Twitter Topic Summarization by Ranking Tweets Using Social Influence and Content Quality

DUAN YaJuan[1*] *CHEN ZhuMin*[2] *WEI FuRu*[3]
ZHOU Ming[3] *Heung − Yeung SHUM*[4]
(1) University of Science and Technology of China, No. 96, Jinzhai Road, Hefei, Anhui, P.R.China
(2) Shandong University, No. 27, Shanda South Road, Jinan, Shandong, P.R.China
(3) Microsoft Research Asia, No. 5, Danling Street, Haidian District, Beijing, P.R.China
(4) Microsoft Corporation, Redmond, USA
dyj@mail.ustc.edu.cn, chenzhumin@sdu.edu.cn
{fuwei, mingzhou, hshum}@microsoft.com

ABSTRACT
In this paper, we propose a time-line based framework for topic summarization in Twitter. We summarize topics by sub-topics along time line to fully capture rapid topic evolution in Twitter. Specifically, we rank and select salient and diversified tweets as a summary of each sub-topic. We have observed that ranking tweets is significantly different from ranking sentences in traditional extractive document summarization. We model and formulate the tweet ranking in a unified mutual reinforcement graph, where the social influence of users and the content quality of tweets are taken into consideration simultaneously in a mutually reinforcing manner. Extensive experiments are conducted on 3.9 million tweets. The results show that the proposed approach outperforms previous approaches by 14% improvement on average ROUGE-1. Moreover, we show how the content quality of tweets and the social influence of users effectively improve the performance of measuring the salience of tweets.

TITLE AND ABSTRACT IN ANOTHER LANGUAGE (CHINESE)

基于用户社会影响力和文本质量的推特话题摘要

本文提出了一个基于时间轴的推特话题自动摘要框架。话题按照时间顺序分成子话题，各子话题中根据文本的重要度和多样性对推特文本进行排序和抽取以生成摘要。我们使用相互增强式图模型同时考虑文本内容、作者社会影响力和文本质量进行排序。实验结果表明：1）本文提出的模型在基准模型上ROUGE-1平均提高了14%；2）作者社会影响力和文本质量有效地改进了文本重要度的度量。

KEYWORDS: Topic summarization, Twitter, Timeline summarization, Content quality, Social influence.

KEYWORDS IN CHINESE: 话题摘要，推特，时间轴上的摘要，文本质量，用户社会影响力 .

[*] This work was done when the first author was an intern at Microsoft Research Asia.

Proceedings of COLING 2012: Technical Papers, pages 763–780,
COLING 2012, Mumbai, December 2012.

1 Introduction

Recently, Twitter [1] has become one of the most popular social networking sites. It enables people to freely post short messages (called tweets) up to 140 characters. Twitter has rapidly gained worldwide popularity, with over 140 million active users generating over 340 million tweets daily in March 2012 [2]. The rapid proliferation of Twitter posts presents a big obstacle for efficient information acquisition. It is impossible for a user to get an overview of important topics on Twitter by reading all tweets everyday. In addition, because of information redundancy and the informal writing style, it is time consuming to find useful information about a topic from a huge number of tweets. The tremendous volume of tweets suggests summarization as the key to facilitating the requirements of topic exploration, navigation, and search from hundreds of thousands of tweets. Specifically, a summary that provides representative information of topics with no redundancy and well-written sentences would be preferred.

In this paper, we focus on the problem of topic summarization in Twitter, which aims to provide a short and compact summary for a collection of tweets on the same or similar topics. We take individual tweets as the basic constituents to compose the summary. Here, tweets are to a certain extent analogous to sentences in traditional extractive document summarization, which has been extensively studied in past decades (Ani Nenkova, 2011). However, we argue that summarizing tweets is substantially different from summarizing news documents owing to the following reasons. First, tweets are streamed with detailed time-stamps delivering real-time information of users' continuous updates and comments that, provide temporal information for tracing topic evolution over time. This timeliness feature motivates us to summarize tweets along time line. Second, tweets are commonly expressed in an informal way. Only some of them obey standard grammar requirements, while others are written in arbitrary styles. For instance, tweets may contain many abbreviations, spelling errors, or information fragments. This requires the summarization algorithm to be aware of the content quality of the tweets when ranking and selecting salient tweets as summaries.

Furthermore, tweets are created on and spread through social networks. The authority of the author of the tweets, as well as the social networks (e.g. follower-followee relationship) of the author, usually plays an important role in demonstrating the salience of the tweets. People may be particularly interested in tweets from celebrities or opinion leaders. To be specific, if there are two tweets stating the same information, and one is published by an influential user and the other is not, we assume the former is more important than the latter on account of two interesting observations about Twitter users. First, users with a high influence have a larger audience. Their tweets are apt to be read by more users than those of non-influential users. Second, encouraged by the interactions with their followers, influential users are more likely to publish informative tweets of better readability, less error, and preferable completeness than common users. This guides us to develop a unified framework to simultaneously model the information from the content and the authors of tweets.

We therefore propose modeling and formulating tweet ranking in a unified mutual reinforcement graph, where the social influence (i.e. authority) of users and content quality of tweets are taken into consideration simultaneously in a mutually reinforcing manner. Particularly, we leverage the follower-followee relationship connecting different authors, upon which the social influence of users can be inferred. We define the content quality of tweets, including readability and

[1] http://twitter.com
[2] http://blog.twitter.com/2012/03/twitter-turns-six.html

content richness, as a measure of the regularity of written language and the pointless degree of the content. The above information is jointly employed in a graph-based ranking algorithm. In order to avoid redundancy in the result, the final summary is generated by selecting tweets from the previous ranking results with the traditional Maximal Marginal Relevance(MMR) algorithm (Carbonell and Goldstein, 1998). We conduct experiments on a real data set containing 3.9 million tweets. Compared with two popular graph-based summarization approaches, namely Lexrank (Erkan and Radev, 2004) and phrase graph (Sharifi et al., 2010a), the experimental results show that:

- The reinforcement summarization model integrating social influence and content quality achieves a considerable performance and outperforms the standard LexRank and the phrase graph summarization approaches.
- The social influence of users and the content quality of tweets help to more effectively measure the salience of tweets.

The rest of this paper is organized as follows. Related work is introduced in Section 2. Next, we present a detailed introduction of our approach in Section 3. Section 4 shows the experiments and results, and we conclude this work with directions for future study in section 5.

2 Related work

2.1 Extractive document summarization

A substantial amount of work has been done on extractive text summarization (Lloret and Palomar, 2012). Many text features, such as term frequency, sentence position, query relevance and sentence dependency structure, have been investigated for sentence salience estimation. They are usually weighted automatically by applying certain learning-based mechanisms or tuned experimentally to build a feature-based summarization system (Fuentes et al., 2007) (Wong et al., 2008). Previous research shows that a combination of sentence position, fixed-phrase and sentence length give the best results in learning-based sentence selection (Ani Nenkova, 2011). Meanwhile, feature-based approaches have been widely used in the top five participating systems in DUC[3] 2005-2007.

In addition, different types of links among sentences and documents are employed by graph-based approaches to measure sentence salience, such as LexRank (Erkan and Radev, 2004), TextRank (Mihalcea, 2004), and Mutual Reinforcement Chain(MRC) (Wei et al., 2008). LexRank and TextRank make use of pairwise similarity between sentences, hypothesizing that the sentences similar to most of the other sentences in a cluster are more salient. In contrast to the single level PageRank in LexRank and TextRank, MRC considers both internal and external constraints on three different levels, document, sentence, and term and achieves promising improvement.

2.2 Micro-blog summarization

Recently, researchers have conducted a number of investigations on micro-blog(e.g. Twitter) summarization. Instead of ranking sentences in traditional document summarization, micro-blog posts are ranked to select salient ones for the generation of topic-sensitive and query-sensitive summary. Both feature-based and graph-based approaches are exploited to measure the salience of posts under an extractive summarization framework. Taking into consideration

[3] http://www-nlpir.nist.gov/projects/duc/data.html

the evolutionary characteristic of topics along time line, researchers have also started to explore the evolutionary summarization of events in micro-blog.

In feature-based approaches, a variety of statistical and linguistic features have been extensively investigated, such as, language model (O'Connor et al., 2010), tweet frequency (Shiells et al., 2010), term frequency (Liu et al., 2011) (Takamura et al., 2011) (Parthasarathy, 2012), TF-IDF (Frederking, 2011) (Chakrabarti and Punera, 2011), hybrid TF-IDF (Sharifi et al., 2010b), KL-divergence (Zubiaga et al., 2012), time delay (Takamura et al., 2011), and topic relevance (Long et al., 2011). Among them, simple term frequency has proven to be extremely extraordinary for topic-sensitive micro-blog summarization because of the unstructured and short characteristics of micro-blog posts according to Inouye and Kalita (2011). As for micro-blog summarization, some micro-blog specific features such as text normalization, the content of shared web pages (Liu et al., 2011), and user behavior in conveying relevant content (Harabagiu and Hickl, 2011), have proven useful for result improvement.

The phrase graph algorithm is the most frequently studied graph-based approach in micro-blog summarization. Sharifi et al. (2010a), Beaux Sharifi and Kalita (2010), and Sharifi (2010) propose a phrase reinforcement summarization algorithm on leverage of trending phrase or phrases specified by a user in micro-blog posts. It achieves substantial improvements on ROUGE results by taking advantage of the link structure among words. Nichols et al. (2012) generate journalistic summary for events in world cup games by employing phrase graph algorithm only on the longest sentence in each tweet. Additionally, PageRank-like algorithms such as LexRank and TextRank have also been investigated by Inouye and Kalita (2011).

Evolutionary summarization approach segments post stream into event chains(usually along time line) and produces the final summary by incorporating the summary extracted for each event. A simple and effective method for detecting events from a post stream is to separate the stream according to the bursty period of post volume (Nichols et al., 2012) (Zubiaga et al., 2012). Long et al. (2011) provide an even particular separation according to the topical words that are recognized by their frequency in #hashtags and the entropy in the corpus. Chakrabarti and Punera (2011) argue that different types of events should be detected even when they are temporally close. They utilize a modified Hidden Markov Model to detect the event chain by learning the underlying hidden state representation of repeated events.

3 Topic summarization by time line with mutual reinforcement model

We begin this section with a formal definition of the work presented in this paper. We formulate the problem of topic summarization in Twitter as follows: given a topic τ depicted by a #hashtag [4], we can obtain a tweet collection $T = \{t_1, t_2, \cdots, t_N\}$ containing the #hashtag, where N is the number of tweets in a time span. Each tweet is attached with a time mark. We take a time-line approach to summarize T to fully capture the rapid topic evolution over time. Specifically, the summary for the topic consists of:

- a set of sub-topics $\omega = \{\omega_1, \omega_2, \cdots, \omega_K\}$ based on the distribution of tweets over time, where K is the number of sub-topics;
- at most M salient tweets as the summary for each sub-topic $\omega_i, 1 \leq i \leq K$.

[4] #hashtags are user-annotated topics in tweets.

There are three major steps generating a time-line summary for a topic. First, we conduct a topic segmentation that segments the tweet stream of the topic into sub-topic clusters in terms of the posting time, in which each cluster describes a sub-topic. Second, tweets in each sub-topic cluster are ranked according to tweet salience by a reinforcement ranking model taking advantage of the content quality of tweets and social influence of the authors. Third, we generate the summary for each sub-topic upon the tweet ranking results by removing the redundant tweets at the whole topic level. In the following sub-sections, we will present the three steps respectively.

3.1 Sub-topic segmentation

The key issue of sub-topic segmentation is to detect the breaking point of sub-topics in the tweet stream. We observe that in comparison to the internal portion of a sub-topic, some statistics of the stream make dramatic changes, called bursty, at breaking points, for example, tweet volumes, term frequency, participating users, and variety of #hashtags. Among them, term frequency bursty is most correlated to topic evolution according to our analysis. Therefore, we segment the sub-topics by detecting term frequency bursty. Typically, a sub-topic occurs associated with several words and covers their bursty period. We break sub-topic segmentation down to the bursty period identification of associated words and the lifetime detection for sub-topics.

First, we track every word in the stream to identify their bursty period. Terms without a bursty period will be discarded. We assume that the term frequency satisfies a binomial distribution in the bursty period (Fung et al., 2005).

$$tf(w) = \binom{I}{i} p(w)^i (1-p(w))^{I-i} \tag{1}$$

$$p(w) = \sum_{i=1}^{I} \frac{tf_i(w)}{Count_i(tweet)} \cdot \frac{1}{I}$$

Where I is the total length of time in days in the stream, $p(w)$ represents the bursty probability of word w, and $tf_i(w)$ is the term frequency of w at i^{th} time. $Count_i(tweet)$ denotes the number of tweets at i^{th} time. We compute the mean value of $tf(w)$. The period in which the term frequency is larger than $2 * mean$ is a bursty period of the corresponding word.

The lifetime of sub-topics is detected depending on the bursty period of associated words. Terms whose half bursty period overlap with each other are recognized as a set of sub-topic-associated words, denoted by w_a. The lifetime of the sub-topic covers a continuous period where

$$tf_i(w_a) > \alpha \cdot mean_a + \beta \cdot var_a \tag{2}$$

$$tf_i(w_a) = \sum\nolimits_{w \in w_a} tf_i(w)$$

where $tf_i(w_a)$ is the term frequency of associated words at i^{th} time. $mean_a$ and var_a denote the expectation and variance of term frequency respectively by blending words in associated words set during the overlapped time period. α and β adjust the length of the sub-topic lifetime, which are selected empirically. In the experiment, we set α as 1.4 and β as 0.4. The sub-topic consists of tweets published in the corresponding lifetime. Time periods contain no bursty words and those with an associated term frequency below the threshold are regarded as noisy periods and kept out from the later summary generation.

3.2 Mutual reinforcement model based sub-topic summarization

We define salient tweets as those similar to most of the tweets in the sub-topic, published by influential users, and presented in a good writing style. The first is consistent with the salience in document summarization. We emphasize the second because Twitter is a social network, in which influential users have a wider audience and more affirmative interaction with others. Tweets published by influential users are more likely to dominate the topic. The last is set against the informal writing style of Twitter.

Inspired by Wei et al. (2008), we propose a unified mutual reinforcement summarization model taking advantage of relations among tweets, words, and users for tweet salience measurement. Figure 1 shows the overview of the proposed model. The similarity of tweets to the sub-topic benefits from both the content similarity among tweets and the word coverage in the sub-topic cluster. In document summarization, the contribution of relationships among sentences to the performance improvements has been recognized (Wan et al., 2007). Sharifi et al. (2010a) found that sequences of words that encompassed the topic phrase highly overlapped when considering a large number of tweets for a single topic. The social influence of users contributes to salience measurement via author relation. And the content quality of tweets is incorporated at the tweet level.

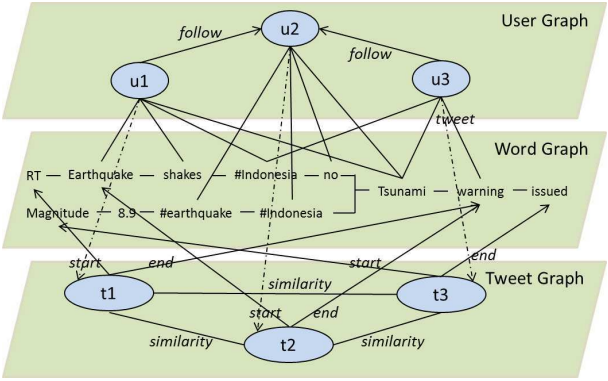

Figure 1: The Unified Mutual Reinforcement Graph Model

The mutual reinforcement model is formed with three PageRank-like models for word, tweet, and user respectively, but in a unified and interrelated way. The ranking of one of them is derived not only from the relationships with instances of itself, but is also affected by the other two. In the model, tweets connect to each other through syntactic similarities. If two tweets have a non-zero cosine similarity, there is a link between them. Words are linked through their co-occurrence in the same tweet. And users are naturally associated by following-followee relationship. If user u_i follows user u_j, a direct edge from u_i to u_j is created. Furthermore, we create edges among users, words, and tweets through authorship and consisting relationship. If user u_i published tweet t_j, we connect u_i with t_j and all words in t_j according to authorship. And t_j is linked to every word it consists of. In Figure 1, we illustrate the connection between

tweet and words through *start* and *end* links to avoid too many lines. *Start* specifies the first word of a tweet and *end* denotes the last word.

3.2.1 Mutual reinforcement based tweet ranking

We can then formulate the ranking algorithm for the mutual reinforcement model.

A tweet is salient if it is published by an influential user, written in regular language style, contains important terms and connects to other salient tweets. The ranking score of a tweet in sub-topic collection ω_k is defined as,

$$\begin{aligned} Score^{(r+1)}(t_i) = \alpha_1 \cdot [(1-d) \cdot \frac{quality(t_i)}{\sum_{t_j \in \omega_k} quality(t_j)} + d \cdot \sum_{t_j \in adj[t_i]} \frac{Sim(t_i, t_j)}{\sum_{t \in adj[t_j]} Sim(t_j, t)} \cdot Score^{(r)}(t_j)] \\ + \beta_1 \cdot \sum_{w \in t_i} Score^{(r)}(w) + \gamma_1 \cdot Score^{(r)}(u_i) \end{aligned} \tag{3}$$

where $Score^{(r)}(t)$, $Score^{(r)}(w)$, and $Score^{(r)}(u)$ denote the ranking score of tweet t, word w and user u in r^{th} iteration respectively. $adj[t_i]$ represents tweets connecting to t_i directly, $Sim(t_i, t_j)$ denotes the cosine similarity between t_i and t_j, u_i is the author of t_i, and $quality(t_i)$ refers to the content quality of t_i which will be detailed in Section 3.2.2. In order to assign high scores for tweets of good quality, we diversify the random walk probabilities (Brin and Page, 1998) of tweets using content quality. High quality tweets will be selected at high probabilities, while low quality ones are selected at low probabilities. Here, d is the damping factor, set to 0.85 as described in Brin and Page (1998).

A user is influential if he publishes salient tweets, uses important terms, and connects to other influential users. The ranking score of a user is defined as,

$$\begin{aligned} Score^{(r+1)}(u_i) = \alpha_2 \cdot \sum_{t \in \omega_k \cap T_{u_i}} Score^{(r)}(t) + \beta_2 \cdot \sum_{w \in t, t \in \omega_k \cap T_{u_i}} Score^{(r)}(w) \\ + \gamma_2 \cdot [(1-d) \cdot \frac{F_{static}(u_i)}{\sum_{u \in U_{\omega_k}} F_{static}(u)} + d \cdot \sum_{u_j \in flw[u_i]} \frac{1}{|frd[u_j]|} \cdot Score^{(r)}(u_j)] \end{aligned} \tag{4}$$

where T_{u_i} denotes the tweets published by u_i, and U_{ω_k} refers to all users who published tweets in ω_k. $flw[u_i]$ represents followers of u_i, and $frd[u_j]$ refers to the users u_j follows. $F_{static}(u_i)$ is regarded as a topic-irrelevant static influence of user u_i. It is predicted by a linear SVM model using features derived from several statistics, including the number of followers, the number of messages, the number of lists, the follower/following ratio, the zombie followers in proportion to all followers, and the number of mentions and retweets the user received. Similar to content quality, we use the static influence of users to differentiate the random walk probabilities of users with the purpose of assigning high scores for users with high static influence.

A phrase term is important if it is contained in salient tweets, used by an influential user, and connects to other important terms.

$$\begin{aligned} Score^{(r+1)}(w_i) = \alpha_3 \cdot \sum_{t \in \omega_k \cap T_{w_i}} Score^{(r)}(t) + \gamma_3 \cdot \sum_{u \in U_{\omega_k} \cap U_{w_i}} Score^{(r)}(u) \\ + \beta_3 \cdot [(1-d) \cdot \frac{df(w_i)}{\sum_{w_j \in t, t \in \omega_k} df(w_j)} + d \cdot \sum_{w_j \in adj[w_i]} \frac{1}{|adj[w_j]|} \cdot Score^{(r)}(w_j)] \end{aligned} \tag{5}$$

where T_{w_i} denotes tweets containing word w_i and U_{w_i} refers to users who used w_i. $df(w_i)$ is the document frequency of term w_i. And $adj[w_i]$ represents the words connnecting to w_i. The random walk probability of terms is initialized by their normalized document frequency with the purpose of assigning high scores for words with high document frequency. α_i, β_i, and γ_i are used to balance the relative weight of tweets, words, and users. They are selected in accordance with Wei et al. (2008)'s work. Wei et al. (2008) proved that the three level unified mutual reinforcement model will converge on unique $Score(t_i)$, $Score(u_i)$, and $Score(w_i)$. We will not repeat the demonstration here. Finally, a ranked tweet list $\omega_k^{'}$ is formed by ranking tweets in ω_k according to the score assigned by the model.

3.2.2 Content quality estimation

We employ a logistic regression model to estimate the content quality used in Section 3.2.1. The content quality of tweets is estimated according to two aspects: readability and content richness. The former measures how easy a tweet is to read. In particular, tweets have high readability if they are written in regular language style using ordinary vocabulary, and have low readability if not. The latter quantifies how much useful information a tweet contains. Instead of calculating the absolute readability and the content richness, we learn models to compare the values of two tweets. Given tweets t_i and t_j, we set their relative quality as

$$quality(t_i, t_j) = \begin{cases} 1, quality(t_i) > quality(t_j) \\ 0, quality(t_i) = quality(t_j) \\ -1, quality(t_i) < quality(t_j) \end{cases} \quad (6)$$

Naturally, for a given group of tweets, if the relative quality of any two tweets has been identified, a ranking list of those tweets with respect to the quality can be generated, and vice visa. Consequently, the relative quality estimation problem is converted into quality ranking. We learn a unified logistic regression model to rank the quality of tweets according to their regression score. The model considers readability and content richness simultaneously. Table 1 and Table 2 present the feature set we used.

Feature	Description
OOV words	The proportion of OOV words in tweet
#hashtags	The proportion of #hashtags in tweet
Mentions	The number of user names appearing in the tweet
Capital letter	Normalized length of capital letters fraction
Punctuation	Normalized length of punctuation character
Emoticon	The number of emoticons
Ellipsis	The number of ellipsis
Stop word	Normalized length of stop word fraction
Character per word	Average character per word
Length	Number of characters in tweet

Table 1: Features for readability estimation

Feature	Description
Celebrity	Normalized word length of celebrities
URL	Share URL or not
URL rank	Alexa rank of shared URL
InfoToNoise	Normalized word length after removing stop word
Word length	The number of words
NE length	Normalized word length of named entities

Table 2: Features for content richness estimation

3.3 Summary generation by removing redundancy

Finally, we will generate a summary for each sub-topic from its corresponding ranked tweets. A straightforward method would be to select the top-ranked tweets of the sub-topics. However, this method would generate a redundant summary both at the tweet and sub-topic levels. At the tweet level, using a similarity link between tweets when measuring the salience of a tweet leads to close ranking scores being assigned to similar tweets. Tweets with high similarity may be chosen simultaneously. At the sub-topic level, the tweet stream is segmented into sub-topics and noisy tweet collections along time line. Two sub-topics describing the same content may be separated by several noisy collections. To avoid redundancy, we employ the MMR algorithm to generate the final summary.

$$S_k = \underset{t_i \in \omega_k' \setminus S_k}{argmax} [\lambda \cdot Sim_1(t_i, w_{a,k}) - (1-\lambda) \cdot \underset{t_j \in S_k \cap S_{k-1}}{max} Sim_2(t_i, t_j)] \quad (7)$$

Where S_k is the summary generated for sub-topic ω_k. Sim_1 represents the cosine similarity between current tweets and the associated terms $w_{a,k}$ of ω_k. And Sim_2 is the cosine similarity between current tweet and tweets selected in S_k and S_{k-1}, based on the observation that a topic changes continuously. λ is the weight emphasizing the importance of the two types of similarities, which is set as 0.5 in the experiment.

4 Experiment

4.1 Data set

We take earthquake event as an example in our experiment. Note that our approach does not leverage topic (e.g. earthquake) specific features. It can generate summary for topics beyond earthquake as well. We obtain 12.7 million English tweets containing the keyword *earthquake* published between September 2010 and April 2012 using the public Twitter API [5]. As #hashtags are regarded as user annotated topics in Twitter, we select the 30 most popular #hashtags related to earthquake as the topics from the collected tweet corpus. Then, we pick out tweets relevant to each topic using two methods of keyword matching. First, tweets containing the corresponding #hashtag are chosen. Second, if the #hashtag is a compound word concatenated of several words, it split into a phrase. Tweets containing the phrase are also selected. In total, we obtain 3.9 million tweets for the 30 topics, which are segmented into 84 sub-topics. Each sub-topic contains 66,416 tweets on average because of the co-occurrence of the #hashtags.

We ask two Ph.D. students(who have not participated in this work) to label 10 salient tweets as

[5] http://windowsphone.uservoice.com

reference for each sub-topic. Given that it is fairly difficult to select 10 tweets from thousands, we filter some tweets beforehand if they satisfy one of the following conditions:

- The length of the word is less than 3.
- It contains no words other than topic words, keyword *earthquake* and stop words.
- It contains no words other than URLs

After filtering, we are left with 26,874 tweets on average for each sub-topic. To make it easy for labeling, the remaining tweets are ranked according to syntactic similarity with the sub-topic. We further remove repetition in the ranking list in case different users post identical tweets. Finally, there are 10,616 tweets for each sub-topic in the ranking list. Annotators select the top tweet in the ranking list, and decide if it should be added to the summary, until 10 tweets have been chosen. We merge their annotation by maintaining the same tweets as the final reference summary. The inter-annotator agreement of annotation is 73%.

In the experiment, we remove spam tweets and extremely short tweets to clear the data. Some users post very similar tweets for certain kinds of promotion. For example, identical content was posted with different URLs attached for a backpack promotion. If a user continuously posts tweets with a content similarity of more than 0.9, all those tweets published by him or her are regarded as spam tweets and removed. We also ignore extremely short tweets less than 3 words in length because they are too short to deliver complete information. In addition, to get a better estimation of similarity between tweets, we conduct some preprocessing before the similarity computation: 1) all words in tweets are stemmed; 2) the topic terms and keyword *earthquake* are neglected due to their occurrence in nearly every tweet; 3) some functional #hashtags and news agencies such as #fb and #BBC are excluded because they are usually irrelevant to the topics and only provide an information source; 4) the stop words are removed, including standard stop words and stop word abbreviations commonly used in Twitter, for instance, *you are* shortens to *ur*.

In the following sections, we present the experimental results on a real-life data set to verify the effectiveness of our summarization approach. We conduct four experiments to evaluate the performance of the reinforcement summarization model, the contribution of social influence and content quality to the model, the correctness of sub-topic segmentation, and the accuracy of the content quality estimation model. The performance of the social influence of users has not been evaluated due to the difficulty of data acquisition. It is hard to label the influence of users, or to collect such information automatically. Therefore, in this paper, we evaluate the end-to-end contribution of social influence under the whole summarization framework.

4.2 Evaluation metric

We evaluate the performance of our summarization system using ROUGE (Lin and Hovy, 2003), which is widely-used in summarization evaluation. It measures the overlap of N-grams between the predicted summary and the reference, which is defined as,

$$ROUGE = \frac{\Sigma_{t \in S_{ref}} \Sigma_{gram_n \in t} Count_{match}(gram_n)}{\Sigma_{t \in S_{ref}} \Sigma_{gram_n \in t} Count(gram_n)}$$

where n is the word length of n-gram, S_{ref} denotes the reference summary, $Count(gram_n)$ is the number of n-grams comprising sentences in the reference summary and $Count_{match}(gram_n)$ computes the maximum number of n-grams appearing both in the summary generated by our system and the reference summary.

4.3 Evaluation of the reinforcement model

In order to evaluate the effectiveness of the reinforcement model, we construct two baselines, namely the phrase graph model (Sharifi et al., 2010a) and the LexRank algorithm (Erkan and Radev, 2004). Both of them are graph-based algorithms, homologous with the reinforcement model. And they are currently the most popular two graph-based algorithms for summarization. Phrase graph measures sentence similarity depending on word frequency as well as the word distance from the topical word in a sentence. LexRank is a PageRank-like summarization algorithm, which calculates sentence salience using the random walk model. Table 3 shows the comparison results of our approach and the baseline methods in terms of ROUGE-1 values. In

Algorithm	ROUGE-1
Phrase graph	0.4286
LexRank	0.3865
Our approach	**0.4617**

Table 3: A comparison of three summarization algorithms[6]

the experiment, α_i, β_i, and γ_i are set as $[\alpha_1, \alpha_2, \alpha_3] = [1, 0.5, 0.25]$, $[\beta_1, \beta_2, \beta_3] = [0.5, 1, 0.5]$, and $[\gamma_1, \gamma_2, \gamma_3] = [0.25, 0.5, 1]$, the same as that in Wei et al. (2008). As seen from Table 3, the proposed approach outperforms both the LexRank and phrase graph methods. It obtains a relative improvement of 20% and 8% compared with the two baselines respectively. We also find that LexRank and phrase graph tend to assign high salience scores to long sentences containing several #hashtags. Some of them are pointless tweets and irrelevant to the topic. For example: *@lightskintess77 =o you could check to see if you had an earthquake by looking it up here: 4 minutes ago #TAB##TAB#5.8 #TAB##TAB#Virginia*. It contains #hashtag *#Virginia*, but the content is not very relevant to the main topic *Virginia earthquake*. It is more like a personal message.

Because LexRank accumulates the information from neighboring sentences (or tweets), its similarity estimation mainly depends on commonly used words, not words with very high IDF scores, especially in short sentences, which increase the chances of selecting pointless information such as personal messages. In comparison to LexRank, phrase graph performs better because it computes the salience of tweets relying on the topical words in the cluster and takes word order into consideration.

Pointless and low quality tweets hurt the performance of both of the baseline models. In comparison to them, our proposed reinforcement model incorporating content quality and social influence emphasizes tweet salience by not only considering the similarity, but also the influence of its author, its readability, and its content richness. These factors penalize the pointless tweets and tweets of low quality when measuring their salience and achieve substantial results.

4.4 Evaluation of social influence and content quality for summarization

We further investigate the effectiveness of the social influence of authors and the content quality of tweets for tweet summarization in detail. To verify the usefulness of content quality, we test

[6]Our approach outperforms LexRank with a significance level of p=0.07. The improvement on Phrase graph is not significant

its contributions in the reinforcement model, LexRank, and phrase graph separately. We remove content quality from the reinforcement model by replacing the random walk probabilities of tweets with a unified weight. The more the performance decreases, the more useful the content quality is. Meanwhile, the content quality is integrated into LexRank and phrase graph. In LexRank, we initialize the random walk probability with the quality score of tweets. And in phrase graph, we change the weight of each word into the weighted summation of its term frequency in every tweet it appears, with the content quality of the tweet as the weight. Table 4 shows the results.

Similar to content quality, we evaluate the contribution of social influence by removing it from the reinforcement model. The salience of tweets is then inferred in the two-level PageRank-like iteration derived from related tweets and words. γ_1, γ_2, γ_3, α_2, and β_2 in section 3.2.1 are set as zero. We set $[\alpha_1,\alpha_3] = [1,0.5]$, and $[\beta_1,\beta_3] = [0.5,1]$ according to Wei et al. (2009). We have not integrated social influence into the LexRank and phrase graph as verification due to the high complexity. Table 5 shows the results.

Model	ROUGE-1
Reinforcement model	0.4617
w/o content quality	**0.4503(-2.5%)**
LexRank	0.3865
LexRank + content quality	**0.3956(+2.4%)**
Phrase graph	**0.4286**
Phrase graph + content quality	0.3959(-7.7%)

Table 4: Effectiveness of content quality

Model	ROUGE-1
Reinforcement model	0.4617
w/o social influence	**0.4310(-6.6%)**

Table 5: Effectiveness of social influence

We can see from Table 4 that the content quality of tweets proves useful in both the reinforcement model and LexRank. The ROUGE-1 is improved by about 2.4% by integrating content quality. We observe in the result that tweets selected by every model pair overlap except for individual pointless tweets. For example, *OMG! This footage on TV now of the earthquake and ensuing tsunami in #Japan is so crazy. My thoughts are with everyone there! Stay strong!*. Content quality excludes such pointless tweets from the final summary by reducing the transition probability. However, the ROUGE-1 in the phrase graph model decreases after it is combined with content quality. It is probably because of the rough assignment of the quality score to words. The content quality of tweet is a measurement based on all words contained in the tweet. It is not so fair to distribute this quality to every word equally.

Table 5 indicates that social influence is an important part of the reinforcement model. The performance decreases rapidly in ROUGE-1 after removing it. The comparison between the results(before and after removing social influence) suggests that more tweets posted by influential users are selected by the former.

4.5 Evaluation of sub-topic segmentation

It is difficult for editors to segment a tweet stream containing hundreds of thousands of tweets into several sub-topics. We have noticed that each sub-topic is a cluster of tweets. Adjacent sub-topics are required to focus on different subjects, which make it a special text clustering task. Thus, we treat each sub-topic as a cluster and automatically compute the average cosine similarity between adjacent ones as an indicator of segmentation performance. The lower the average cosine similarity is, the better performance the segmentation achieves. Table 6 shows the results.

Approach	Cosine similarity
Uniform segmentation	0.835
Nichols et al. (2012)	0.674
Ours	**0.670**

Table 6: Evaluation of sub-topic segmentation

We compare our method with uniform segmentation and the method in Nichols et al. (2012). Uniform segmentation separates the tweet stream into identical length time slots. Tweets in each time slot form a sub-topic. The number of sub-topics equals the number generated by our method. Nichols et al. segment the tweet stream by detecting the extreme changes in update volume per minute. The moment in which the update volume per minute exceeds 3*median(update volume) is a segment. In the experiment, 2.8 sub-topics are produced on average for each topic by our method, and 3 by Nichols et al. (2012). From Table 6 we can see that our method achieves comparable result to Nichols et al. (2012). Both of them lead to lower average cosine similarities between adjacent sub-topics in comparison to uniform segmentation.

4.6 Evaluation of content quality

In this sub-section, we evaluate the performance of the content quality classifiers. We randomly select 3,000 tweets from an additional tweet corpus for content quality training and evaluation. One of two labels, H(high quality) and L(low quality), are assigned to every tweet. Tweets that have good readability and cover rich content are labeled H, otherwise L. A tweet is regarded as having good readability if it does not include too many abbreviations, ellipses, spelling errors, #hashtags, or mentions. In addition, a tweet is viewed as containing rich content if it 1) describes an event or an interesting topic, 2) contains crisp, clear, and effective text that is easy to understand, 3) provides some insight about the event or topic beyond simply stating that it occurred, or 4) reflects a useful opinion on some topics. We learn two logistic regression models using features described in section 3.2.2 and unigrams respectively, denoted as LRO and LRU. Five-fold cross-validation is conducted to evaluate the performance. Table 7 shows the results as well as the result of the Random classification model.

The LRO model obtains promising results for both high and low quality tweets. It achieves substantial improvements in comparison to the LRU model and random classification model. The result of random classification reflects the imbalanced distribution of data. Tweets of high quality in a tweet stream are rare, leading to a proportion of 13% in the annotated data. Low quality tweets account for 87%, which makes it difficult to achieve a very good prediction. However, our concern is the performance on tweets of high quality because we want to select good tweets for summary generation. The precision of high quality tweets increases by 0.429

Method	Data	Precision	Recall	F-score	Accuracy
Random	H	0.130	0.500	0.206	0.500
	L	0.870	0.500	0.635	
LRU	H	0.321	0.636	0.427	0.765
	L	0.931	0.786	0.852	
LRO	H	**0.559**	**0.714**	**0.627**	**0.883**
	L	**0.952**	**0.910**	**0.931**	

Table 7: Evaluation of Content Quality

and 0.238 separately, which dramatically increases the probability of high quality tweets being selected.

A deep analysis of the features reveals several interesting facts. Whether a tweet includes a URL is highly valued by the model. This is mainly because tweets with URLs are usually news titles or shared information, such as videos and games from corresponding websites. They are written in formal language and deliver rich information. Another highly useful feature is the number of ellipsis. Ellipsis is used frequently in Twitter. They are usually used to leave out a part of sentences, or as a break between two irrelevant sentences. Tweets involving more than one ellipsis become difficult to read. In addition, the word length of tweets, the number of emoticons, and the stop word ratio are useful for measuring content quality.

4.7 Discussion

To make it easier for annotators to label the reference, we rank all the tweets before annotation. The ranking procedure used in selecting the reference tweets makes tweets most similar to the corpus have a higher priority for selection. However, rank is not the only aspect the annotators considered (e.g. tweets with low readability or content richness will not be selected). It is just used to reduce the annotation workload. Annotators go through the top N tweets in the ranked list until 10 reference tweets are selected (N may be 100, 1000, or more). A summarization algorithm considering similarity will benefit from this corpus. But both the mutual reinforcement model and the two baselines have not integrated the heuristics to rank tweets to create the corpus. They are compared in fair head start.

5 Conclusion

In this paper, we propose summarizing tweet streams with regard to topics along time line to produce an overview of topic evolution, which is expressed by sub-topics in chronological order. For each sub-topic, a set of salient tweets is selected to produce the summary by ranking them according to salience. Different from traditional documents, tweets suffer a great deal from pointless information and irregular writing style. We thus model the salience of a tweet by using a unified mutual reinforcement graph to incorporate the social influence of users and the content quality of tweets. The experimental results show that the proposed approach achieves substantial improvements in comparison to LexRank and phrase graph. Furthermore, the content quality and the social influence show great effectiveness in measuring the salience of tweets. In the future, we plan to exploit more specific relationships among tweets, such as retweeting in the reinforcement model to rank tweets for summary generation.

References

Ani Nenkova, K. M. (2011). *Automatic Summarization*, pages 103–233.

Beaux Sharifi, M.-A. H. and Kalita, J. (2010). Automatic summarization of twitter topics. In *National Workshop on Design and Analysis of Algorithm*.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, WWW7, pages 107–117, Amsterdam, The Netherlands, The Netherlands. Elsevier Science Publishers B. V.

Carbonell, J. and Goldstein, J. (1998). The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *In Research and Development in Information Retrieval*, pages 335–336.

Chakrabarti, D. and Punera, K. (2011). Event summarization using tweets. In *ICWSM'11*, pages –1–1.

Erkan, G. and Radev, D. R. (2004). Lexrank: graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Frederking, K. D. R. R. S. B. L. A. G. R. (2011). Topical clustering of tweets. In *Proceedings of the ACM SIGIR 3rd Workshop on Social Web Search and Mining*.

Fuentes, M., Alfonseca, E., and Rodríguez, H. (2007). Support vector machines for query-focused summarization trained and evaluated on pyramid data. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 57–60, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fung, G. P. C., Yu, J. X., Yu, P. S., and Lu, H. (2005). Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pages 181–192. VLDB Endowment.

Harabagiu, S. M. and Hickl, A. (2011). Relevance modeling for microblog summarization. In Adamic, L. A., Baeza-Yates, R. A., and Counts, S., editors, *ICWSM*. The AAAI Press.

Inouye, D. and Kalita, J. K. (2011). Comparing twitter summarization algorithms for multiple post summaries. In *SocialCom/PASSAT*, pages 298–306. IEEE.

Lin, C.-Y. and Hovy, E. (2003). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA. Association for Computational Linguistics.

Liu, F., Liu, Y., and Weng, F. (2011). Why is "sxsw" trending?: exploring multiple text sources for twitter topic summarization. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 66–75, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lloret, E. and Palomar, M. (2012). Text summarisation in progress: a literature review. *Artif. Intell. Rev.*, 37(1):1–41.

Long, R., Wang, H., Chen, Y., Jin, O., and Yu, Y. (2011). Towards effective event detection, tracking and summarization on microblog data. In *Proceedings of the 12th international conference on Web-age information management*, WAIM'11, pages 652–663, Berlin, Heidelberg. Springer-Verlag.

Mihalcea, R. (2004). Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, ACLdemo '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nichols, J., Mahmud, J., and Drews, C. (2012). Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, IUI '12, pages 189–198, New York, NY, USA. ACM.

O'Connor, B., Krieger, M., and Ahn, D. (2010). Tweetmotif: Exploratory search and topic summarization for twitter. In *ICWSM'10*, pages –1–1.

Parthasarathy, X. Y. A. G. Y. R. S. (2012). A framework for summarizing and analyzing twitter feeds. In *In Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, SIGKDD.

Sharifi, B., Hutton, M.-A., and Kalita, J. (2010a). Summarizing microblogs automatically. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 685–688, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sharifi, B., Hutton, M.-A., and Kalita, J. K. (2010b). Experiments in microblog summarization. In *Proceedings of the 2010 IEEE Second International Conference on Social Computing*, SOCIALCOM '10, pages 49–56, Washington, DC, USA. IEEE Computer Society.

Sharifi, B. P. (2010). Automatic microblog classification and summarization. In *Master's thesis*.

Shiells, K., Alonso, O., and Lee, H. J. (2010). Generating document summaries from user annotations. In *Proceedings of the third workshop on Exploiting semantic annotations in information retrieval*, ESAIR '10, pages 25–26, New York, NY, USA. ACM.

Takamura, H., Yokono, H., and Okumura, M. (2011). Summarizing a document stream. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 177–188, Berlin, Heidelberg. Springer-Verlag.

Wan, X., Yang, J., and Xiao, J. (2007). Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 552–559, Prague, Czech Republic. Association for Computational Linguistics.

Wei, F., Li, W., Lu, Q., and He, Y. (2008). Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 283–290, New York, NY, USA. ACM.

Wei, F., Li, W., Lu, Q., and He, Y. (2009). Applying two-level reinforcement ranking in query-oriented multidocument summarization. *J. Am. Soc. Inf. Sci. Technol.*, 60(10):2119–2131.

Wong, K.-F., Wu, M., and Li, W. (2008). Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 985–992, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zubiaga, A., Spina, D., Amigó, E., and Gonzalo, J. (2012). Towards real-time summarization of scheduled events from twitter streams. In *Proceedings of the 23rd ACM conference on Hypertext and social media*, HT '12, pages 319–320, New York, NY, USA. ACM.