TOTEM: Personal Tweets Summarization on Mobile Devices

Jin Yao Chin
Nanyang Technological University
Singapore
s160005@e.ntu.edu.sg

Sourav S Bhowmick
Nanyang Technological University
Singapore
assourav@ntu.edu.sg

Adam Jatowt
Kyoto University
Japan
adam@dl.kuis.kyoto-u.ac.jp

ABSTRACT

Tweets summarization aims to find a group of representative tweets for a specific 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* (*i.e.,* 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 demonstration we present a lightweight, personalized, on-demand, topic modeling-based tweets summarization engine called TOTEM, designed for such devices. Specifically, TOTEM summarizes most recent tweets on a user's timeline and enables her to visualize and navigate representative topics and associated tweets in a user-friendly tap-and-swipe manner.

1 INTRODUCTION

Similar to several online social networking platforms such as *Facebook* and *Instagram*, *Twitter* has also adopted a reverse chronological timeline to display tweets. Users are only able to scroll through posts on her timeline one by one, beginning with the most recent post, and it can often be daunting to catch up with the most recent contents due to high volume and velocity of tweets. Specifically, the reverse chronological timeline is inadequate due to two main reasons: (a) the most recent posts may be repeating the same information as users tend to retweet the tweets which are of interest to them, and (b) it can be challenging to comprehend overall summary of the topics being discussed in a user's most recent posts.

To alleviate the aforementioned issues, *Twitter* introduced a new concept called ~~*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 comprehend the overall summary of the topics being discussed in her recent posts.

In this demonstration, we present a novel recent *tweets summarization* (*i.e.,* a group of representative tweets for a topic) framework called TOTEM (**T**opic M**O**deling-based Recen**T** Tw**E**et Su**M**marizer) that enables a user to obtain on demand an overview of the most salient topics present in the recent tweets on her timeline. Furthermore, it assists a user to easily identify topics that she is most interested in and zoom into a specific topic and representative tweets. Since it is estimated that 80% of active *Twitter* users access it on mobile platforms[1], TOTEM is designed specifically for mobile devices. Consequently, it is *lightweight* in design in order to tackle limited memory, limited processing power, limited network connectivity, and small screen size of mobile devices. A user can simply visualize the summary by using a tap-and-swipe approach.

SIGIR '17, August 7–11, 2017, Shinjuku, Tokyo, Japan
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5022-8/17/08…$15.00.
http://dx.doi.org/10.1145/3077136.3084138

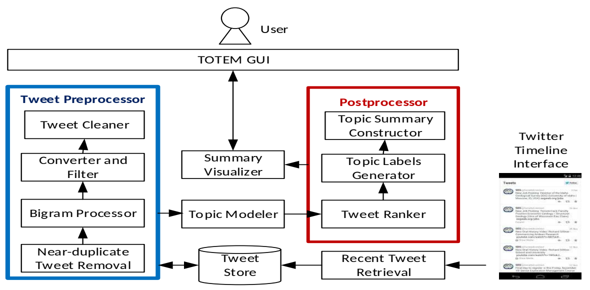


Figure 1: Architecture of TOTEM.

2 SYSTEM ARCHITECTURE

Figure 1 shows the system architecture of TOTEM and mainly consists of the following modules.

Recent Tweet Retrieval Module. This module is responsible for retrieving recent tweets from a user's account by utilizing the functionalities provided by the *Twitter* REST API. To this end, it utilizes the *Fabric Software Development Kit* (Fabric SDK) created by *Twitter* for the mobile platform. Note that the API only allows the retrieval of 800 most recent tweets. These tweets are stored locally on the user's mobile device in a *SQLite* database.

Tweet Preprocessor Module. The aim of this module 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 module is to strike a balance between the amount of preprocessing that needs to be performed against the time it takes to perform them. Naturally, if preprocessing takes too long to execute then TOTEM will not only consume a lot of resources of the mobile device but also fail to provide real-time summary.

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 posed 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, it uses the following submodules to perform a variety of preprocessing tasks.

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

Tweet Cleaning Submodule. This submodule is responsible for "cleaning" the content of retrieved tweets. Many tweets may either contain user mentions or URLs from the linked web pages or embedded media elements that cannot be exploited by popular topic modelling techniques. Hence, they are removed. Furthermore, as tweets may arrive from different sources, some of them may contain special characters such as words with accents or emoticons. This submodule removes the accents appropriately (*e.g.,* "café" to "cafe") to ensure that words are correctly interpreted by the topic model. Additionally, it removes all non-printable ASCII characters. Finally, the cleaned text is tokenized.

Note that this submodule does not undertake spelling correction and hashtag segmentation as part of the preprocessing step. Although the presence of hashtags (which are often multi-word expressions) in tweets may mislead the topic model, both spelling correction and hashtag segmentation are expensive to perform especially in a mobile device.

Converter and Filter Submodule. The 140 characters constraint on *Twitter* has led users to come up with different strategies to get messages across to other users in a most succinct way. This includes the use of acronyms or abbreviations. This submodule converts such acronyms and abbreviations to its original form (*e.g.,* "govt" to "government", "SOA" to "service oriented architecture") by leveraging on a conversion dictionary [2]. It also eliminates common stop words.

Bigram Processor Submodule. Next, it identifies phrases that have better semantic meaning when treated as a single entity. Some examples include "new year", "rocket launch", and "surface water". This also helps to eliminate ambiguity associated with words that are quite common but are not stop words. For example, the word "water" can be used in many different scenarios and might cause semantically unrelated tweets to be put together under a same topic due to the lack of information to resolve ambiguity. This submodule identifies phrases that tend to co-occur together by leveraging a list of over 50,000 of the most common bigrams [1] and join them using an underscore character (*e.g.,* "surface water" to "surface_water").

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

Near-duplicate Tweet Removal Submodule. Often tweets on a user's timeline may contain identical or nearly identical textual content but different URLs (*i.e.,* near-duplicates). Hence, the goal of this submodule is to remove such near-duplicate tweets. To this end, it leverages on the *Cross-Sentence Informational Subsumption* (CSIS) technique [9][2], which measures the informational content of sentences. CSIS reflects the occurrence of sentences which repeat some of the information contained in other sentences, and hence can be omitted without any loss of information. Therefore, if the information content of a tweet is contained in some other tweet, then it is redundant and can be removed. Note that it is also possible for two tweets to subsume each other, which means that they are equivalent and can be substituted with the other without any loss of information. Hence, one of them can be removed. Specifically, this enables TOTEM to remove the following categories of tweets: (a) those which only differ in their usage of stop words or URLs; and (b) tweets that subsume some other tweets. We experimentally observe that after performing the aforementioned pre-processing steps, around 87% tweets are typically retained in a dataset.

[2]Note that the choice of near-duplicate tweet detection technique is orthogonal to our framework. Any superior technique can be used in TOTEM.

Topic Modeler Module. The goal of this module is to identify a set of topics associated with the preprocessed tweets and label each tweet with the most relevant topic. To this end, we utilize the Latent Dirichlet Allocation (LDA) model [3]. Particularly, this module extends the MALLET toolkit in order to use it within an Android application. Since the Android framework does not support the full set of Java packages, various code segments in the MALLET toolkit which relied on the unsupported Java packages are refactored to make it Android compatible.

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 pre-processing. 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 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 datasets, β is set to 0.01.

LDA model requires the number of topics (denoted as N) to be modelled. A small N value would cause the topic model to identify very broad topics, while a large N value might result in very detailed and fine-grained topics. We experimentally investigated the impact of different values of N on the tweet summarization quality for different datasets and decided to use 8 as the default value for N. Note that a user can change this value as desired.

Lastly, the number of iterations used for the topic modelling 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 1500 to 2000, this module uses 2000 iterations.

Note that given the limited length of each preprocessed tweet and small α value, this module allocates each tweet to the topic which has the maximum probability, resulting in a hard clustering.

Post-Processor Module. Intuitively, it is more palatable to users if they can view a set of *most relevant* tweets for a given topic. The goal of this module is to generate a ranked list of tweets, a meaningful topic label, and a topic summary (*i.e.,* group of representative tweets) for each topic produced by the topic modeling module. It consists of the following submodules.

Tweet Ranker Submodule. Each tweet is assigned a *topic score* that is based on the following scores. Each tweet is converted to a feature vector using 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 as $S_c(t)$), 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:

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

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

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 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 numbers of times hashtags have been used in the topic, R_c and F_c are total number of times a tweet has been retweeted and selected as favorite, respectively, in the topic.

The *topic score* $S(t)$ for each tweet is computed as follows[3]:

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

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 [5], lower weight is assigned to hashtag score. Similarly, as far as retweets and favourites 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. TOTEM retains only the top-k tweets (k is 30 in our system) based on $S(t)$ for each topic.

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 which have relatively higher cosine similarity with the topic centroid. The topics produced by the topic model are then ranked in descending order of the topic quality.

Topic Labels Generator Submodule. Although each topic can now be represented by a set of most relevant tweets, the most salient topic in these tweets may not be immediately apparent. Therefore, this submodule aims to generate a meaningful topic label of a set of relevant tweets by extending the graph-based *TextRank* algorithm [7]. To generate topic labels for each topic, it first computes the frequencies of the bigrams (2-grams), 3-grams, and 4-grams in the collection of tweets related to a topic. These *n*-grams shall be used later to find best permutations of a set of words. It treats the entire collection of tweets for each topic as a single document, whereby each tweet is essentially a sentence in the document, and constructs an undirected, weighted graph where nodes represent words and word co-occurrences are emphasized through the weighted edges. Next the value of each node is iteratively updated (until it converges) based on the current values of its neighbours and weights of the edges connecting them, as well as the total weight of the outgoing edges of each of these neighbouring nodes. These nodes are then sorted in descending order of their final values, and by applying a graph reduction factor of 0.1, only the top 10% of the nodes are retained. These nodes are used as *seeds* to generate candidate topic labels. Specifically, it finds neighbours of the seeds that have a score not less than the initial score of 1 as these nodes can be considered as important in the graph. By taking the permutations of the words contained in these nodes, and using word frequencies obtained based on the collection of tweets, it can then find the best permutation of the given words. 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 (words). 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. Figure 2 shows an example of a list of topic labels generated by TOTEM.

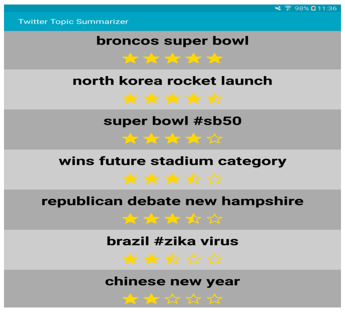
Topic Summary Constructor Submodule. The final task is to generate a *topic summary* for each of the topics. A topic summary is essentially a small set of the most representative tweets. Together with the topic label, it aims to provide a user with a good representation of the most salient topics and tweets.

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. Therefore, this submodule exploits the notion of *Maximal Marginal Relevance* (MMR) [4] concept to select a coherent and diverse set of tweets to form the topic summary. The MMR metric enables us to compute a linear combination of relevance and novelty. The relevance of the tweets are measured based on its cosine similarity with the topic centroid. In summary, this metric ensures that the tweets selected are relevant to the most salient topic, while maintaining sufficient diversity in the topic summary.

Summary Visualizer Module. This module enables a user to view the retrieved recent tweets and generate a topic summary of these tweets on the visual interface of a mobile device. Note that the tweets summarization is performed on demand as requested by a user instead of continuous summarization of tweets on a user's timeline. This design choice is made due to the fact that a user may not always intend to view a summary. Hence, continuous summarization will lead to unnecessary consumption of resources.

Once the collection of recent tweets in a user's personal timeline is retrieved, she can generate the summary by tapping on the `Generate Topics` button on the GUI. Each topic is shown as a

[3]We set w_c, w_r, w_f, w_h, and w_p to 0.6, 0.15, 0.25, 0.015, and 0.025, respectively, as our experimental study demonstrates that this combination gives good results on a variety of datasets.



broncos super bowl ★★★★★
north korea rocket launch ★★★★⯪
super bowl #sb50 ★★★★☆
wins future stadium category ★★★⯪☆
republican debate new hampshire ★★★⯪☆
brazil #zika virus ★★⯪☆☆
chinese new year ★★☆☆☆

Figure 2: Overview of topics.

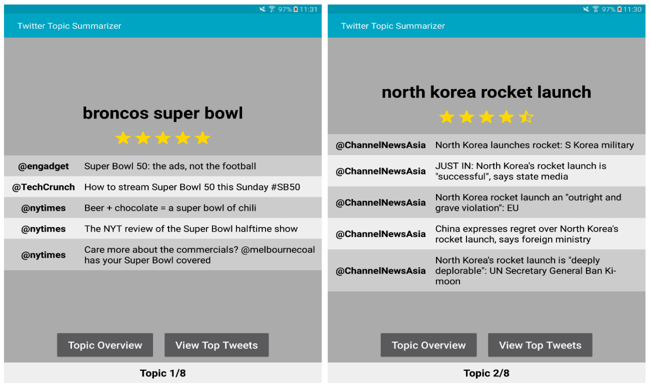
single page along with its label, topic quality, and summary. Figure 3 depicts the visualization of two topics. The topic quality is represented by the number of stars (0 to 5). Users can double-tap on a particular entry within the topic summary to retrieve and display the corresponding tweet. Users can swipe to the left or right to go to the next or previous topic, respectively. Tapping on the `Topics Overview` button returns a page containing an overview of all topics as shown in Figure 2. By default, this module displays 8 topics for the given collection of most recent tweets. However, a user can easily change this number through the interface. A user can navigate to any topic easily by tapping on the topic itself (*e.g.,* tapping on `north korea rocket launch` will take the user to the page shown in Figure 3). Tapping on the `View Top Tweets` button (Figure 3) allows a user to view the ranked list of tweets related to the given topic (Figure 4).

3 RELATED SYSTEMS AND NOVELTY

Recently, several tweets summarization techniques have been proposed in the literature [6, 8, 10, 11]. To the best of our knowledge, these existing techniques either do not focus only on personal tweets (*i.e.,* tweets that are in the timeline of a specific user) while extracting representative tweets for a topic or they are not designed for mobile devices. Techniques described in [6, 10, 11] demand significant computing resources and time and hence are not suitable for real-time summarization on a mobile device. TOTEM can finish computing summary of personal tweets in 2 minutes. On the other hand, techniques in [6, 8, 11] do not focus only on the recent tweets on a user's timeline. Additionally, none of these techniques have been demonstrated as an interactive, working system in a major information retrieval or data mining venue. In summary, *TOTEM is the first personal tweets summarization system on a mobile device to be demonstrated in a major research venue.*

4 DEMONSTRATION OBJECTIVES

TOTEM can be used with any personal *Twitter* account. In our demonstration we shall be using an Android mobile device loaded with 800 most recent tweets from the timeline of our default account. Users can also use their personal accounts through our mobile device to experience TOTEM. A video of TOTEM is available at https://www.youtube.com/watch?v=esENCgAowGI&feature=share&app=desktop. One of the key objectives of the demonstration is to enable the audience to interactively experience the proposed recent tweets summarization framework in real-time. Through the visual interface, the user will be able to invoke summarization of



broncos super bowl ★★★★★

@engadget	Super Bowl 50: the ads, not the football
@TechCrunch	How to stream Super Bowl 50 this Sunday #SB50
@nytimes	Beer + chocolate = a super bowl of chili
@nytimes	The NYT review of the Super Bowl halftime show
@nytimes	Care more about the commercials? @melbournecoal has your Super Bowl covered

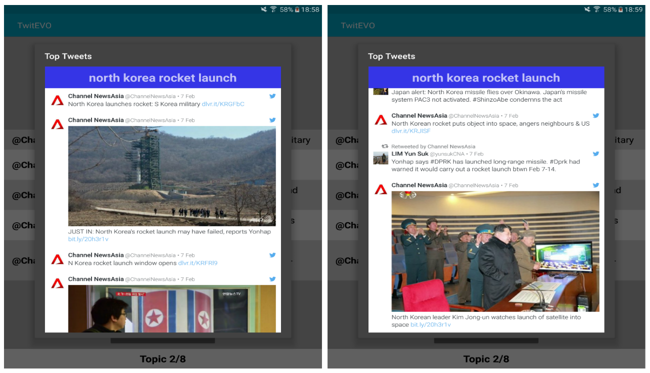
Topic 1/8

north korea rocket launch ★★★★⯪

@ChannelNewsAsia	North Korea launches rocket: S Korea military
@ChannelNewsAsia	JUST IN: North Korea's rocket launch is "successful", says state media
@ChannelNewsAsia	North Korea rocket launch an "outright and grave violation": EU
@ChannelNewsAsia	China expresses regret over North Korea's rocket launch, says foreign ministry
@ChannelNewsAsia	North Korea's rocket launch is "deeply deplorable": UN Secretary General Ban Ki-moon

Topic 2/8

Figure 3: Components of two topics.



north korea rocket launch

Channel NewsAsia @ChannelNewsAsia • 7 Feb
North Korea launches rocket: S Korea military dlvr.it/KRGFbC

Channel NewsAsia @ChannelNewsAsia • 7 Feb
JUST IN: North Korea's rocket launch may have failed, reports Yonhap bit.ly/20h3r1v

Channel NewsAsia @ChannelNewsAsia • 7 Feb
N Korea rocket launch window opens dlvr.it/KRFRl9

Channel NewsAsia @ChannelNewsAsia • 7 Feb

Topic 2/8

north korea rocket launch

Japan alert: North Korea missile flies over Okinawa. Japan's missile system PAC3 not activated. #ShinzoAbe condemns the act

Channel NewsAsia @ChannelNewsAsia • 7 Feb
North Korean rocket puts object into space, angers neighbours & US dlvr.it/KRJISF

Retweeted by Channel NewsAsia
LIM Yun Suk @yunsukCNA • 7 Feb
Yonhap says #DPRK has launched long-range missile. #Dprk had warned it would carry out a rocket launch btwn Feb 7-14.

Channel NewsAsia @ChannelNewsAsia • 7 Feb
North Korean leader Kim Jong-un watches launch of satellite into space bit.ly/20h3r1v

Topic 2/8

Figure 4: Ranked list of tweets for the `north korea rocket launch` topic. The left and right tweets show top-ranked and bottom-ranked tweets, respectively.

recent tweets on her timeline, browse the topics overview and corresponding summary as well as information about their labels and quality (Figures 2 and 3). Going a step further, the user may immediately retrieve and browse ranked list of tweets on a specific topic of interest (Figure 4). Additionally, by setting different values for N (*i.e.,* number of topics), she can view changes to the summaries.

REFERENCES

[1] Natural Language Corpus Data: Beautiful Data. http://norvig.com/ngrams/count_2w.txt, 2016.
[2] Slang Dictionary - Text Slang & Internet Slang Words. http://www.noslang.com/dictionary/, 2016.
[3] D. M. Blei, A. Y. Ng, M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, vol. 3, 2003.
[4] J. Carbonell, J. Goldstein. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. *In SIGIR*, 1998.
[5] L. Hong, G. Convertino, E. Chi. Language Matters in Twitter: A Large Scale Study. *In ICWSM*, 2011.
[6] X. Liu, Y. Li, F. Wei, M. Zhou. Graph-Based Multi-Tweet Summarization using Social Signals. *In COLING*, 2012.
[7] R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts, *In EMNLP*, 2004.
[8] B. O'Connor, M. Krieger, D. Ahn. TweetMotif: Exploratory Search and Topic Summarization for Twitter. *In ICWSM*, 2010.
[9] D. Radev, et al. Centroid-based Summarization of Multiple Documents. *In NAACL-ANLP Workshop on Automatic Summarization*, 2000.
[10] Z. Ren, et al. Personalized Time-aware tweets summarization. *In SIGIR*, 2013.
[11] Z. Wang, L. Shou, et al. On Summarization and Timeline Generation for Evolutionary Tweet Streams. *In IEEE TKDE*, 27(5), 2015.