Neurocomputing 152 (2015) 274–286

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Search engine reinforced semi-supervised classification and graph-based summarization of microblogs

Yan Chen [a,*], Xiaoming Zhang [a], Zhoujun Li [a], Jun-Ping Ng [b]

[a] State Key Laboratory of Software Development Environment, Beihang University, China
[b] Bloomberg L.P., USA

ARTICLE INFO

Article history:
Received 29 April 2014
Received in revised form
14 October 2014
Accepted 31 October 2014
Communicated by Y. Chang
Available online 11 November 2014

Keywords:
Microblog
Topic classification
Summarization
Probabilistic graphical model
Semi-supervised
Pagerank

ABSTRACT

There is an abundance of information found on microblog services due to their popularity. However the potential of this trove of information is limited by the lack of effective means for users to browse and interpret the numerous messages found on these services. We tackle this problem using a two-step process, first by slicing up the search results of current retrieval systems along multiple possible genres. Then, a summary is generated from the microblog messages attributed to each genre. We believe that this helps users to better understand the possible interpretations of the retrieved results and aid them in finding the information that they need. Our novel approach makes use of automatically acquired information from external search engines in each of these two steps. We first integrate this information with a semi-supervised probabilistic graphical model, and show that this helps us to achieve significantly better classification performance without the need for much training data. Next we incorporate the extra information into graph-based summarization, and demonstrate that superior summaries (up to 30% improvement in ROUGE-1) are obtained.

1. Introduction

Microblog services have provided a platform for users to convey their thoughts, share their experience and perform virtual social activities. One of the better-known platforms include Twitter,[1] which has more than 140 million active users and 1 billion new microblog messages (or *tweets*) posted every 3 days[2] as of March 2012. Over time, the tremendous number of microblog messages that have been generated makes up a large and informative repository from which users can query to retrieve information.

However the sheer volume of these messages is a double-edged sword. To get the information they want, users have to wade through voluminous search results to locate the information they are interested in. While textual markups like hashtags can help users to zoom-in quickly on messages of interests, the open-ended nature and free styling of these markups affect the effectiveness of such searches. Typical microblog platforms display search results in a ranked list sorted in order of relevance to query keywords or hashtags. Unfortunately, these search terms are very short, potentially ambiguous, or even vague, leading to unsatisfactory search results. For example, searching for the keyword “apple” on Twitter returns a set of messages that is diversified and varied. Fig. 1 shows an example of the results obtained from Twitter. The results span several different genres, ranging from albums (presumably sold on the Apple iTunes store), to the band Beatles, to a reference to the voice recognition feature *Siri* on a phone made by the company named “Apple”. If luck has it, we may probably retrieve tweets on the fruit itself!

We believe that to fully tap on the potential of the large repository of microblog messages, it is important to make it easier for users to retrieve meaningful search results. For example if the search results are presented based on meaningful, loosely structured genres, along with a summary of the information within each genre, users can potentially see at a glance which of these genres are relevant to their information needs. All these are more important if the user is making use of a mobile device with limited screen estate (such as smartphones). In fact recent surveys [1] have confirmed that the use of smartphones and tablets for internet access has increased multi-fold.

A similar problem already exists — news reading, where the huge number of news sources reporting on the same event may overwhelm a user. News aggregators (*e.g.*, Google News[3]) have evolved to tackle this problem by clustering reports on the same events together, while multi-document summarization systems present a short snippet of the highlights in the cluster of reports to

* Corresponding author.
E-mail addresses: chenyan@cse.buaa.edu.cn (Y. Chen), yolixs@buaa.edu.cn (X. Zhang), lizj@buaa.edu.cn (Z. Li), email@junping.ng (J.-P. Ng).

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

http://dx.doi.org/10.1016/j.neucom.2014.10.068
0925-2312/

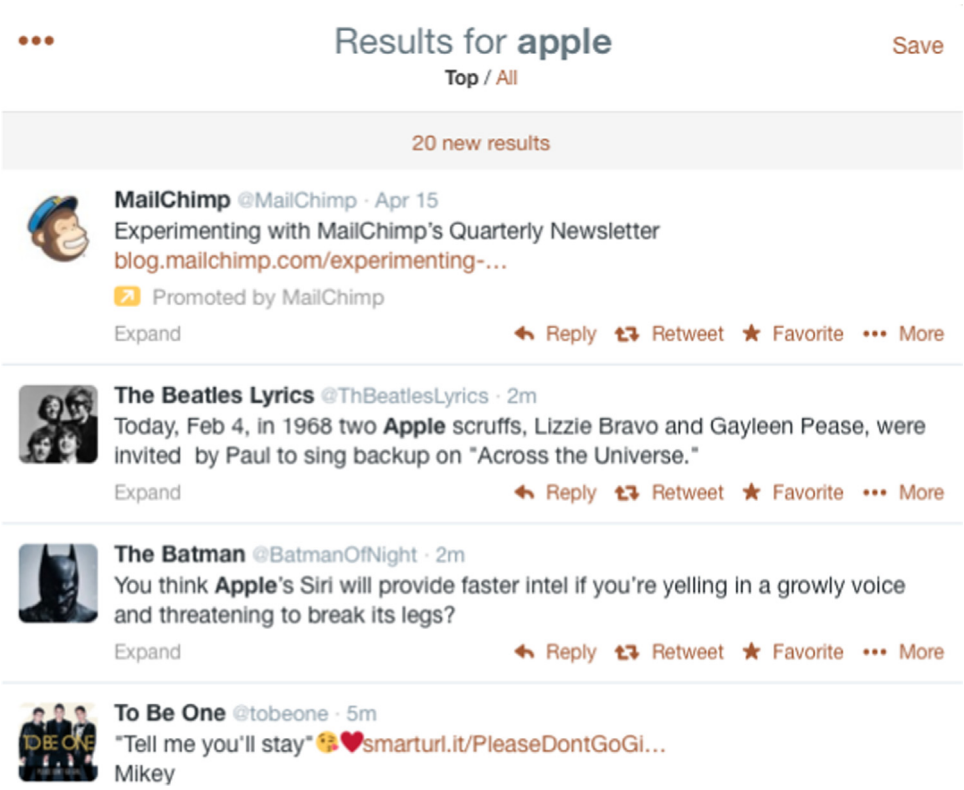


Fig. 1. Extract of search results for "apple" using Twitter's search function.

the user. We are proposing a similar solution here for microblog services. However the challenges involved in dealing with long news articles are different from those faced when working with the typically short text snippets in microblogs.

The main challenge is due to the length limitations imposed on microblog messages. These messages are typically short, consisting of no more than 140 characters. This data sparsity is a problem when trying to classify these messages into different genres. The lack of sufficient contextual information means that traditional similarity measures such as the use of word co-occurrences are ineffective [2]. A secondary problem has to do with the training corpora that are required by popular supervised machine learning-based solutions to achieve good performance. While there is a good number of such corpora for traditional domains like news-wire articles, building up and annotating similar corpora for microblog messages is laborious and time-consuming.

We tackle these two key challenges in this paper by making innovative use of search engines to automatically acquire extra documents and text to enrich the collection of microblog messages we are working with. In doing so, we are able to obtain more relevant text content to overcome the problem of data sparsity. It also allows us to adopt a semi-supervised methodology which requires far less training material than traditional supervised machine learners.

To classify microblog messages into one of the several genres, we propose the use of a semi-supervised probabilistic graphical model which combines textual content from microblog messages and automatically acquired content from search engine results. The model learns a suitable genre distribution with which we can assign individual microblog messages into the most likely genres. To generate a summary for each genre, we evaluate several graph-based summarization algorithms, built on the popular PageRank [3] and HITS [4] algorithms. We further modified these algorithms to take in additional text content from relevant web search results and show that this helps improve summarization performance.

The key contribution of this work is our novel proposal to incorporate the use of external resources to overcome the lack of contextual information inherent with microblog messages. These external resources are obtained automatically and efficiently, and we show that they help to (1) improve the performance of a semi-supervised classification model significantly, thereby reducing our reliance on large annotated datasets, and (2) improve the quality of summaries generated from microblog messages.

2. Related work

Our work overlaps two key areas of research: (1) topic classification and (2) microblog summarization. In this section, we explore related literature for each of them in turn. We also review existing work which adopt a similar "classify-then-summarize" approach to ours and share our aim of easing the information overload that users face today.

2.1. Topic classification

The task of topic classification of microblog messages (or what we refer to subsequently as *genre classification* in the rest of this paper) is to assign messages to one of the several pre-identified class labels. Topic classification is a fundamental task for many applications, including query disambiguation [5], location prediction [6] and hot topic tracking [7].

A common approach to topic classification is with the use of topic models. Of significance here is the work of Hong and Davison [8], where the use of latent Dirichlet allocation (LDA) [9] and author-topic models [10] is explored to automatically detect hidden topic structures within Twitter messages. Several variants of LDA have been proposed [11,12] and have been shown to be competitive for the classification of microblog messages. Although these LDA-based topic models work well generally, they are however hampered by the length limits imposed on most microblog messages.

The shorter pieces of text mean that similarity comparisons for good topic classification are less effective than they should have been. While our model for topic classification draws similarly on a probabilistic graphical model, we overcome similar deficiencies by augmenting the microblog messages with additional text snippets retrieved automatically from external search engines.

Lee et al. [13] adopted a different approach, choosing instead to construct word vectors with term frequencies and inverse document frequencies (TF-IDF). These vectors are paired with a naive Bayesian multinomial classifier for topic classification. Zubiaga et al. [14] adopted a similar approach, but with the use of support vector machines (SVM). Going beyond the use of lexical frequencies, Sriram et al. [15] proposed the use of a small set of domain-specific features extracted from user profiles to represent short messages. Their methodology, however, requires extensive pre-processing for effective feature analysis. This has an adverse impact on performance and is less useful should real-time classification of a large number of microblog messages be required.

All of these described approaches depend on the availability of large amounts of labeled training data to a large extent. Preparing such datasets however is laborious and time-consuming. To break the current impasse between effort and effectiveness, we tap on additional information cues from automatically acquired resources and fuse them seamlessly with our proposed probabilistic model. In doing so, our model requires only a fraction of the training data, compared to traditional supervised methods.

2.2. Microblog summarization

Summarization is a well-studied problem and has been well researched over the past many years. Good overviews of the advances in summarization are given in [16,17]. We will not survey the entire field here, but instead zoom-in specifically onto work relating to microblog summarization.

Given the wealth of prior art looking at traditional text summarization, it is understandable that many of these techniques and methodologies were applied to microblog summarization. The use of frequency measures to compute the saliency of text [18] for example is commonly adapted and explored [19–21]. Another popular measure of word relevance — the Kullback–Leibler divergence (KLD) — which was used to great effect in Ng et al. [22] had also been transplanted and shown to be useful for microblog summarization [23]. Integer linear programming (ILP) has also been shown to give rise to effective summarization systems [24,25]. Takamura et al. [26] adopted this approach and showed that it is also effective for microblog summarization.

Sharifi [27,28] proposed a phrase reinforcement summarization algorithm. This is a graph-based approach which leverages on trending phrases specified by users in microblog messages. By taking advantage of the link structure among words, this approach is shown to achieve significant improvements in the quality of the summaries generated. Harabagiu and Hickl [29] also adopted a graph-based approach based on events detected from microblog messages. Summaries are then generated by scoring each message based on the computed relevance of these events. Following from these good results, we are convinced of the benefits of graph-based approaches. Therefore in this work we make use of both the PageRank [3] and HITS [4] algorithms, combined with the use of similarity measures LexRank [30] and TextRank [31], for summarization.

2.3. Classification with summarization

Several researchers have worked on the same vein of work we present in this paper, *i.e.*, breaking up a collection of microblog messages into smaller groups or clusters, before producing a summary for each cluster.

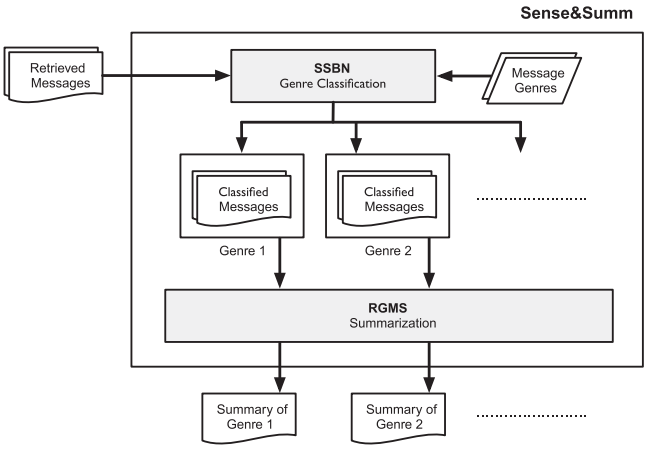


Fig. 2. Overview of the architecture of `Sense&Summ`.

In particular, the work of Olariu [32] is closely related to ours. He had also proposed grouping microblog messages into related clusters, before summarizing each grouping of messages separately. A key difference between our work and his however is our novel use of external resources in the form of search engine results to overcome the shortage of sufficient contextual information inherent to microblog messages.

Xu et al. [33], Chakrabarti and Punera [34] and Duan et al. [35] similarly generate summaries for identified sub-topics in a collection of microblog messages. However their sub-topics are defined based on the clustering and segregation of messages along a timeline. While this is useful to help users track the evolution of and changes to an event as it progresses, it is less useful when dealing with a more general corpus, where microblog messages from different senses of a query word are mixed together. Our proposal is geared towards the latter.

3. General framework

An overview of our proposed genre classification and summarization pipeline, `Sense&Summ`, is illustrated in Fig. 2. We assume that the input to the system is a set of microblog messages that has been retrieved using standard retrieval techniques with a user-supplied keyword. This could be results returned by existing search interfaces of microblog platforms for example.

A pre-processing step (not shown in the figure) is performed for each microblog message. This step is important because it helps us to alleviate problems associated with the informal language used in microblog messages. Language used in composing these messages tends to be more casual, and the character limit imposed by microblog services accentuates this problem. To maximize the information that they can convey within the tight character limits, users often make use of abbreviations. For example instead of "*tomorrow*", they may just use an abbreviated form of the word "*tmr*". We make use of the approach used in [36,37] to correct or normalize this use of informal language. We construct a lexicon containing possible abbreviations from existing resources including Twitternary[4] and Twitterforteachers.[5] This lexicon consists of 727 unique words. Given an input microblog message, individual word tokens in this input message are compared against our constructed lexicon. Matching word tokens are then normalized into the corresponding formal equivalent.

[4] http://www.twittonary.com/
[5] http://twitterforteachers.wetpaint.com/page/Twitter+Dictionary

As seen in the figure, there are two key phases in Sense&Summ:

1. *Genre classification*: Sense&Summ first attempts to classify each microblog message into one of the several specific genres. As explained earlier, our goal is to enhance the presentation of retrieved microblog messages to make it easier for users to interpret the retrieval results.
   The set of genres to classify each message can potentially be dynamically generated, or manually pre-identified. In this work, we have decided to identify the genres into which messages can be classified into. The motivation behind this decision is to standardize the presentation of the eventual retrieval results to users. Users will be shown a summary for each of these genres, and we believe that a constant set of genres for all search requests helps improve user familiarity. In future work, it will definitely be worth investigating the use of a variable set of genres which is dynamically and automatically generated given a specific query.
   Our proposed approach using a semi-supervised probabilistic graphical model (PGM), as well as our novel solution to overcome the lack of contextual information inherent with the short microblog messages we are dealing with, is explained in detail in Section 3.1.
2. *Genre summarization*: Having classified the retrieved messages into one of the several genres, Sense&Summ generates a summary for each of these genres. These summaries give users a high-level overview of the nature and content of the retrieved microblog messages, broken down along the identified genres. We believe that this will help users get a better understanding of the possible facades of the retrieved results. This can then guide subsequent searches to help users zoom-in on the information that they are seeking for example. More details of the graph-based summarization solution we propose will be given in Section 3.2.

3.1. Semi-supervised PGM genre classification

Genre classification is an important phase in Sense&Summ. As reviewed earlier, existing work targeted at the processing of traditional, long text documents (*e.g.* newswire articles) perform well. However the use of informal language prevalent in microblog messages and the character limit imposed on these messages are significant challenges which limit the applicability of these methodologies to the classification of microblog messages [37,38]. Knights et al. [39] also note that models trained to work on microblog messages suffer from "topic drift". Topic drift occurs when the underlying topics being discussed in a collection of messages change over time. Models learnt in the past become irrelevant, and thus new, updated training data is required to keep the models updated.

In Sense&Summ, a probabilistic graphical model is used to model the genre distribution of microblog messages and classify the genre to which each microblog message belongs to. To overcome the problems highlighted earlier, our semi-supervised Bayesian network (SSBN) model automatically acquires relevant external resources to reduce the amount of training data that is required. Specifically, we propose retrieving a set of document snippets from existing web search engines (*e.g.*, Google[6]), and augmenting these to the original microblog messages to build up a more accurate and dynamic model for genre classification. The motivation for this is to overcome the length limitations of microblog messages with the additional text content. As we will show later, this allows us to significantly reduce the amount of training data required by our approach. It also has the advantage of helping to keep our model up-to-date with the incorporation of additional fresh data.

[6] http://www.google.com

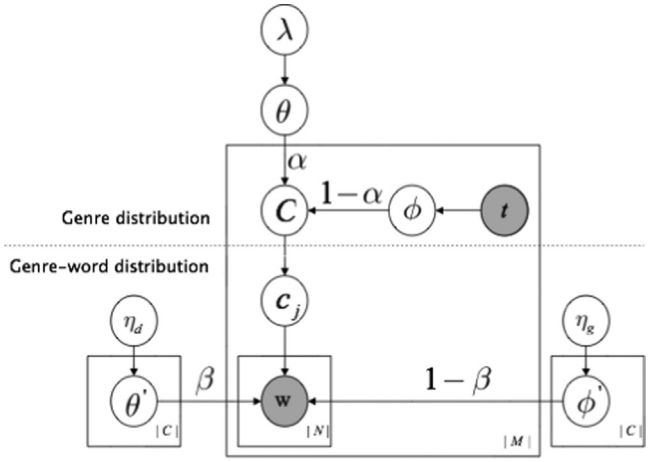


Fig. 3. Plate notation describing SSBN, our proposed PGM for genre classification.

Tapping on external resources: We start from a query phrase, and a set of microblog messages retrieved with the phrase. For each pre-identified genre, we create a search phrase by concatenating the query phrase with the genre name. Suppose the query phrase is "Apple", and we have two genres named "Sports" and "Science". We can then obtain two search phrases "Apple Sports" and "Apple Science".

We query the Google search engine with each of the search phrases. For each query, we retrieve the snippets of the top 20 document matches from Google. Basic pre-processing is performed on these snippets, such as the removal of stop words. Having obtained these sets of snippets, one for each genre, we can then use them to compute a word distribution for every genre. These distributions will be used by the SSBN model which we will describe next.

Construction of the SSBN model: It is best to illustrate SSBN with the use of the plate notation, as is done in Fig. 3. The variables used in the figure are explained in Table 1. The model attempts to model two different distributions: (1) a genre distribution and (2) a word distribution within a genre. For brevity, we will also refer to the latter as a genre-word distribution:

1. *Genre distribution*: SSBN captures two different types of genre distributions. Let θ denotes the genre distribution obtained from the input collection of messages, M. This is a weight vector representing the weight of each genre. Similarly, let ϕ denote the genre distribution for the additional information we obtain using external search engines. The genre distribution for the combined collection is then assumed to be a linear combination of θ and ϕ. Parameter α is employed as a weight to adjust the contributions of each source of information (*i.e.*, original microblog message, or additional information pulled from search engine). λ is used to denote the contribution of unlabeled data when generating the genre distribution for M.
2. *Genre-word distribution*: There are also two facets to the genre-word distribution: θ' denotes the distribution of different words over different genres in the input collection of messages, which is a $|C| \times |N|$ matrix. Here, $|C|$ is the number of genres, and $|N|$ is the number of words in the collection. Similarly, ϕ' denotes the genre-word distribution from information derived from external search engines. The genre-word distribution for the combined collection is again assumed to be a linear combination of θ' and ϕ', weighted by the parameter β.

SSBN formally denotes the probability of a message m falling into a genre c as

$$P(c|m) = \frac{P(c)P(m|c)}{\sum_c P(c)P(m|c)}, \quad (1)$$

Table 1
Important notations used in describing our probabilistic graphical model SSBN.

Variable	Description
θ	The vector indicating weights for each genre in the input collection of messages
ϕ	The vector indicating weights for each genre for a specific message
θ', ϕ'	The $\|C\| \times \|N\|$ matrix indicating genre-word distribution
λ	The contribution of unlabeled data to prior probability
α	The contribution of prior knowledge from θ
$1-\alpha$	The contribution of prior knowledge from ϕ
β	The contribution of likelihood probability from θ'
$1-\beta$	The contribution of likelihood probability from ϕ'
η_d, η_g	Hyperparameters and priors of Dirichlet distributions
C	The genre vector
c_j	The jth genre
M	The input collection of messages
m	The message being classified
N	The vocabulary from the input collection of messages
w	Word from message being classified
y	The genre of a specified message

where $P(c)$ is the prior probability of a genre in the collection of messages. To compute $P(m|c)$, we can assume that the presence of a word, w, is independent of the presence of any other word in m. In this case we can compute it as

$$P(m|c) = \prod_{w \in m} P(w|c) \tag{2}$$

Parameter inference: To infer the latent parameters in SSBN, we make use of the expectation-maximization (EM) algorithm. In the expectation step, the distributions θ, ϕ, $\hat{\theta'}_{c_j}^{w_k}$ and $\hat{\phi'}_{c_j}^{w_k}$ are estimated. Besides the labeled data and external resource, the parameter estimation step also taps on unlabeled data. Initially we assign genre labels to unlabeled data with a uniform distribution, *i.e.*, the probability is $1/|C|$ for each genre. In subsequent iterations, labels of the unlabeled data and the statistical model are alternatively updated and reinforced until they converge.

Estimating θ: θ represents the probability of each genre in the input microblog messages. It is proportional to the expected number of messages that is assigned to each genre:

$$\hat{\theta}_{c_j} \equiv P(c_j|\hat{\theta}) = \frac{1 + \sum_{i=1}^{|M|} \Lambda(i) P(y_i = c_j|m_i)}{|C| + |M^l| + \lambda |M^u|} \tag{3}$$

The input microblog messages consist of both labeled messages, M^l, and unlabeled ones, M^u. The function $\Lambda(i)$ defined in Eq. (4) controls their respective contributions to the estimation of the genre probability:

$$\Lambda(i) = \begin{cases} \lambda & \text{if } m_i \in M^u; \\ 1 & \text{if } m_i \in M^l. \end{cases} \tag{4}$$

The parameter λ can range from 0 to 1. When λ is close to 1, unlabeled data is taken to be as important as labeled data. Conversely when λ approaches 0, SSBN behaves more like a supervised learning algorithm. In earlier work [36], it has been found that setting the value of λ to within $[0.3, 0.5]$ gives optimal performance.

Estimating ϕ: ϕ denotes the prior probability distributions of the various genres over the external resources obtained from Google search results. The prior probability of a genre c_j for a query q depends on the relationship between q and the pre-defined genre names:

$$\hat{\phi}_{c_j} \equiv P(c_j|\hat{\phi}) = \frac{\frac{1}{NGD(q, c_j) + \mu}}{\sum_{j=1}^{|C|} \frac{1}{NGD(q, c_j) + \mu}} \tag{5}$$

where μ is a smoothing factor and $NGD(q, c_j)$ is the Normalized Google Distance [40], which is employed to calculate the distance between q and c_j.

Estimating θ' and ϕ': θ' and ϕ' denote the genre-word distributions over the input microblog messages and the search results from Google respectively. Both of them are $|C| \times |N|$ matrices. They can be estimated using the following formulae:

$$\hat{\theta'}_{c_j}^{w_k} \equiv P(w_k|c_j, \hat{\theta}') = \frac{n_{dc_j}^{w_k} + \eta_d}{\sum_{r'=1}^{|N|} n_{dc_j}^{w_{r'}} + |N|\eta_d} \tag{6}$$

$$\hat{\phi'}_{c_j}^{w_k} \equiv P(w_k|c_j, \hat{\phi}') = \frac{n_{gc_j}^{w_k} + \eta_g}{\sum_{s'=1}^{|N|} n_{gc_j}^{w_{s'}} + |N|\eta_g} \tag{7}$$

where $n_{dc_j}^{w_k}$ and $n_{gc_j}^{w_k}$ are the number of times that the word w_k has occurred in the genre c_j within the input microblog messages and the search results from Google respectively. η_d and η_g are hyperparameters that are assigned to small, non-zero values for the purpose of smoothing.

The genre assigned to a given message, m_i, can be computed with the maximum likelihood estimator:

$$\begin{aligned} y_i &= \arg\max_{c_j} P(c_j|m_i, \hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}') \\ &= \frac{P(c_j|\hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}') P(m_i|c_j, \hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}')}{P(m_i|\hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}')} \end{aligned} \tag{8}$$

where $P(m_i|\hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}')$ is formally defined as follows:

$$P(m_i|\hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}') = \sum_{c_j} P(c_j|\hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}') P(m_i|c_j, \hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}') \tag{9}$$

The prior probability of genre c_j is obtained as a linear combination of the estimates from both the input microblog messages and the external search results from Google:

$$P(c_j|\hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}') = P(c_j|\hat{\theta}, \hat{\phi}) = \alpha P(c_j|\hat{\theta}) + (1-\alpha) P(c_j|\hat{\phi}) \tag{10}$$

α here is a trade-off parameter to balance the contributions between $\hat{\theta}$ and $\hat{\phi}$. Finally, the maximum likelihood probability for the each message m_i can be derived as

$$\begin{aligned} P(m_i|c_j, \hat{\theta}, \hat{\phi}, \hat{\theta}', \hat{\phi}') &= P(m_i|c_j, \hat{\theta}', \hat{\phi}') = \prod_{k=1}^{|m_i|} P(w_k|c_j, \hat{\theta}', \hat{\phi}') \\ &= \prod_{k=1}^{|m_i|} \left\{ \beta P(w_k|c_j, \hat{\theta}') + (1-\beta) P(w_k|c_j, \hat{\phi}') \right\} \end{aligned} \tag{11}$$

β plays a similar role here as α did earlier in Eq. (10). It is used to control the relative contributions of the genre-word distributions from both the input microblog messages and the external search results from Google.

3.2. Resource-reinforced graph-based microblog summarization

To generate a summary of the microblog messages attributed to each genre, we propose a novel way to tap on external resources to improve the quality and richness of the content of the generated summary. As our proposal builds on top of a graph-based summarization approach, we christened it as Resource-Reinforced Graph-based Microblog Summarization (RGMS).

The length limit imposed on each microblog message means that there is a limited amount of contextual information that can be obtained from each message. Messages also tend to be phrased casually with abundant informal language use. As we are focusing on extractive summarization, these two factors limit the quality of the summaries that can be generated. To overcome these, we augment the microblog messages with search results from a web search engine as was done earlier in SSBN. We generate automatically a set of keywords from the original microblog messages to be summarized, and use these as a query to retrieve relevant and related documents. These documents are then combined with the microblog messages before extractive summarization is performed. We believe that the motivation for this process is compelling and just, as the additional content from the search results:

1. provides more data to improve the statistical measures used to estimate the saliency of the microblog messages,
2. helps enrich the text we have available to generate a summary with,
3. lends more background and circumstantial information to the existing microblog messages.

RGMS is a two-stage approach. In the first stage, as explained earlier, text content found within the microblog messages to be summarized is augmented with search results from a web search engine (*i.e.*, Google). Then in the next stage, we implement a graph-based summarization system. These two stages are detailed in Fig. 4:

*Stage*1 — *Resource-reinforcement*: Starting from a set of microblog messages (all of which belong to the same genre), a set of keywords is first identified and extracted. These keywords are then composed into a single search engine query and passed to the Google search engine. The top five results from the search are then downloaded, and the respective text content extracted. The text content is broken down into individual sentences and added to the set of microblog messages.

An important question here is what we can use to query the search engine with. An intuitive choice is to use each microblog message as a query. However this is not ideal. We argue that since the purpose of tapping on the search engine is to help us obtain more contextual information for microblog messages, using each microblog message as input to the search engine will likely lead to overly specific search results. Further, it is not computationally feasible to send each microblog message to the search engine. The large number of microblog messages will mean that this process will take too long a time. In our experiments that we describe in the next section, there can be hundreds of messages per genre. As such, we propose instead to extract a set of keywords from the input microblog messages. These keywords will then be appended together and used as a query to the search engine.

To automatically select a set of keywords which can accurately represent the key ideas described by a set of microblog messages, we adopt the work of Cataldi et al. [41]. In their work, they proposed an effective algorithm making use of the notions of the authoritativeness of both users and the microblog messages they post. These are referred to as user authority and message authority respectively.

Let $U_t = \{u_1, u_2, \ldots, u_i, \ldots, u_m\}$ denote users who post microblog messages about a particular topic t. We define user authority as a measure of the number of followers each user has. Therefore we can compute the user authority for $u_i \in U_t$ as follows:

$$auth_t(u_i) = \frac{f_{u_i} + 1}{\sum_j f_{u_j}}, \tag{12}$$

where f_{u_i} is the total number of u_i's followers.

Let the set of microblog messages posted by users in U_t be defined as $Tw_t = \{tw_1, tw_2, \ldots, tw_i, \ldots, tw_n\}$. Note that all the message in Tw_t discuss a common topic t. For a microblog message tw_i, we define its message authority by leveraging on user authority, that is

$$auth_t(tw_i) = \log(1 + auth_t(u_{tw_i})), \tag{13}$$

where $auth_t(u_{tw_i})$ is the user authority of the author of tw_i.

Let $W_t = \{w_1, \ldots, w_i, \ldots, w_r\}$ be the set of words appearing in topic t. For each word w_i in topic t, we compute its weight $Weight_t(w_i)$ through the influence of tweets that it appears in, as

$$Weight_t(w_i) = \frac{\sum_{\forall tw_j \in Tw_t \wedge w_i \in tw_j} auth_t(tw_j)}{\sum_{\forall w \in W_t} \sum_{\forall tw \in Tw_t \wedge w \in tw} auth_t(tw)}. \tag{14}$$

With these keyword candidates, we filter out words which do not appear at least three times in the whole dataset. This is to ensure that the eventual selected keywords are sufficiently represented. Finally, we obtain a ranked list of keyword candidates based on the values of $Weight_t(w_i)$. The top n (here, $n=6$) keyword candidates are regarded as keywords for this specific topic. These keywords are then appended to form a query string and submitted to a web search engine (Google in our case). The text content from the top m (here, $m=5$) search results is next extracted and broken down into sentences. Finally, these sentences are added to the collection of input microblog messages.

*Stage*2 — *Graph-based summarization*: Typical extractive summarization approaches such as [42] assign a measure of saliency to each individual sentence in the input collection to be summarized. This saliency measure is used to obtain a ranked list of the sentences. Subsequently the top k sentences are extracted to compose the final summary. There is a whole myriad of methods with which the saliency measure can be computed. State-of-the-art methodologies include the use of frequency-based measures [22,43] and graph-based approaches [30,31] among others.

We draw on the work of [44] and elect to implement graph-based summarizers, which have been shown to turn in state-of-the-art performance. We make use of the popular graph algorithms

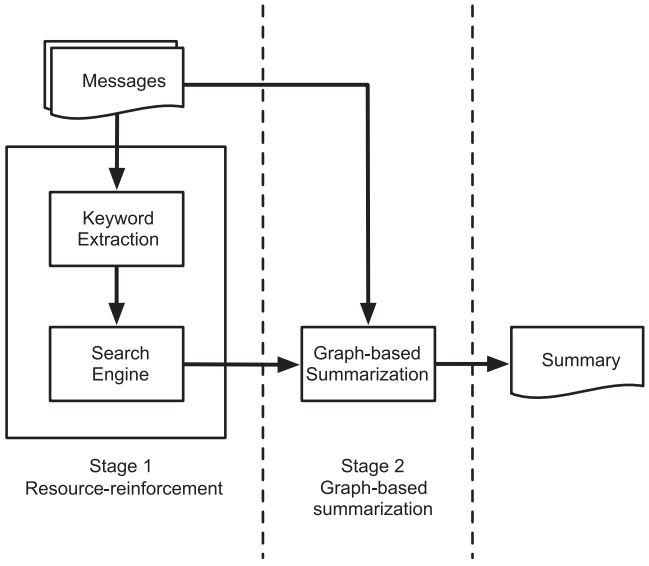


Fig. 4. Proposed two-stage resource-reinforced graph-based microblog summarization.

Table 2
Pre-identified set of genres which retrieved microblog messages are assigned to for our two datasets.

Datasets	
D_{Sina}	D_{Tweet}
Sports	Sports
Politics	Politics
Science & Technology	Science & Technology
Games	Entertainment
Movie	Business
Music	Education
Others	Others

PageRank [3] and HITS [4]. In this representation, each sentence in the input collection is represented as a node in a graph. Edges linking up pairs of nodes are weighted by the similarity scores of the nodes. To compute these similarity scores, we compare two well-known and popular similarity measures LexRank [30] and TextRank [31]. The final summary is then generated by selecting the highest scoring sentences until a 50-word limit is reached.

4. Results and analysis

We validate our proposal via two sets of evaluations: (1) classification of tweets into the various genres, and (2) summarization performance for tweets in each genre. We will first describe the datasets that we use for our evaluation experiments, before explaining the results that we have obtained.

4.1. Datasets

We assembled two real-world datasets from two popular microblog platforms, Sina Weibo and Twitter. In this section, we will explain how these datasets were built up, as well as highlight the differences between them.

Sina Weibo Dataset (D_{Sina}). Sina Weibo publishes a list of trending topics. We select 23 topics from this list and crawled a collection of messages related to these topics using the Sina Weibo API. To assemble the ground truth necessary to evaluate genre classification, we invited 15 people (not including the authors of this paper) of varying backgrounds and training to manually assign each message to one of the seven genres explained in Table 2. The 15 manual annotators are divided into three teams. Each team of five annotators is tasked to annotate the entire set of messages. A majority-voting scheme was then used to compare and aggregate the annotations of each team, such that each message is only assigned to one of the seven genres. In cases where a message is assigned to three different genres by each of the three teams, the teams were requested to discuss and arbitrate their decisions until a consensus is reached. Finally, messages in the "*Others*" genre are discarded to give a total of 15,811 messages.

We note that building up D_{Sina} is a laborious process, involving a large team of annotators and a total of nearly 120 man-hours. While we could repeat the same process for the subsequent dataset described below, we are also keen to explore how this endeavor can be made easier. We identified that the most time-consuming task of the dataset collection is the manual annotation of messages to their corresponding genres. Therefore in the next dataset, we experimented with an approximate and less cost-intensive approach to this annotation.

Twitter Dataset (D_{Tweet}). We first identified 10 general hot topics from Google Trends,[7] and searched through the `Tweets2011` dataset from the Microblog track of the Text Retrieval Conference (TREC) [45] . Then for each of these topics, we manually identified related sub-topics based on the content of relevant messages found within `Tweets2011`. Each of these sub-topics are subsequently assigned to one of the seven genres described earlier in Table 2. A subset of these topics and their related sub-topics are shown in Table 3. The sub-topics and the genres they are attributed to are also shown in the same table.

To illustrate, let us consider the topic "*Apple*". Sub-topics that were identified include "*stock*" and "*ipad*". By combining the main topic title (*i.e.*, "*Apple*") with a sub-topic title (*i.e.*, "*stock*"), we get a query that is then used to retrieve relevant messages from the `Tweets2011` corpus. We made an assumption that messages retrieved in this fashion falls under the same genre as its parent sub-topic. So this means that messages retrieved with the query "*Apple stock*" belongs to the genre "*Business*" while messages from the query "*Apple ipad*" belongs to "*Science & Technology*". In this way we are able to avoid the laborious annotation process required to assign messages to their respective genres and easily built up a dataset of approximately 16,000 messages.

Table 4 illustrates the distribution of messages over the pre-identified genres (excluding "*Others*") for each of the two datasets D_{Sina} and D_{Tweet}. The datasets are well-balanced, containing more or less the same amount of messages in each genre.

To prepare the ground truth for summarization evaluation, we also invited an annotator (who is not an author of this paper, and did not participate in the earlier genre classification annotation) to write summaries for every genre in every topic for the two datasets. The annotator selected a representative, ordered set of messages from the input microblog messages in each case to form a 50-word summary.

4.2. Evaluating genre classification

The first set of evaluation experiments we will describe looks at how well our system does in assigning microblog messages to the pre-identified genres.

Evaluation metrics: We utilize several widely used performance metrics to evaluate genre classification, including average accuracy, precision, recall, and F_1 [46,21]. Average accuracy evaluates the mean effectiveness of our classifier for each genre. Precision is the proportion of microblog messages that are relevant to the search, while recall is the percentage of the relevant messages that are successfully retrieved. Finally the F_1 measure is the harmonic mean of both precision and recall. We also provide the macro-averaged and micro-averaged results to give better insights into our results. The macro-averaged results assign an equal weight to each genre and give a good picture of the overall performance of our classifier. The micro-averaged results give a sense of the performance of our classifier over the entire message collection, and tend to allocate more weight to genres which contain more messages than others.

Classification performance: Tables 5 and 6 show the results of our proposed SSBN genre classification model on D_{Sina} and D_{Tweet} respectively. Parameters used for our model are $\alpha=0.9$, $\beta=0.9$, $\lambda=0.3$ for D_{Sina}, and $\alpha=0.5$, $\beta=0.9$, $\lambda=0.4$ for D_{Tweet}. In both cases, 5% of the dataset is used as training data and the remaining 95% is used as testing data.

Despite only using a fraction of both datasets for training, our proposed scheme achieved good precision, recall and F_1 scores. For D_{Sina}, remarkable performances of more than 0.8 is obtained across all evaluation measures. Our classifier also performs very well on D_{Tweet}, although so less when compared to that on D_{Sina}. Precision fares better generally across most genres, and excellent F_1 scores were obtained for up to half the genres, including "*Sports*", "*Science & Technology*" and "*Politics*". D_{Tweet} is a slightly noisier dataset than D_{Sina}, so this relative difference in performance is not un-expected.

[7] http://www.google.com/trends/

Table 3
Subset of topics and sub-topics identified from the `Tweets2001` dataset, together with the genres each sub-topic is assigned to.

Topic	Sub-topic	Assigned genre
android	app	Science & Tech
	phone	Science & Tech
apple	ipad	Science & Tech
	stock	Business
nba	celtics	Sports
	kobe	Sports

Table 4
Distribution of messages over pre-identified genres for our two assembled datasets. The "#" column denotes the number of messages for a particular genre.

D_{Sina}		D_{Tweet}	
Genres	**#**	**Genres**	**#**
Sports	2602	Sports	2720
Politics	2654	Politics	2937
Science & Tech	2647	Science & Tech	2827
Games	2605	Entertainment	2816
Movies	2694	Business	2912
Music	2609	Education	2723
Total	15,811	**Total**	16,935

Table 5
Genre classification results on D_{Sina}.

Genre	Precision	Recall	F_1
Sports	**0.9318**	0.8747	**0.9023**
Politics	0.8661	**0.9324**	0.8980
Science & Technology	0.8688	0.8323	0.8502
Games	0.8090	0.9283	0.8646
Movies	0.8848	0.8207	0.8515
Music	0.8819	0.8699	0.8759
Micro-averaged	0.8798	0.8798	0.8798
Macro-averaged	0.8737	0.8764	0.8738

The bolded fields denote the best performance for a given column in a set of comparisons.

In fact, surveying the results obtained, we note that our classifier is sufficiently robust to the noise introduced into D_{Tweet} by the automatic annotation that was performed.

Looking at the genres in D_{Tweet} for which we do not perform as well (*i.e.*, "*Entertainment*", "*Business*" and "*Education*"), we believe that this could be due to the fact that these genres are more general and diverse than say the "*Music*" and "*Movies*" genres used in the D_{Sina} dataset. We may not have identified a sufficiently complete set of sub-topics for these genres, and the varied nature of the sub-topics may have caused the under-performance. We hope to study this limitation in future work and see how we can better address this.

To get a notion of how well our proposed SSBN model does with respect to the state-of-the-art, we also ran an experiment to compare it against other classifiers surveyed in [47], including: (1) support vector machines (SVM) [48], (2) Naive Bayesian (NB) [49], (3) K-Nearest-Neighbors (KNN) [50], (4) Rocchio [51], (5) Labeled LDA (L-LDA) [11], (6) Transductive SVM (Trans-SVM) [52], and (7) Semi-Naive Bayesian (Semi-NB) [53]. The input to these classifiers includes only the microblog messages to be classified, and not the external data that was acquired for SSBN.

Tables 7 and 8 show the results obtained for each of the aforementioned approaches, vis-a-vis our proposed approach. For each of these approaches, all required parameters are empirically tuned and the best obtained performances are shown in the tables. The parameters used for our SSBN model are $\alpha=0.9$, $\beta=0.9$, $\lambda=0.3$ for D_{Sina}, and $\alpha=0.5$, $\beta=0.9$, $\lambda=0.4$ for D_{Tweet}.

Table 6
Genre classification results on D_{Tweet}.

Genre	Precision	Recall	F_1
Sports	**0.9322**	0.9483	**0.9402**
Entertainment	0.9000	0.5625	0.6923
Business	0.8043	0.5323	0.6382
Science & Technology	0.6937	**0.9801**	0.8124
Politics	0.9096	0.9640	0.9360
Education	0.5000	0.5519	0.5165
Micro-averaged	0.7979	0.7979	0.7979
Macro-averaged	0.7934	0.6043	0.6128

The bolded fields denote the best performance for a given column in a set of comparisons.

Table 7
Comparison of genre classification performance with various state-of-the-art classifiers on D_{Sina}. 90% and 5% of the dataset are used as training data when comparing SSBN with other supervised and un-supervised classifiers respectively.

Classifier	Micro-averaged			Macro-averaged		
	Precision	Recall	F_1	Precision	Recall	F_1
Supervised						
SSBN (90%)	**0.9020**	**0.9020**	**0.9020**	0.8976	**0.9045**	**0.9004**
SVM	0.8991	0.8991	0.8991	**0.9017**	0.8971	0.8991
NB	0.9015	0.9015	0.9015	0.8990	0.9024	0.9003
KNN	0.8565	0.8565	0.8565	0.8589	0.8486	0.8526
Rocchio	0.8803	0.8802	0.8802	0.8769	0.8832	0.8781
L-LDA	0.8905	0.8905	0.8905	0.8876	0.8989	0.8932
Un-supervised						
SSBN (5%)	**0.8798**	**0.8798**	**0.8798**	**0.8737**	**0.8764**	**0.8738**
Trans-SVM	0.8084	0.8084	0.8084	0.8049	0.8085	0.8052
Semi-NB	0.8198	0.8198	0.8198	0.8225	0.8217	0.8204

The bolded fields denote the best performance for a given column in a set of comparisons.

The same set of underlying features is used for all approaches for a fair comparison. Further, we distinguish between supervised approaches and un-supervised approaches. When comparing against supervised approaches, we make use of 90% of the size of the dataset as training data, and the remaining 10% as testing data. When comparing against un-supervised approaches, we make use of only 5% of the original dataset for training, and the remaining 95% for testing.

Looking at both the micro- and macro-averaged results, we see that SSBN out-performs all other supervised approaches for both D_{Sina} and D_{Tweet}. SVM turns in a higher precision score on D_{Tweet}, however due to lower recall values, SSBN still out-performs the SVM classifier in terms of the F_1 measure.

Making use of just a small amount of training data, SSBN is able to out-perform the two un-supervised approaches for both datasets. This significant result underscores the robustness of our approach, and in particular, how little training data it requires to achieve good performance. As we have explained earlier in describing our dataset D_{Sina}, building high-quality datasets takes a lot of effort and labeled data is more of an exception than the norm. Our proposed approach is valuable in that it allows us to make do with just a small amount of labeled data. In fact looking at the results, using just 5% of the dataset for training, SSBN (5%) even out-performs supervised approaches such as KNN on both datasets, which employs as much as 90% of the dataset for training.

We believe that this out-performance is due to the appropriate incorporation of external knowledge sources, as we have done with the search results from Google. This extra information

Table 8
Comparison of genre classification performance with various state-of-the-art classifiers on D_{Tweet}. 90% and 5% of the dataset are used as training data when comparing SSBN with other supervised and un-supervised classifiers respectively.

Classifier	Micro-averaged			Macro-averaged		
	Precision	Recall	F_1	Precision	Recall	F_1
Supervised						
SSBN (90%)	**0.8875**	**0.8875**	**0.8875**	0.8282	0.7627	0.7845
SVM	0.8670	0.8670	0.8670	0.8768	0.7611	**0.7860**
NB	0.8696	0.8722	0.8722	**0.8879**	0.7329	0.7587
KNN	0.7268	0.7268	0.7268	0.6721	0.6471	0.6516
Rocchio	0.8204	0.8180	0.8192	0.7361	**0.8384**	0.7605
L-LDA	0.8605	0.8605	0.8605	0.8467	0.7223	0.7532
Un-supervised						
SSBN (5%)	**0.7979**	**0.7979**	**0.7979**	**0.7934**	**0.6043**	**0.6128**
Trans-SVM	0.6707	0.6707	0.6707	0.6602	0.5108	0.4491
Semi-NB	0.7156	0.7156	0.7156	0.7308	0.5653	0.549

The bolded fields denote the best performance for a given column in a set of comparisons.

enriches the contextual information we have surrounding each microblog message and allows us to make a more informed decision as to its correct genre.

Though the acquisition of external resources incurs an extra computation step, this does not impose a significant burden during genre classification. For a given query and input set of microblog messages, assuming that we have $|C|$ genres, only a constant number of $|C|$ searches are required in this extra search step. This is regardless of the size of the input set of microblog messages. Each search consumes just a small amount of time as we only retrieve the top 20 snippets returned by the Google search engine. We believe that this slight increase in cost is worthwhile considering the gains that can be obtained in the accuracy of genre classification.

Value of external resources: To validate the usefulness of the use of external resources, we repeated our experiments with SSBN, this time without the use of external resources. This can be achieved by setting the value of $\alpha=1$ in SSBN. Recall that α is a weight which allows us to specify the contributions of both the original microblog messages and external resources (see Fig. 3). When $\alpha=1$, the contributions from external resources are not considered.

Table 9 shows the accuracy results that are obtained. We ran the experiment with two configurations of SSBN, one using 90% of the dataset as training data and the other using 5%. The suffix '-NoExt' denotes the use of SSBN without external resources (*i.e.*, $\alpha=1$). Otherwise, the values for α are 0.9 and 0.5 respectively for D_{Sina} and D_{Tweet}, as was used for earlier experiments.

It can be seen that the use of external resources helps us to boost the accuracy of SSBN-NoExt. Without external resources, SSBN-NoExt is not better than the other classifiers shown earlier in Tables 7 and 8.

The use of external resources gives a much larger performance increase when less training data is available (*i.e.*, SSBN (5%) benefits more than SSBN (90%)). This is intuitive: SSBN (90%) has access to more training data which allows it to better model the underlying probability distributions. This means that it does not require a lot of additional information to perform well. SSBN (5%), on the other hand, has access to much less information and thus is not able to generalize as well. Having access to additional information from external resources thus results in a larger relative gain in performance. However labeled data is an expensive resource, and these results show that external resources (which are much cheaper to obtain) are a very viable alternative.

We also note that the improvement from using external resources is more significant in the case of D_{Tweet}. This could be due to the language differences between D_{Sina} and D_{Tweet}. The former consists of Chinese tweets while the latter consists of English tweets. Chinese is arguably a more compact language and more information can be represented in 140 characters, as compared to English [54]. The use of external resources is thus more beneficial for D_{Tweet} as it contributes more contextual information to the limited content found in English tweets.

Table 9
Accuracy of SSBN with and without the use of external resources for D_{Sina} and D_{Tweet}. Results for both the use of 90% and 5% of the dataset as training data are shown. The suffix '-NoExt' denotes the use of SSBN without external resources.

Configuration	Accuracy (%)	
	D_{Sina}	D_{Tweet}
SSBN (90%)	**0.9020**	**0.8875**
SSBN-NoExt (90%)	0.9009	0.8507
SSBN (5%)	**0.8978**	**0.7979**
SSBN-NoExt (5%)	0.8741	0.6865

The bolded fields denote the best performance for a given column in a set of comparisons.

Sensitivity to amount of training data available: Having seen the out-performance that our proposed model delivers over other state-of-the-art un-supervised approaches, we are interested to investigate the influence of the amount of available training data on overall classification performance.

We progressively increase the amount of training data available to SSBN, at incremental steps of 10%. The effect this has on classifier performance is plotted in Fig. 5.

On the whole, we can observe an upward trend in classifier performance as more training data is made available. This is intuitive, and is evident also from the higher F_1 results we see for SSBN (90%) vis-a-vis SSBN (5%). The bigger disparity between micro- and macro-averaged results seen in the results for D_{Tweet} is likely due to the noise introduced by the automatic labeling we have performed while building the dataset.

4.3. Evaluation of genre summarization

The next set of experiments we conduct aims to evaluate the summarization performance of our RGMS system. Recall that having classified retrieved microblog messages into one of the several pre-identified genres, the final output of our system is a summary for each of these genres.

Evaluation metric: To evaluate the quality of the summaries generated by our approach, we employ the widely used ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric [55]. ROUGE is an n-gram based metric which evaluates the likeness of an automatically generated summary to a manually written one. The ROUGE-1 (R-1) variant in particular has been shown to co-relate well with human assessments [56] for short summaries. Since the summaries we are generating for each genre here is typically in the order of 50 words or less, we decide to make use of the R-1 metric here.

Value of resource-reinforcement: Table 10 sums up the R-1 results obtained for the experiments we have conducted to assess summarization performance. Two sets of results are shown in the table, obtained both with (*i.e.*, RGMS) and without the use of resource-reinforcement. The first important observation is that the use of resource-reinforcement is very effective in improving R-1 scores. Using PageRank with LexRank for example, we get relative improvements of approximately 8% and 30% for D_{Sina} and D_{Tweet} respectively. The performance gain especially for the case of D_{Tweet} is impressive.

The next important observation is that LexRank outperforms TextRank as a similarity measure. Higher R-1 scores are obtained for both PageRank and HITS when LexRank is used. LexRank pays more thought to inverse document frequency (IDF) scores than TextRank. We believe that this helps in the case of the short microblog messages we are working with here.

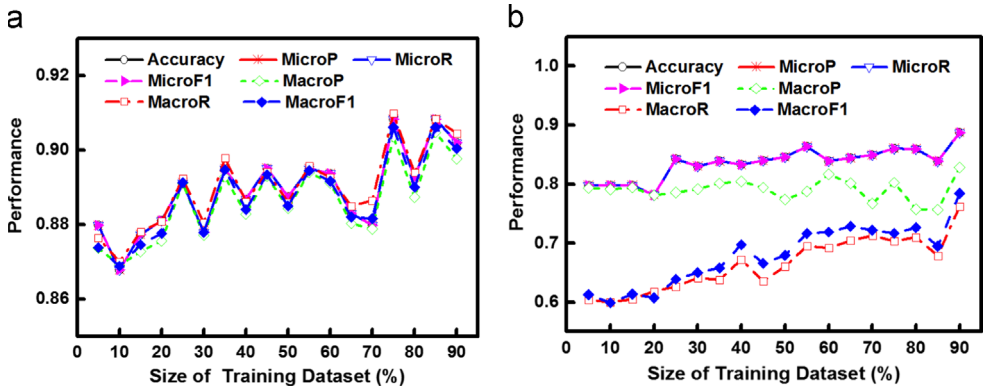


Fig. 5. Effect on classifier performance from varying the amount of data used for training: (a) results on D_{Sina} and (b) results on D_{Tweet}.

Table 10
R-1 results obtained for RGMS as well as a plain graph-based summarization system without the use of external resources (*i.e.*, under the *Without* column).

Algorithm	Without		RGMS	
	D_{Sina}	D_{Tweet}	D_{Sina}	D_{Tweet}
PageRank + LexRank	0.5372	**0.4135**	**0.5821**	**0.5371**
HITS + LexRank	**0.5396**	0.4123	0.5796	0.5248
PageRank + TextRank	0.5028	0.3937	0.5634	0.5017
HITS + TextRank	0.5083	0.3886	0.5667	0.4966

The bolded fields denote the best performance for a given column in a set of comparisons.

Table 11
Human assessment results for generated summaries, graded on a scale of 1–5, where higher scores are better.

Algorithm	D_{Sina}	D_{Tweet}
PageRank + LexRank	**3.8257**	**3.7667**
HITS + LexRank	3.8014	3.7628
PageRank + TextRank	3.7891	3.3500
HITS + TextRank	3.8002	3.3117

The bolded fields denote the best performance for a given column in a set of comparisons.

As is the case with genre summarization, RGMS incurs some additional cost for resource-reinforcement (*i.e.*, Stage 1 of the summarization process). This cost is however kept small and manageable. Given $|C|$ genres to generate summaries for, resource-reinforcement is performed once for each of these $|C|$ genres. The required keywords are collected each time, before a single search engine query is made. To further limit the total cost incurred, Sense&Summ is restricted to only query the top five documents returned by the search engine. This prevents Sense&Summ from spending too much time polling for and retrieving web pages. The un-predictable latencies of accessing webpages over the Internet, as well as the multitude of factors that can affect these latencies, makes it hard for us to arrive at a rigorous estimate of the additional time incurred for resource-reinforcement. However from empirical observations, we find that Sense&Summ is able to produce the required summaries quickly.

Human assessment: While the R-1 scores are indicative, they are not perfect measures of summarization performance. The community has been working on various alternative evaluation measures. For example the Automatically Evaluating Summaries of Peers (AESOP) task [57] in the Text Analysis Conference (TAC) is an annual evaluation workshop targeted at promoting efforts into building new automatic evaluation measures. However there has yet to be a consensus on an effective, automatic evaluation metric. Therefore, to augment the results we have in Table 10, we also carried out a manual assessment of the quality of the generated summaries.

We invited three independent assessors to review and score the generated summaries. Recall that one summary is generated for each genre of each topic. The assessors were instructed to evaluate each summary based on a scale of 1–5, where higher scores are better. They are tasked to consider (1) how well the generated summary corresponds to the content of the microblog messages classified under the genre for which the summary is for, as well as (2) how representative the content within the summary is.

Table 11 shows the human assessed scores averaged out over all generated summaries for each dataset. It can be seen that the human assessors grade the generated summaries very favorably. The summaries for D_{Sina} are assessed to be slightly better than those for D_{Tweet}. This is consistent with the earlier observations we had from the R-1 measure.

Query formulation for resource-reinforcement: Recall that keywords derived as part of RGMS are used as queries for resource-reinforcement. We are interested to investigate whether the type of keywords used to formulate the queries has an effect on the quality of the documents retrieved from the search engine, thereby affecting summarization performance. The intuition behind this is that perhaps nouns (in particular proper nouns) may help to retrieve text that is more specific while verbs which are less specific and general in nature may perhaps retrieve content which is more generic.

With this mind, we performed several additional experiments to study the influence of the part-of-speech of keywords that are used to formulate queries used for resource-reinforcement. We varied the keyword generation step in RGMS, and imposed selection constraints based on the part-of-speech of the keywords. Specifically we explored three cases, using (1) only keywords which are nouns, (2) keywords which are either nouns or verbs, and (3) keywords which are nouns, verbs and adjectives. The effect that each of these cases has on R-1 is plotted in Fig. 6.

We see that selecting keywords which are either nouns or verbs gives the best R-1 scores, while using only nouns results in the worst performance. This observation holds regardless of whether PageRank or HITS is used with either similarity measure. We postulate that the under-performance of using only nouns for keywords could be due to the specificity of nouns. The retrieved search results during resource-reinforcement may not be sufficiently diverse, leading to less effective summarization.

4.4. Discussion

The evaluation results above show that the use of external resources in Sense&Summ is of tremendous value, both for genre

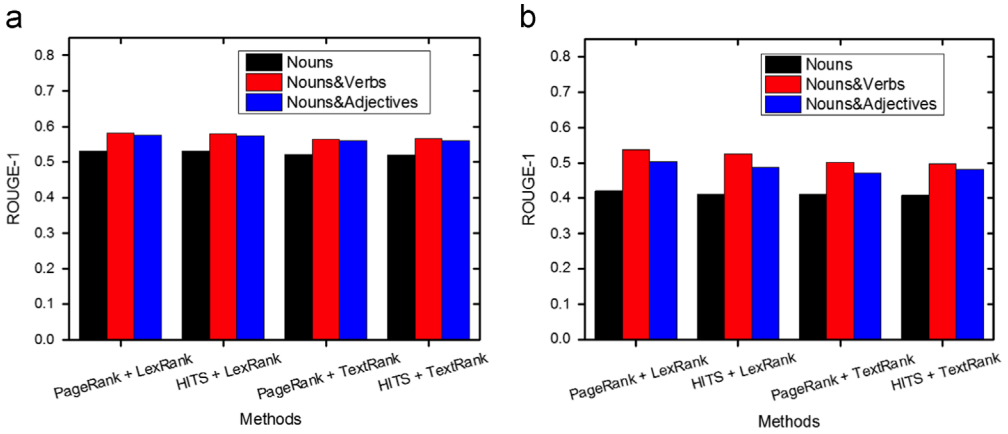


Fig. 6. Effect of varying the type of keywords used to construct queries for resource-reinforcement: (a) results on D_{Sina} and (b) results on D_{Tweet}.

classification and for genre summarization. Before ending this section, it is useful to discuss one limitation of our proposal.

The use of external resources helps us to overcome the lack of sufficient contextual information in short microblog messages. It is therefore very useful when we can obtain a lot of information on the topic being discussed from search engines. However if the topic being discussed is very recent (*e.g.*, emerging topic), or became popular in a very short amount of time (*e.g.*, bursty topic), there may not be sufficient information about it. In this case, our proposal may not be as effective.

Due to the way our evaluation datasets are constructed, it is not possible for us to validate the hit on system performance in this situation. It will be interesting to investigate in future work the effectiveness of our proposal for such emerging or bursty topics.

5. Conclusion

There is a lot of information that can be gleaned from the millions of microblog messages that have been posted and archived. However we have identified a problem with existing retrieval interfaces that severely limits how users can gain access to this information. We propose tackling two fundamental limitations in this paper with `Sense&Summ`, that of (1) the inability of current search results to distinguish between different, possibly ambiguous senses of search queries, and (2) the information overload caused by the huge number of microblog messages returned with each search.

`Sense&Summ` solves these two limitations by (1) classifying retrieved microblog messages into different genres, and (2) displaying a summary of each genre to help give users a better idea of the content within each genre. The problems that we are tackling are not novel, however we believe that our approach is innovative. In particular, we have shown that we can tap on resources obtained automatically with search engines to provide us with more contextual information to better solve these two tasks. For genre classification, we integrate these resources with a semi-supervised PGM. This allows us to obtain significantly improved performance without the need for large amounts of training data. We also make further use of these resources to improve graph-based summarization, in a process we refer to as resource-reinforcement, and demonstrate that it allows us to obtain enhanced summaries for each genre.

This is an exciting area of research which tackles a very real problem. Going ahead we believe that a lot more promising work can be done. Importantly, we posit that the use of externally acquired data can be useful beyond `Sense&Summ`. In fact, it can be a potential solution to overcome the lack of sufficient training data and corpora. One interesting extension will be to investigate the effectiveness of this approach in improving the performance of un-supervised or semi-supervised machine learners in a more general setting, beyond just genre classification and summarization as we have done here.

We can also take a closer look at the `Sense&Summ` pipeline. In the case of genre classification, our work currently proposes the use of static, pre-identified genres. It is possible and definitely interesting to consider the use of dynamic genres which are automatically generated from the microblog messages. We also hope to explore better ways to summarize the content within each genre. While our proposed resource-reinforcement approach is helpful in improving the quality of the generated summaries, it will be worthwhile to consider alternative ways of presenting the content within a genre to users, perhaps even without the use of text.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant nos. 61170189, 61370126, 61202239), the Fund of the State Key Laboratory of Software Development Environment (Grant no. SKLSDE-2013ZX-19), the Innovation Foundation of Beihang University for Ph.D. Graduates (YWF-13-T-YJSY-024), the Fund for the Doctoral Program of Higher Education of China (Grant no. 20111102130003), and a Research Fund from Microsoft Research Asia (No. FY14-RES-OPP-105).

References

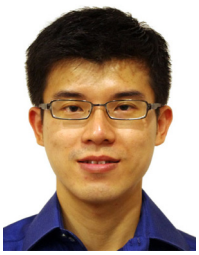
[1] D. Bosomworth, Mobile Marketing Statistics 2014, Technical Report, SmartInsights, 2014.
[2] X. Hu, N. Sun, C. Zhang, T.-S. Chua, Exploiting internal and external semantics for the clustering of short texts using world knowledge, in: Proceedings of the ACM Conference on Information and Knowledge Management (CIKM), 2009, pp. 919–928.
[3] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, Technical Report, Stanford University, 1999.
[4] J.M. Kleinberg, Hubs, authorities, and communities, ACM Comput. Surv. 31 (4es).
[5] J. Teevan, D. Ramage, M.R. Morris, #TwitterSearch: a comparison of microblog search and web search, in: Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM), 2011, pp. 35–44.
[6] H. Gao, J. Tang, H. Liu, Exploring social-historical ties on location-based social networks, in: Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM), 2012, pp. 114–121.
[7] J. Weng, B.-S. Lee, Event detection in twitter, in: Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM), 2011, pp. 401–408.
[8] L. Hong, B.D. Davison, Empirical study of topic modeling in twitter, in: Proceedings of the Workshop on Social Media Analytics (SOMA), 2010, pp. 80–88.
[9] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[10] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth, M. Steyvers, Learning author-topic models from text corpora, ACM Trans. Inf. Syst. 28 (2010) 1–38.
[11] D. Ramage, D. Hall, R. Nallapati, C.D. Manning, Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora, in: Proceedings of

International Conference on Empirical Methods in Natural Language Processing (EMNLP), 2009, pp. 248–256.
[12] D. Ramage, S. Dumais, D. Liebling, Charaterizing microblog with topic models, in: Proceedings of International Conference on Weblogs and Social Media (ICWSM), 2010, pp. 130–137.
[13] K. Lee, D. Palsetia, R. Narayanan, M.M.A. Patwary, A. Agrawal, A. Choudhary, Twitter trending topic classification, in: Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW), 2011, pp. 251–258.
[14] A. Zubiaga, D. Spina, V. Fresno, R. Martinez, Classifying trending topics: a typology of conversation triggers on twitter, in: Proceedings of ACM Conference on Information and Knowledge Management (CIKM), 2011, pp. 2461–2464.
[15] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, M. Demirbas, Short text classification in twitter to improve information filtering, in: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), 2010, pp. 841–842.
[16] I. Mani, M.T. Maybury, Advances in Automatic Text Summarization, MIT Press, Cambridge MA, USA, 1999.
[17] A. Nenkova, K. McKeown, Automatic summarization, Found. Trends Inf. Retr. 5 (2011) 103–233.
[18] C.-Y. Lin, E. Hovy, From single to multi-document summarization: a prototype system and its evaluation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), 2002, pp. 457–464.
[19] F. Liu, Y. Liu, F. Weng, Why is "SXSW" trending? Exploring Multiple Text Sources for Twitter Topic Summarization, in: Proceedings of the Workshop on Language in Social Media (LSM), 2011, pp. 66–75.
[20] X. Yang, A. Ghoting, Y. Ruan, S. Parthasarathy, A framework for summarizing and analyzing twitter feeds, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2012, pp. 370–378.
[21] K.D. Rosa, R. Shah, B. Lin, A. Gershman, R. Frederking, Topical clustering of tweets, in: Proceedings of the SIGIR Workshop on Social Web Search and Mining (SWSM), 2011, pp. 223–232.
[22] J.-P. Ng, P. Bysani, Z. Lin, M.-Y. Kan, C.-L. Tan, Exploiting category-specific information for multi-document summarization, in: Proceedings of the International Conference on Computational Linguistics (COLING), 2012, pp. 2093–2108.
[23] A. Zubiaga, D. Spina, E. Amigo, J. Gonzalo, Towards real-time summarization of scheduled events from twitter streams, in: Proceedings of the ACM Conference on Hypertext and Social Media, 2012, pp. 230–319.
[24] R. McDonald, A study of global inference algorithms in multi-document summarization, in: G. Amati, C. Carpineto, G. Romano (Eds.), Advances in Information Retrieval, Lecture Notes in Computer Science, vol. 4425, Springer, Berlin, Heidelberg, 2007, pp. 557–564.
[25] K. Woodsend, M. Lapata, Multiple aspect summarization using integer linear programming, in: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP), 2012, pp. 233–243.
[26] H. Takamura, H. Yokono, M. Okumura, Summarizing a document stream, in: P. Clough, C. Foley, C. Gurrin, G. Jones, W. Kraaij, H. Lee, V. Mudoch (Eds.), Advances in Information Retrieval, Lecture Notes in Computer Science, vol. 6611, Springer, Berlin, Heidelberg, 2011, pp. 177–188.
[27] B. Sharifi, M.-A. Hutton, J. Kalita, Summarizing microblogs automatically, in: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), 2010, pp. 685–688.
[28] B. Sharifi, M.-A. Hutton, J. Kalita, Experiments in microblog summarization, in: Proceedings of the IEEE International Conference on Social Computing (SocialCom), 2010, pp. 49–56.
[29] S.M. Harabagiu, A. Hickl, Relevance modeling for microblog summarization, in: Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM), 2011, pp. 514–517.
[30] G. Erkan, D.R. Radev, LexRank: graph-based lexical centrality as salience in text summarization, J. Artif. Intell. 22 (2004) 457–479.
[31] R. Mihalcea, P. Tarau, Textrank: bringing order into texts, in: Proceedings of International Conference on Empirical Methods in Natural Language Processing (EMNLP), 2004, pp. 404–411.
[32] A. Olariu, Hierarchical clustering in improving microblog stream summarization, in: A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science, vol. 7817, Springer, Berlin, Heidelberg, 2013, pp. 424–435.
[33] W. Xu, R. Grishman, A. Meyers, A. Ritter, A preliminary study of tweet summarization using information extraction, in: Proceedings of the Workshop on Language in Social Media (LASM), 2013, pp. 20–29.
[34] D. Chakrabarti, K. Punera, Event summarization using tweets, in: Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM), 2011, pp. 66–73.
[35] Y. Duan, Z. Chen, F. Wei, M. Zhou, H.-Y. Shum, Twitter topic summarization by ranking tweets using social influence and content quality, in: Proceedings of the International Conference on Computational Linguistics (COLING), 2012, pp. 763–780.
[36] Y. Chen, Z. Li, L. Nie, X. Hu, W. Xiangyu, T.-S. Chua, Z. Xiaoming, A semi-supervised bayesian network model for microblog topic classification, in: Proceedings of the International Conference on Computational Linguistics (COLING), 2012, pp. 561–576.
[37] C. Zhang, T. Baldwin, H. Ho, B. Kimelfeld, Y. Li, Adaptive parser-centric text normalization, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL), 2013, pp. 1159–1168.
[38] Y. Chen, A. Hadi, Z. Li, T.-S. Chua, Emerging topic detection for organizations from microblogs, in: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), 2013, pp. 43–52.
[39] D. Knights, M.C. Mozer, N. Nicolov, Detecting topic drift with compound topic models, in: Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM), 2009, pp. 242–245.
[40] R.L. Cilibrasi, P.M. Vitanyi, The google similarity distance, IEEE Trans. Knowl. Data Eng. 19 (3) (2007) 370–383.
[41] M. Cataldi, L.D. Caro, C. Schifanella, Emerging topic detection on twitter based on temporal and social terms evaluation, in: Proceedings of the International Workshop on Multimedia Data Mining (MDMKDD), 2010.
[42] R. Ferreira, L. de Souza Cabral, R.D. Lins, G. Pereira e Silva, F. Freitas, G.D. Cavalcanti, R. Lima, S.J. Simske, L. Favaro, Assessing sentence scoring techniques for extractive text summarization, Expert Syst. Appl. 40 (14) (2013) 5755–5764.
[43] L. Vanderwende, H. Suzuki, C. Brockett, A. Nenkova, Beyond sumbasic: task-focused summarization with sentence simplification and lexical expansion, Inf. Process. Manag. 43 (6) (2007) 1606–1618.
[44] D. Inouye, J.K. Kalita, Comparing twitter summarization algorithms for multiple post summaries, in: Proceedings of the International Conference on Social Computing (SocialCom), 2011, pp. 298–306.
[45] I. Ounis, C. Macdonald, J. Lin, I. Soboroff, Overview of the TREC 2011 microblog track, in: Proceedings of the 20th Text Retrieval Conference (TREC), 2011.
[46] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (2009) 427–437.
[47] T.N. Phyu, Survey of classification techniques in data mining, in: Proceedings of International MultiConference of Engineers and Computer Scientists (MECS), 2009, pp. 727–731.
[48] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.
[49] Y. Yang, J. Pederson, A comparative study on feature selection in text categorization, in: Proceedings of International Conference on Machine Learning (ICML), 1997, pp. 412–420.
[50] R.H. Creecy, B.M. Masand, S.J. Smith, D.L. Waltz, Trading MIPS and memory for knowledge engineering, in: Communication of the ACM, vol. 35, 1992, pp. 48–64.
[51] R.E. Schapire, Y. Singer, A. Singhal, Boosting and Rocchio applied to text filtering, in: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), 1998, pp. 215–223.
[52] T. Joachims, Transductive inference for text classification using support vector machines, in: Proceedings of International Conference on Machine Learning (ICML), 1999, pp. 200–209.
[53] K. Nigam, A.K. Mccallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, in: Machine Learning—Special Issue on Information Retrieval, vol. 39, 2000, pp. 103–134.
[54] H.-T. Liao, K.-W. Fu, M. Jiang, N. Wang, Chinese web data: definition, uses, and scholarship, in: Proceedings of the Annual Chinese Internet Research Conference, 2013.
[55] C.-Y. Lin, E. Hovy, Automatic evaluation of summaries using N-gram co-occurrence statistics, in: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL), 2003, pp. 71–78.
[56] C.-Y. Lin, ROUGE: a package for automatic evaluation of summaries, in: Proceedings of the Workshop on Text Summarization Branches Out (WAS), 2004, pp. 74–81.
[57] H.T. Dang, K. Owczarzak, Overview of the TAC 2009 summarization track, in: Proceedings of the Text Analysis Conference (TAC), 2009.

Yan Chen received her Bachelor's degree (2007) in Computer Science from the Central South University and Masters degree (2010) from the Hohai University. She is now pursuing her Ph.D. degree in Beihang University under the supervision of Prof. Zhoujun Li. Her research interests include text mining and social media analysis.

Xiaoming Zhang was born in Hunan, China, on December 7, 1980. He received his B.Sc. degree and his M.Sc. degree in Computer Science and Technology from the National University of Defence Technology, China, in 2003 and 2007 respectively. He received his Ph.D. degree in Computer Science from Beihang University, in 2012. He is currently working at the School of Computer Science and Engineering, Beihang University, where he has been a lecturer since 2012. His major interests are in social network analysis and image tagging.

Zhoujun Li received his B.S. degree in the School of Computer Science from Wuhan University, in 1984, and his M.S. and Ph.D. degrees in the School of Computer Science from the National University of Defense Technology. He is currently working as a Professor in Beihang University. His research interests include data mining, information retrieval and information security. He is a member of the IEEE and ACM.



Jun-Ping Ng graduated with a Ph.D. in Computer Science from the National University of Singapore, in 2014, and is currently a software engineer at Bloomberg L.P., USA. His research interests include improving natural language applications such as question-answering and summarization, as well as temporal information extraction.