

# Data Summarization with Social Contexts

Hao Zhuang\*, Rameez Rahman\*, Xia Hu†, Tian Guo\*, Pan Hui‡, Karl Aberer\*

\* LSIR, École Polytechnique Fédérale de Lausanne (EPFL)

† Computer Science and Engineering, Texas A&M University

‡ SyMLab, Hong Kong University of Science and Technology

hao.zhuang@epfl.ch, rameez.rahman@epfl.ch, hu@cse.tamu.edu, tian.guo@epfl.ch,  
panhui@cse.ust.hk, karl.aberer@epfl.ch

## ABSTRACT

While social data is being widely used in various applications such as sentiment analysis and trend prediction, its sheer size also presents great challenges for storing, sharing and processing such data. These challenges can be addressed by data summarization which transforms the original dataset into a smaller, yet still useful, subset. Existing methods find such subsets with objective functions based on data properties such as representativeness or informativeness but do not exploit social contexts, which are distinct characteristics of social data. Further, till date very little work has focused on topic preserving data summarization, despite the abundant work on topic modeling. This is a challenging task for two reasons. First, since topic model is based on latent variables, existing methods are not well-suited to capture *latent* topics. Second, it is difficult to find such social contexts that provide valuable information for building effective topic-preserving summarization model. To tackle these challenges, in this paper, we focus on exploiting social contexts to summarize social data while preserving topics in the original dataset. We take Twitter data as a case study. Through analyzing Twitter data, we discover two social contexts which are important for topic generation and dissemination, namely (i) *CrowdExp* topic score that captures the influence of both the crowd and the expert users in Twitter and (ii) *Retweet* topic score that captures the influence of Twitter users' actions. We conduct extensive experiments on two real-world Twitter datasets using two applications. The experimental results show that, by leveraging social contexts, our proposed solution can enhance topic-preserving data summarization and improve application performance by up to 18%.

## Keywords

Data Summarization, Social Context, Submodular Optimization, Topic Model

## 1. INTRODUCTION

Enormous amounts of social data that are generated on sites such as Twitter, Facebook, Yelp, hold great value for

both researchers and practitioners in understanding social behaviors. For instance, social data has been applied on various applications such as social event detection and sentiment analysis. Though social media data provides great opportunities, it also brings about many challenges due to its sheer size. Firstly, it is very expensive to store, share and process data of such large scale [38]. Secondly, it is computationally expensive to build analytical models to power social services and applications. Thirdly, more data does not necessarily mean more useful data [34]. Sometimes additional training data could result in worse performance [37], and it is challenging to identify those data that are truly useful.

Data summarization [18, 26, 30] has shown its effectiveness in preparing large-scale data for data analytics. Different from data compression (that reduces cost of storage and communication by compressing dataset into smaller size) or data sampling (that reduces the cost of training analytical models by randomly selecting a subset of data samples), data summarization aims to maintain (or improve) the performance of applications by selecting a subset of data that is considered to be useful. Generally, data summarization is done by transforming the problem into selecting a subset of data instances, with an objective function that quantifies properties of the selected subset. These properties could be representativeness [26, 30], diversity [28], informativeness [22], etc., which are defined in the context of different applications. However, existing methods design the objectives of data summarization purely based on attribute-value contents, ignoring social contexts which could provide valuable information. We take Twitter as a case study. Social contexts can refer to demographical information about users (e.g., age, gender, location), social status (e.g., number of followers) and their actions (e.g., reply, retweet, favorite, block). Intuitively, it can be conjectured that incorporating those tweets which are posted by influential users, or those that are retweeted by more users, into the summarized dataset, could be more beneficial to a learning task. Thus, we are motivated to investigate how social contexts could be utilized for social data summarization.

In addition, topical information plays an essential role in many text mining tasks. Despite many work on summarizing social data in different aspects such as sentiments [6] or events [9, 29], very little work has yet focused on topic preserving data summarization. Therefore, our aim is to summarize social media data into a small subset while preserving topics in the original dataset. To this end, we analyze statistical topic models [7] and present three challenges in topic-preserving summarization for social media data. The

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

CIKM'16, October 24-28, 2016, Indianapolis, IN, USA

© 2016 ACM. ISBN 978-1-4503-4073-1/16/10...\$15.00

DOI: <http://dx.doi.org/10.1145/2983323.2983736>

first challenge is that a topic is a *latent* concept that needs to be learned from the dataset. Most of the existing work [26, 28, 30] in data summarization have explicit objectives with known parameters while, because of latent topics, our work needs to estimate unknown parameters based on the dataset. The second challenge is that even if we fix parameters in the objective function, we would still have to search over an exponential number of possible subsets to find the optimal subset. The last challenge is that the subset obtained from data summarization should maintain (or even improve) the performance of applications. Put bluntly, the performance of a trending topic application should not be much worse when it works on a summarized dataset as compared to its performance on the original dataset.

To address these challenges, we explore the use of social contexts in social media data to help topic-preserving summarization. Specifically, we study this problem using two real-world Twitter datasets based on two different applications, namely *Topic Discovery* and *Tweet Classification*. In doing so, we are confronted with the following questions: how to design an objective function for preserving latent topics in Twitter dataset? Is the social context in Twitter datasets helpful for topic-preserving summarization? Does the performance of applications on summarized subsets degrade as compared to their performance on the original datasets? In addressing the above questions, this paper makes the following contributions:

- Through analyzing statistical topic models, we present an objective function for preserving topics in Twitter data. We also present two challenges related to parameter estimation and the size of the search space for this function. To solve these two challenges, we first design a submodular model, called *E-model*, based on information entropy. Apart from solving the parameter estimation challenge, *E-model* also provides a lower bound for searching the optimal subset with a greedy algorithm (Section 2).
- Based on the *E-model*, we further devise our novel model, *S-model*, which incorporates two important social contexts that influence topic generation and dissemination, namely **CrowdExp** and **Retweet** topic score. The former considers the influence of both the experts and the crowd (the majority of the users) while the latter captures the influence of Twitter users’ actions (i.e., retweet in our case) (Section 3).
- We conduct experiments on real-world Twitter datasets using two different applications, Topic Discovery and Tweet Classification. The experimental results demonstrate that, by leveraging social contexts, *S-model* can help topic-preserving data summarization and further improve application performance by up to 18% (Section 4).

## 2. DATA SUMMARIZATION FOR TWITTER TOPICS

In this paper, we use Twitter data, as an example of social data, to explore topic-preserving data summarization. Thus, in this section, we begin by formally defining our Twitter data model. Then, we discuss statistical topic models in a case of Twitter. Finally, we will introduce our data summarization framework for Twitter topics.

### 2.1 Twitter Data

Here we first give a data model for tweets and then we define topics in the Twitter dataset. Generally, a tweet is represented as a textual feature vector, where each dimension can be constructed using different models, e.g., N-grams, tf-idf scheme, Part of Speech. In this paper, given we are targeting to preserve topics, we employ the “bag-of-words” model, following common simplification in most work in information retrieval and topic modeling [7, 23], to construct the feature space using term frequency as the feature weight. Formally, we define a tweet as:

*Definition 1. (Tweet):* a text tweet  $t$  in a Twitter dataset  $\mathcal{T}$  is a sequence of words  $w_1, w_2, \dots, w_{|t|}$ , where  $w_i$  is a word from a fixed vocabulary  $\mathcal{W}$ . We represent a tweet with a bag of words, i.e.,  $t = \{w_1, w_2, \dots, w_{|t|}\}$ .

Based on the definition 1, we can define a Twitter dataset (i.e., a collection of tweets) as a Tweet-word matrix:

*Definition 2. (Tweet-word matrix):* Given a fixed Twitter dataset  $\mathcal{T}$  with a collection of  $N$  tweets, we can easily build a Tweet-word matrix  $R_{|\mathcal{T}| \times |\mathcal{W}|}$ , in which each entry  $r_{w,t} = n(w,t)$  represents the frequency of word  $w$  in tweet  $t$ . We use  $n(w,t)$  to denote the frequency of word  $w$  in  $t$ .

A topic in the Twitter dataset is modeled as a distribution over words. Formally,

*Definition 3. (Topic):* a semantic topic  $\tau$  in a Twitter dataset  $\mathcal{T}$  with a fixed vocabulary  $\mathcal{W}$  is represented by a topic model  $\theta$ , which is a probabilistic distribution of words  $\{p(w|\theta)\}_{w \in \mathcal{W}}$ . Clearly, we have  $\sum_{w \in \mathcal{W}} p(w|\theta) = 1$  and we assume there are altogether  $K$  topics in  $\mathcal{T}$ .

In the following, we will further discuss statistical topic models in detail.

### 2.2 Statistical Topic Models

Compared to tweet-word matrix in Definition 2 that can be derived directly from a Twitter dataset, topics in Definition 3 are latent variables that need to be learned from the dataset. For topic models such as the Probabilistic Latent Semantic Indexing (pLSI) and the Latent Dirichlet Allocation (LDA), the assumptions are that each document is a distribution over topics and each topic is a distribution over words [7, 8]. Specifically, the generative process of a tweet is represented as :

$$p(w, t) = p(t)p(w|t) = p(t) \sum_{\theta \in \Theta} p(w|\theta)p(\theta|t) \quad (1)$$

where  $p(w, t)$  denotes the probability of observing a word  $w$  in a tweet  $t$  and can be further interpreted as the production of  $p(t)$ , the probability distribution of tweets, and  $p(w|t)$ , the probability distribution of words given a tweet. For topic modeling, the assumption is that there is a latent topic  $\theta$  for each word  $w$ . Thus,  $p(w|t)$  can be further modeled as the multiplication of  $p(w|\theta)$ , the probability distribution of words given a topic, and  $p(\theta|t)$ , the probability distribution of topics given a tweet. Given there are a total of  $K = |\Theta|$  topics, we sum the multiplication over a set of all independent topics. In this way, we can see that the set of topics  $\Theta$  is an additional (latent) layer between tweets and words, whose parameters needs to be estimated from the dataset.

Given Eq. 1, we can infer  $p(w|\theta)$  and  $p(\theta|t)$  by maximizing the log-likelihood function of the observed tweet-word matrix

$$L(\mathcal{T}) = \log\left(\prod_{t \in \mathcal{T}} \prod_{w \in \mathcal{W}} p(w, t)^{n(w, t)}\right) \quad (2)$$

$$= \sum_{t \in \mathcal{T}} \sum_{w \in \mathcal{W}} n(w, t) \log p(w, t) \quad (3)$$

$$= \sum_{t \in \mathcal{T}} \sum_{w \in \mathcal{W}} n(w, t) \log\left(\sum_{\theta \in \Theta} p(w|\theta)p(\theta|t)\right) \quad (4)$$

The goal of a topic model is to estimate the parameters  $p(w|\theta)$  and  $p(\theta|t)$  which measure the correlation between a word  $w$  and a topic  $\theta$  and that between a topic  $\theta$  and a tweet  $t$ , respectively. For example, if we have  $p(w_{\text{football}}|\theta_{\text{sports}}) > p(w_{\text{football}}|\theta_{\text{entertainment}})$ , it means that the word *football* is more likely to occur in the topic *sports* than in the topic *entertainment*. The parameter space depends on the complexity of different topic models. For pLSI, the parameter space is  $O(Kn + Kd)$ , where  $n$  is the size of Twitter dataset,  $d$  is the size of vocabulary and  $K$  is the number of latent topics. We can see that the number of parameters shows linear growth in the number of tweets  $n$ , which suggests that the model is prone to overfitting [7]. To overcome the overfitting problem, LDA treats the topic weights as a  $K$ -parameter hidden random variable (i.e., follows Dirichlet distribution) rather than a large set of individual parameters that are linked to each dataset. In this way, the parameter space of LDA model is  $O(K + Kd)$  which does not increase linearly in the size of dataset. Therefore, LDA does not suffer from the overfitting problem like pLSI.

### 2.3 Topic-preserving Data Summarization

Given a Twitter dataset  $\mathcal{T}$ , we aim to select a subset  $\mathcal{S} \subseteq \mathcal{T}$  of bounded size  $|\mathcal{S}| = k$ , which maximizes the objective function  $F : 2^{|\mathcal{T}|} \rightarrow \mathbb{R}$ .

$$S^* = \arg \max_{\mathcal{S} \subseteq \mathcal{T}, |\mathcal{S}|=k} F(\mathcal{S}) \quad (5)$$

where we define  $S^*$  as the *summarized subset* of the original dataset  $\mathcal{T}$ . If we know the utility function  $F$ , the problem as shown in Eq. 5 is the classical knapsack problem which can be solved via greedy approximation algorithm illustrated in Algorithm 1. In our case, our target is to select a subset that preserves the topics in the original dataset. Thus, based on Eq. 4 of topic modeling, we formulate the objective function of topic-preserving data summarization as:

$$F(\mathcal{S}) = L(\mathcal{S}) = \sum_{t \in \mathcal{S}} \sum_{w \in \mathcal{W}} n(w, t) \log\left(\sum_{\theta \in \Theta} p(w|\theta)p(\theta|t)\right) \quad (6)$$

where the parameters  $p(w|\theta)$  and  $p(\theta|t)$  need to be estimated based on the Twitter dataset as discussed before. Thus, from Eq. 6, we discover that there are mainly two challenges to preserve latent topics in the Twitter dataset: **C1-parameter challenge**: topic-related parameters (i.e.,  $p(w|\theta)$ ,  $p(\theta|t)$ ) in the objective function are unknown, which need to be estimated based on each subset; **C2-combination challenge**: the search space of the number of possible subsets is exponential and even with the cardinality constraint of size  $k$ , we would still need to search over  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$  to determine  $S^*$ .

The first challenge **C1** is caused by the complexity of topic models. Recall that for Eq. 3, we can see that the objective

---

#### Algorithm 1 Greedy algorithm for E-model

---

**Input:**  $\mathcal{T}$ : original dataset,  $k$ : cardinality constraint

**Output:**  $\mathcal{S}$ : summarized subset

```

1:  $\mathcal{S} \leftarrow \emptyset$ 
2: while  $|\mathcal{S}| \leq k$  do
3:    $L = \{t \in \mathcal{T} \setminus \mathcal{S}\}$ 
4:    $t = \arg \max_{t \in L} F(\mathcal{S} \cup \{t\}) - F(\mathcal{S})$ 
5:    $\mathcal{S} = \mathcal{S} \cup \{t\}$ 
6: end while
7: return  $\mathcal{S}$ 

```

---

is to maximize total log-likelihood  $\log p(w, t)$  of picking the cell  $n(w, t)$  from the tweet-word matrix. Eq. 4 incorporates the topic modeling of pLSI in Eq. 1. As is discussed earlier, the parameter space of pLSI is  $O(Kn + Kd)$  while that of LDA is  $O(K + Kd)$ . The immediate idea to overcome the parameter challenge is that we can employ a simple topic model, which has fewer parameters. In [7], LDA and pLSI are compared with other simpler models, i.e., unigram and mixture of unigrams. Inspired by this, we first consider exploiting the unigram model for data summarization to preserve the topics in Twitter datasets. The unigram model assumes that the words of every tweet are drawn independently from a single multinomial distribution:

$$p(w, t) = \prod_{w \in t} p(w) \quad (7)$$

where  $p(w) = \frac{\sum_{t \in \mathcal{T}} n(w, t)}{\sum_{t \in \mathcal{T}} \sum_{w \in \mathcal{W}} n(w, t)}$ . Here, we can see that the parameter space of the unigram model is  $O(d)$  where  $d$  is the size of vocabulary. By incorporating the unigram model of Eq. 7 into Eq. 3, we have unigram-weighted summarization model

$$F(\mathcal{S}) = \sum_{t \in \mathcal{S}} \sum_{w \in \mathcal{W}} n(w, t) \log p(w) \quad (8)$$

Compared to pLSI formulation in Eq. 6, Eq. 8 reduces the parameter space from  $O(Kn + Kd)$  to  $O(d)$  i.e., only bounds to the size of vocabulary. However, we notice that unigram-weighted summarization model prefers to select the tweets with high-frequency words. This leads to summarization bias, as only those topics which correspond to high frequency words will be preserved. Motivated by information theory, we argue that topics are evenly distributed across tweets (i.e., high entropy) rather than concentrated in a few tweets (i.e., low entropy). Thus, we design an objective function based on information entropy to evaluate the information that is contained in each selected tweet and we formulate this data summarization model as:

$$F(\mathcal{S}) = \sum_{t \in \mathcal{S}} \sum_{w \in \mathcal{W}} n(w, t) p(w) \log \frac{1}{p(w)} \quad (\text{E-model})$$

With this formulation, *E-model* is more likely to select a subset with high entropy. It can be interpreted as the higher the entropy of a subset, the more information or topics it contains. Meantime, entropy is also a measure of *uncertainty* or *diversity*, which can reduce summarization bias compared to unigram-weighted model in Eq. 8. We can solve *E-model* by greedy algorithm as shown in Algorithm 1. In addition, the *E-model* presents three nice properties that help us solve the combination challenge **C2**.

The first property of *E-model* is monotonicity. That is, addition of more tweets to an existing subset will increase the utility of the overall selection.

PROPOSITION 1. (*Monotonicity*). Let  $\mathcal{T}$  be a collection of tweets,  $\mathcal{S}^* = \langle t_1, \dots, t_n \rangle, t_i \in \mathcal{T}, 1 < i < n$  a selection, and  $t' \in (\mathcal{T} \setminus \mathcal{S}^*)$  is from a set of non-selected tweets. Then it holds that:  $F(\mathcal{S}^* \cup \{t'\}) \geq F(\mathcal{S}^*)$

PROOF. We denote all the words that occurs in  $t'$  but not in current selection  $\mathcal{S}^*$  as the set of words  $\{t' \setminus \mathcal{S}^*\}$ . Then, we have  $F(\mathcal{S}^* \cup \{t'\}) - F(\mathcal{S}^*) = \sum_{w \in \{t' \setminus \mathcal{S}^*\}} n(w, t') p(w) \log \frac{1}{p(w)} \geq$

0. If all the words in the  $t'$  have already occurred in the current selection  $\mathcal{S}^*$ , i.e.,  $\{t' \setminus \mathcal{S}^*\} = \emptyset$ , the utility will not increase.  $\square$

Second, *E-model* shows submodularity which refers to the property that marginal gains start to diminish due to saturation of objective. That is, adding a tweet to a smaller set helps more than adding it to a larger set, w.r.t. the size of summarized subset.

PROPOSITION 2. (*Submodularity*). Let  $\mathcal{T}$  be a collection of tweets,  $\mathcal{S}^* = \langle t_1, \dots, t_n \rangle, t_i \in \mathcal{T}, 1 < i < n$  a selection, and  $t, t' \in (\mathcal{T} \setminus \mathcal{S}^*)$  a set of non-selected tweets. Then it holds that:  $F(\mathcal{S}^* \cup \{t\}) - F(\mathcal{S}^*) \geq F(\mathcal{S}^* \cup \{t'\} \cup \{t\}) - F(\mathcal{S}^* \cup \{t'\})$ .

PROOF. since we have  $\{t \setminus \mathcal{S}^*\} = \{t \setminus (t \cap \mathcal{S}^*)\} \supseteq \{t \setminus (t \cap (\mathcal{S}^* \cup \{t'\}))\} = \{t \setminus (\mathcal{S}^* \cup \{t'\})\}$ , then we can derive

$$F(\mathcal{S}^* \cup \{t\}) - F(\mathcal{S}^*) = \sum_{w \in \{t \setminus \mathcal{S}^*\}} n(w, t) p(w) \log \frac{1}{p(w)} \geq \sum_{w \in \{t \setminus (\mathcal{S}^* \cup \{t'\})\}} n(w, t) p(w) \log \frac{1}{p(w)} = F(\mathcal{S}^* \cup \{t'\} \cup \{t\}) - F(\mathcal{S}^* \cup \{t'\}).$$

Here we can see that the equality holds if and only if  $\{t \cap t'\} = \emptyset$ .  $\square$

Thus, based on Proposition 1 and 2, Algorithm 1 provides the performance guarantee for searching the optimal subset with *E-model*.

PROPOSITION 3. (*Near-Optimality*). Algorithm 1 is a  $(1 - 1/e)$ -approximation to the optimal value for *E-model*.

PROOF. For any monotone, submodular function  $F$  with  $F(\emptyset) = 0$ , it is known that a greedy algorithm that selecting the element  $t$  with the maximal value of  $F(\mathcal{S} \cup \{t\}) - F(\mathcal{S}^*)$  with  $T$  as the elements selected so far has a performance guarantee of  $(1 - 1/e) \approx 0.63$  [12]. This result is applicable to Algorithm 1, since the objective function of *E-model* is monotonic ( Proposition 1 ) and submodular ( Proposition 2 ) with  $F(\emptyset) = 0$ .  $\square$

In the next section, we will further discuss that how we use these properties to solve the combination challenge **C2**.

### 3. OPTIMIZATION

Under the previous *E-model*, we design objective functions with only considering the correlation between words and topics. However, Twitter data not only has rich text information but also the contextual information about users (e.g., number of followers, location) and their actions (e.g., reply, retweet, favorite, block). Summarizing Twitter data with *E-model* *neglects* the correlation between topics and the users who generate and disseminate topics. In the following, we will firstly enhance *E-model* with leveraging social contexts. Then we improve Algorithm 1 via lazy evaluation to improve search efficiency over an exponentially space.

### 3.1 An Enhanced Model with Social Contexts

Our task is to summarize Twitter dataset with a small subset while preserving the topics in the original dataset. The task is challenging since a topic is a latent concept. In the previous *E-model*, we measure the correlation between words and topics with unigram weights and entropy. However, we did not consider that a social topic is generated from tweets that are written and retweeted by users in a social network. Thus, the generative process of a topic is highly correlated to users and their actions such as retweet, like and reply. In the following, we firstly discuss two straightforward social contexts in the Twitter, namely: user influence and tweet influence. Then based on understanding of these two simple social contexts, we present our design of two topic scores.

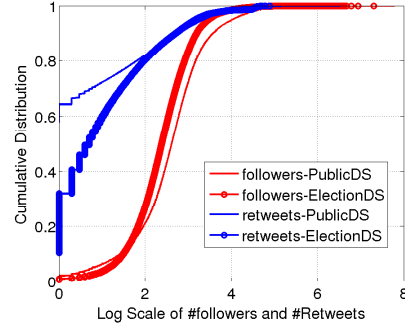


Figure 1: Cumulative distribution of the number of followers and the number of retweets in log scale (base 10)

User influence can be measured in various ways including PageRank, Granger Causality, Node’s Indegree, etc [10] while tweet influence can be measured by the number of retweets or replies, etc. In this paper, we model user influence in terms of the number of followers while tweet influence in terms of the number of retweets. To better understand user and tweet influence, we plot the cumulative distributions of both the number of retweets of a tweet message and the number of followers of a user based on two Twitter datasets (detailed in Section 4.1). It clearly shows that only about 10% of users have more than 3000 followers (with the highest having about 3.5 million followers) while the majority of users (about 80%) have less than 1200 followers. As for the number of retweets, we observe that only about 5% of tweets are retweeted more than 3000 times while the majority of tweets (about 60%) are retweeted less than 10 times. From Figure 1, it can be also seen that there are two extremes in the number of followers, while the majority of the users lies in the middle (70% of them have followers in the range [50,1200]). Thus, we can deduce that *topics are largely generated by a crowd of users who have about 50-1200 followers and their tweets are normally retweeted less than 10 times*. In other words, we should assign more weights to the crowd (the majority of the users) for their effects on the topic generation and dissemination, not just giving more weights to those users with a high number of followers and those tweets with a high number of retweets.

Base on these observations from Figure 1, we further design two topic scores to capture the impact of the crowd, the experts and their retweet actions on topic generation and dissemination, namely:

(1) **CrowdExp topic score**: We use CrowdExp topic score to measure the contribution of a user on generating the

latent topics. Intuitively, the higher the influence of a user, the more contribution she makes. In other words, we would like to preserve those tweets that are posted by topic experts who have the maximum number of followers. Meanwhile, according to [33], a group of diverse crowd can outperform a group of experts due to the effect of collective wisdom. Also, based on our discussion before, we know that most of the topics in Twitter are generated by the majority of the users. Thus, we seek to devise a topic score that considers the influence of both topic experts and the majority of the users, on topic generation and dissemination. We name our topic score, *CrowdExp*. As the name suggests, *CrowdExp*, is a combination of the influence that the crowd of users has on topics in a dataset, and the experts, those that have a high number of followers. The influence of users in social network can be evaluated by various algorithms such as PageRank and Granger Causality. However, these algorithms are too expensive to compute. In this paper, we simply model user influence as the number of followers  $x$ , i.e., in-degree of a node in a user relation graph. There are two steps to compute the *CrowdExp* topic score: distribution estimation of user influence and topic score computation.

**Step 1: Distribution estimation of user influence.** Figure 1 shows that the number of followers of Twitter users  $x$  follows a power-law distribution, the probability density function of which is defined by

$$f_u(x) = (\alpha - 1)(1 + x)^{-\alpha} \quad (9)$$

where  $\alpha$  is the exponent parameter and can be derived by maximum likelihood estimators (MLEs)

$$\alpha = 1 + |X| \left[ \left( \sum_{x \in X} \ln(1 + x) \right) \right]^{-1} \quad (10)$$

in which  $X$  is the set of observed values, i.e., the number of followers for users that write tweets. Furthermore, we compute user influence based on the cumulative distribution function over  $f_u(x)$

$$F_u(x) = \int_0^x f_u(z) dz = 1 - (1 + x)^{(1-\alpha)} \quad (11)$$

In this way, the user influence is mapping to the interval  $[0, 1]$  in which a topic expert has the value of influence approaching to 1.

**Step 2: CrowdExp topic score computation.** In this step, we aim to design a score function that models the influence of both the majority crowd and the experts. We apply the following piecewise function which takes user influence as the input to compute *CrowdExp* topic score:

$$u_t = \begin{cases} -F_u(x) \log F_u(x), & x \leq F_u^{-1}(\eta) \\ \log(F_u(x) + \phi), & x > F_u^{-1}(\eta) \end{cases} \quad (12)$$

where  $\eta$  is the cut-point of experts and non-experts and  $\phi$  is a location parameter. For example, if  $\eta = 0.9$ , it means that the experts are those top 10% of users who have a high number of followers. In practice, we tune this parameter from 0.85 to 0.95 in order to find the suitable value that has the best performance in terms of application metrics. With this function, we can give more weights for the crowd who are majority and the experts who are important.

**(2) Retweet topic score:** Similar to user influence, we can also evaluate the importance of a tweet by the number of retweets. ‘Retweet’ happens when a user is interested in

some topics that are contained in a tweet and thus decides to further disseminate it. In other words, retweet reflects approval or recommendation that a tweet has in a community, which is an important social context [4]. We infer that there are two correlations that between topics and retweet actions: 1) most topics are generated from the majority of tweets (from Figure 1 it can be seen that the majority of tweets is retweeted that less than 10 times); 2) the higher the number of retweets, the more likely a tweet can become a topic. We apply similar process to compute the tweet importance based on the number of retweets  $y$ . Firstly, we estimate the scale parameter  $\beta$  for probability density function:

$$f_r(y) = (\beta - 1)(1 + y)^{-\beta} \quad (13)$$

Then, we compute tweet importance based on the cumulative distribution function over  $f_r(y)$

$$F_r(y) = \int_0^y f_r(z) dz = 1 - (1 + y)^{(1-\beta)} \quad (14)$$

Finally, we design retweet score function as an absolute log function to capture two correlations that discussed above.

$$re_t = |\log(F_r(y) + \gamma)| \quad (15)$$

where  $\gamma$  is a location parameter and is determined by cut-point of majority and non-majority. From this function, we can observe that more weights are given to the tweets which are important and those which are the majority.

**(3) Put All Together.** Finally, we incorporate the *CrowdExp* and retweet topic scores, given by Equations 12 and 15, into *E-model* based on the product rule. Then, we have an enhanced model with exploiting social contexts *S-model* that

$$F(\mathcal{S}) = \sum_{t \in \mathcal{S}} \sum_{w \in \mathcal{W}} n(w, t) \cdot \left[ u_t \cdot re_t \cdot p(w) \cdot \log \frac{1}{p(w)} \right] \quad (\text{S-model})$$

Thus, we can see that *S-model* considers the impact of both experts and majority users, and their retweet actions on topic generation and dissemination, as well as the content diversity based on entropy measures for topic-preserving data summarization.

Also, it is easy to discover that *S-model* preserves all the properties of *E-model* and thus we can also solve *S-model* by Algorithm 1. However, for a large dataset with the huge number of items, the time complexity of Algorithm 1 is high. Next, we will discuss how we optimize Algorithm 1 by applying lazy evaluation.

## 3.2 Lazy Greedy

Given a tweet dataset  $\mathcal{T}$ , Algorithm 1 needs  $O(|\mathcal{T}| * k)$  times evaluation on function  $F(\cdot)$  in order to find an optimal summarized subset of size  $k$ . However, when the size of  $\mathcal{T}$  increases greatly, the standard greedy algorithm becomes more expensive. Fortunately, Minoux [24] *et.al* developed lazy greedy algorithm, which exploits submodularity to prune the search space and accelerate the searching. Since our objective function is submodular, we can apply this technique and Algorithm 2 shows our implementation of lazy greedy. Instead of computing  $F(\mathcal{S}^* \cup \{t\}) - F(\mathcal{S}^*)$  for each tweet  $t \in \mathcal{T}$ , the lazy greedy algorithm keeps an upper bound  $\rho(t)$  on the marginal gain sorted in decreasing order (i.e., store in a priority queue in Algorithm 2). In each iteration, the lazy greedy evaluates the element on top of the list

---

**Algorithm 2** Lazy greedy algorithm

---

**Input:**  $\mathcal{T}$ : original dataset,  $k$ : cardinality constraint**Output:**  $\mathcal{S}$ : summarized subset

```
1:  $\mathcal{S} \leftarrow \emptyset$ 
2:  $PriorityQueue < Node > queue$  //store marginal gains
3: for  $t \in \mathcal{T}$  do
4:    $marginalGain = F(\mathcal{S} \cup \{t\}) - F(\mathcal{S})$ 
5:    $queue.add(t, marginalGain)$  //initial  $\rho(t)$ 
6: end for
7: while  $|\mathcal{S}| \leq k$  do
8:    $t = queue.poll()$  // get the top tweet
9:    $marginalGainCurrent = F(\mathcal{S} \cup \{t\}) - F(\mathcal{S})$ 
10:  if  $marginalGainCurrent > \rho(t)$  then
11:     $\mathcal{S} = \mathcal{S} \cup \{t\}$ 
12:  else
13:     $queue.add(t, marginalGainCurrent)$ 
14:  end if
15: end while
16: return  $\mathcal{S}$ 
```

---

(line 8-9), say  $t$ , and updates its upper bound (line 10-14). If after the update  $\rho(t) \geq \rho(t')$  for all  $t' \neq t$ , submodularity guarantees that  $t$  is the element with the largest marginal gain. Even though the exact cost (i.e., number of function evaluations) of lazy greedy is unknown, this algorithm leads to orders of magnitude speedups in practice [25].

## 4. EXPERIMENTS

In this section, we empirically evaluate the  $S$ -model in terms of both computational cost and performance using two real-world Twitter datasets. Through extensive experiments, we aim to answer the following questions:

- How effective is the proposed  $S$ -model, compared to other existing solutions which do not consider social contexts, in preserving topics in the Twitter dataset?
- How is the performance of different applications based on topic models influenced by different summarization models?
- What are the tradeoffs between cost and performance entailed in data summarization?

In the following, we first introduce the datasets and then present our experimental settings and results.

### 4.1 Datasets

We use two datasets crawled from Twitter based on two different public APIs: Sample streaming API and Search REST API [2]. The former focuses on completeness while the later on relevance [3]. The first dataset, which we denote as **PublicDS**, is collected using the public Sample streaming API which returns a small random sample (about 1%) of all public real-time tweets. We select 1 million Tweets which are written in English. The second dataset, which we denote as **ElectionDS**, is about the US 2012 Presidential Election, collected using the public Search API. The Search API allows users to specify one or more search terms to obtain historical tweets and only those tweets that match

	<b>PublicDS</b>	<b>ElectionDS</b>
#Users	716,958	634,814
#Tweets	1 million	1 million
#Max in-degree	61,803,119	20,143,264
#Max retweets	3,330,699	76,096

Table 1: Statistics of two Twitter datasets

the specified terms are returned. For example, for the 2012 presidential election, we can use political keywords such as “Barack Obama”, “Mitt Romney” or “Joe Biden”, etc. ElectionDS also contains 1 million tweets. Table 1 summarizes the statistics of two datasets.

### 4.2 Experimental Setup

**Evaluated techniques:** We compare  $S$ -model with existing solutions that do not exploit social contexts:

- *E-model*: It considers the information entropy as also shown in Section 2.3.
- *Random*: It is common used sampling methodology that selects items randomly from original big datasets, as it preserves certain important statistical properties of the entire dataset. It is also used in Twitter’s public stream API which returns about 1% random sample of all tweets.
- *ClusterSum*: In this technique, the objective function of  $K$ -medoids clustering is used, which defines representativeness of an item (i.e., how well a selected item represents the other items in the dataset) based on distance measures. For example, the loss function  $L(S) = \frac{1}{|V|} \sum_{v \in V} \min_{s \in S} d(v, s)$  is to find a subset  $S \subset V$  of medoids that has a minimum average distance (e.g., cosine distance in our evaluation) to each element in the original dataset  $V$ . For details, please refer to [5].
- *KLSum*: KLSum generates a summary  $S$  given a tweet collection  $D$  via objective function  $S = \min_{s \in D} KL(P_D || P_S)$  where  $KL$  is Kullback-Lieber (KL) divergence [14]. This method greedily selects tweets to a summary set so long as it decreases the KL divergence.
- *SvdSum*: This method represents tweet collection  $D$  as a matrix and applies singular value decomposition (i.e.,  $D = U\Sigma V^T$ ) to text summarization [32]. This method selects the most informative tweets according to the matrix  $V^T$ .
- *LexSum*: LexSum assesses the eigenvector centrality of each tweet and extracts the most important ones to include in the summary [11].

**Metrics:** we evaluate data summarization models in terms of both computational cost and performance. From the cost perspective, we focus on time cost and break it down into two phases: 1) *Summarization Time* to complete data summarization and 2) *Training Time* to finish training topic models based on summarized subsets. Both summarization and training time costs depend on the size of summarized subsets. Besides the time cost, we still need to consider the performance of different data summarization models, where performance is defined by different applications. A good data summarization model shall not only spend less time on summarizing the dataset but also provide good performance for applications based on summarized subsets.

From the performance perspective, we analyze two representative applications based on topic models. The first application is Topic Discovery. In this application, the baseline performance is represented by the topics generated from the original dataset. The performance of different summarized subsets (based on different summarization techniques)

is evaluated by comparing their divergence from the baseline. Since a topic is a probability distribution over words, we use Jensen-Shannon divergence (JSD) to measure the distance between two topics i.e., divergence between two probability distributions. Specially, we compute JSD as

$$JSD(\tau_1||\tau_2) = \frac{1}{2}[KL(\tau_1||M) + KL(\tau_2||M)] \quad (16)$$

where  $M = \frac{1}{2}(\tau_1 + \tau_2)$ ,  $\tau_1$  and  $\tau_2$  are two topics, and  $KL$  represents the Kullback-Leibler divergence. A low value of JSD means that two topics (probability distributions) are highly similar, i.e., topics are well preserved.

The second application is Tweet Classification. The goal of this application is to classify tweets with the same topics together. We use hashtags and search terms as the ground truth for classification and further compute the classification accuracy. The higher the classification accuracy, the better the summarized subset.

### 4.3 Evaluation

In the following, we will detail our evaluation from the perspective of both computational cost and performance.

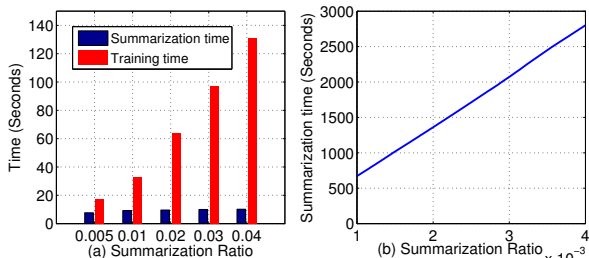


Figure 2: (a) Time cost on data summarization with S-model and topic model training with LDA; (b) Time cost without lazy evaluation of S-model

#### 4.3.1 Computational Cost

The time cost largely depends on the size of summarized subsets. We define *summarization ratio* as the size of summarized subset  $\mathcal{S}$  to that of the (entire) original dataset  $\mathcal{T}$ , i.e.  $summarization\ ratio = |\mathcal{S}|/|\mathcal{T}|$ . We vary summarization ratio from 0.005 to 0.04 with *S-model* and Figure 2 shows the results based on the PublicDS. We can see that with the increase in the size of summarized subset, the summarization time increases slowly. Let’s consider the case in Figure 2 (a) where the summarization ratio varies from 0.01 to 0.04. We can see that the summarization time only increases by 1 second, from about 9.1 seconds to 10.1 seconds while the size of summarized subset increases three times from about 1 percent to 4 percent of the original dataset. On the other hand, we observe the training time cost increases linearly in the size of summarized subsets from 24 seconds at 0.01 to about 97 seconds at 0.04. If we train the topic models based on the entire original dataset, the training time is about 3820 seconds (not shown in Figure 2). In addition, Figure 2 (b) also shows that if we do not apply lazy evaluation, the time cost of data summarization is highly expensive. As we can observe, it takes almost 3000 seconds to select 4000 tweets from a total of 1 million tweets (i.e., summarization ratio=0.004). Thus, our results confirm that lazy evaluation brings about orders of magnitude improvement in the summarization time cost. Thus, based on our observations, we discover that sum of the time cost on data summarization and training times on the summarized subsets of our proposed *S-model* is negligible compared to the high training

cost without data summarization. Also, we notice that the time cost on training models based on summarized subsets increases proportionally to the summarization ratio.

Meanwhile, we note that the time costs of *E-model* is highly similar to that of *S-model*. The overhead of *S-model* to compute CrowdExp and Reteet topic score is negligible (less than 1 second on both datasets). In addition, the results of time cost of *S-model* based on ElectionDS show a similar trend, which we omit for saving space. Finally, we discover that the summarization time cost of all other baselines (except Random) are an order of magnitude more computationally intensive, due to lack of optimizations for summarizing large-scale datasets.

#### 4.3.2 Performance

In the above experiments, we demonstrated that data summarization can reduce the time cost on training models. Next, we will evaluate the performance of different summarization models for two applications, namely Topic Discovery and Tweet Classification. For performance evaluation of Topic Discovery, we compute the difference between topics generated on the basis of summarized subsets, and those that are generated based on the original dataset. For Tweet classification, we evaluate how different data summarization methods influence application’s performance in terms of classification accuracy.

**Topic Discovery:** LDA is a widely used topic model to discover the topics in the Twitter dataset [15, 35, 36]. It takes three parameters as its input: the number of topics  $K$ , a hyperparameter for the Dirichlet prior topic distribution  $\alpha$  and a hyperparameter for the Dirichlet prior word distribution  $\beta$ . Choosing optimal parameters is a very challenging problem, and it is not the focus of our work. In this paper, we set  $K = 100$  and use priors of  $\alpha = 50/K$ , and  $\beta = 0.01$  as suggested by [7, 27]. We use Mallet software library [1] to discover the topics in the dataset. We firstly discover the topics based on the entire original dataset and we denote this set of topics as  $\Gamma^{full} = \{\tau_1^{full}, \dots, \tau_{100}^{full}\}$ . Then based on different summarization models, we discover the topics respectively and we denote this set as  $\Gamma^{sum} = \{\tau_1^{sum}, \dots, \tau_{100}^{sum}\}$ .

Since there is no implicit orderings of LDA’s topics, we first match them based on the similarity of the words in the distribution. To match the topics, we construct a weighted bipartite graph between the topics from  $\Gamma^{full}$  and  $\Gamma^{sum}$ . Since each topic is a bag of words, we can apply the Jaccard similarity between the words in two topics from different topic sets as the weight of the edges in the graph.

$$d(\tau_i, \tau_j) = \frac{|\tau_i \cap \tau_j|}{|\tau_i \cup \tau_j|}, \tau_i \in \Gamma^{full}, \tau_j \in \Gamma^{sum}$$

After constructing the graph, we use maximum weight matching algorithm proposed in [13] to find the best matches between topics from  $\Gamma^{full}$  and  $\Gamma^{sum}$ . After matching, we can compute Jensen-Shannon divergence using Eq. 16 between topics since each topic is a probability distribution over words.

Before we evaluate the performance of different models, we firstly evaluate the effect of summarization ratio by varying its value from 0.01 to 0.04 only with *S-model*. Then, we compute the JSD between topics that are generated based on the original dataset and those generated from summarized subsets of different summarization ratios. Figure 5 shows

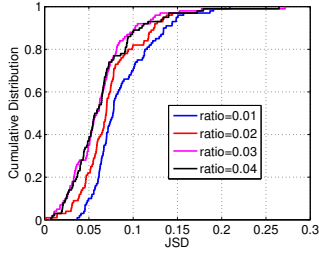


Figure 5: JSD with varying summarization ratio of S-model

the results. It is clear that with more data in the summarized subsets, the topic distribution generated by summarization is closer to that generated based on the original dataset. However, we can also observe that more data provides diminishing benefits since the summarized subsets with  $ratio = 0.03$  and  $ratio = 0.04$  show very similar results.

For the following set of experiments, we fix the summarization ratio to 0.02 to compare the performance of different models. For each model, we compared the topics generated from the subset (with ratio 0.02) to the topics generated from the original dataset. Figure 3 and Figure 4 show the histograms of JSD of different models based on PublicDS and ElectionDS, respectively. We observe clearly that both *E-model* and *S-model* outperform other models. In detail, we can see that *E-model* has the lowest average JSD (i.e.,  $\mu = 0.0666$  for PublicDS and  $\mu = 0.0684$  for ElectionDS) while *S-model* has the lowest standard deviation of JSD (i.e.,  $\sigma = 0.035$  second lowest for PublicDS and  $\sigma = 0.0384$  for ElectionDS). In addition, we observe that only all JSD of *S-model* in Figure 3 (a) and Figure 4 (a) are less than 0.2 while there always exist some topics (i.e., with  $JSD > 0.2$ ) that are far from the topics generated based on  $\Gamma_{all}$  for all other models. We infer the reason is that compared to other models like *E-model* which only consider the correlation between words and topics, the performance of *S-model* is further regularized by social contexts that are exploited in *S-model*. That is also why other models have higher probability to choose some outliers, which can be observed in Figure 3 (b)-(g) and 4 (b)-(g) with some JSDs larger than 0.2. Another interesting observation is that the performance of *LexSum* and *ClusterSum* are even worse than Random model. We conjecture this is because the topics are evenly distributed in the dataset rather than being clustered around some representative tweets (in terms of similarity or centrality measures). In such a scenario, it can be expected that Random model would do better than *LexSum* and *ClusterSum*.

From above results, we can safely reach two conclusions:

- 1) *S-model* shows very similar performance to the *E-model*

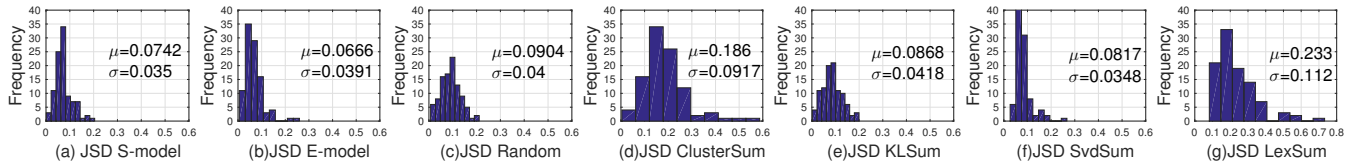


Figure 3: Histograms for JSD based on PublicDS

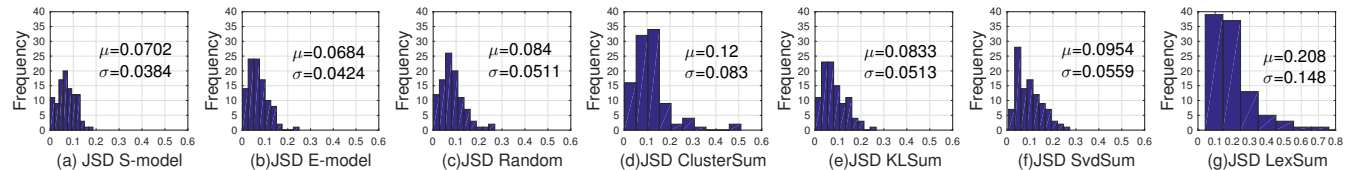


Figure 4: Histograms for JSD based on ElectionDS

with the added benefit that it is less likely to select the outliers from the original dataset; 2) Representativeness-based approaches which are based on similarity (used in ClusterSum) or centrality (used in LexSum) measures show poor performance in preserving topics in the dataset.

**Tweet Classification:** Based on an already trained topic model, we can estimate the topic probability for a new tweet and further classify tweets within the same topic together. For PublicDS, we select a set of 100 hashtags, each of which is a label for a unique category. Further, we select 500 new tweets for each hashtag (i.e., tweets labeled with particular hashtags). Then, given a new tweet, we infer its topic distribution and assign it to a topic that has a maximum probability distribution. We thus have tweets which have originally been labeled by particular hashtags, and subsequently, have topics assigned to them. Finally, for each hashtag, we count the number of tweets belonging to that hashtag, which have been assigned to the same topic. Ideally, if all the tweets with the same hashtag are assigned to the same topic, the accuracy is 100%. In practice, for tweets belonging to each hashtag, we consider the top 5 topics with the highest probability distribution (as opposed to only one maximum). Then we count the number of tweets that have been assigned to any of these top 5 topics. After this we can define classification accuracy as the number of tweets in the top 5 topics to the total number of tweets (with the same hashtag). As for ElectionDS, instead of using hashtags, we use search terms as category labels. For example, we regard the set of tweets that are retrieved with the same search term as in the same category.

We firstly evaluate the performance of different models by fixing summarization ratio as 0.02 and Table 2 shows the accuracy of tweet classification. It is clear that the *S-model* outperforms other data summarization methods for both datasets. In addition, we compare the results of different models with the results based on the entire original dataset (i.e., *AllData* in the Table 2). We compute p-value using Student's t-test for statistical significance analysis (with 5% significance level). From the Table 2, we can observe that (i) both results of *S-model* are significantly better than the results from the entire original dataset for both datasets (with both p-values less than 0.05); (ii) both the results of ClusterSum are significantly worse than that of AllData (with p-values  $< 0.05$ ) while those of other baselines are not significantly different from the results of AllData (with p-values  $> 0.05$ ). Based on above observations, we can safely conclude that our model that exploits social contexts can effectively preserve topics in the original dataset and also outperforms



	PublicDS		ElectionDS	
	accuracy	p-value	accuracy	p-value
AllData	0.6660	—	0.4648	—
S-model	<b>0.7133</b>	<0.0001	<b>0.5295</b>	<0.0001
E-model	0.6605	0.5685	0.4702	0.0724
Random	0.6545	0.2166	0.4496	0.0730
ClusterSum	0.6514	0.0193	0.4311	<0.0001
KLSum	0.6553	0.7758	0.4533	0.6437
SvdSum	0.6546	0.6841	0.4532	0.6017
LexSum	0.6511	0.7519	0.4362	0.3138

Table 2: Classification accuracy that are based on different summarization models and significance analysis with p-values of t-test ( $\alpha = 0.05$ )

the other existing summarization methods (those using similarity or centrality measures) for our target applications.

Furthermore, we vary the summarization ratio from 0.005 to 0.04 and Figure 6 shows the results (we omit *ClusterSum* and *LexSum* since their performance are worse than Random model). It is clear that the performance of *S-model* always outperforms other models at any summarization ratio. With the summarization ratio increasing from 0.005 to 0.02, the overall performance of each model increases slightly for both datasets. However, we still observe that the performance of different models fluctuate as the summarization ratio increases from 0.02 to 0.04, which confirms that more data does not always improve the application’s performance.

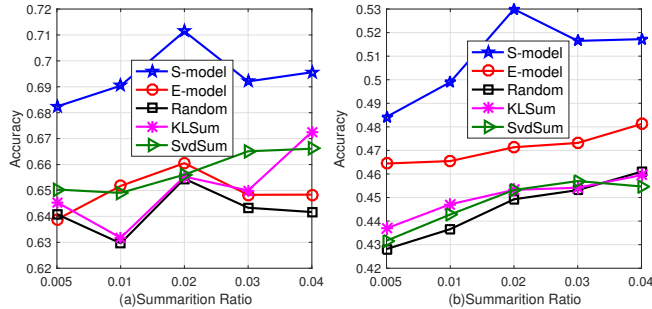


Figure 6: Classification accuracy of different models with varying summarization ratios for (a) PublicDS and (b) ElectionDS

#### 4.4 Summary

Based on extensive experiments, we can answer the three questions raised in the beginning of this section. From the first application of Topic Discovery, we confirm that our proposed *S-model* which leverages social contexts is effective in preserving topics in the Twitter dataset and is less likely to select outliers than other models which do not consider social contexts. For the second application of Tweet Classification, we observe that the *S-model* outperforms other models. In particular, we observe that the performance of Tweet Classification based on summarized subsets via *S-model* is significantly better than that based on the entire dataset. It means that our *S-model* can effectively select the data that is truly useful for the application as well as being effective in removing noise.

There are two reasons why our proposed *S-model* outperforms other existing methods for preserving topics. First, our model exploits social contexts, i.e., CrowdExp and Retweet topic scores, which model the impact of both centrality and majority in users and their retweet actions, on topic gen-

eration and dissemination. This is different from the existing methods which separately utilize the user centrality (*LexSum*), or similarity-based representativeness (*ClusterSum*) for data summarizations. Second, *S-model* is enhanced based on *E-model*, which is more likely to generate a subset of highly diverse content. In other words, our proposed *S-model* is novel since it integrates both experts and majority users, their retweet actions, and content diversity based on entropy measures, for social data summarization.

However, as for making sound tradeoff decisions between cost and performance, it largely depends on the performance requirements of different applications. For instance, if the application designer considers 66% accuracy for PublicDS to be good enough for Tweet Classification, then *Random* would be the best solution since the cost of random sampling is almost zero. For our *S-model*, the cost of data summarization is about 10 seconds as shown in Figure 2 but the performance of Tweet Classification can be improved by about 9% for PublicDS and 18% for ElectionDS compared to Random method. Thus, our proposed *S-model* will be preferred by applications that have high performance requirements.

## 5. RELATED WORK

**Topic model.** Topic model [7] is a popular statistical model for discovering topics in text corpora. Previous work [15, 35, 36] discovered the topics with considering both geographic location and time. In addition, based on trained topic models, many applications were developed such as tweet classification [16] and collaborative filtering [7]. In this paper, we do not focus on new topic models and applications. Instead, we focus on topic-preserving data summarization and we further validate the effectiveness of our approach using two applications that are based on topic models.

**Text and Data Summarization.** There is a lot of related work on text summarization, which focuses on constructing summaries for a natural language text such as documents [18] and news articles [30]. Unlike text summarization, data summarization aims to identify a truly useful subset of structured data from the original dataset. The ‘usefulness’ is defined based on data properties such as representativeness [26, 30], diversity [28], informativeness [22] and coverage [18]. However, these work do not exploit social contexts. As for text summarization, many related work focus on different dimensions such as user’s sentiments [6], events[9, 29] and contextual information [10]. Specifically, Chang *et.al* [10] proposed a supervised learning framework to summarize contextual information using different user influence models. Unlike their work on text summarization to generate contextual information, we focus on data summarization to generate a subset of the original dataset while preserving topics. In addition, we consider cost-performance tradeoffs in data summarization, which avoids using compute-intensive learning models.

**Exploiting social contexts.** Social contexts are widely exploited in recommender systems [17, 19, 21] and prediction models [20]. Both [19] and [21] added social regularization into matrix factorization recommendation model to constrain the taste difference while Lu [20] incorporated social contexts to a linear regression model for review quality predictions. However, it was not clear if social contexts can help data summarization and our work is the first work that incorporates social contexts into data summarization model for preserving topics in the original dataset.

**Submodular Optimization.** Submodularity [31] is a property of set functions which models natural diminishing returns property. This property states that adding an element to a smaller set has more value than adding it to a larger set. In data summarization, many previous studies [18, 26, 30, 34] designed their objective functions as submodular functions in order to achieve the performance guarantee (i.e.,  $1 - 1/e$  approximation guarantee to the optimum solution) for their greedy algorithms. In addition, many variants such as lazy greedy [24] and stochastic greedy [25] exploited submodularity to implement accelerated versions of classical greedy algorithm. In this work, we also design our objective function as a submodular function and apply the lazy greedy algorithm to accelerate the search performance.

## 6. CONCLUSION AND FUTURE WORK

Summarizing social data gives us the opportunity to exploit social contexts for data summarization. In this paper, we select Twitter as an example social data site, and focus on summarizing Twitter datasets while preserving topics. We firstly design a simple summarization model, namely *E-model*, that preserves topics without leveraging social contexts. Then, we propose our *S-model* which exploits two social contexts that are important for topic generation and dissemination. Finally, our experimental results demonstrate the effectiveness of our *S-model* for two different applications that are based on topic models.

This work suggests some interesting future directions. One direction is that we can extend our models by incorporating other social contexts. For example, we can explore friendships in the social network to help data summarization. Another direction is that we can explore other applications that are based on different analytical models other than topic model (which is what we focus on in this paper). Also, it is an interesting question whether there exists a general model that can summarize data for different types of applications.

## Acknowledgment

This research is funded by the EU project CloudSpaces: Open Service Platform for the Next Generation of Personal clouds (FP7-317555) and NanoTera.ch OpenSense2 project.

## 7. REFERENCES

- [1] Machine learning for language toolkit. <http://mallet.cs.umass.edu/>.
- [2] Twitter public apis. <https://dev.twitter.com/overview/documentation>.
- [3] Twitter public search apis. <https://dev.twitter.com/rest/public/search>.
- [4] S. Auty and R. Elliott. Being like or being liked: identity vs. approval in a social context. *Advances in Consumer Research*, 28(1), 2001.
- [5] A. Badanidiyuru, B. Mirzasoleiman, A. Karbasi, and A. Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 671–680. ACM, 2014.
- [6] S.-A. Bahrainian and A. Dengel. Sentiment analysis and summarization of twitter data. In *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*, pages 227–234. IEEE, 2013.
- [7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of machine Learning research*, 3:993–1022, 2003.
- [8] Y. Cha, B. Bi, C.-C. Hsieh, and J. Cho. Incorporating popularity in topic models for social network analysis. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 223–232. ACM, 2013.
- [9] D. Chakrabarti and K. Punera. Event summarization using tweets. *ICWSM*, 11:66–73, 2011.
- [10] Y. Chang, X. Wang, Q. Mei, and Y. Liu. Towards twitter context summarization with user influence models. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 527–536. ACM, 2013.
- [11] G. Erkan and D. R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, pages 457–479, 2004.
- [12] S. Fujishige. Polymatroidal dependence structure of a set of random variables. *Information and Control*, 39(1):55–72, 1978.
- [13] Z. Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)*, 18(1):23–38, 1986.
- [14] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics, 2009.
- [15] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsoulis. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, pages 769–778. ACM, 2012.
- [16] L. Hong and B. D. Davison. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.
- [17] X. Hu, L. Tang, J. Tang, and H. Liu. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 537–546. ACM, 2013.
- [18] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.
- [19] X. Liu and K. Aberer. Soco: a social network aided context-aware recommender system. In *Proceedings of the 22nd international conference on World Wide Web*, pages 781–802. International World Wide Web Conferences Steering Committee, 2013.
- [20] Y. Lu, P. Tsaparas, A. Ntoulas, and L. Polanyi. Exploiting social context for review quality prediction. In *Proceedings of the 19th international conference on World wide web*, pages 691–700. ACM, 2010.
- [21] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 287–296. ACM, 2011.
- [22] R. Mehrotra and E. Yilmaz. Representative & informative query selection for learning to rank using submodular functions. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 545–554. ACM, 2015.
- [23] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceedings of the 17th international conference on World Wide Web*, pages 101–110. ACM, 2008.
- [24] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978.
- [25] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause. Lazier than lazy greedy. *arXiv:1409.7938*, 2014.
- [26] B. Mirzasoleiman, A. Karbasi, R. Sarkar, and A. Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems*, pages 2049–2057, 2013.
- [27] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley. Is the sample good enough? comparing data from twitter’s streaming api with twitter’s firehose. *arXiv preprint arXiv:1306.5204*, 2013.
- [28] T. T. Nguyen, Q. V. H. Nguyen, M. Weidlich, and K. Aberer. Result selection and summarization for web table search. In *31st IEEE International Conference on Data Engineering*, number EPFL-CONF-203577, 2015.
- [29] J. Nichols, J. Mahmud, and C. Drews. Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 189–198. ACM, 2012.
- [30] F. Pan, W. Wang, A. K. Tung, and J. Yang. Finding representative set from massive data. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.
- [31] B. Sankaran, M. Ghazvininejad, X. He, D. Kale, and L. Cohen. Learning and optimization with submodular functions. *arXiv preprint arXiv:1505.01576*, 2015.
- [32] J. Steinberger and K. Jezek. Using latent semantic analysis in text summarization and summary evaluation. In *Proc. ISIM’04*, pages 93–100, 2004.
- [33] J. Surowiecki. *The wisdom of crowds*. Anchor, 2005.
- [34] H. P. Vanchinathan, A. Marfurt, C.-A. Robelin, D. Kossmann, and A. Krause. Discovering valuable items from massive data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1195–1204. ACM, 2015.
- [35] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–613. ACM, 2013.
- [36] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Advances in Information Retrieval*, pages 338–349. Springer, 2011.
- [37] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection?. In *BMVC*, volume 3, page 5. Citeseer, 2012.
- [38] H. Zhuang, I. Filali, R. Rahman, and K. Aberer. Coshare: A cost-effective data sharing system for data center networks. In *2015 IEEE Conference on Collaboration and Internet Computing (CIC)*, pages 11–18. IEEE, 2015.