Embracing Information Explosion without Choking: Clustering and Labeling in Microblogging

Xia Hu, *Member, IEEE*, Lei Tang, *Member, IEEE*, and Huan Liu, *Fellow, IEEE*

Abstract—The explosive popularity of microblogging services produce a large volume of microblogging messages. It presents great difficulties for a user to quickly gauge his/her followees' opinions when the user interface is overwhelmed by a large number of messages. Useful information is buried in disorganized, incomplete, and unstructured text messages. We propose to organize the large amount of messages into clusters with meaningful cluster labels, thus provide an overview of the content to fulfill users' information needs. Clustering and labeling of microblogging messages are challenging because that the length of the messages are much shorter than conventional text documents. They usually cannot provide sufficient term co-occurrence information for capturing their semantic associations. As a result, traditional text representation models tend to yield unsatisfactory performance. In this paper, we present a text representation framework by harnessing the power of semantic knowledge bases, i.e., Wikipedia and Wordnet. The originally uncorrelated texts are connected with the semantic representation, thus it enhances the performance of short text clustering and labeling. The experimental results on Twitter and Facebook datasets demonstrate the superior performance of our framework in handling noisy and short microblogging messages.

Index Terms—Microblogging, clustering, labeling, semantic knowledge

1 Introduction

Microblogging services such as Twitter and Facebook have become one of the most important communication platforms in people's daily life. They are widely used for commenting on breaking news, participating online events and connecting with each other. The services trace what people are thinking and talking in real time, providing valuable information to understand human behavior in a new dimension. For example, Twitter messages are already archived in the US Library of Congress.[1] In many scenarios, microblogging services have become an important information source for Internet and social media users.

With the increasing popularity, microblogging services produce a large amount of data every second. The information overload presents great difficulties for users to fulfill their information needs. Take Twitter as an example. While many Twitter users only have the patience to glance the latest and sometimes redundant tweets, many tweets of their interests may be buried in the large amount of streaming data. Given the huge number of tweets, it is hard for users to efficiently gauge the main topics from their tweets. As a result, the large volume and short noisy text messages hinder the accessibility of information for users to conveniently search, navigate and locate the specific topics one might be interested in. It significantly discourages user engagement, and subsequently the microblogging service can become poorly accessible and less interesting.

Hence, it is appealing to provide users an efficient way of determining the topics/subtopics contained in the microblogging messages of their followees. For example, when Apple Inc. released iPad Air, many microblogging users posted messages related to this event. Among the messages posted by followees, some users might be interested in many fans queueing overnight at "Apple Store", while others might want to read messages related to new features of "Apple iPad Air". Without effective navigation, many valuable and interesting posts may be buried in disorganized messages. To make a large collection of microblogging messages accessible to users, current web systems need to provide not only accurate clusters for subtopics in microblogging messages, but also meaningful labels for each cluster. Then users are able to quickly identify messages of interest by examining an overview of subtopics. Under this scenario, we propose to explore clustering [1] and labeling to embrace the information explosion.

The distinct characteristics of social media data present great challenges to directly apply existing text analytics methods to process microblogging messages. First, microblogging messages are short. For example, Twitter only allows users to post tweets that are no more than 140 characters. Thus it cannot provide sufficient statistical evidence for similarity measurement, which is essential in text processing methods. Second, unstructured form of textual data is popular in

1. http://blogs.loc.gov/loc/2010/04/how-tweet-it-is-library-acquires-entire-twitter-archive/

- *X. Hu is with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX. E-mail: hu@cse.tamu.edu.*
- *H. Liu is with the Computer Science and Engineering, Arizona State University, Tempe, AZ 85281. E-mail: huan.liu@asu.edu.*
- *L. Tang is with Clari Inc., Mountain View, CA 94041. E-mail: leitang@acm.org.*

Manuscript received 19 Feb. 2015; revised 26 May 2015; accepted 17 June 2015. Date of publication 8 July 2015; date of current version 17 Sept. 2015. Recommended for acceptance by W. Zhu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TBDATA.2015.2451635

microblogging. Many slang words, such "cooool" and "Good 9t", are widely used in microblogging. While the informal words provide convenience for users' communication, they also bring in difficulties for text precessing methods.

Bag of words (BOW) representation has been widely used by practitioners to represent text for topic modeling [2], [3] or text mining. It has been extended to microblogs as well [4], [5]. However, this representation is often inadequate to perform finer textual analysis if the task involves the use of varying syntactic structures or complex semantic meanings. For example, consider two tweets: "For which movie did Colin Firth get Oscar nominations?" and "Which Oscar nominations did Colin Firth get for the movie?". They have different topic focuses ("movie" and "Oscar nominations"). However, it is hard to distinguish them due to their high overlap of words. In order to capture the order information, n-gram can be exploited as features, but they lead to an explosion of terms, among which many are meaningless. Besides, microblogging messages are very short. For instance, Twitter allows up to 140 characters only for each message. Such sparsity as presented in microblogging messages poses thorny challenges to map and connect related phrases in a corpus. There is a pressing need to enrich each message's representation for more accurate clustering and labeling.

In this paper, we present a novel framework to enhance the accessibility of microblogging messages. The proposed framework improves message representation by mapping messages from an unstructured feature space to a semantically meaningful knowledge space. First, in order to reduce the noise yet keep the key information as expressed in each message, we propose to use natural language processing (NLP) techniques to analyze the message and extract informative words and phrases. Then, to overcome the extreme sparsity of microblogging messages, we map the selected terms to structured concepts derived from external knowledge bases that are semantically rich. After these two steps, the message representation is expanded substantially with meaningful semantics. By conducting feature selection to refine the feature space, we are able to cluster all messages more accurately and generate human-comprehensible labels efficiently from related concepts. Our main contributions are summarized as follows:

- Formally define the problem of enhancing accessibility of large number of microblogging messages, and propose to employ clustering and labeling to solve the problem;
- Present a novel framework that decomposes microblogging messages to parse tree fragments and integrates external knowledge bases to improve clustering of microblogging messages;
- Provide an efficient and effective way to simultaneously generate textual labels for each cluster by ranking structured concepts from Wikipedia.

The rest of this paper is organized as follows: Section 2 defines the problem formally. Section 3 introduces our proposed framework of mapping unstructured data to structured and meaningful concepts to enhance the accessibility of microblogging messages. Experimental results are presented in Section 4 . Section 5 reviews related work. Section 6 concludes the paper with directions for future work.

2 Related Work

Recent studies in various domains show great interest in microblogging services. Existing studies on microblogging messages include influence study [6], sentiment analysis [7], [8], [9], event or topic tracking [10], [11], text summarization [12] etc. However, to our best knowledge, this paper tackles a novel problem of clustering and labeling on microblogging messages in a unified way.

The idea of enhancing the accessibility of microblogging messages is related to aggregated search technologies. In [13], [14], the authors clustered search snippets and then generated summarization for each cluster. Hu et al. [15] aims to provide a better solution for clustering of search results by finding contextual clues from internal and external knowledge. Different from the semi-structured search snippets, microblogging messages are more like unstructured natural language, which presents more challenges and opportunities. Comparing with methods in organizing search snippets, our method differs by harnessing the power of finer level syntactic analysis. This would be difficult in other domains due to its expensive computational cost. But thanks to the shorter length microblogging messages, the computational difference becomes affordable. In addition, structured concepts from Wikipedia provide an effective way to generate human-comprehensible textual cluster labels, thus avoids the readability problem of segmentation and frequency based labeling methods.

Besides web texts clustering, many methods based on "bag of words" or "bag of features" model achieved satisfactory results by improving representation of standard documents for clustering and classification. Features from different perspectives, including terms [16], phrases [17] and segments [18], were extracted to construct the feature space. Unfortunately, microblogging messages only consist of a few phrases or 1-2 sentences. Context shared information contained in the messages is insufficient for effective similarity measure, which is the basis of clustering methods.

It has been found to be useful to enhance text representation by incorporating semantic knowledge [19], [20]. For document clustering and classification, [21] analyzed documents and found related ontology concepts from Wikipedia and Open Directory Project (ODP), which in turn induced a set of features that augment the standard BOW. Towards improving the management of Google snippets, existing methods focus either on classifying the web texts into smaller categories [15] or assigning labels for each category [22] with the help of Wikipedia and WordNet. Hu et al. [23] presented a novel document clustering method by enriching document representation with concepts from Wikipedia. Phan et al. [24] use LDA to sample topics from both large-scale data collection and a sparse testing dataset. For web texts applications, Banerjee et al. [25] proposed a method to enrich short texts representation with features from Wikipedia. Although this method only used the titles of Wikipedia articles as additional external features, it showed improvement in the accuracy of short texts clustering. For word sense disambiguation, Kohomban and Lee [26] built a word sense disambiguation system to tackle the data scarcity problem. This system trains the classifier using grouped senses for verbs and nouns according to the top-level synsets from

WordNet and effectively pool the training cases across senses within the same synset. We explored to integrate semantic knowledge for the clustering and labeling of microblogging messages, and achieved significant improvement as compared to existing methods.

3 Problem Statement

In microblogging websites, tweets and retweets of followees are listed in reverse chronological order for a user to read. With a large number of messages appearing in the interface, people often do not have patience to skim every message. A collateral problem with many messages is that there are often different focal topics given diverse interests of users. It prevents a user from jumping to what she is interested in. Take Twitter as an example. When "Arizona Shooting" occurs, a user may be interested in the current status of congresswoman Gabby Giffords or the investigation of the event. It is very difficult for the user to find accurate information when there are more than 100 tweets posted by her followees. As a result, we try to provide an efficient way to reduce the gap between user's information needs and a large volume of microblogging messages. To present a clear structure of subtopics, we propose to cluster the posted microblogging messages into several groups and assign semantically meaningful labels for each cluster.

We now formally define two major tasks in the problem of enhancing accessibility of microblogging messages.

Task 1: Microblogging Message Clustering. Let $M = \{m_1, m_2, \ldots, m_n\}$ be a corpus of n microblogging messages. Among these n messages, there are k latent topics or subtopics. We aim to cluster the n messages into k clusters $\{c_1, c_2, \ldots, c_k\}$ with their latent topics as centroids.

Task 2: Cluster Labeling. For each cluster c_i, we aim to generate human readable cluster labels $\{l_{i1}, l_{i2}, \ldots, l_{ik}\}$, which are semantically similar to the latent topic of c_i.

Text representation of microblogging messages for clustering and labeling is important and challenging in many ways. In the next section, we present a novel framework by improving text representation to achieve the task of microblogging message clustering and labeling.

4 Managing Microblogging Messages

In this section, we introduce the proposed framework for clustering and labeling microblogging messages.

4.1 Overview of the Framework

Our proposed framework consists of three phases, *Syntactic Decomposition*, *Semantic Mapping*, *Clustering & Labeling*, as shown in Fig. 1. We use an example to illustrate how to perform clustering and labeling of microblogging messages. Suppose we have a microblogging message as follows:

> "I just voted for Colin Firth for Best Actor in Movieline's Statuesque Contest. Who do you pick?"

4.1.1 Syntactic Decomposition

The original document itself always contains the most important information. However, it is difficult to mine useful information from microblogging messages as they can be short and noisy. Thus a method which can make better use

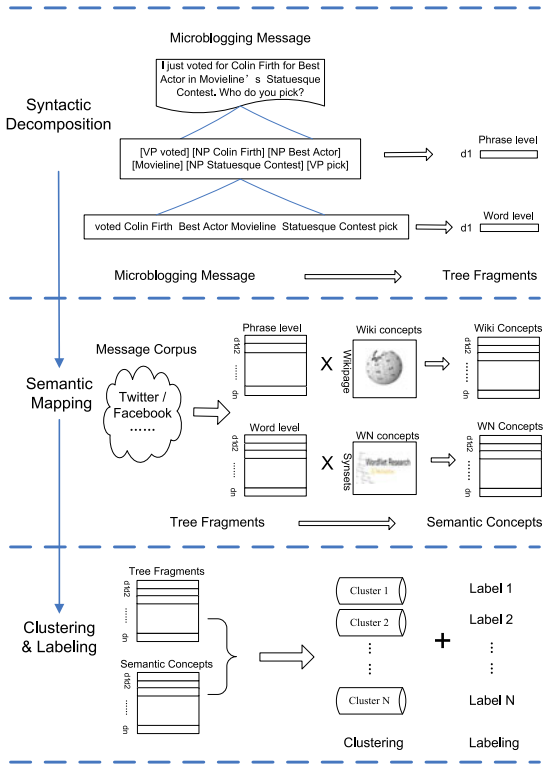


Fig. 1. The proposed framework for managing microblogging messages. It consists three components: (1) Syntactic Decomposition; (2) Semantic Mapping; and (3) Clustering & Labeling.

of the limited text is necessary for message representation. We propose to utilize a parse tree to model the message. Given a corpus of microblogging messages, we apply Syntactic Decomposition to extract a number of tree fragments for each message. The extracted fragments can be divided into two categories: phrase-level and word-level fragments, as shown in Fig. 1. Particularly, the phrase-level fragments serve as part of the feature space for clustering, and provide an informative basis for Semantic Mapping in the next phase.

4.1.2 Semantic Mapping

The tree fragments from the first phase are inadequate for message representation due to the "semantic gap". Two semantically related texts cannot be connected by only considering their word or phrase co-occurring information. Thus, we propose to map the original unstructured message to structured semantic space. With tree fragments extracted in Syntactic Decomposition, our framework maps the fragments to their corresponding semantic concepts. These generated semantic concepts will also serve as part of the feature space.

4.1.3 Clustering and Labeling

With the combination of tree fragments from Syntactic Decomposition and semantic features from Semantic Mapping, we construct the original feature space for clustering and labeling. However, one message may generate a number of tree fragments and semantic concepts, which will introduce noise and may harm the feature space for clustering. In addition, to avoid "curse of dimensionality", feature selection is employed to ensure the feature space is compact

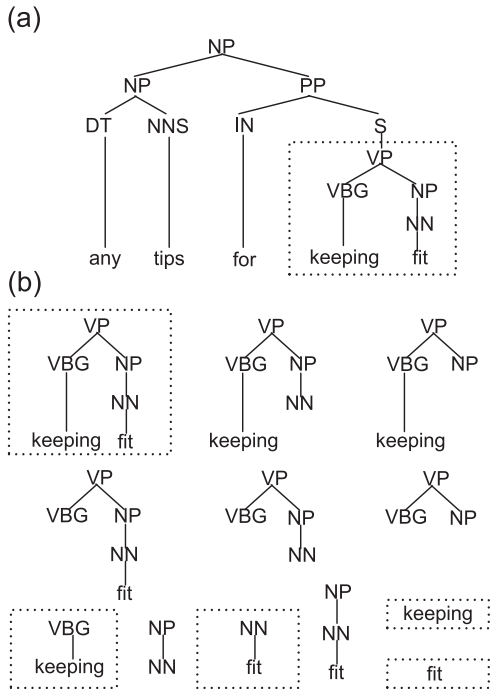


Fig. 2. (a) The parse tree of "any tips for keeping fit?". (b) Tree fragments of the subtree covering "keeping fit"; the fragments with dotted line frame are extracted tree fragments for "keeping fit".

and effective for clustering. The most representative semantic concept in the cluster of messages is extracted as the meaningful cluster label.

4.2 Syntactic Decomposition

Traditional methods exploit phrases [17] contained in the original text to preserve the valuable contextual information. However, these methods did not apply NLP techniques such as parsing to analyze the structure of documents in detail. As a result, they fail to perform a deep analysis of original text. Many NLP techniques have achieved great success by extracting tree fragments that occur in a parse tree [27] to enrich text representation. Hence, we employ parsing to analyze the syntactic structure contained in microblogging messages.

A parse tree (or syntactic tree) is an ordered and rooted tree that represents the syntactic structure of a string according to a formal grammar [28]. Fig. 2a illustrates an example of a parse tree generated by OpenNLP.[2] In the Figure, "VP" is for verb phrase and "NP" represents noun phrase.[3] From the syntactic structure above, we can see that the tweet contains abundant lexical information at phrase-level and word-level.

Given a microblogging message, a parse tree has been constructed to retain the syntactic information. Furthermore, we need to extract useful information from the parse tree to improve message representation. To better utilize the syntactic structure of a parse tree, Wang et al. [29] proposed to employ tree fragments as syntactic features.

The tree fragments of a parse tree are all its subtrees which include at least one terminal symbol (word) or one production rule [27], with the restriction that no production rules can be broken into incomplete parts. For example, any subtree containing a part of the production rule "VP→VB:NP", such as"VP→VB", is considered invalid. Fig. 2b presents an illustration of all of the valid tree fragments for the subtree "keeping fit".

The tree fragments generation algorithm is illustrated in Algorithm 1. Basically, we divided our algorithm into two steps: *Subtree Selection* and *Fragment Selection*.

Algorithm 1. Tree Fragments Generation

Input: microblogging message corpus M
Output: a set of *tree fragments* (F)

```
F ← null
for m ∈ M do
    m' ← Lexical Tokens (m)
    T ← Parse Tree Construction (m')
    T' = {t_1, t_2, ..., t_n} ← SubTrees Selection(T)
    for sub tree t ∈ T' do
        if t ∈ {NP, VP, NN, VB} then
            {f_1, f_2, ..., f_n} ← Fragments Selection(t)
            F ← F + {f_1, f_2, ..., f_n}
        end if
    end for
end for
return F
```

4.2.1 Subtree Selection

As shown in Fig. 2a, given a microblogging message, we first construct a parse tree according to its lexical tokens. Note that the number of subtrees is extremely large, which leads to the "curse of dimensionality" [30] and expensive computational cost for real web applications. Thus, we need to develop an efficient way to ensure the generated subtrees are not only informative but also effective. As we know, when people speed-read through a text, they do not fully parse the sentence but instead look for "key phrases" contained in the text [31]. Among these key phrases, the nouns and verbs are considered to be more important than articles, adjectives or adverbs [29]. Thus, we utilize VP (Verb Phrase), NP (Noun Phrase), VB (Verb) and NN (Noun) rooted subtrees to extract tree fragments in next step.

4.2.2 Fragment Selection

As shown in Fig. 2b, one subtree may generate a lot of tree fragments, which will result in redundancies. To avoid introducing redundant information to text representation, we only choose the tree fragments whose leaf nodes are constructed by words or phrase. Take the subtree [VP [VBG keeping] [NP [NN fit]]] as an example. Five tree fragments with dotted line frame in Fig. 2b are extracted. Particularly, the tree fragments can be categorized into two groups phrase-level and word-level, according to their leaf nodes, as shown below:

phrase-level: [VP [VBG keeping] [NP [NN fit]]]
word-level: [VBG keeping] [NN fit] [keeping] [fit]

4.3 Semantic Mapping

The information extracted from an original message is inadequate to build semantic connection with other related

2. http://incubator.apache.org/opennlp/
3. Full list of the abbreviations can be found in http://en.wikipedia.org/wiki/Parse_tree/

messages. For example, without the integration of background knowledge, it is difficult to build connections between "Oscar" and "The King's Speech" related tweets. To explore latent topics hidden in the original message, we propose to map the text into a semantic space.

In order to transform syntactic feature space to semantic feature space, we need to collect original message information as a basis and construct semantic space for mapping. From the Syntactic Decomposition phase, we obtain abundant tree fragments, which are informative to cover the subtopics and structure information in the message. Thus, we employ phrase level and word level tree fragments to construct an informative basis for mapping.

Wikipedia, as a collaborative knowledge base, is regularly updated to reflect recent events. Compared to WordNet, it has better knowledge coverage and time series coverage. On the other hand, as an expert knowledge base, WordNet follows theoretical model or corpus evidence and contains rich lexical knowledge. Hence, we use Wikipedia as the primary semantic knowledge and WordNet as a complementary one. Previous work [32] preprocessed Wikipedia data to collect useful concepts. However, preprocessing of Wikipedia leaves out the valuable textual information of Wikipedia pages. In addition, without information from original Wikipedia pages, it is difficult to map messages to their corresponding semantic concept accurately. We download Wikipedia XML corpus, remove XML tags and create a Solr[4] index for all Wikipedia articles. It facilitates to map the tree fragments to the semantic space. For WordNet, we employ the synsets for mapping a specific word to the corresponding WordNet concepts.

The Semantic Mapping algorithm is illustrated in Algorithm 2. Given a tree fragment, we apply semantic knowledge according to its syntax property. For the phrase-level tree fragments, they are informative to represent a subtopic of the microblogging message. Therefore, we can retrieve accurate Wikipedia pages for these tree fragments. The word-level tree fragments are too general to map to accurate concepts in Wikipedia. We thus utilize WordNet as complement to deal with the word level tree fragments.

Algorithm 2. Semantic Space Mapping

Input: a set of *Tree Fragments*, Wikipedia, WordNet
Output: *Semantic Feature Space* (SF)

```
SF ← null
for tree fragment f ∈ F do
  if f ∈ Phrase-Level then
    f.Query ← SolrSyntax(f, AND)
    Wiki_p ← RetrievePages(f.Query)
    SF ← SF + WikiConcepts(Wiki_p)
    SF ← SF + WikiTopics(Wiki_p)
  else
    WNconcept ← WordNet.Synsets(f)
    SF ← SF + WNconcept
  end if
end for
return SF
```

Particularly, if a tree fragment is from the phrase-level, we build "AND" query[5] which requires the retrieved pages to contain every term in the phrase. For each query, top w articles are retrieved from Wikipedia. Besides title and bold terms (links) contained in the retrieved articles, we also utilize the key phrases which appear in the articles as semantic concepts. We adopted an effective key phrase extraction algorithm, Lingo [33], to extract these key phrases. For example, for the actor "Colin Firth", we may obtain extrinsic concepts "The King's Speech" and intrinsic concepts "England" by mining the related Wikipedia pages. For the tree fragments from the word-level, we employ WordNet synsets to extract similar concepts. For example, we can obtain "auto", "automobile" and "autocar" for the fragment "car".

With a semantic mapping, we can handle phrase-level synonymy problems by mapping two different phrases into the same semantic concept. For example, we can easily map "Colin Firth" and "The King's Speech" to highly overlap semantic features due to their strong semantic connection with each other.

4.4 Clustering & Labeling

4.4.1 Feature Selection

We conduct feature selection to avoid the "curse of dimensionality". A message contains a large number of tree fragments, including phrase-level (t_1) and word-level (t_2) tree fragments. We empirically set the upper bound of selected tree fragments as the number of non-stop words (N) in the message. Then the top N tree fragments are extracted from the original t fragments based on their frequency in the whole corpus. Note that N is different for different tweets according to their number of non-stop words.

We then collect m tree fragments from Syntactic Decomposition and n semantic concepts from semantic knowledge bases, construct a $(m+n)$ dimensional feature space for clustering. As a large number of external features would bring in negative impact on the text representation quality, the number of semantic concepts is determined by:

$$n = \frac{m \times \theta}{1-\theta}, \tag{1}$$

where θ is the fraction of semantic concepts to the feature space for clustering. Apparently, θ is in an interval $[0, 1)$, where $\theta = 0$ means the feature space is constructed of tree fragments and $\theta = 1$ indicates the features are all from semantic concepts. Top n semantic concepts are extracted based on their frequency.

4.4.2 Text Representation for Clustering

To normalize the weight of each feature, we reformulate the weighting policy proposed by Zhang and Lee [34]. For tree fragments f_i extracted from original parse tree, f_i is weighted according to the size and depth of a tree fragment:

$$W_{f_i} = \frac{tf \times idf}{(s(i)+1) \times (d(i)+1)}, \tag{2}$$

4. http://lucene.apache.org/solr/

5. For more detail about query syntax, please refer to http://wiki.apache.org/solr/SolrQuerySyntax

where $s(i)$ is the number of generated tree fragments considering the tree fragment as a subtree and $d(i)$ is the depth of the tree fragment root in the entire parse tree. For example, the tree fragment in Fig. 2b has $s(i) = 3$ and $d(i) = 3$. With this weighting scheme, focus of the message can be measured according to its depth. The key idea here is that a tree fragment will be less important if the tree can generate more fragments or the fragment is in a deeper level. For example, the focus of the sentence "For which movie did Colin Firth get Oscar nominations?" (Section 1) is "movie" but not "Oscar nomination", since the depth of "movie" is less than the latter. Weight scores for all tree fragments are normalized. In addition, weights of semantic features from external knowledge bases are determined by their $tf * idf$ values. Weight scores for all semantic concepts are normalized. At the end, messages are represented in a refined feature space.

4.4.3 Labeling

Traditional labeling methods are based on frequent word, phrase or sentence extraction. There is no guarantee for readability of the extracted labels. It is a natural and effective way to generate textual label from the generated Wikipedia concepts, which have wide knowledge coverage and stably high quality.

Text representation of messages can be conducted offline, however, the label ranking has to be done online. Hence, we propose an efficient informativeness metric to rank our extracted labels.

As shown in Algorithm 2, we can map each tree fragments f_i to several semantic concepts, which are extracted as label candidates $\{l_{i1}, l_{i2}, \ldots, l_{in}\}$. To select the most informative label, the weight function is developed based on the assumption that the informativeness of a label is strongly correlated with which tree fragment the label is generated from, how frequent the label is and how unique the label is. This idea is motivated by the widely used tf-idf weight metric which is employed to measure how important a feature is in various NLP and IR applications [35], [36], [37]. For each labeling candidate l_{ij}, the informativeness score is measured by:

$$Info_{l_{ij}} = W_{f_i} \times tf_{ij} \times idf_{ij}, \tag{3}$$

where W_{f_i} is a weight of the "parent" tree fragment defined in Equation (2), tf_{ij} and idf_{ij} measure the weights among all the candidates. Finally, the labels with highest Info score are extracted as cluster labels.

4.5 Time Complexity Analysis

In this section, we discuss time complexity of the text representation phases. Given a collection of N messages with average length l, we conduct Syntactic Decomposition and Semantic Mapping. For the Syntactic Decomposition phase, each message is modeled as a parse tree, then M_1 phrase-level and M_2 word-level tree fragments are extracted from its subtree. We leverage widely used Chonmsky normal form [38] to construct the parse tree with the complexity of $O(l^3N)$. For the Semantic Mapping phase, we map the extracted tree fragments to the semantic feature space, the time complexity is $O(M_1t_1 + M_2t_2)$, where t_1 and t_2 are retrieving time of Wikipeida and WordNet respectively. Thus we have $O(l^3N + M_1t_1 + M_2t_2)$ time complexity for text representation. Empirically, number of tree fragments $M_1 + M_2 \approx al * N(a < 5)$, we can induce:

$$\begin{aligned} O(l^3N + M_1t_1 + M_2t_2) &\approx bO(l^3N + alN(t_1 + t_2)) \\ &= O(N(l^3 + al(t_1 + t_2))), \end{aligned} \tag{4}$$

As we discussed earlier, microblogging messages are very short (i.e., a small value of l) and we have built local index for Wikipedia and WordNet (i.e., a small value of t_1 and t_2), thus the time complexity of text representation is efficient and affordable in practice. In addition, we can conduct these data preprocessing work offline, which will not affect the time efficiency of the online application.

5 Experiments

In this section, we empirically evaluate the effectiveness of the proposed Microblogging Message Management (M^3) framework. In particular, we conduct experiments from two perspectives. First, we compare eight different text representation methods on two microblogging message datasets to verify how effective our approach is for microblogging message clustering; Second, we conduct extensive experiments to see how effective our texts labeling scheme is by comparing with the state-of-the-art labeling methods for microblogging messages.

5.1 Datasets

Given a set of microblogging messages, it is impractical to manually determine the cluster label of each message and corresponding meaningful textual label for each cluster. Thus the dataset is built indirectly to simulate real-world microblogging applications. "Hot Searches" produced by Google Trends[6] provides a snapshot of what's on the public's collective mind by viewing the fastest-rising queries for different points of time. We crawled the hot queries published by Google Trends, and chose hot queries of different length according to statistical results. According to the percentage of query length and domain distribution, thirty hot queries of diverse topics are selected from Google Trends, as shown in Table 1. Each hot query is considered to be a trending topic, and we crawl top five query suggestions from Google as subtopics of this topic. For example, the topics such as "Apple Store", "Apple Support" etc. are considered as subtopics of the trending topic "Apple". The ground truth is obtained based on the following assumption: the messages returned by a query suggestion construct a cluster and the query suggestion is highly semantically associated with the correct label of this cluster. Thus, we have 150 topics from two levels (30 groups and five subtopics in each group).

5.1.1 Twitter Dataset

Based on the 150 query suggestions (subtopics) from Google, we use Twitter Search API[7] to crawl 100 tweets for each query suggestion and construct a dataset containing

6. http://www.google.com/intl/en/trends/about.html/
7. http://search.twitter.com/api/

TABLE 1
The Selected Hot Topics in Two Datasets

Apple	Jared Allen	Bin Laden
Eyedea	Herman Cain	Diddy Dirty Money
Green Bay	Sidney Poitier	The Dark Knight
Black Friday	Amazing Grace	Fox News Channel
Bloom Box	Aretha Franklin	Sugarloaf Mountain
Bill T Jones	Anjelah Johnson	Teddy Pendergrass
Total Eclipse	Russian National Anthem	
Merle Haggard	Giants Stadium Demolition	
Family Watch Dog	Sue Sylvester Vogue	
New York Giants	National Economic Council	
Victoria Beckham	Kennedy Center Honors	
Pro Bowl	West Memphis Three	

150 categories to testify the effectiveness of clustering algorithm and the quality of the cluster label. As the API will not return exactly 100 tweets for each query, it leaves 11,362 tweets after text preprocessing.

5.1.2 Facebook Dataset

Facebook is not a purely microblogging site, however, it allows a user to post status ("what's on your mind?"), which is very similar to other microblogging sites. Therefore, we employ Facebook Graph API[8] to crawl 200 messages for each subtopic, thus construct a dataset containing 14,322 messages. Different from Twitter dataset, the messages in this dataset may not contain keywords in the query due to their ranking policy.

The statistics of these two datasets are presented in Table 2. Both datasets consist of very short messages, which have average length with 17 and 22.

5.2 Evaluation of Clustering

In this section, we conduct experiments to verify the effectiveness of our proposed clustering module.

5.2.1 Experimental Setup

To evaluate the performance of the proposed clustering module, we use F_1*measure* [39] and *Accuracy* [40] as the performance metrics, and compare the following methods:

- *BOW*: Traditional "bag of words" model with the *tf-idf* weighting schema.
- *BOP*: Widely used "bag of phrase" model with the *tf-idf* weighting schema. We employ key phrases extracted from original microblogging messages as features to construct the feature space.
- *BOT*: Modification of Tree Kernel model with the *tf-idf* weighting schema. We employ the tree fragments extracted from Section 4.2 to construct the feature space.
- *WN_Method*: *BOW* model integrated with additional features from WordNet as presented in [41].
- *Wiki_Method*: *BOW* model integrated with additional features from Wikipedia as presented in [25].
- *WikiWN_Method*: Only use semantic concepts from WordNet and Wikipedia as features. The feature generation methods are the same as in [41] and [25].

8. http://developers.facebook.com/docs/api/

TABLE 2
Statistics of the Datasets

Dataset	Twitter	Facebook
# of Messages	11,362	14,322
# of Clusters	150	150
# of Unique Words	9,626	10,886
Ave Message Length	17	22
Max Clustering Size	100	191
Ave Clustering Size	75.75	95.48
Min Clustering Size	2	6

- *SemKnow*: *BOP* integrated with additional features from external knowledge. We follow the feature generation and selection methods discussed in [15].
- M^3: Clustering module of the proposed framework.

Note that our proposed text representation framework is independent of any specific dimensionality reduction and clustering methods. Dimensionality reduction methods can be used to improve quality of feature space for clustering, and its effect will be further discussed in Seciton 5.2.4. Similarly, we can easily apply this text representation framework to many clustering methods, such as *K-means*, *LDA* [42], *NMF* [43] etc. Specifically, a clustering package from Weka3 [44] is used in the experiments. Two clustering algorithms, *K-means* and *Expectation Maximization (EM)*, are employed to test effectiveness of the eight text representation methods. As determining the number of clusters in the algorithms is beyond the scope of this study, for general experimental purposes, we set the K as the number of clusters (150) in the datasets for both algorithms.

This experiment involves two parameters, w and θ. As mentioned in Section 4.3, we retrieved top w Wikipedia articles to extract semantic concepts. Given the scale of Wikipedia corpus and the reliable ranking provided by Solr search engine, our experimental result is not sensitive to the number of retrieved documents [45]. We empirically set the value $w = 20$ in the experiment. As discussed in Section 4.4.1, θ controls the influence of semantic features to the whole feature space. We set $\theta = 0.5$ which means that the number of semantic features is the same as the tree fragments from original message.

5.2.2 Evaluation Metrics

In the experiments, two widely used measures, F_1measure [39] and *Accuracy* [40], are employed as the performance metrics, as shown below:

F_1*measure:* A combination of both $precision$ and $recall$ that measures the extent to which a cluster contains only objects of a particular class and all objects of that class. The F_1measure of cluster i with respect to class j is defined as follows:

$$F_1(i,j) = \frac{2 \times precision(i,j) \times recall(i,j)}{precision(i,j) + recall(i,j)}, \tag{5}$$

$$F_1(j) = max_i F_1(i,j). \tag{6}$$

Overall effectiveness is computed as:

$$F_1 measure = \frac{\sum_i (|j| F_1(j))}{\sum_j |j|}. \tag{7}$$

TABLE 3
Clustering Results Using Different Text Representation Methods on Twitter Dataset

	K-means		*Expectation Maximization*	
	F_1measure (Impr)	Accuracy (Impr)	F_1measure (Impr)	Accuracy (Impr)
BOW	0.493 (N.A.)	0.543 (N.A.)	0.530 (N.A.)	0.545 (N.A.)
BOP	0.493 (+0.02%)	0.549 (+1.02%)	0.521 (−0.18%)	0.549 (+0.81%)
BOT	0.504 (+2.27%)	0.556 (+2.29%)	0.538 (+1.51%)	0.554 (+1.70%)
WN_Method	0.499 (+1.28%)	0.553 (+1.85%)	0.531 (+0.13%)	0.550 (+0.92%)
Wiki_Method	0.525 (+6.37%)	0.576 (+5.97%)	0.555 (+4.81%)	0.573 (+5.26%)
WikiWN_Method	0.513 (+4.08%)	0.569 (+4.70%)	0.554 (+4.70%)	0.573 (+5.27%)
SemKnow	0.529 (+7.36%)	0.578 (+6.46%)	0.585 (+10.36%)	0.575 (+5.53%)
M^3	**0.554**(+**12.27%**)	**0.628**(+**15.55%**)	**0.615**(+**16.02%**)	**0.617**(+**13.39%**)

TABLE 4
Clustering Results Using Different Text Representation Methods on Facebook Dataset

	K-means		*Expectation Maximization*	
	F_1measure (Impr)	Accuracy (Impr)	F_1measure (Impr)	Accuracy (Impr)
BOW	0.454 (N.A.)	0.465 (N.A.)	0.503 (N.A.)	0.533 (N.A.)
BOP	0.455 (+0.11%)	0.470 (+0.94%)	0.506 (+0.57%)	0.537 (+0.58%)
BOT	0.464 (+2.22%)	0.474 (+1.95%)	0.513 (+1.95%)	0.544 (+1.93%)
WN_Method	0.461 (+1.54%)	0.471 (+1.31%)	0.513 (+1.95%)	0.544 (+2.04%)
Wiki_Method	0.488 (+7.44%)	0.498 (+7.11%)	0.535 (+6.36%)	0.564 (+5.79%)
WikiWN_Method	0.473 (+4.16%)	0.499 (+7.35%)	0.534 (+6.14%)	0.571 (+7.11%)
SemKnow	0.489 (+7.55%)	0.504 (+8.39%)	0.538 (+6.79%)	0.578 (+8.41%)
M^3	**0.527**(+**15.90%**)	**0.548**(+**17.84%**)	**0.578**(+**14.91%**)	**0.605**(+**13.38%**)

Accuracy: A statistical measure, which is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \tag{8}$$

where TP, TN, FP, FN are defined in Table 5.

In Table 3, TP (true positive) denotes that two texts are manually labeled with the same class and clustered into same cluster; FN (false negative) denotes that two texts are manually labeled with different classes but clustered into same one. TN (true negative) and FP (false positive) are defined in a similar manner.

5.2.3 Clustering Results and Discussion

The experimental results of the different methods on Twitter and Facebook datasets are displayed in Tables 3 and 4. In the tables, Impr represents the percentage improvement of the methods as compared with the BOW model. In the experiment, each result denotes an average of 10 test runs by randomly choosing the initial parameters for the clustering method. By comparing the results of different methods, we draw the following observations:

1) In most cases, *BOP* and *BOT* augment the performance of *BOW* model on both datasets using the two clustering algorithms. We believe that this is because of the utilization of syntactic information from original messages. *BOT* achieves better performance than *BOP*, which is because of the parse tree based Syntactic Decomposition perform finer analysis of original text than shallow parsing based method.
2) From the tables, we note that *WN_Method*, *Wiki_Method*, *SemKnow* also achieve better performance as compared to *BOW* model. The performance of *BOW* is improved by incorporating semantic features from WordNet, Wikipedia and both of them respectively. It demonstrates the integration of semantic concepts from external knowledge bases improved the quality of microblogging messages representation for clustering. Among these three methods, *Wiki_Method* achieves better results than *WN_Method*; we conjecture that the abundant up-to-date concepts from Wikipedia are more informative than others to model the real-time messages.
3) An interesting finding is that *WikiWN_Method* achieves comparable results with other baselines, which is beyond the observation of previous work [21]. *WikiWN_Method* works well without the integration of features from original message. It shows that the combination of semantic features complement each other and contribute to the overall result.
4) Comparing with the other seven methods, M^3 achieves best $F_1 measure$ and $Accuracy$ scores on both datasets using *K-means* and *EM* clustering algorithms. The highest improvement with respect to *BOW* is obtained on Facebook Dataset using *K-means*. We apply t-test to compare M^3 with the best baselines *WikiWN_Method* and *SemKnow*. The results demonstrate our approach significantly outperforms the two

TABLE 5
Average Accuracy Test Condition

	Same Class	**Different Class**
Same Cluster	$TruePositive$	$FalsePositive$
Different Cluster	$FalseNegative$	$TrueNegative$

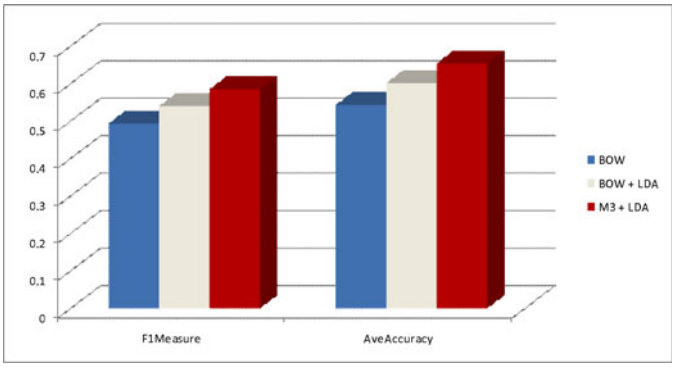


Fig. 3. Clustering results on twitter using LDA.

methods with *p-value* < 0.01. M^3 obtains superior performance as compared with *SemKnow*; we believe the improvement stems from our proposed framework performs better syntactic analysis and map the unstructured text to high quality semantic space.

5.2.4 Effect of Dimensionality Reduction

Latent topic models, such as Principal Component Analysis (PCA), Latent Dirichlet Allocation (LDA) [42] etc., have been successfully applied as an unsupervised dimensionality reduction technique in document collections. Due to their computational cost [46], these methods are not introduced to our framework directly. To better understand the effect of dimensionality reduction methods to clustering performance, we conduct experiments based on different text representation methods, as shown in Fig. 3. As compared to *BOW*, both text representation methods achieve better performance. This result demonstrates that dimensionality reduction method (LDA) is useful to improve the feature space for clustering. M^3 + *LDA* achieves better performance than *BOW* + *LDA*, which indicates that the text representation quality of our proposed M^3 is much better than *BOW*. Therefore, our proposed text representation method is independent of dimensionality reduction methods and can be easily combined with them to achieve better performance, when the application is computation cost insensitive.

5.3 Evaluation of Labeling

In this section, we conduct experiments to verify the effectiveness of our proposed cluster labeling framework.

5.3.1 Experimental Setup and Criteria

We followed the evaluation framework proposed in [47] to assess the quality of the cluster labels. Specifically, we treat the cluster labeling task as a ranking problem, which is to rank all of the concepts from Wikipedia and find the best matched label for a cluster of microblogging messages. We treat the subtopics used for crawling microblogging messages as ground truth for cluster labeling. In our experiment, a generated label is considered correct if it is an inflection, a WordNet synoym of, or identical to the correct label.

Based on the definition above, we evaluate the performance of cluster labeling task by three metrics:

Precision@K: The percentage of labels returned that are correct.

TABLE 6
Labeling Results (Precision @ 10) on Two Dataset

	Twitter	Facebook
Kphrase	0.314 (N.A.)	0.325 (N.A.)
WN	0.322 (+2.55%)	0.303 (−6.77%)
Wiki	0.382 (+21.66%)	0.423 (+29.92%)
M^3	**0.459**(+**46.07%**)	**0.468**(+**43.85%**)

Match@K: Match@K indicates whether the top K generated labels match the correct label. It is a binary indicator, and monotonically increases as K increases.

NDCG: Normalized Discounted Cumulative Gain [48] considers both relevance and position in the ranked list for scoring, which is a widely used metric for measuring performance of ranking problem.

In this experiment, we compare the performance of four methods, as defined below:

- *Kphrase*: Traditional "bag of phrases" model is used to generate the most frequent phrase as cluster label.
- *WN*: The concepts extracted from WordNet are used to generate the cluster label. The feature generation method is proposed in [41].
- *Wi-ki*: The concepts extracted from Wikipeida are used to generate the cluster label. The feature generation method is proposed in [25].
- M^3: Labeling module of the proposed framework.

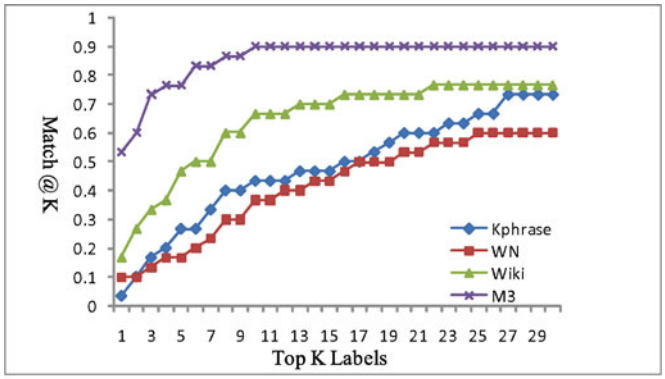
5.3.2 Labeling Results and Discussion

The Precision@10 results of the different labeling methods on Twitter and Facebook datasets are reported in Table 6.

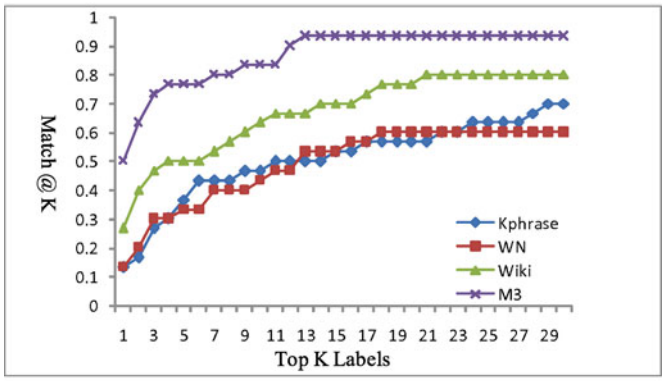
The results of *Kphrase* and *WN* are unsatisfactory, which is mainly because the frequent phrases from original microblogging messages and synonymy words from WordNet are always noisy and meaningless. The labeling quality of *Wi-ki* is more effective than the first two baselines. It shows that, by providing meaningful concepts, Wikipedia has its natural advantage to tackle labeling problems. In addition, M^3 further improves the labeling precision as compared with *Wi-ki*. We believe it stems from our framework providing a better mapping from original unstructured space to structured semantic space.

Fig. 4 depicts the Match@K results on the two datasets, respectively. From Fig. 4a, M^3 is much more effective than the other three baselines. The performance of M^3 increases along with the number of labels K increase, and it achieves best performance at $K = 10$ and $Match@K = 83.3\%$. When $K > 10$, the performance becomes stable, which indicates that no more clusters that can be covered by the correct label with the number of labels increasing. Among the baselines, *Wi-ki* is the most effective method and *Kphrase* achieves close results to *WN*. Similar phenomena have been observed on Facebook Dataset; we omit the results owing to lack of space.

It is noted that the curve of M^3 peaks and stabilizes at a much smaller number of labels $K = 10$ (Twitter) and $K = 12$ (Facebook) compared with other baselines. This indicates that our framework is much more robust than others when only choosing a small number of labels.



(a) Twitter



(b) Facebook

Fig. 4. Labeling results (Match@K) on two dataset.

5.3.3 Ranking Results

We compare the ranking performance of our proposed framework with the other three methods. Table 7 shows the NDCG@10 score on the two datasets respectively.

From Table 7, we can observe that M^3 outperforms the other three baselines. It demonstrates that the generated labels from M^3 not only cover more potential topics hidden in the microblogging messages, but also assign the most relevant labels at a higher position. Among the three baselines, *Wiki* achieves the best performance. We believe that the improvement stems from the structure and meaningful concepts providing by Wikipedia.

5.4 A Usability Case Study

To illustrate the usability of our proposed framework, we show an example of top-five generated textual labels and their corresponding sample tweets for a trending topic "Apple" in Table 8. In the table, subtopics listed in the left side are considered as "correct labels". The underlined labels mean "identical" to correct labels and the ones with daggers mean "inflection" of correct labels. We observe that while the labels for all clusters seem to represent the subtopics well, only the last cluster fails to achieve correct label within top-five labels, although most of the generated labels ("Apple Care", "Customer Support" etc.) are highly related to subtopic "Apple Support". The failure is mainly because that there is no corresponding Wikipedia page named "Apple Support".

TABLE 7
Ranking Results (NDCG@10) on Two Dataset

	Twitter	Facebook
Kphrase	0.342 (N.A.)	0.322 (N.A.)
WN	0.338 (−1.17%)	0.335 (+4.04%)
Wiki	0.436 (+27.49%)	0.448 (+39.13%)
M^3	**0.498**(+**45.61%**)	**0.506**(+**57.14%**)

6 Conclusions and Future Work

In this paper, we proposed a novel framework to enhance the accessibility of microblogging messages by utilizing semantic knowledge. In particular, we improved the quality of microblogging message clustering and labeling. By analyzing the structure of microblogging messages, the original short and noisy texts were mapped into a semantic space to improve the quality of text representation. The features from original text and semantic knowledge bases tackled the problem of data sparseness and semantic gap well in natural microblogging messages. With help of abundant structured features from Wikipedia, the task of cluster labeling was solved without introducing much computational cost. Empirical evaluations demonstrated that our framework significantly outperformed all the baselines including previously proposed linguistic based and knowledge based methods on two real-world datasets.

This work suggests some interesting directions for future work. As this work is for improving the management of microblogging messages, which are connected from their authors' point of view. It is interesting to explore if integrating social network information can improve the quality of message clustering. Moreover, different from other web texts, microblogging messages are really natural language produced by users. Thus we can introduce more NLP techniques to tackle the problems in Text Mining area. NLP and

TABLE 8
Lists of Top-Five Labels Generated From M^3 and Corresponding Sample Tweets

Subtopics	Generated Top-5 Labels	Sample Tweets
Apple Store	AppleStore, Retail Store, Apple Inc., Steve Jobs, iPad	Apple can make a great phone, but Steve Jobs needs to reign in the App Store developers.
Apple TV	AppleTV, iTunes, Apple Inc., iTunes Store, Digital Media Receiver	RT @igiveaway: Checkout Face Quiz for iPhone for a chance to win a Apple TV
Apple iPad	iPad Air, iPad†, Tablet Computer, Apple A5 Processor, Foxconn	The Foxconn Explosion Might've Just Made iPad Air Lines Millions of People Longer
Apple Trailers	Trailer†, QuickTime, Mac OS, Trailer Film, Apple Inc.	Bill Cunningham New York—Movie Trailers—iTunes
Apple Support	Apple Care, Apple Inc., iPod Customer Support, Apple Store	Apple continues to tell support reps: do not help customers with Mac malware

external knowledge bases can be valuable to help understand microblogging messages, if we can find effective ways of using them.

ACKNOWLEDGMENTS

This material is based upon work supported by, or in part by, the National Science Foundation (NSF) under grant number IIS-1217466, and the Office of Naval Research (ONR) under grant number N000141410095.

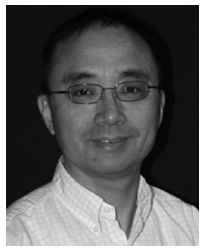
REFERENCES

[1] Y. Ko and J. Seo, "Automatic text categorization by unsupervised learning," in *Proc. 18th Conf. Comput. linguistics-Vol. 1*, 2000, pp. 453–459.
[2] Y. Song, S. Pan, S. Liu, M. X. Zhou, and W. Qian, "Topic and keyword re-ranking for LDA-based topic modeling," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1757–1760.
[3] Y. Hu, A. John, F. Wang, and S. Kambhampati, "ET-LDA: Joint topic modeling for aligning events and their twitter feedback," in *Proc. 26th Conf. Artif. Intell.*, vol. 12, pp. 59–65, 2012.
[4] L. Hong and B. D. Davison, "Empirical study of topic modeling in Twitter," in *Proc. 1st Workshop Social Media Analytics*, 2010, pp. 80–88.
[5] D. Ramage, S. Dumais, and D. Liebling, "Characterizing microblogs with topic models," in *Proc. Int. AAAI Conf. Weblogs Soc. Media*, vol. 5, no. 4, pp. 130–137, 2010.
[6] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Identifying influencers on Twitter," in *Proc. 4th ACM Int. Conf. Web Seach Data Mining*, 2011, pp. 65–74.
[7] B. O Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," in *Proc. Int. AAAI Conf. Weblogs Soc. Media*, 2010, pp. 122–129.
[8] X. Hu, J. Tang, H. Gao, and H. Liu, "Unsupervised sentiment analysis with emotional signals," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 607–618.
[9] X. Hu, L. Tang, J. Tang, and H. Liu, "Exploiting social relations for sentiment analysis in microblogging," in *Proc. 6th ACM Int. Conf. Web Search and Data Mining*, 2013, pp. 537–546.
[10] C. X. Lin, B. Zhao, Q. Mei, and J. Han, "PET: A statistical model for popular events tracking in social communities," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 929–938.
[11] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: Real-time event detection by social sensors," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 851–860.
[12] H. Huang, S. Anzaroot, H. Ji, H. Le, D. Wang, and T. Abdelzaher, "Free-form text summarization in social sensing," in *Proc. 11th Int. Conf. Inf. Process. Sens. Netw.*, 2012, pp. 141–142.
[13] D. R. Cutting, D. R. Karger, and J. O. Pedersen, "Constant Interaction-time scatter/gather browsing of very large document collections," in *Proc. 16th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1993, pp. 126–134.
[14] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma, "Learning to cluster web search results," in *Proc. 27th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2004, pp. 210–217.
[15] X. Hu, N. Sun, C. Zhang, and T.-S. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 919–928.
[16] D. D. Lewis and W. B. Croft, "Term clustering of syntactic phrases," in *Proc. 13th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1989, pp. 385–404.
[17] H. Chim and X. Deng, "Efficient phrase-based document similarity for clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1217–1229, Sep. 2008.
[18] A. Tagarelli and G. Karypis, "A segment-based approach to clustering multi-topic documents," in *Proc. Text Mining Workshop, SIAM Datamining Conf.*, 2008, pp. 563–595.
[19] J. Hu, L. Fang, Y. Cao, H.-J. Zeng, H. Li, Q. Yang, and Z. Chen, "Enhancing text clustering by leveraging Wikipedia semantics," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 179–186.
[20] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen, "Short text conceptualization using a probabilistic knowledgebase," in *Proc. 22nd Int. Joint Conf. Artif. Intell.-Volume Three*, 2011, pp. 2330–2336.
[21] E. Gabrilovich and S. Markovitch, "Feature generation for text categorization using world knowledge," in *Proc. Int. Joint Conf. Artif. Intell.*, vol. 19, p. 1048, 2005.
[22] D. Carmel, H. Roitman, and N. Zwerdling, "Enhancing cluster labeling using Wikipedia," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 139–146.
[23] X. Hu, X. Zhang, C. Lu, E. K. Park, and X. Zhou, "Exploiting Wikipedia as external knowledge for document clustering," in *Proc. 15th ACM SIGKDD*, 2009, pp. 389–396.
[24] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to classify short and sparse text & web with hidden topics from large-scale data collections," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 91–100.
[25] S. Banerjee, K. Ramanathan, and A. Gupta, "Clustering short texts using Wikipedia," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007, pp. 787–788.
[26] U. S. Kohomban and W. S. Lee, "Learning semantic classes for word sense disambiguation," in *Proc. 43rd Annu. Meeting Assoc. Comput. Linguistics*, 2005, pp. 34–41.
[27] M. Collins, N. Duffy, et al., "Convolution kernels for natural language," in *Proc. 14th Conf. Neural Inf. Process. Syst.*, 2001, vol. 14, pp. 625–632.
[28] C. Manning, H. Schutze, and MITCogNet, *Foundations of Statistical Natural Language Processing*. Cambridge, MA, USA: MIT Press, 1999, vol. 59.
[29] K. Wang, Z. Ming, and T. Chua, "A syntactic tree matching approach to finding similar questions in community-based QA services," in *Proc. 32nd ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 187–194.
[30] Y. Song and D. Roth, "Unsupervised sparse vector densification for short text similarity," in *Proc. North Am. Chapter Assoc. Computat. Linguistics*, 2015, pp. 1275–1280.
[31] T. Marinis, "Psycholinguistic techniques in second language acquisition research," *Second Lang. Res.*, vol. 19, no. 2, p. 144, 2003.
[32] P. Wang, J. Hu, H.-J. Zeng, L. Chen, and Z. Chen, "Improving text classification by using encyclopedia knowledge," in *Proc. 7th IEEE Int. Conf. Data Mining*, 2007, pp. 332–341.
[33] S. Osinski, J. Stefanowski, and D. Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition," in *Proc. IIS Conf.*, 2004, p. 359.
[34] D. Zhang and W. S. Lee, "Question classification using support vector machines," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval.*, 2003, pp. 26–32.
[35] X. Hu and H. Liu, "Text analytics in social media," *Mining Text Data*, pp. 385–414, 2012.
[36] X. Hu, J. Tang, Y. Zhang, and H. Liu, "Social spammer detection in microblogging," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2633–2639.
[37] Y. Chen, Z. Li, L. Nie, X. Hu, X. Wang, T.-S. Chua, and X. Zhang, "A semi-supervised Bayesian network model for microblog topic classification," in *Proc. 24th Int. Conf. Comput. Linguistics*, 2012, pp. 561–576.
[38] J. Martin, *Introduction to Languages and the Theory of Computation*. New York, NY, USA: McGraw-Hill, 2002.
[39] B. Larsen and C. Aone, "Fast and effective text mining using linear-time document clustering," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 1999, pp. 16–22.
[40] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2006.
[41] A. Hotho, S. Staab, and G. Stumme, "Ontologies improve text document clustering," in *Proc. 3rd IEEE Int. Conf. Data Mining*, 2003, pp. 541–544.
[42] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet allocation," *The J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[43] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2003, pp. 267–273.
[44] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2005.
[45] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," in *Proc. 16th Int. Conf. World Wide Web*, 2007, vol. 7, pp. 757–786.

[46] D. Sontag and D. Roy, "Complexity of inference in topic models," in *Proc. NIPS Workshop Appl. Topic Models: Text and Beyond*, 2009, pp. 1–5.
[47] P. Treeratpituk and J. Callan, "Automatically labeling hierarchical clusters," in *Proc. Int. Conf. Digit. Government Res.*, 2006, pp. 167–176.
[48] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, pp. 422–446, Oct. 2002.

Xia Hu received the BS and MS degrees in computer science from Beihang University, China and the PhD degree in computer science and engineering from Arizona State University. He is currently an assistant professor at the Department of Computer Science and Engineering, Texas A&M University. His research interests are in data mining, social network analysis, machine learning, etc. As a result of his research work, he has published nearly 40 papers in several major academic venues, including WWW, SIGIR, KDD, WSDM, IJCAI, AAAI, CIKM, SDM, etc. One of his papers was selected in the Best Paper Shortlist in WSDM'13. He is the recipient of the 2014 ASU's Presidents Award for Innovation, and Faculty Emeriti Fellowship. He has served on program committees for several major conferences such as IJCAI, SDM and ICWSM, and reviewed for multiple journals, including *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Information Systems* and *Neurocomputing*. His research attracts wide range of external government and industry sponsors, including US National Science Foundation (NSF), ONR, AFOSR, Yahoo!, and Microsoft. Updated information can be found at http://www.public.asu.edu/∼xiahu. He is a member of the IEEE.

Lei Tang received the BS degree in computer science from Fudan University, China in 2004, and the PhD degree in computer science from Arizona State University in 2010. He is the Chief Data Scientist at Clari Inc., a leader in predictive analytics and sales management solutions. His research interests include social computing, data mining, and computational advertising. He has published four book chapters and more than 30 peer-reviewed papers at prestigious conferences and journals related to data mining. His book on *Community Detection and Mining in Social Media* is the top download in the data mining and knowledge discovery lecture series. For more information, please visit his homepage: http://leitang.net. He is a member of the IEEE.



Huan Liu received the BEng degree in computer science and electrical engineering at Shanghai JiaoTong University and the PhD degree in computer science at the University of Southern California. He is a professor of computer science and engineering at Arizona State University. Before he joined ASU, he worked at Telecom Australia Research Labs and was on the faculty at the National University of Singapore. He was recognized for excellence in teaching and research in computer science and engineering at Arizona State University. His research interests are in data mining, machine learning, social computing, and artificial intelligence, investigating problems that arise in many real-world, data-intensive applications with high-dimensional data of disparate forms such as social media. His well-cited publications include books, book chapters, encyclopedia entries as well as conference and journal papers. He serves on journal editorial boards and numerous conference program committees, and is a founding organizer of the International Conference Series on Social Computing, Behavioral-Cultural Modeling, and Prediction (http://sbp.asu.edu/). Updated information can be found at http://www.public.asu.edu/∼huanliu. He is a fellow of the IEEE and an ACM distinguished scientist.