are well differentiated by time. This motivates us to address a deficiency in the Decay Topic Model. Since we have already modeled the temporal correlation of tweets by adding the exponential decay function between the tweets' topic distributions, we could also add additional parameters to the topic word distributions to model the assumption that words specific to certain topics has higher chance of appearing at specific times. This leads us to introduce the Gaussian Decay Topic Model.

Gaussian Decay Topic Model (GDTM)

The generative process for the Gaussian Decay Topic Model (GDTM) follows that of DTM with the addition of the time stamp generation for each noun phrase,

1. In addition to topic word distribution ϕ_z, each topic z has an additional topic time distribution $\mathcal{G}_z$ approximated by the Gaussian distribution with mean μ_z and variance σ_z^2.

$$\mathcal{G}_z \sim \mathcal{N}(\mu_z, \sigma_z^2)$$

2. Then the time t for a noun phrase np is given by the following,

$$P(t_{np}|z, \mathcal{G}_z) = \frac{1}{\sqrt{2\pi\sigma_z^2}} exp\left(-\frac{(t_{np}-\mu_z)^2}{2\sigma_z^2}\right)$$

Since every topic z is now associated with a Gaussian distribution $\mathcal{G}_z$, we can use the shape of the distribution curve to determine the decay factors $\delta_z, \forall z \in Z$. The δ_z which was previously used for transferring the topic distribution from previous tweets to the subsequent tweets can depend on the variances of the Gaussian distributions. Topics having small values of variance σ_z^2 indicate aspects that have short lifespans and should decay quickly (larger δ_z), while topics with large variance represent aspects that should decay slowly giving it a smaller δ_z. The inference algorithm for GDTM is as follows,

$$P(z_{d,np} = k | w_{d,np}, \alpha, \beta, \delta_k, \mu_k, \Sigma_k^2) \propto \frac{\Gamma(V\beta + q_k)}{\Gamma(V\beta + |np| + q_k)}$$
$$\left[\prod_{v \in np} \frac{\Gamma(\beta + q_{k,v} + |v \in np|)}{\Gamma(\beta + q_{k,v})}\right] \left[\frac{1}{\sqrt{2\pi\sigma_k^2}} exp\left(-\frac{(t_{np}-\mu_k)^2}{2\sigma_k^2}\right)\right]^{|np|}$$
$$\left[\alpha + \sum_{i=1}^{n-1} p_{i,k} \cdot exp\left(-\delta_k(t_n - t_i)\right)\right]$$

where V is the total size of vocabulary, $|np|$ is the number of words in the noun phrase, $|v \in np|$ is the number of times v appear in np, $q_{k,v}$ is the number of times v is inferred as topic k and q_k is the number of words that are in topic k.

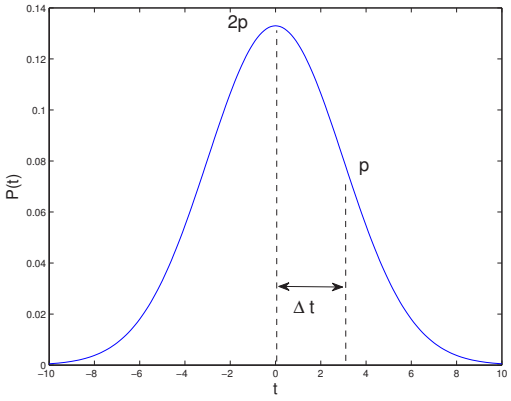


Figure 7: Find Δt

We use the concept of half-life to estimate the value of δ_z. Given that we want to find the δ_z value that causes a tweet to discard half of the topic from previous tweet,

$$exp(-\delta \cdot (t_n - t_{n-1})) = 0.5$$
$$\delta \cdot \Delta T = \log 2$$
$$\delta = \frac{\log 2}{\Delta T}$$

Figure 7 shows a Gaussian distribution with an arbitrary mean and variance. The value of ΔT is affected by the variance (width) of the distribution. To estimate ΔT, let $\Delta T = \tau \Delta t$ where τ is a parameter and Δt is estimated as follows,

$$\frac{P(0)}{P(\Delta t)} = \frac{2p}{p}$$
$$\frac{exp(0)}{exp(-\frac{(\Delta t)^2}{2\sigma^2})} = 2$$
$$\frac{(\Delta t)^2}{2\sigma^2} = \log 2$$
$$\Delta t = \sqrt{2\sigma^2 \log 2}$$

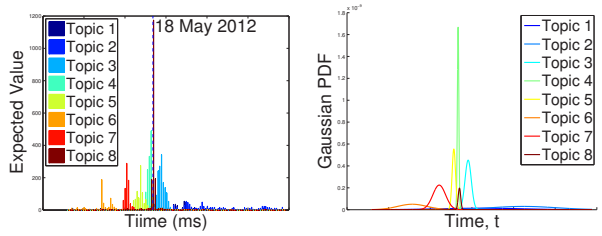
Finally, δ is given by,

$$\delta = \frac{\log 2}{\tau\sqrt{2\sigma^2 \log 2}} \quad (1)$$

As shown in Equation 1, the larger the variance σ^2 is, the smaller the δ (decay) and vice versa.

Figure 8(a) shows the intensity (y-axis) of all the topics each differentiated with different colors with respect to time (x-axis) found by the Gaussian Decay Topic Model (GDTM). Figure 8(b) shows the corresponding Gaussian components for each topic.

While some other probability distributions can also be used to describe the time distribution of words in an event, we choose the Gaussian distribution because of the ease



(a) Chronological Intensity of Gaussian Decay Topics (b) Gaussian Components

Figure 8: Intensity and the Gaussian Components

of computing sufficient statistics for inference using Gibbs Sampling. We also exploit the symmetry of Gaussian distribution in estimating δ. This symmetric property cannot be observed in most continuous distributions.

Additional Tweets from the Tweet Sample

After finding the topics from the initial set of relevant tweets D_e^1, the next step is to find additional tweets D_e^2 from the tweet stream D using the trained model. A related method of achieving this is to perform query expansion by using the top words in a topic for keyword search. Instead of applying a threshold on selecting the top-k keywords for query expansion, we compute a perplexity score for each tweet $d \in D, d \notin D_e^1$. Tweets relevant to the event e are then ranked in ascending order with lower perplexity being more relevant to event e. Using the perplexity score instead of keyword search from each topic allows us to differentiate between the importance of different words using the inferred probabilities. The perplexity of tweet d is given by the exponential of the log likelihood normalized by the number of words in a tweet.

$$perplexity(d) = exp\left(\frac{-\log P(d|\theta, \phi, \mathcal{G})}{N_d}\right) \quad (2)$$

where N_d is the number of words in tweet d. Since tweets with fewer words tend to have higher probabilities and therefore lower perplexity, we normalized by N_d in order to favour tweets with more words.

Summarization

Our goal is to use the extracted topics to summarize the event e. Summarizing the event is a multi-objective problem. On one hand we want to select tweets such that we maximize the total perplexity using as few tweets as possible. But we also want the topic overlap between the selected tweets to be as low as possible.

The models described earlier are designed to provide us diverse topics representing the various different aspects of the event that are being discussed on Twitter. Using the topics learned from the set of relevant tweets D_e, we can obtain the most representative tweet from each topic to summarize the target event e.

To choose the most representative tweet for topic z, we compute the perplexity with respect only to topic z for all tweets $d \in D_e$ and choose the tweet that has lowest perplexity with respect to z.

$$perplexity(d, z) = exp\left(\frac{-\log P(d, z|\theta, \phi_z, \mathcal{G}_z)}{N_d}\right)$$

The list of representative tweets for each topic forms the summary of the event e. Note that, depending on the size of the summary required, we could extract additional representative tweets for each topic, based on the perplexity scores. Since we choose one tweet from each topic, then the number of topics K determines the number of selected tweets for the summarized event.

Experiments

To validate our choice of using the temporal correlation between tweets to extract topics, we evaluate the two models DTM and GDTM with respect to the LDA baseline[1]. Unlike our DTM and GDTM models, the LDA baseline does not consider the use of noun phrases and assumes that every tweet has no temporal relation to other tweets. One possible way to evaluate the temporal correlation is to compare the convergence log-likelihoods of these models and assume that the model with the highest log-likelihood during convergence is a better model. Alternatively, we can also compute the perplexity score of a held-out test set.

However these approaches have the following problems, 1) The models make different assumptions on the generative process of the data, especially LDA which considers tweets as a bag-of-words while DTM and GDTM consider noun phrases. 2) Tweets contain a great deal of noise in them. Many of the tweets containing keywords such as "Facebook" and "IPO" are found to be spam instead. These tweets try to gain attention and visibility by riding on the popularity of these trending keywords during the occurrence of these events. A model that optimizes for the log-likelihood or perplexity score risks over-fitting the parameters to these noisy tweets.

We therefore evaluate the temporal correlation and the two derivative models by comparing 1) the quality of the summaries generated from these models and 2) their utility towards finding additional tweets from the tweet sample that are related to the event and yet do not contain the keywords from the original queries.

Events and Data Set

We perform our experiments for four real-world events, selected to cover natural disasters, politics and company events. For each event, we apply a set of queries on the sample of tweets D to obtain the relevent set of tweets, D_e^1. The events used in this study are:

1. **Facebook IPO**: The Initial Public Offer (IPO) of Facebook Inc. (web c). We use { { (Facebook — FB), IPO }, { (Facebook — FB), Initial, Public, (Offer — Offering) } } as queries to obtain a set of 9,570 tweets.
2. **Obamacare**: The Patient Protection and Affordable Care Act (web d). We use { { Obamacare }, { PPACA }, {

[1]The constants for the prior distributions are set as 0.1

Obama, Health, Care }, { Obama (Healthcare — Healthcare) } } to obtain a set of 136,761 tweets.

3. **Japan Earthquake**: The earthquake that occurred near Tokyo, Japan in 2011 (web a). We use { { Fukushima }, { (Japan — Tokyo), (Earthquake — Quake — Tsunami } } to obtain a set of 251,802 tweets.
4. **BP Oil Spill**: The oil spill resulted from British Petroleum (BP) drilling in the Gulf of Mexico (web b). We use { { BP, Oil, Spill }, { Gulf, Mexico, Oil, Spill } } to obtain a set of 79,676 tweets.

Note that the number of tweets for these events ranges from a small 9,570 tweets for Facebook IPO to a mammoth 251,802 tweets for the Japan Earthquake.

Summarization Results

Fair evaluations of our summaries require both a quantitative comparison with simulated true summaries and qualitative assessment from human readers. Due to the difficulty of obtaining human generated summaries from our data sets, we construct the true summaries by using the headlines of news articles found in the reference section of the events' Wikipedia articles. The human readers are crowdsourced from Amazon Mechanical Turk (MTurk).

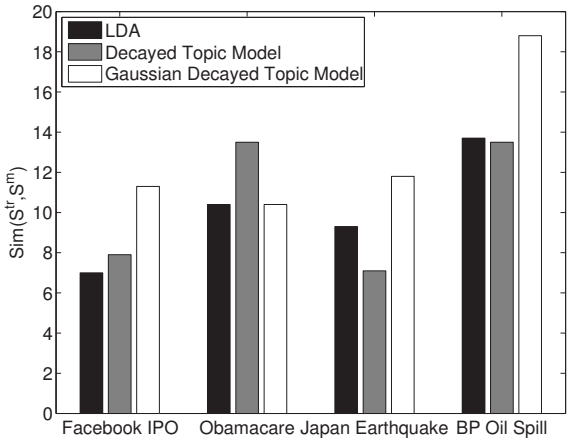
Quantitative Comparison with Wikipedia: Wikipedia forms a comprehensive resource for all manner of real-world content including the events that we consider in this paper. Each Wikipedia article for an event contains a section that references the relevant news articles which contributed to the article. These news articles thus can be considered as proxies for each of the important pieces of news about the event. Since Wikipedia articles are edited and discussed by the general public, the news articles that are referenced represent the popular choices of the internet public. For each of the Wikipedia references for our events, we extract the headline text which gives a one-line summary of the corresponding news article. The headline also has an advantage of resembling the language style used in tweets. To construct a true summary for each event from its corresponding Wikipedia article, we aggregate the one-line summaries of all the news articles referenced in the Wikipedia article. We then compare the true summary with the summaries we generated from each model using a similarity metric.

The similarity metric we use for the comparison of summaries is adapted from the ROUGE metric proposed by Lin and Hovy (Lin and Hovy 2003). The ROUGE metric counts the total number of matching n-grams (excluding stop-words) between the true summary S^{tr} and the summary S^m generated from model m. We let NG_n^{tr} denote the set of n-grams from the true summary and NG_n^m denote the n-grams from summaries generated by the model m.

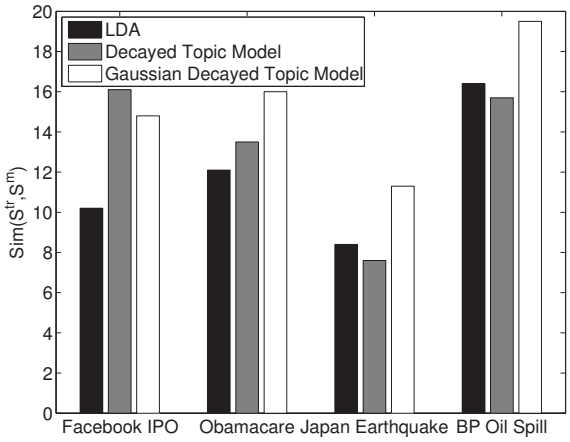
$$g_n = \sum_{ng \in NG_n^m} min(|ng \in NG_n^{tr}|, |ng \in NG_n^m|) \quad (3)$$

$$Sim(S^{tr}, S^m) = 0.2 \cdot g_1 + 0.3 \cdot g_2 + 0.5 \cdot g_3 \quad (4)$$

Equation 3 first calculates the number of n-grams common to both S^{tr} and S^m. In order not to let a few frequent n-gram dominate the counts, each n-gram is limited to the minimum number of counts between the true summary and the generated summary. Equation 4 calculates the final similarity score between the summaries by aggregating the number of matched 1,2 and 3 grams. The weights allocated are meant to give a higher importance to 3-grams and lower importance to 1-grams.



(a) Results for 8 Topics



(b) Results for 10 Topics

Figure 9: Results of $Sim(S^{tr}, S^m)$ Score

Figures 9(a) and 9(b) show two sets of results for different number of topics. The y-axis gives the similarity score between the model-generated summaries and the true summary, while the x-axis differentiates between the various events. Figures 9(a) and 9(b) show that GDTM consistently gives better performance than LDA. DTM is shown to be better than GDTM for "Obamacare" at K=8 in Figure 9(a) and "Facebook IPO" at K=10 in Figure 9(b). Overall, GDTM shows better performance over DTM and LDA with DTM showing inconsistent performance. DTM is sometimes better than LDA and sometimes slightly worse-off than LDA. This suggests that estimating the appropriate decay parameters is important for using the temporal correlation features.

Since we extract the most representative tweet for each

topic, the use of K topics gives K tweets as the summary for each event. In our experiments, we use $K = 8$ and $K = 10$, to obtain 8-tweet summaries and 10-tweet summaries for each event. We choose these values of K to avoid generating long summaries for the events so that the human evaluation task in Mechanical Turk will be easier for our mechanical turk workers.

Qualitative Evaluation on Mechanical Turk: We used Amazon's Mechanical Turk to find human evaluators for this task. To each mechanical turk worker (mturker), we presented the generated summaries of the four events from each model. Since each event has three summaries from the three models, mturkers were instructed to choose 1 or 2 out of the 3 summaries as the best representations of the event. We also provided a feature for mturkers to leave comments. To avoid bias to any one model, we did not show which model generated the summaries and we randomized the order of presented summaries. We required mturkers to fulfil three criteria before they could participate in our experiment: 1) have approval rating of $\geq$ 95%, 2) have completed more than 1000 tasks, and 3) are located in United States (USA) because the events "Facebook IPO", "Obamacare" and "BP Oil Spill" are more relevant to the population of USA. Although "Japan Earthquake" did not occur in USA, the event can be considered to be of interest to everyone worldwide. We employed a total of 100 different mturkers. Each mturker spent an average of 7 minutes and 10 seconds to complete the task which translated to an hourly wage of approximately USD$4.2.

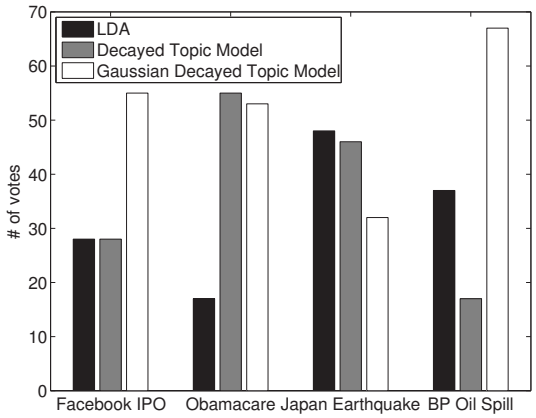


Figure 10: Mechanical Turk Results

Figure 10 shows the results from Mechanical Turk with the x-axis differentiating between the four events and the y-axis showing the number of votes for the respective models. Aggregating the total votes for all events, GDTM has the most number of votes with 207 votes followed by DTM with 146 votes and finally LDA with 130 votes.

Some comments given by our mturkers were : 1) "*I think these (GDTM) best summarize Facebook IPO because it shows a broad range of information related to the event.*" 2) "*Well I learned that their co-founder renounced his US citizenship just now!*" 3) "*I believe that other summaries (non-GDTM) had a large amount of personal opinion and not fact.*" 4) "*If I wanted to find information, Summaries (DTM) and (GDTM) had the most.*" 5) "*There is too much garbage posts in the other summaries (non-GDTM), and not true news.*" These comments show that readers appreciated the GDTM summaries and felt that it was a good representation of the event.

Search Results

Next we evaluate the models on their ability to retrieve additional tweets that are relevant to the event but do not contain the keywords in the queries. The relevance of a tweet is measured based on perplexity, as given by Equation 2. We calculate the perplexity score for each of the tweets with respect to each of the models from the sample of tweets and then rank it accordingly. This way, we obtain three ranked lists from the three models.

Traditional Information Retrieval (IR) evaluation is done by going through each list from top to bottom to compute the precision and recall curve at each k. In our evaluation here, we do not make a binary decision as to whether each tweet is relevant to the event. Instead, we compute the number of n-grams that matched between the top-k tweets and the true summary. Because we vary n from 1 to 3, we obtain three sets of precision PR_n and recall RC_n values. The precision and recall are calculated as follows,

$$g_n = \sum_{np \in NG_n^{top-k}} min(|ng \in NG_n^{tr}|, |ng \in NG_n^{top-k}|)$$

$$PR_n^k = \frac{g_n}{|D^{top-k}|}$$

$$RC_n^k = \frac{g_n}{|NG_n^{tr}|}$$

$$PR^k = 0.2 \cdot PR_1^k + 0.3 \cdot PR_2^k + 0.5 \cdot PR_3^k$$

$$RC^k = 0.2 \cdot RC_1^k + 0.3 \cdot RC_2^k + 0.5 \cdot RC_3^k$$

Varying k from 1 to the size of the tweet sample gives us the Precision-Recall curve (PR-curve) as shown in Figure 11. Figure 11 shows us that the results given by GDTM is sig-

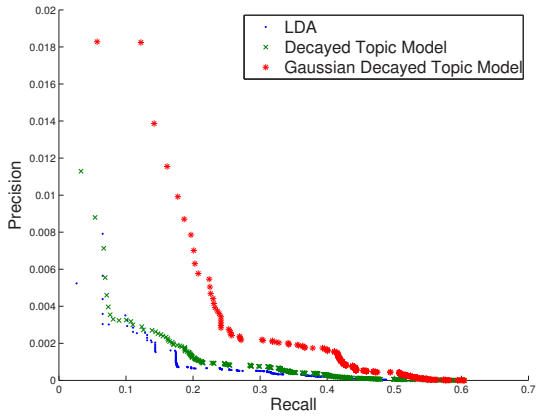


Figure 11: Precision Recall Curves of D_e^2

nificantly better than DTM and DTM is in turn better than LDA. The precision and recall does not give high values because the true summaries are keywords from Wikipedia articles which are different corpus from the generate summaries.

Discussion and Future Work

We have presented our framework for summarizing events from a sample of tweets. We have developed two topic models, the Decay Topic Model (DTM) and Gaussian Decay Topic Model (GDTM) that leverage the temporal correlation that exists among tweets written around the same time, to extract meaningful topics that capture different aspects of the underlying event. We have shown how representative tweets with low perplexity can be selected from the extracted topics to generate a concise and information-rich summary of the events. Our experiments evaluating the summaries using Wikipedia links as well as the qualitative evaluation using Mechanical Turk have demonstrated that both our topic models generated summaries that outperformed traditional LDA in almost all cases with GDTM having the highest performance overall and also receiving the highest overall votes from the Turk workers. The Search and Summarize framework also proceeds in an iterative loop, with newer search queries being generated from the extracted topic models and the resultant tweets used to refine the topic model and the summary.

Since our approach relies on computing topic models, the running time would depend on training the model as well as rounds of iterations for improving them by adding more tweets. In this paper, we have assumed that relevant keywords are easy to determine for the initial search. This is generally true in most cases, where the search is for a particular event or brand or product. Alternatively, one may use an independent event detection system such as (Becker, Naaman, and Gravano 2011) to find the relevant keywords. We have used the Twitter search API to extract the tweets relevant to our four events. This can also be done in an incremental manner to gain full coverage of an ongoing event.

We believe that the area of social media summarization has lots of scope for future work. To that end, we have received insightful feedback from Mechanical Turk workers. Some workers preferred summaries that fit their own beliefs and opinions. Thus personalized summaries could be extracted that are tailored to suit particular sentiments or beliefs. And conversely, factual tweets could be weighted higher to generate objective summaries. We wish to extend our work in these directions.

Acknowledgements

The authors will like to thank all members of the Social Computing Research Group, Hewlett Packard Research Lab for their helpful comments and discussions.

References

Ahmed, A.; Low, Y.; Aly, M.; Josifovski, V.; and Smola, A. J. 2011. Scalable distributed inference of dynamic user interests for behavioral targeting. KDD '11.

Becker, H.; Naaman, M.; and Gravano, L. 2011. Beyond trending topics: Real-world event identification on twitter. In *ICWSM*.

Chakrabarti, D., and Punera, K. 2011. Event summarization using tweets. ICWSM '11. The AAAI Press.

Chua, F. C. T.; Cohen, W. W.; Betteridge, J.; and Lim, E.-P. 2012. Community-based classification of noun phrases in twitter. CIKM '12. ACM.

Ganesan, K.; Zhai, C.; and Viegas, E. 2012. Micropinion generation: an unsupervised approach to generating ultra-concise summaries of opinions. WWW '12.

Lin, C.-Y., and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. NAACL '03.

Lu, Y.; Zhai, C.; and Sundaresan, N. 2009. Rated aspect summarization of short comments. WWW '09.

Meng, X.; Wei, F.; Liu, X.; Zhou, M.; Li, S.; and Wang, H. 2012. Entity-centric topic-oriented opinion summarization in twitter. KDD '12.

Metzler, D.; Cai, C.; and Hovy, E. 2012. Structured event retrieval over microblog archives. In *NAACL-HLT*, 646–655. Montréal, Canada: Association for Computational Linguistics.

Nenkova, A., and McKeown, K. 2012. A survey of text summarization techniques. In Aggarwal, C. C., and Zhai, C., eds., *Mining Text Data*. Springer US.

Popescu, A.-M., and Pennacchiotti, M. 2010. Detecting controversial events from twitter. CIKM '10.

Saha, A., and Sindhwani, V. 2012. Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. WSDM '12.

Sakaki, T.; Okazaki, M.; and Matsuo, Y. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. WWW '10.

Sayyadi, H.; Hurst, M.; and Maykov, A. 2009. Event detection and tracking in social streams. ICWSM, '09.

Sharifi, B.; Hutton, M.-A.; and Kalita, J. 2010. Summarizing microblogs automatically. HLT '10.

Wang, Y.; Agichtein, E.; and Benzi, M. 2012. Tm-lda: efficient online modeling of latent topic transitions in social media. KDD '12.

Wang, X., and McCallum, A. 2006. Topics over time: a non-markov continuous-time model of topical trends. KDD '06.

Watanabe, K.; Ochi, M.; Okabe, M.; and Onai, R. 2011. Jasmine: a real-time local-event detection system based on geolocation information propagated to microblogs. CIKM '11.

2011 tōhoku earthquake and tsunami. http://en.wikipedia.org/wiki/2011_Tōhoku_earthquake_and_tsunami.

Deepwater horizon oil spill. http://en.wikipedia.org/wiki/Deepwater_Horizon_oil_spill.

Facebook ipo. http://en.wikipedia.org/wiki/Facebook_IPO.

Patient protection and affordable care act. http://en.wikipedia.org/wiki/Patient_Protection_and_Affordable_Care_Act.

Weng, J., and Lee, B.-S. 2011. Event detection in twitter. ICWSM '11.

Yang, X.; Ghoting, A.; Ruan, Y.; and Parthasarathy, S. 2012. A framework for summarizing and analyzing twitter feeds. KDD '12.