On-demand recent personal tweets summarization on mobile devices

Chin, Jin Yao; Bhowmick, Sourav S.; Jatowt, Adam

2019

Chin, J. Y., Bhowmick, S. S. & Jatowt, A. (2019). On-demand recent personal tweets summarization on mobile devices. Journal of the Association for Information Science and Technology, 70(6), 547-562. doi:10.1002/asi.24137

https://hdl.handle.net/10356/144489

https://doi.org/10.1002/asi.24137

On-Demand Recent Personal Tweets Summarization on Mobile Devices

Jin Yao Chin
Nanyang Technological University, Singapore 639798. E-mail: s160005@ntu.edu.sg

Sourav S. Bhowmick
Nanyang Technological University, Singapore 639798. E-mail: assourav@ntu.edu.sg

Adam Jatowt
Kyoto University, Kyoto 606-8501, Japan. E-mail: adam@dl.kuis.kyoto-u.ac.jp

Tweets summarization **aims to find a group of representative tweets for a specific set of input tweets or a given topic. In recent times, there have been several research efforts toward devising a variety of techniques to summarize tweets in Twitter. However, these techniques are either not *personal* (that is, consider only tweets in the timeline of a specific user) or are too expensive to be realized on a mobile device. Given that 80% of active Twitter users access the site on mobile devices, in this article we present a lightweight, personal, on-demand, topic modeling-based tweets summarization engine called TOTEM, designed for such devices. Specifically, TOTEM first preprocesses recent tweets in a user's timeline and exploits Latent Dirichlet Allocation-based topic modeling to assign each preprocessed tweet to a topic. Then it generates a ranked list of relevant tweets, a topic label, and a topic summary for each of the topics. Our experimental study with real-world data sets demonstrates the superiority of TOTEM.**

Introduction

Twitter is an online social networking service, in which users post short messages called *tweets*. Similar to several online social networking platforms such as Facebook and Instagram, Twitter has also adopted a reverse chronological timeline to display tweets.[1] Users are typically able to scroll through posts on their timelines one by one, beginning with the most recent post. Consequently, it can often be daunting to get an overview of recent contents being discussed due to the high volume and velocity of tweets.

With the explosive growth of mobile devices, interestingly, it is estimated that 80% of active Twitter users access it on mobile platforms.[2] Most mobile users intermittently visit Twitter with varying frequencies (that is, several times an hour to once in few days). They mostly consume information but rarely post tweets of their own (Pennacchiotti, Silvestri, Vahabi, & Venturini, 2016). Consequently, interesting or relevant tweets on a user's timeline can easily be missed using the aforementioned tweets display scheme.

To alleviate the above-mentioned issues, Twitter introduced a new concept called the *algorithmic timeline* in February 2016. When a user invokes Twitter after being away for a while, it aims to show tweets that the user is most likely to care about at the top of the timeline. These tweets are selected by analyzing user interaction history with tweets and followers. However, it still fails to address the inability of a user to get a bird's-eye view of topics being discussed in her/his recent posts.

A palatable way to address the aforementioned challenge is to provide a summary of the topics being discussed in the timeline of a user's recent tweets. In this article, we present a novel *tweets summarization* (that is, a group of representative tweets for a specific set of input tweets or a given topic) framework called TOTEM[3] (**T**opic M**O**deling-based Recen**T** Tw**E**et Su**M**marizer) that enables a user to obtain an overview of the most salient topics present in the recent tweets on his/her timeline. Furthermore,

[1] We omit the discussion of algorithmic timeline (https://support.twitter.com/articles/164083)

[2] https://about.twitter.com/

[3] TOTEM has been demonstrated in SIGIR 2017 (Chin, Bhowmick, & Jatowt, 2017).

it assists a user to easily identify topics that s/he is most interested in and zoom into a specific topic and representative tweets.

TOTEM has three distinguishing features. First, it focuses on summarizing a user's recent personal tweets. That is, it summarizes tweets that are on the timeline of a user's Twitter account. We advocate that users are typically not interested in arbitrary tweets but mostly in those from their followers (Kwak, Lee, Park, & Sue, 2010). Hence, it is important to summarize these personal tweets instead of summarizing any arbitrary collection of tweets. Second, TOTEM realizes on-demand summarization instead of continuous summarization of tweet streams. Since users intermittently connect to their Twitter account, they may not always be interested in viewing the summaries. Sometimes they may simply intend to browse some of their recent tweets and at other times they may wish to invoke the summarization feature to get an overview of the salient topics. On-demand summarization enables users to control when summaries should be displayed to them. A byproduct of this feature is the reduction of unnecessary computation, as summaries do not need to be computed and maintained continuously. Third, TOTEM is designed specifically for mobile devices, as the majority of users access Twitter using such devices. Consequently, it is lightweight in design in order to tackle limited memory, limited processing power, limited network connectivity, and the small screen size of mobile devices. A user can simply visualize the summary by using a tap-and-swipe approach (Chin et al., 2017).

TOTEM summarizes recent tweets in a user's timeline as follows. First, it utilizes the Twitter API to retrieve the most recent tweets in a user's home timeline and store them locally on a mobile device. Next, it preprocesses these tweets to extract various textual features from the tweets and perform various "cleaning" steps such as removal of near-duplicate tweets, conversion of acronyms and abbreviations, and removal of stop words. Then Latent Dirichlet Allocation (LDA)-based topic modeling (Blei, Ng, & Jordan, 2003) is leveraged to assign each preprocessed tweet to a topic. Since each topic may contain many tweets, not all of which are relevant to the most salient topic, TOTEM ranks these tweets to find the most relevant tweets for a given topic and generate a meaningful topic label for it. Finally, a topic summary (that is, representative tweets) is generated for each topic.

Preprocessing Personal Tweets

In this section, we describe the steps we take to preprocess recent personal tweets so that they can be subsequently summarized.

Recent Tweet Retrieval

In order to summarize recent personal tweets, we need to retrieve them from a user's account by utilizing the functionalities provided by the Twitter rest API. To this end, we utilize the Fabric Software Development Kit (Fabric SDK) created by Twitter for the mobile platform. Note that the API only allows the retrieval of the 800 most recent tweets. These tweets are stored locally on the user's mobile device in an SQLite database.

Preprocessing Tweets

The aim is to preprocess the retrieved tweets so that topic modeling and high-quality summarization can be performed on them effectively. An important issue in realizing this goal is to strike a balance between the amount of preprocessing that needs to be performed against the time it takes to perform them. Figure 1 depicts an overview of our preprocessing steps.

First, textual contents of the tweets are extracted. In this context, retweets are handled separately, as the original content might be truncated due to the 140 characters limit imposed by Twitter. Since a retweet includes the "retweeted_status," the original textual content can easily be obtained from it. Next, all textual contents are converted to lowercase. Subsequently, we perform the following preprocessing tasks.

Tweet cleaning. We first "clean" the content of retrieved tweets. First, user mentions, URLs from the linked web pages or embedded media elements are removed. Additionally, we remove all nonprintable ASCII characters. Finally, the cleaned text is tokenized.

Note that this step does not undertake spelling correction and hashtag segmentation. Although the presence of hashtags (which are often multiword expressions) in tweets may mislead the topic model, and both spelling correction and hashtag segmentation are expensive to perform, especially in a mobile device. Fortunately, as we shall see in the Performance Study section, such omissions do not have significant adverse impact on the quality of summary.

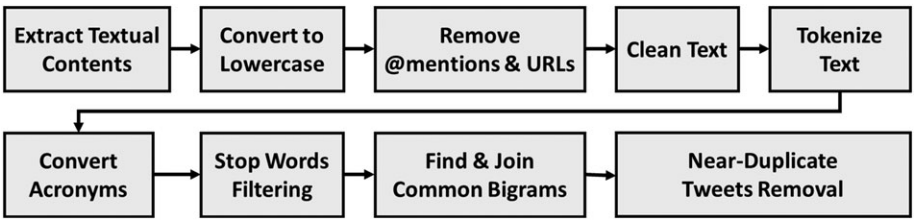

FIG. 1. Preprocessing pipeline for recent personal tweets.

FIG. 2. Examples of near-duplicate tweets. Top: Information content of the second tweet is contained in the first. Bottom: Both tweets are semantically equivalent. [Color figure can be viewed at wileyonlinelibrary.com]

Convert acronyms and stop words filtering. The 140 characters constraint on Twitter has led users to come up with different strategies to get messages across to other users in the most succinct way. This includes the use of acronyms or abbreviations. We convert such acronyms and abbreviations to its original form (for instance, "govt" to "government") by leveraging a conversion dictionary (Slang, 2010). We also eliminate common stop words using a predefined list.

Bigram processing. Next, we identify phrases that have better semantic meaning when treated as a single entity. Some examples include "rocket launch" and "surface water." This also helps to eliminate ambiguity associated with words that are quite common but are not stop words. We identify phrases that tend to co-occur together by leveraging a list of over 50,000 of the most common bigrams (Norvig, 2016) and we join them using an underscore character (for instance, "surface water" to "surface_water").

Tweets with less than three tokens at the end of the aforementioned preprocessing steps are removed, as it is unlikely for the topic model to be able to utilize them effectively.

Near-duplicate tweets removal. Often tweets on a user's timeline may contain identical or nearly identical textual content but different URLs (that is, near-duplicates). Figure 2 shows examples of near-duplicate tweets. Hence, in our last step we remove such near-duplicate tweets. To this end, we leverage the Cross-Sentence Informational Subsumption (csis) technique (Radev, Jing, & Budzikowska, 2000), which measures the informational content of sentences. Specifically, this enables us to remove the following categories of tweets: (i) those which only differ in their usage of stop words or URLs; and (ii) tweets that subsume some other tweets. As we shall see in the Performance Study section, after performing the aforementioned preprocessing steps, around 85% or more tweets are typically retained in a data set.

Tweets Topics Generation

In this section we present a topic modeling-based technique to identify a set of topics associated with the collection of preprocessed tweets and label each tweet with the most relevant topic.

LDA-Based Topic Modeling

We utilize the Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) for identifying topics associated with the preprocessed tweets. Particularly, we extend the mallet toolkit in order to use it within an Android application.

The generative process of the LDA topic model uses two hyperparameters, α and β, for Dirichlet distribution. It should be noted that the LDA implementation in the mallet toolkit uses symmetric Dirichlet distributions. For such distribution, a high α value implies that each document (tweet in this case) is likely to be made up of a large number of different topics. On the other hand, a low α value suggests that it is more likely that a document may contain a mixture of just a few topics. Since tweets are constrained by the 140 characters limit, there are only 3–10 words left in each tweet after preprocessing. Hence, it is reasonable to assume that each tweet contains at most one or two topics. Therefore, α is set to 0.005.

Likewise, a high β value means that each topic is likely to contain a mixture of a majority of the words within the vocabulary, and not any specific set of words. A low β value means that a topic may contain only a mixture of just a few of the words, and topics tend to be less similar in terms of their topic word distribution. Given the sparsity of terms in our data sets, β is set to 0.01.

Number of Topics and Number of Iterations

The LDA model requires the number of topics (denoted T) to be modeled. A small T value would cause the topic model to identify very broad topics, while a large T value might result in very detailed and fine-grained topics. The value of T is often estimated based on the size of the data set, and different values of T can be experimented with to find the "optimal" number of topics. However, since the data set to be used for topic modeling would in fact consist of the most recent tweets retrieved from a Twitter user's home timeline, the accurate number

of topics that would exist in such a data set cannot be determined a priori.

Recall that after the preprocessing steps, around 85–90% of the 800 most recent tweets are retained. Out of these tweets, some might just be chatter and do not contain any topical meaning. Furthermore, while certain tweets might be semantically related, the number of such tweets might by insufficient for forming a topic. Therefore, based on the observed number of tweets that would be used as input for topic modeling and experimenting with different values, we decided to use 8 as the default value for T. Note that a user can change this value as desired.

Lastly, the number of iterations used for the topic modeling process would in fact be a trade-off between the running time and the accuracy or quality of the topic model. Since the recommended number of iterations required to obtain a good topic model is between 1,500 to 2,000, we use 2,000 iterations.

Allocation of Tweets to Topics

The output of the topic modeling is in fact a soft clustering, as each tweet can be viewed as a probability distribution over the set of topics. However, since α is small, each tweet has a higher probability of being in only one or two topics. Given the limited length of each tweet and the number of content words remaining in each tweet after preprocessing, we make a reasonable assumption that each tweet can in fact only belong to a single topic. Therefore, we allocate each tweet to the topic that has the maximum probability, resulting in a hard clustering.

Ranking Tweets

The aforementioned topic modeling technique enables us to assign to each tweet a topic that has a maximum probability. Each topic generated by the topic modeling step may contain many tweets and it is highly unlikely that all these tweets are relevant to the most salient topic. Intuitively, it is more palatable to users if they can view a set of *most relevant* tweets for a given topic. In this section, we present a technique to generate a ranked list of tweets.

Each tweet is assigned a *topic score* that is based on the following scores. Each tweet is converted to a feature vector using the tf-idf weighting scheme to enable comparison between different tweets using cosine similarity as the distance measure. The first score, known as the *coherence score* (denoted $S_c(t)$), is derived by finding the centroid of the topic, and calculating the cosine similarity between the tweet t's feature vector and the topic centroid. Since the centroid is derived from the set of tweets related to a topic, having a higher cosine similarity with the centroid would imply that the tweet t is more coherent to the given topic.

The next two scores are calculated based on the list of words within each topic, sorted by their word frequencies. For a topic with V unique words, the *word rank score* $S_{wr}(t)$ and *word frequency score* $S_{wf}(t)$ of a tweet t are calculated as follows:

$$S_{wr}(t) = \sum_{k=1}^{K} \frac{1}{K}(V - WordRank(w_k)) \quad (1)$$

$$S_{wf}(t) = \sum_{k=1}^{K} \frac{1}{K}\left(M_{wf} - WordFreq(w_k)\right) \quad (2)$$

where K is the number of words in t and M_{wf} is the frequency of the most frequent word in the topic. The functions *WordRank* and *WordFreq* are based on word frequencies. The former finds the rank of a specific word w_k (that is, the rank position of the word in the list of words ordered by their frequency within the particular topic) whereas the latter returns the frequency count of w_k.

The last two scores, *hashtag score* (S_{ht}) and *popularity score* (S_{pop}), are derived using the metadata associated with each tweet and are computed as follows:

$$S_{ht}(t) = \sum_{h=1}^{H} \frac{hashTagFreq(W_h)}{H_f} \quad (3)$$

$$S_{pop}(t) = 0.5\frac{retweetCount(t)}{R_c} + 0.5\frac{favoriteCount(t)}{F_c} \quad (4)$$

where H is the number of hashtags in a tweet t, H_f is the total number of times hashtags have been used in the topic, R_c and F_c are the total number of times a tweet has been retweeted and selected as favorite, respectively, in the topic. The function $hashTagFreq(W_h)$ returns the number of times that a given hashtag in the target tweet was used in tweets corresponding to a given topic. Functions *retweetCount* and *favoriteCount* return the numbers of times t has been retweeted and marked favorite, respectively.

The *topic score* $S(t)$ for each tweet is computed as follows:

$$S(t) = w_c S_c(t) + w_r S_{wr}(t) + w_f S_{wf}(t) + w_h S_{ht}(t) + w_p S_{pop}(t) \quad (5)$$

Note that the coherence score is given the largest weight, as a high-ranked tweet should be most coherent to a specific topic. Since most tweets do not contain hashtags (Hong, Convertino, & Chi, 2011), lower weight is assigned to hashtag score. Similarly, as far as retweets and favorites are concerned, most recent tweets may be at a disadvantage due to the lack of exposure time even when they are representative of the most salient topic. Hence, the popularity score is also given a relatively lower weight. We retain only the top-k tweets (k is set to 30 in our framework) based on $S(t)$ for each topic. In Performance Study, we shall investigate how to determine "good" weights for each component of $S(t)$.

At the same time, the topic quality of each topic is also derived by calculating the average cosine similarity of each tweet associated with a topic with the topic centroid. This is based on the intuition that a highly coherent, and therefore high-quality topic, will have tweets that have relatively higher cosine similarity with the topic centroid. The

Algorithm 1: TWEETGRAPHCONSTRUCTOR()

```
   Input: Collection of tweets T
   Output: Tweet graph G = (V, E)
1  Initialize w_d, i
2  for t ∈ T do
3      for adjacent words (w_1, w_2) ∈ t do
4          if w_1 ∉ V then
5              add w_1 to V
6          if w_2 ∉ V then
7              add w_2 to V
8          if edge(w_1, w_2) ∉ E then
9              addEdge(w_1, w_2, w_d)
10         else
11             incrementEdge(w_1, w_2, i)
12 return G
```

topics produced by the topic model are then ranked in descending order of the topic quality.

Topics Label Generation

Although each topic can now be represented by a set of the most relevant tweets, the most salient topic in these tweets may not be immediately apparent. Therefore, in this section we present a technique to generate a meaningful topic label of a set of relevant tweets by extending the *TextRank* algorithm (Mihalcea & Tarau, 2004). *TextRank* is a graph-based algorithm for finding the most important nodes (that is, word, sentence) in a graph by taking into account the global information in it.

Our algorithm for topics label generation consists of three key phases, namely, the *tweet graph construction* phase, the *top tweet words identification* phase, and the *label extraction* phase. We discuss them in turn.

The Tweet Graph Construction Phase

We first construct a tweet graph. We treat the entire collection of tweets for each topic as a single document, whereby each tweet is essentially a sentence in the document, and we construct an undirected, weighted tweet graph where nodes represent words and word co-occurrences are reflected through the weighted edges. Algorithm 1 outlines the procedure for tweet graph construction. It involves looking at the adjacent words in a given tweet and adding a new edge with a default weight w_d (for instance, 0.75) if it is not already connected in the tweet graph using the *addEdge* procedure (Lines 3–9). If an edge already exists for a pair of adjacent words, the weight of the existing edge is incremented by i (for instance, 0.10) using the *incrementEdge* procedure (Line 11).

Figure 3 depicts some sample tweets together with the corresponding tweet graph. Observe that edges between words that tend to co-occur together have larger weights. Specifically, word pairs (north, korea), (korea, rocket), (rocket, launch) tend to co-occur frequently in the sample tweets, and this is reflected by their edge weights.

The Top Tweet Words Identification Phase

Algorithm 2 outlines the steps for this phase. The value, denoted by $WS(v_i)$, of each node in a tweet graph, is iteratively updated (until it converges) based on the current values of its neighbors and weights of the edges connecting them, as well as based on the total weight of the edges incident to each of these neighboring nodes (Lines 5–7). The algorithm converges when the differences in the node values between consecutive iterations are less than the predefined threshold δ (Lines 9–10). These nodes are then sorted in descending order of their final values, and by applying a graph reduction factor R, only the top $m\%$ (m = 10 in our setting) of the nodes are retained (Lines 13–14). These nodes are used as *seeds* to generate candidate topic labels in the next phase.

The Label Extraction Phase

Algorithm 3 outlines the last phase. In this phase, we find neighbors of the seeds in the tweet graph that have a score not less than the initial score of 1, since these nodes can be considered as important in the graph (Line 9). By taking the permutations of the words contained in these nodes, and using word frequencies obtained based on the collection of tweets,[4] we can then find the best permutation of the given words by invoking the *bestWordSeq* procedure (Lines 11, 18, 25). This is quantified by multiplying the individual scores together with the edge weights connecting those nodes, and then normalizing it by the number of nodes (that is, words) as encapsulated by the *calculateScore* procedure (Lines 12, 19, 26). This ensures that every word sequence will take into account the likelihood of these words appearing together by including the edge weights, and the normalization process ensures that longer but not necessarily more important word sequences will not be preferred over shorter ones. Consequently, this step generates the best topic label for a given topic, and the best topic can simply be a single word, or a phrase that can be

[4] We compute the frequencies of bigrams (2-g), 3-g, and 4-g in the collection of tweets related to a topic.

Note: Words with strikethrough applied are removed during the pre-processing phase.

Sample tweets
Tweet 1: "~~just~~ ~~in~~ north korea rocket launch failed reports yonhap"
Tweet 2: "destabilising ~~and~~ provocative ~~us~~ condemns north korea rocket launch"
Tweet 3: "north korea rocket launch ~~may~~ spur ~~us~~ missle defence buildup ~~in~~ asia"
Tweet 4: "north korea rocket launch ~~an~~ outright ~~and~~ grave violation ~~eu~~"
Tweet 5: "north korea launches rocket south korea military"

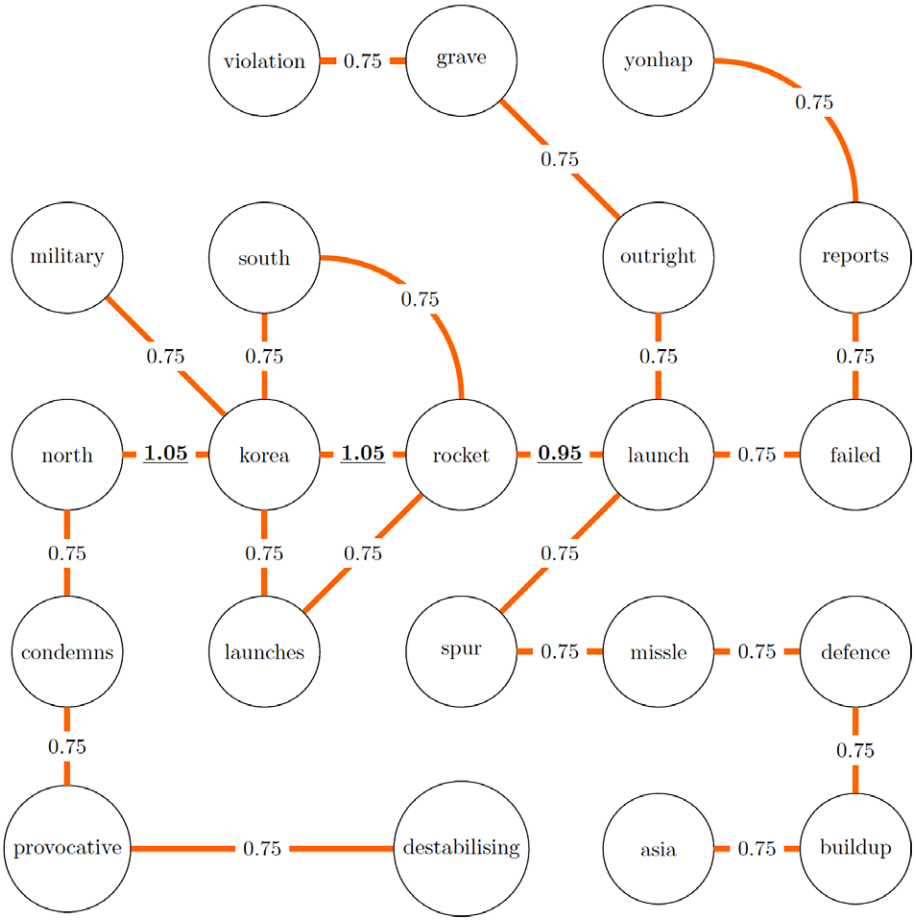

FIG. 3. Sample tweets and the corresponding tweet graph. [Color figure can be viewed at wileyonlinelibrary.com]

found in the collection of tweets. Specifically, the *getBestTopicLabel*(*M*) procedure in Line 29 selects the top-ranked label out of *M* sorted candidate labels to represent a given topic by a single word or phrase. Note that this step preserves the original word order as much as possible.

Figure 4 shows a tweet graph with corresponding word scores for each node. Based on the word scores and the edge weights, the best topic label generated using the aforementioned algorithm is "north korea rocket launch." Observe that since our approach preserves word order, labels such as "rocket korea north launch" will not be generated.

Importance of the Topic Label Generation Algorithm

Lastly, we justify why the aforementioned label generation technique is superior to simple alternative approaches to address this issue, such as extraction of top-*k* words from topic word distribution and top-*k* frequent words. Table 1 shows a comparison of the top-10 words derived from the topic word distribution, 10 most frequent words in the collection of tweets, and the generated topic label for a subset of the topics. It can be observed that the generated topic label is more meaningful and better at capturing the most salient topic. This is because the algorithm for deriving the topic label is based on the observation that words do not appear together in a random fashion. Instead, there are certain syntactic or semantic guidelines that result in various co-occurrences of words.

Topic Summary Generation

The final task is to generate a *topic summary* for each of the topics. A topic summary is essentially a small set of

Algorithm 2: TopTweetWordIdentifier()

Input: Tweet graph $G = (V, E)$
Output: The set of nodes V' with highest scores in G

$V' \leftarrow V$
$d = 0.85$ /* Damping factor */
$R = 0.1$ /* Graph reduction factor */
$\delta = 0.0001$ /* convergence threshold */
for $iter = 1$ *to* $|V|$ **do**
  for $v_i \in V'$ **do**
    $WS(v_i) = (1 - d) + d \times \sum_{v_j \in I(v_i)} \frac{EdgeWeight_{ji}}{\sum_{v_k \in O(v_j)} EdgeWeight_{jk}} WS(v_j)$
  /* Check for convergence */
  if $error(V', V) < \delta$ **then**
    break
  else
    $V \leftarrow V'$
$sort(V')$ /* Sort V' in descending order of scores */
$reduce(V', R)$ /* Graph reduction by retaining only vertices with highest scores */
return V'

the most representative tweets. Together with the topic label, it aims to provide a user with a good representation of the most salient topics and tweets. In this section, we describe the technique for generating such a summary.

Naïvely, a topic summary can be obtained by simply taking the top-*k* ranked tweets within each topic. However, such a simplistic approach does not ensure diversity of the tweets selected for the summary and would fail to provide

Algorithm 3: LabelExtraction()

Input: The set of nodes V' with highest scores in G
Output: Best topic label ℓ

/* M is the set of candidate labels L and their corresponding scores S */
Map $M = (L, S)$
/* $W(v_i)$ returns the word represented by v_i */
/* $WS(v_i)$ returns the word score computed using Algorithm 2 */
for $v_i \in V'$ **do**
  $addToMap(M, W(v_i), WS(V_i))$
  /* 2-words sequence */
  for $v_j \in Out(v_i)$ **do**
    if $W(v_i) == W(v_j)$ *or* $WS(v_j) < 1$ **then**
      continue
    $L \leftarrow bestWordSeq(v_i, v_j)$ /* Derive permutations of words */
    $S \leftarrow calculateScore(v_i, v_j)$
    $addToMap(M, L, S)$
    /* 3-words sequence */
    for $v_k \in Out(v_j)$ **do**
      if $W(v_k) == W(v_j)$ *or* $W(v_k) == W(v_i)$ *or* $WS(v_k) < 1$ **then**
        continue
      $L \leftarrow bestWordSeq(v_i, v_j, v_k)$
      $S \leftarrow calculateScore(v_i, v_j, v_k)$
      $addToMap(M, L, S)$
    /* 4-words sequence */
    for $v_m \in Out(v_k)$ **do**
      if $W(v_m) == W(v_k)$ *or* $W(v_m) == W(v_j)$ *or* $W(v_m) == W(v_i)$ *or* $WS(v_m) < 1$ **then**
        continue
      $L \leftarrow bestWordSeq(v_i, v_j, v_k, v_m)$
      $S \leftarrow calculateScore(v_i, v_j, v_k, v_m)$
      $addToMap(M, L, S)$
$sort(M)$ /* Sort M in descending order of label scores */
$\ell \leftarrow getBestTopicLabel(M)$ /* Get the label in M with the highest score */
return ℓ

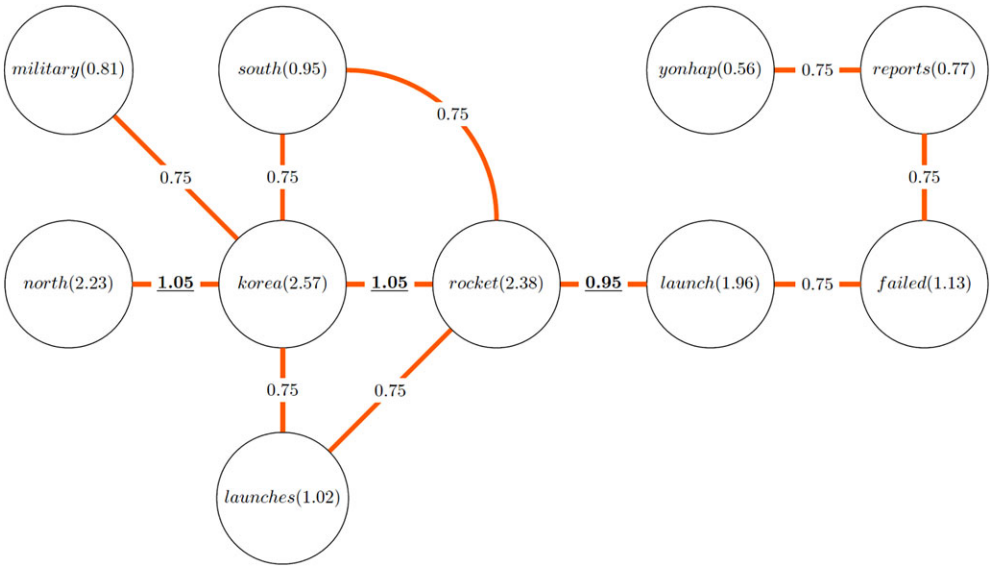

FIG. 4. Topic label extraction. [Color figure can be viewed at wileyonlinelibrary.com]

a good overview of the topic. Therefore, we exploit the notion of *Maximal Marginal Relevance* (MMR) (Carbonell & Goldstein, 1998) to select a coherent and diverse set of tweets to form the topic summary. The MMR of a tweet can be computed as follows.

$$MMR = \underset{T_i \in R-S}{\arg\max}[\lambda \times CosSim(T_i, TC) - (1-\lambda) \times \max_{T_j \in S} CosSim(T_i, T_j)] \quad (6)$$

where TC is the topic centroid, R is the set of most relevant tweets, S is a subset of tweets in R already selected, and $0 \le \lambda \le 1$. T_i represents a tweet from $R - S$ for which the MMR score is to be computed, while T_j is a tweet in S. The MMR metric enables us to compute a linear combination of relevance and novelty. The constant λ can be used to prioritize either relevance or novelty. When $\lambda = 1$, a standard relevance-ranked list can be computed incrementally. On the other hand, when it is 0, a maximal diversity ranking among the documents can be computed. The relevance of the tweets is measured based on their cosine similarity with the topic centroid. In order to select a highly relevant and sufficiently diverse subset of tweets as the topic summary, we set $\lambda = 0.75$. This ensures that the tweets selected are relevant to the most salient topic, while maintaining sufficient diversity in the topic summary.

Algorithm

Algorithm 4 shows the generation of a set of N tweets to be used as the topic summary, using the MMR metric in the above equation. The most relevant tweet in the collection is first added to the set of selected tweets, S (Line 2). Each subsequent tweet is chosen from R and added to the set S, by maximizing both the similarity to the topic centroid and the dissimilarity between the tweet and those that are already selected. This process is repeated until there are N tweets selected as the topic summary (Lines 3–4).

Figure 5 illustrates examples of topic summaries, each made up of five tweets selected using the MMR metric, for a subset of topics. Consider the summary of Topic 2. It is apparent that the topic summary is related to North Korea rocket launch. The first two tweets provide information about the rocket launch itself, with the second tweet mentioning the "success" of the rocket launch as reported by the state media. The remaining tweets in the summary

TABLE 1. Comparison of top words and generated topic labels for a subset of the topics.

Topic 2	Top 10 (Topic Model)	korea, rocket, north, launch, japan, media, parliament, myanmar, missile, space
	Top 10 (Word Freq)	korea, rocket, north, launch, missile, japan, yonhap, space, reports, south
	Generated Topic Label	**north korea rocket launch**
Topic 4	Top 10 (Topic Model)	tcnfl, future, wins, category, startup, sports-themed, stadium, pitch-off, home, football
	Top 10 (Word Freq)	tcnfl, future, category, wins, sports-themed, pitch-off, startup, stadium, live, 1st
	Generated Topic Label	**wins future stadium category**
Topic 6	Top 10 (Topic Model)	zika, virus, brazil, football, mystery, win, waves, doctors, medical, title
	Top 10 (Word Freq)	zika, football, virus, win, brazil, australia, mystery, reached, billion, months
	Generated Topic Label	**brazil #zika virus**
Topic 8	Top 10 (Topic Model)	taiwan, new york, quake, good, people, turkey, president, found, survivors, press
	Top 10 (Word Freq)	taiwan, new york, good, quake, survivors, people, found, turkey, trapped, rubble
	Generated Topic Label	**taiwan quake survivors found**

Algorithm 4: TopicSummaryGenerator()

Input: Set of most relevant tweets R, Number of tweets to be selected for summary N
Output: Selected subset of tweets S

```
λ = 0.75
S ← addToSet(S, getFirst(R))
while |S| < |N| do
    S ← addToSet(S, MMR(S, R, λ))
return S
```

provide the views of other countries with regard to this event, with responses such as "outright and grave violation" and "deeply deplorable."

Performance Study

In this section we investigate the performance of the TOTEM framework. To the best of our knowledge, we are not aware of any existing technique that focuses on summarizing recent personal tweets of a user on a mobile device.

Experimental Setup

We used an Android mobile device for all our experiments. Unless specified otherwise, we set w_c = 0.6, w_r = 0.15, w_f = 0.25, w_h = 0.015, and w_p = 0.025. We set the number of topics T = 8. Table 2 shows the statistics of the five data sets that are used for the evaluation of the proposed solution. These data sets are obtained by retrieving the most recent tweets at different dates using our Twitter account (recall the recent tweets retrieval procedure discussed in the Preprocessing Personal Tweets section). It takes less than 90 seconds to generate a summary in our setup.

User Study

We first conduct a user study to evaluate the quality of the topic summaries using *Data set A*. Twenty unpaid participants (ages 21–27) took part in the user study[5] in accordance with HCI research that recommends at least 10 users (Faulkner, 2003; Lazar, Feng, & Hochheiser, 2010). They all have at least an undergraduate degree in a variety of disciplines. Eighteen (90%) of them are Twitter users and follow 10 to 200 users. The majority of the participants do not check Twitter very frequently. Consequently, it is relatively hard for them to keep track of recent tweets and a summary of their recent tweets will be useful. Note that we use only one data set for user study as each participant evaluates 120 tweets (see below) and our experience suggests that having to evaluate many more strongly deters majority of them from participating in such study.

Methodology. For each topic, participants are required to evaluate the suggested topic labels, rate the first 20 tweets in the collection, as well as evaluate the topic summary. Participants are informed that topic labels should cover most of the topics. As there are a total of eight topics produced by the topic model, we ask each participant to evaluate four out of the eight topics for two key reasons: (i) requiring the participants to evaluate all eight topics may cause considerable cognitive overhead, leading to low-quality judgments; and (ii) some of the topics are about events that all participants may not be familiar with. Consequently, each topic is evaluated by 10 participants. Note that prior to the user study, we demonstrated the use of TOTEM to the participants, and they are allowed to explore it on their own to familiarize themselves with the system.[6]

Participants are then presented with a set of three suggested topic labels for each topic (Figure 6). The first and second suggested labels are the sets of the top-10 words based on the topic word distribution and word frequency within the collection, respectively. The last suggested label is generated based on our technique presented in the Topics Label Generation section. For each suggested topic label, participants are required to provide a score from 0 to 3, with 0 indicating that the suggested label is not a good label for representing the most salient topic, and 3 indicating that it is in fact a very good label.

Next, the participants are asked to rate the first 20 tweets in a topic.[7] Note that the participants are unaware that the tweets are presented in ranked order by TOTEM. For each tweet, a participant is asked to evaluate it based on two factors: (i) relevance to the most salient topic in the collection of tweets and (ii) the quality of the tweet itself. A high-quality tweet should contain meaningful content and provide new information to the user. S/he is allowed to visit a tweet multiple times and update its ranking based on his/her knowledge of the entire list. An example of tweet evaluation is shown in Figure 7.

Finally, the participants are asked to rate the topic summaries based on three criteria, namely, the coverage of the collection of tweets, diversity of the selected tweets in the summary, and the ability to capture the most salient topic present in that collection of tweets. An example is shown in Figure 8.

[5] None of the authors are participants in this study.

[6] A video of TOTEM is available at https://www.youtube.com/watch?v=esENCgAowGI&feature=share&app=desktop.

[7] The alternative strategy is to ask a user to rate all tweets in a topic. However, given the large number of tweets, such an approach strongly deters a user to participate in the study.

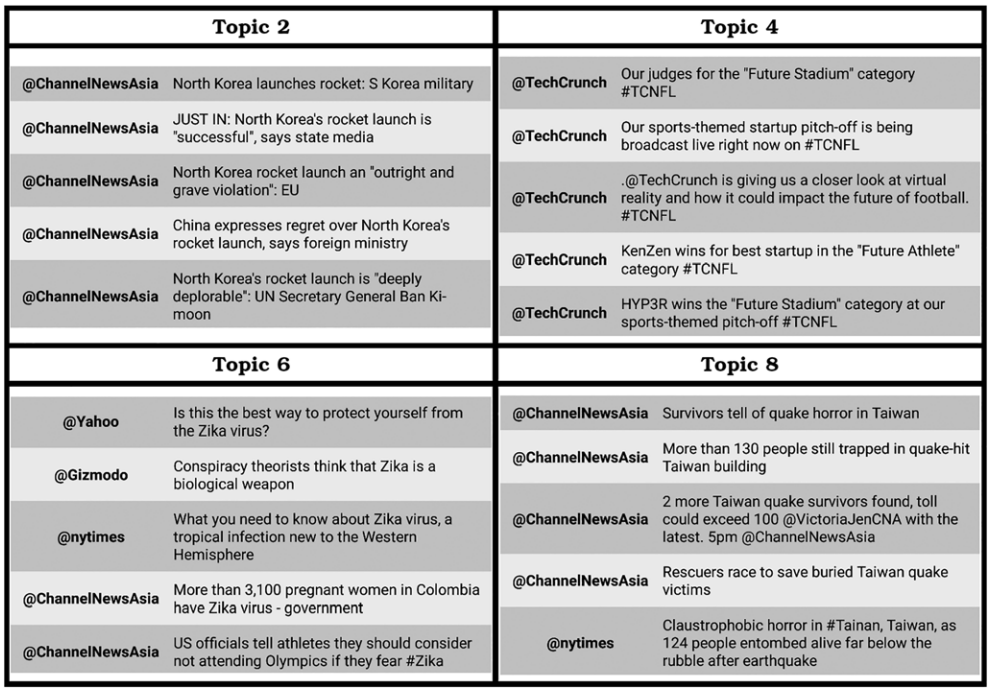

FIG. 5. Generated topic summaries for a subset of the topics.

The user study was carried out in a research lab over a period of 2 weeks. The participants were allowed to take breaks if they felt fatigued. All participants completed the tasks assigned to them.

Results. We use three measures, *Average Rating* (AR), *Average Rating@k* (AR@k) (Manning, Raghavan, & Schütze, 2008), and *Normalized Discounted Cumulative Gain* (NDCG@k) (Järvelin & Kekäläinen, 2000) to evaluate our user study. Since each topic is rated by 10 participants, the AR of a topic is computed as the average of the ratings given by these 10 participants. NDCG@k is defined below:

$$NDCG@k = \frac{\sum_{i=1}^{k} \frac{score_i}{log_2(i+1)}}{\sum_{i=1}^{|SC|} \frac{score_i}{log_2(i+1)}} \quad (7)$$

where, k is particular rank position, $score_i$ denotes the rating score of a tweet at position i, and $|SC|$ represents the list of documents (tweets) ordered by their scores up to the position k, hence the denominator represents the case of ideal ranking. As an NDCG@k score of 1 does not actually reflect the absolute ratings of the tweets, the AR@k is used to complement the NDCG@k score. It is computed as the average rating of the first k tweets in the ranked list (for instance, AR@5 computes the average rating of the top-5 tweets).

First, we present the findings for the evaluation of suggested topic labels. For each topic, we compare the generated topic label with the baselines using the top-10 words from the topic word distribution and the top-10 words based on word frequency. Figure 9 reports the AR scores. Observe that across all topics, the highest-rated topic label is always the one generated using our proposed algorithm. The generated topic labels consistently outperform the baselines by a significant margin. The poorest-performing topic label generated by our proposed algorithm is the one for Topic 4, which is "wins future stadium category." As remarked earlier, our approach of generating topic labels is extractive in nature, which assumes that the "best" topic label can be found in the collection of tweets. Hence, it may produce suboptimal labels when this assumption does not hold (for instance, Topic 4).

Next, we report the findings for ranking of the tweets. Table 3 presents the NDCG@k and AR@k scores. It can be observed that the order in which the tweets are shown to the participants is in fact very close to the ideal ranking. Furthermore, the AR@k scores of the tweets indicate that the top few tweets shown to the participants have relatively good quality.

Lastly, we evaluate the topic summaries based on three criteria, namely, the coverage of the collection of tweets, diversity of the selected tweets in the summary, and the ability to capture the most salient topic present in that collection of tweets. The AR scores shown in Table 4 indicate that the tweets chosen to form the topic summary allow a

TABLE 2. Data sets.

Id	Before preprocessing	After preprocessing	% of preprocessed tweets
A	800	687	85.88
B	800	700	87.5
C	800	710	88.75
D	800	685	86.63
E	800	718	89.75

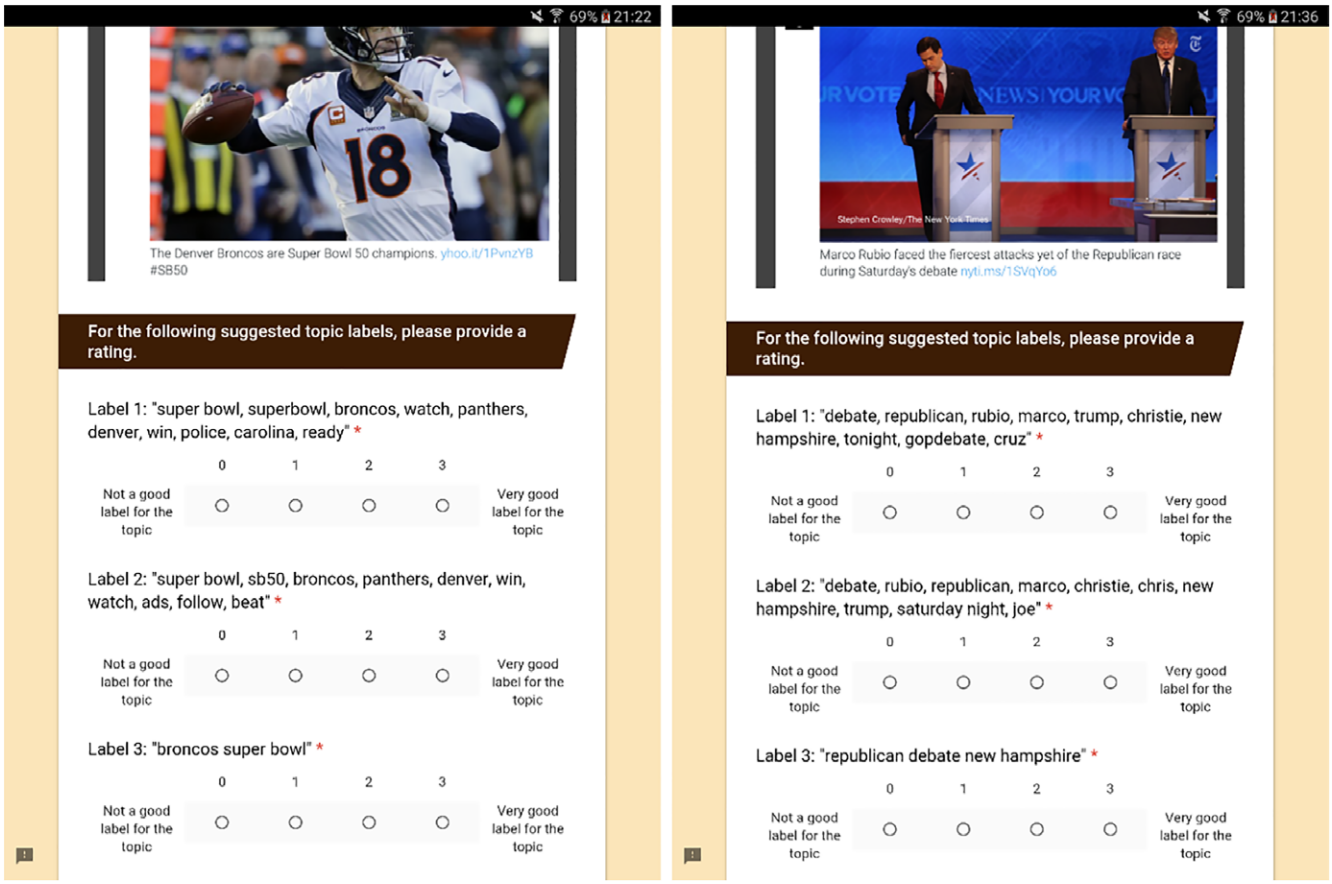

FIG. 6. Evaluation of topic labels for Topics 1 and 5. [Color figure can be viewed at wileyonlinelibrary.com]

user to understand the most salient topic in the full collection of tweets. At the same time, the summary provides a good coverage of the collection, and there is sufficient diversity in the selected tweets.

Automated Evaluation of Topic Summaries

Other than evaluating the topic summaries through a user study, it is also possible to evaluate the generated summaries using a quantitative approach even when a corresponding set of gold standard reference summaries are not available (Louis & Nenkova, 2013; Mackie, McCreadie, Macdonald, & Ounis, 2014). We now evaluate the generated summaries using the Jensen-Shannon Divergence (JSD) (Schütze & Manning, 1999) and the Fraction of Topic Words (FoTW)[8] measures. The JSD is a measure of two probability distributions over words and can be used to compare the text contained in the summary and those in the original documents (that is, tweets). A low divergence indicates that an effective summary is produced (Lin, 1991; Mackie et al., 2014). For example, Mackie et al. (2014) report that JSD scores of good microblog summarization techniques are between 0.21–0.33. On the other hand, to compute FoTW we first determine the topic words (or topic signatures) by applying the log-likelihood test, and then it can be calculated as the number of topic words in a summary divided by the number of topic words in the input documents/tweets. A larger FoTW value is better, as effective summaries contain more topic words from the original documents. Observe that this is a complementary evaluation strategy that allows automated measurement. It is not supposed to simulate a user study.

The results are reported in Table 5. Specifically, observe that the generated summaries for *Data set A* are of good quality based on both JSD and FoTW measures. This is consistent with the findings from the user study. The average JSD is 0.290 and we note that the JSD of Topics 6, 7, and 8 are much larger than the rest. This is likely due to the significant amount of irrelevant tweets contained in these topics.

The FoTW measure shows that the topic summaries contain a substantial amount of the topic words. Although it may seem that the FoTW values are relatively low, given that the ideal value is 1, it should be noted that the topic summaries consist of only five out of the original 30 tweets. That is, we are only using 16.7% of the tweets in the collection as the topic summary for each topic. Therefore, the average FoTW value of 0.44 (44%) shows that a significant amount of topic words from the collection of tweets are in fact contained in these summaries when taking into account the actual number of tweets being utilized as topic summaries.

[8] We use the SIMetrix toolkit (Louis & Nenkova, 2013; SIMetrix, 2012).

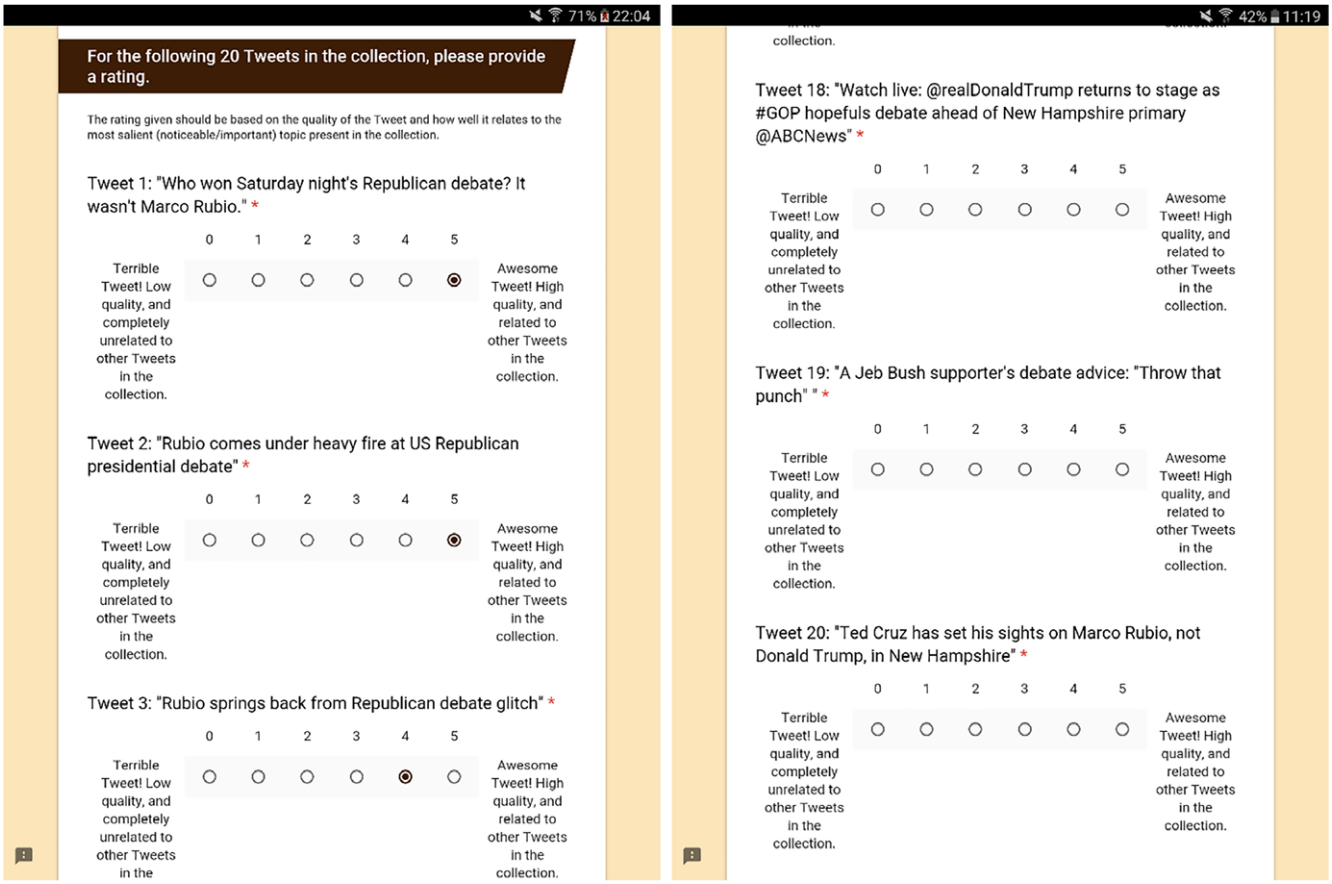

FIG. 7. Evaluation of the tweets in Topic 5. [Color figure can be viewed at wileyonlinelibrary.com]

We also performed the automatic evaluation of topic summaries for the remaining data sets. The overall JSD for these data sets are not much worse than that of *Data set A*. This indicates that the generated summaries are good representations of the original tweets based on the probability distributions over words. Furthermore, the topic summaries also contain a substantial amount of topic words found in the collection of tweets.

Weight Combinations for Topic Score

In this section we investigate the impact of different combinations of weights in Equation 5 on the quality of topic summaries by utilizing the JSD measure.

For each data set, we rank the tweets within a topic using different combinations of weights. We experiment with the following combinations of weights for the five scores: {0.0, 0.25, 0.5, 0.75, 1.0}. Observe that there are a total of 5^5 (that is, 3,125) combinations of these weights. For each combination of weights, we produce the ranked list of tweets by sorting them in descending order of their topic score. For each topic, we retain only the top 30 tweets and produce the topic summary based on the MMR metric. Then we compute the average JSD of the topic summaries to study the impact of the weight combinations.

Tables 6 and 7 show the 10 best linear combination of scores for *Data sets A*, *B*, *C*, and *D* (the results for *Data set E* are qualitatively similar). We can make the following observations. First, the difference between the best and the proposed linear combination is small for all data sets with the exception of *Data set C*. Hence, the aforementioned weight combination can generate good summaries. In fact, there are several weighting schemes that produce relatively good topic summaries (that is, very similar JSD values). In other words, the topic summaries are not extremely sensitive to the weight combinations. Second, it can be observed that assigning a higher weight to the coherence score in most cases tends to produce a summary with the lowest, and hence best, JSD. This is consistent with our intuition stated earlier. Note that only for *Data set C*, the best linear combination involves assigning a higher weight to the word rank and frequency scores. Nevertheless, the difference between the JSD values with our proposed combination is still small for this data set.

Related Work

Tweets summarization techniques focus on selecting a list of meaningful tweets that are most representative for some topic. TweetMotif (O'Connor, Krieger, & Ahn, 2016)

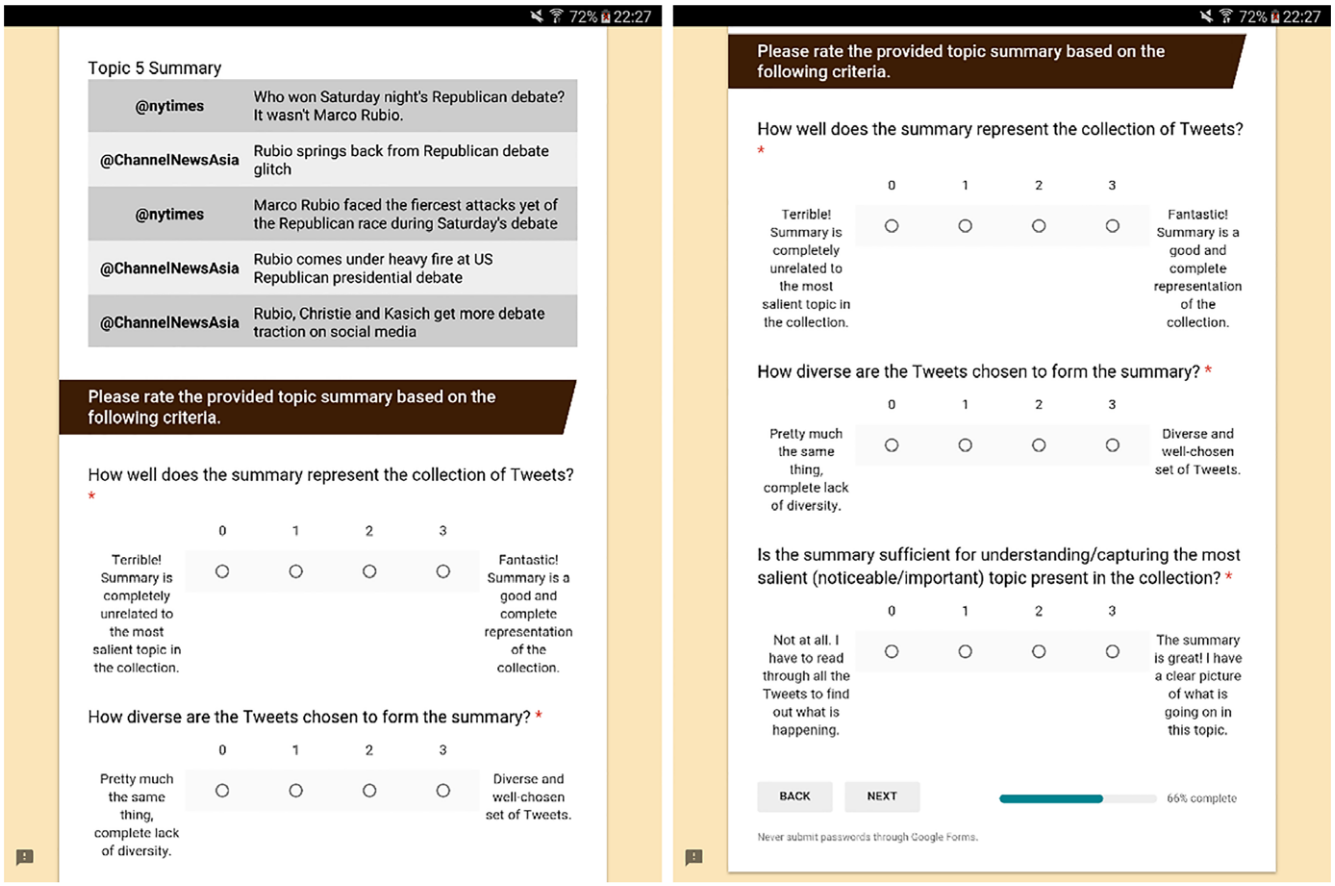

FIG. 8. Evaluation of the topic summaries for Topic 5. [Color figure can be viewed at wileyonlinelibrary.com]

is an exploratory search system for Twitter. A set of topics are extracted in order to group and summarize the messages that form the result set of a given query. Each topic is characterized by a one to three words-long textual label, and a set of messages whose texts contain the textual label. The main difference between TweetMotif and TOTEM is that the former does not focus on summarizing recent personal tweets and the underlying steps for our summarization technique is different from it.

The work in Rosa, Shah, Lin, Gershman, and Frederking (2011) summarized topics and stories being discussed in a cluster of tweets. Specifically, the aim was to find the most representative tweet or the top few tweets to summarize a cluster. These top tweets were selected by ranking the tweets based on their TF-IDF similarity with the cluster centroid, and repeatedly picking the highest-ranked tweet whose TF-IDF similarity with the set of selected tweets is below an empirically determined threshold k. This approach is not designed

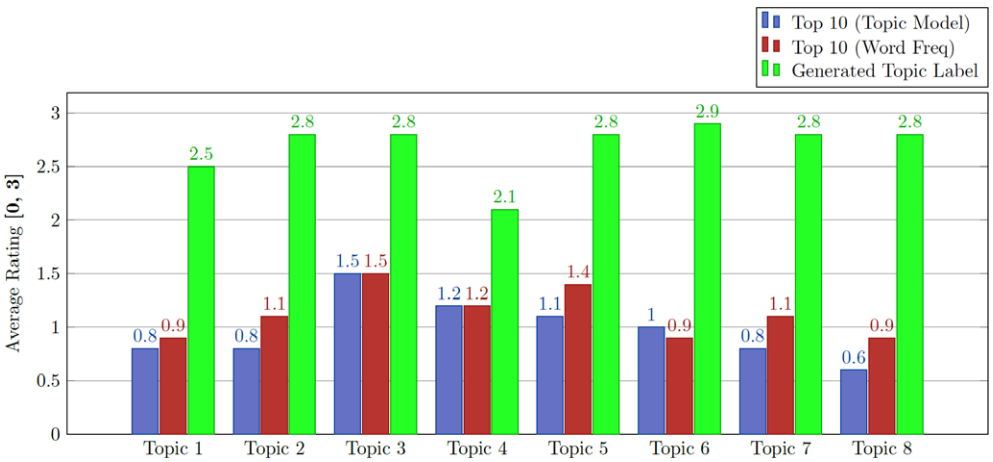

FIG. 9. Average ratings for the topic labels. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3. Evaluation of tweet ranking using NDCG@k and AR@k (that is, using top-k returned tweets).

NDCG@k and AR@k					
Topic	NDCG@3	NDCG@5	NDCG@10	AR@5	AR@10
Topic 1	0.961	0.966	0.953	4.420	4.000
Topic 2	0.898	0.916	0.935	4.240	4.230
Topic 3	0.940	0.926	0.941	3.980	3.820
Topic 4	0.891	0.874	0.926	3.700	3.820
Topic 5	0.979	0.961	0.937	4.280	3.940
Topic 6	0.990	0.991	0.991	4.740	4.650
Topic 7	0.985	0.988	0.987	4.440	4.280
Topic 8	0.980	0.985	0.936	4.820	4.380

Note: [0,1] for NDCG@k, [0,5] for AR@k.

TABLE 4. Evaluation of topic summaries.

Average ratings for different aspects of the topic summaries			
Topic	Coverage of collection	Diversity of tweets	Capturing the most salient topic
Topic 1	2.00	2.40	2.40
Topic 2	2.90	2.50	2.70
Topic 3	2.00	2.40	2.40
Topic 4	2.40	2.30	2.30
Topic 5	2.50	2.30	2.60
Topic 6	2.60	2.70	2.70
Topic 7	2.60	2.60	2.70
Topic 8	2.70	2.80	2.90

for personal tweets on mobile devices. Furthermore, it does not generate a label for each topic or summary and the ranking does not consider diversity of tweets within a topic.

Recently, time-aware summarization in the context of tweets has been studied by several authors (Chakrabarti & Punera, 2011; Ren, Liang, Meij, & de Rijke, 2013; Yan et al., 2011). Our approach is orthogonal to these efforts, as all these techniques focus on evolutionary or temporal aspects and do not generate summary from recent personal tweets on a user's timeline. Furthermore, unlike TOTEM, these approaches generate a summary by considering large volumes of tweets or social relations between users.

More recently, a continuous summarization framework called *Sumblr* was proposed by Wang, Shou, Chen, Chen, and Mehrotra (2015) to summarize large-scale evolutionary tweet streams by clustering them. In contrast, in TOTEM we not only focus on personal tweets but our framework is based on on-demand summarization instead of continuous summarization. As remarked earlier, in a mobile environment a user visits Twitter intermittently, with varying frequency, and may not intend to view the summary in each visit. A continuous summarization strategy may not only be interruptive to a user's interaction with her/his mobile device, but may also lead to wastage of computing resources for continuously computing or fetching summaries even when a user is not interested in it.

In an earlier work, Yang, Ghoting, Ruan, and Parthasarathy (2012) proposed a continuous summarization framework for tweet streams where the tweets are divided into

TABLE 5. Automated evaluation of generated topic summaries.

Topic	Data set A		Data set B		Data set C		Data set D		Data set E	
	JSD	FoTW	JSD	FotW	JSD	FoTW	JSD	FoTW	JSD	FoTW
Topic 1	0.251	0.364	0.219	0.5	0.34	0.312	0.322	0.285	0.315	0.320
Topic 2	0.251	0.4	0.304	0.571	0.213	0.533	0.305	0.304	0.317	0.400
Topic 3	0.286	0.6	0.328	0.250	0.331	0.333	0.265	0.529	0.327	0.333
Topic 4	0.178	0.692	0.336	0.4	0.265	0.64	0.270	0.5	0.328	0.208
Topic 5	0.250	0.5	0.296	0.8	0.313	0.416	0.344	0.363	0.227	0.705
Topic 6	0.387	0.4	0.326	0.571	0.272	0.533	0.326	0.388	0.340	0.384
Topic 7	0.317	0.389	0.397	0.250	0.291	0.636	0.283	0.733	0.298	0.5
Topic 8	0.338	0.583	0.283	0.688	0.318	0.533	0.305	0.333	0.336	0.75

TABLE 6. Top 10 best linear combinations of weights. The last row shows the result of our proposed weight combinations.

Data set A						Data set B					
w_c	w_r	w_f	w_h	w_p	Avg. JSD	w_c	w_r	w_f	w_h	w_p	Avg. JSD
0.75	0.25	0.25	0.0	0.75	0.272	0.75	0.0	0.0	0.75	0.0	0.3
0.75	0.25	0.25	0.0	1.0	0.272	0.75	0.0	0.0	0.75	0.25	0.3
0.75	0.25	0.5	0.0	0.75	0.272	1.0	0.0	0.0	0.75	0.0	0.3
0.75	0.25	0.5	0.0	1.0	0.272	1.0	0.0	0.0	1.0	0.0	0.3
0.75	0.25	0.5	1.0	1.0	0.272	1.0	0.0	0.0	1.0	0.25	0.3
0.75	0.25	0.75	0.0	1.0	0.272	1.0	0.0	0.0	1.0	0.5	0.3
0.75	0.25	0.75	1.0	1.0	0.272	0.75	0.0	0.0	0.75	0.5	0.302
0.75	0.5	0.25	0.0	1.0	0.272	0.75	0.0	0.0	1.0	0.0	0.302
0.75	0.5	1.0	1.0	0.25	0.272	1.0	0.0	0.0	0.75	0.25	0.302
0.75	0.75	0.75	0.75	0.25	0.272	1.0	0.0	0.0	0.75	0.5	0.302
(0.6, 0.15, 0.25, 0.015, 0.025)					**0.283**	**(0.6, 0.15, 0.25, 0.015, 0.025)**					**0.312**

TABLE 7. Top 10 best linear combinations of weights. The last row shows the result of our proposed weight combinations.

Data set C						Data set D					
w_c	w_r	w_f	w_h	w_p	Avg. JSD	w_c	w_r	w_f	w_h	w_p	Avg. JSD
0.75	1.0	0.75	0.25	0.75	0.267	0.75	0.0	0.0	0.0	1.0	0.297
0.75	1.0	0.75	0.25	1.0	0.267	0.75	0.25	0.0	0.25	0.75	0.297
0.75	1.0	0.75	0.5	0.75	0.267	0.75	0.25	0.0	0.25	1.0	0.297
0.75	1.0	0.75	0.25	0.0	0.268	0.75	0.5	0.0	0.75	1.0	0.297
0.75	1.0	0.75	0.25	0.5	0.268	1.0	0.25	0.0	0.25	0.75	0.297
0.75	1.0	0.75	0.5	0.0	0.268	1.0	0.25	0.0	0.25	1.0	0.297
0.75	1.0	0.75	0.5	0.25	0.268	1.0	0.5	0.0	0.5	1.0	0.297
0.75	1.0	1.0	0.25	0.0	0.268	1.0	0.5	0.0	0.75	1.0	0.297
0.75	1.0	1.0	0.25	0.75	0.268	0.75	0.0	0.0	0.0	0.75	0.298
0.75	1.0	1.0	0.25	1.0	0.268	0.75	0.0	0.0	0.25	1.0	0.298
(0.6, 0.15, 0.25, 0.015, 0.025)					**0.293**	**(0.6, 0.15, 0.25, 0.015, 0.025)**					**0.303**

approximately equal-sized batches (for instance, 1 hour per batch) and compress each batch of messages into a summary object which can fit in a constant memory budget. Although the framework is efficient to generate summaries, it does not consider removal of near-duplicates, topic labeling, or ranking tweets in a summary based on novelty and diversity. Furthermore, in contrast to TOTEM, it focuses on a continuous generation of summaries.

Liu, Li, Wei, and Zhou (2012) proposed a graph-based summarization system that aggregates various social signals such as retweeted times and follower numbers to summarize tweets. Furthermore, it considered readability of tweets and diversity of sources to generate summaries. In contrast to TOTEM, it does not preprocess tweets to clean the content and remove near-duplicates. This may lead to a poorer quality of summary. Furthermore, it neither generates a label for each topic nor rank the tweets.

The technique proposed by Rakesh, Reddy, Singh, and Ramachandran (2013) generates a summary of tweets that are specific to a location. Specifically, it leverages on the tweets content and the network information of users to identify location-specific tweets. Similar to our approach, an LDA-based topic model was used that exploits local news database and tweet-based URLs to predict the topics from the location-specific tweets.

Conclusion and Future Work

Motivated by the limitations of the reverse chronological timeline currently deployed in Twitter, as well as many other social networking services, we present an alternative on-demand personal tweets summarization-based approach called TOTEM. Our topic modeling-based approach is designed for mobile devices where users intermittently invoke Twitter at different frequencies and where computing resources are scarce. TOTEM leverages LDA to generate topics associated with preprocessed recent tweets, ranks these tweets, and generates the topic labels as well as the topic summaries associated with them. Evaluation of the generated topic summaries suggest reasonably high consistency and effectiveness of summaries based on results obtained in our experimental data sets. As part of future work, we intend to explore more efficient techniques to improve its runtime performance so that summaries can be generated within a few seconds. Furthermore, it is important to explore how the summaries can be incrementally updated instead of generating them from scratch. Although the latter is imperative for many use case scenarios, incremental update is particularly useful for mobile users who are addicted to Twitter (that is, they check their timeline with very high frequency).

References

Blei, D.M., Ng, A.Y., & Jordan, M.I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, 3, 993–1022.

Carbonell, J. & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In SIGIR (pp. 335–336). Melbourne, Australia: ACM Press.

Chakrabarti, D. & Punera, K. (2011). Event summarization using tweets. In ICWSM 2011 (pp. 66–73). Barcelona, Spain: AAAI Press.

Chin, J.Y., Bhowmick, S.S., & Jatowt, A. (2017). TOTEM: Personal tweets summarization on mobile devices. In SIGIR (pp. 1305–1308). Tokyo, Japan: ACM Press.

Faulkner, L. (2003). Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. Behavior Research Methods, Instruments, & Computers, 35(3), 379–383.

Hong, L., Convertino, G., & Chi, E. (2011). Language matters in twitter: A large scale study. In ICWSM (pp. 518–521). Barcelona, Spain: ACM Press.

Järvelin, K. & Kekäläinen, J. (2000). IR evaluation methods for retrieving highly relevant documents. In SIGIR (pp. 41–48), Athens, Greece: ACM Press.

Kwak, H., Lee, C., Park, H. & Sue, M. (2010). What is Twitter, a social network or a news media? In Proceedings of the International Conference on World Wide Web (pp. 591–600). Raleigh, NC.

Lazar, J., Feng, J.H., & Hochheiser, H. (2010). Research methods in human-computer interaction. Hoboken, NJ: John Wiley & Sons.

Lin, J. (1991). Divergence measures based on the Shannon entropy. IEEE Transaction on Information Theory, 37(1), 145–151.

Liu, X., Li, Y., Wei, F., & Zhou, M. (2012). Graph-based multi-tweet summarization using social signals. In COLING (pp. 1699–1714). Mumbai, India: ACL Anthology.

Louis, A., & Nenkova, A. (2013). Automatically assessing machine summary content without a gold standard. Computational Linguistics, 39(2), 267–300.

Mackie, S., McCreadie, R., Macdonald, C., & Ounis, I. (2014). Comparing algorithms for microblog summarization. Lecture Notes in Computer Science, 8586, 153–159.

Manning, C.D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge, UK: Cambridge University Press.

Mihalcea, R. & Tarau, P. (2004). TextRank: Bringing order into texts. In EMNLP (pp. 1–8). Barcelona, Spain: ACL Anthology.

Norvig, P. (2016). Natural language corpus data. In T. Segaran & J. Hammerbacher (Eds.), Beautiful data. O'Reilly Media, Sebastopol, CA. Retrieved from http://norvig.com/ngrams/count_2w.txt.

O'Connor, B., Krieger, M., & Ahn, D.. (2010). TweetMotif: Exploratory search and topic summarization for twitter. In ICWSM (pp. 384–385). Washington DC: AAAI Press.

Pennacchiotti, M., Silvestri, F., Vahabi, H., & Venturini, R. (2012). Making your interests follow you on Twitter. In CIKM (pp. 165–174). Maui, HI: ACM Press.

Radev, D., Jing, H., & Budzikowska, M. (2000). Centroid-based summarization of multiple documents. In NAACL-ANLP Workshop on Automatic Summarization (pp. 21–30). Seattle, WA: ACL Anthology.

Rakesh, V., Reddy, C.K., Singh, D., & Ramachandran, M.S. (2013). Location-specific tweet detection and topic summarization in Twitter. In ASONAM (pp. 1441–1444) Nigara, Ontario, Canada: ACM Press.

Ren, Z., Liang, S., Meij, E., & de Rijke, M. (2013). Personalized time-aware tweets summarization. In SIGIR.

Rosa, K.D., Shah, R., Lin, B., Gershman, A., & Frederking, R. (2011). Topical clustering of tweets. In ACM SIGIR: Social Web Search and Mining Workshop (SWSM) (pp. 1–8). Beijing, China: ACM Press.

Schütze, H., & Manning, C. (1999). Foundations of statistical natural language processing. Cambridge, MA: MIT Press.

SIMetrix. (2016). Retrieved from http://homepages.inf.ed.ac.uk/alouis/IEval2.html

Slang. 2016. Slang Dictionary—Text Slang & Internet Slang Words. All-Slang Network. Retrieved from http://www.noslang.com/dictionary/

Wang, Z., Shou, L., Chen, K., Chen, G., & Mehrotra, S. (2015). On summarization and timeline generation for evolutionary tweet streams. IEEE Transactions on Knowledge and Data Engineering, 27(5), 1301–1315.

Yan, R., Wan, X., Otterbacher, J., Kong, L., Li, X., & Zhang, Y. (2011). Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In SIGIR (pp. 745–754). Beijing, China: ACM Press.

Yang, X., Ghoting, A., Ruan, Y., & Parthasarathy, S. (2012). A framework for summarizing and analyzing twitter feeds. In KDD (pp. 370–378). Beijing, China: ACM Press.