Enhancing Accessibility of Microblogging Messages Using Semantic Knowledge

Xia Hu
Arizona State University
Tempe, AZ 85287, USA
xiahu@asu.edu

Lei Tang
Yahoo! Labs
Santa Clara, CA 95054, USA
ltang@yahoo-inc.com

Huan Liu
Arizona State University
Tempe, AZ 85287, USA
huan.liu@asu.edu

ABSTRACT

The volume of microblogging messages is increasing exponentially with the popularity of microblogging services. With a large number of messages appearing in user interfaces, it hinders user accessibility to useful information buried in disorganized, incomplete, and unstructured text messages. In order to enhance user accessibility, we propose to aggregate related microblogging messages into clusters and automatically assign them semantically meaningful labels. However, a distinctive feature of microblogging messages is that they are much shorter than conventional text documents. These messages provide inadequate term co-occurrence information for capturing semantic associations. To address this problem, we propose a novel framework for organizing unstructured microblogging messages by transforming them to a semantically structured representation. The proposed framework first captures informative tree fragments by analyzing a parse tree of the message, and then exploits external knowledge bases (Wikipedia and WordNet) to enhance their semantic information. Empirical evaluation on a Twitter dataset shows that our framework significantly outperforms existing state-of-the-art methods.

Keywords

Microblogging, Accessibility, Clustering, Labeling

1. INTRODUCTION

Microblogging services such as Twitter[1] are increasingly used for communicating breaking news, information sharing, and participating in events. This emerging medium has become a powerful communication channel in recent digital revolutions. However, the accessibility of these messages has been very limited so far. Tweets and retweets of a user's followees appear alongside the user's own tweets in reverse chronological order. People often have only enough patience to skim through the first 20 - 50 messages. When the messages become overwhelming, it is impractical for a user to quickly gauge the main subjects from their followees' posts.

[1]http://www.twitter.com/

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. Enhancing the accessibility of microblogging messages entails two tasks: (1) cluster microblogging messages into manageable categories, and (2) assign readable and meaningful labels for each cluster of messages. Unlike standard text with many sentences or paragraphs, microblogging messages are *noisy* and *short*. In addition, microbloggers, when composing a message, may use or coin new abbreviations or acronyms that are uncommon in conventional text documents. Furthermore, these short messages do not provide sufficient contextual information to capture their semantic meanings. Traditional text mining methods, when applied to microblogging messages directly, lead to unsatisfactory results.

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. 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.

2. MANAGING MICROBLOGGING MESSAGES

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

2.1 Problem Statement

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.

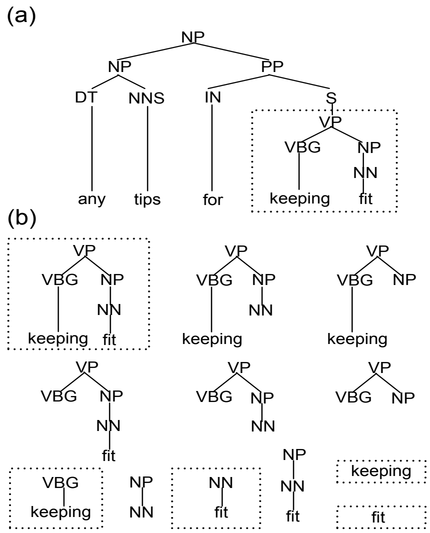

Figure 1: (a) The parse tree of "any tips for keeping fit?". (b) Tree fragments of the subtree covering "keeping fit"; the fragments with dotted line frame are extracted tree fragments for "keeping fit".

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.

2.2 Syntactic Decomposition

Many NLP techniques have achieved great success by extracting tree fragments that occur in a parse tree to enrich text representation. 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. Figure 1 (a) illustrates an example of a parse tree generated by OpenNLP[2]. In the Figure, "VP" and "NP" represent verb phrase and noun phrase[3], respectively.

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. [6] proposed to employ tree fragments as syntactic features.

Figure 1 (b) presents an illustration of the tree fragments for subtree "keeping fit". Basically, we divided our algorithm into two steps: *Subtree Selection* and *Fragment Selection*.

Subtree Selection: As shown in Figure 1 (a), 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" 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. Among these key phrases, the nouns and verbs are considered to be more important than articles, adjectives or adverbs [6]. Thus, we utilize VP (Verb Phrase), NP (Noun Phrase), VB (Verb) and NN (Noun) rooted subtrees to extract tree fragments in next step.

Fragment Selection: As shown in Figure 1 (b), 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.

2.3 Semantic Mapping

In order to transform the syntactic feature space to a semantic feature space, we collect the extracted tree fragments as a basis and construct semantic space for mapping. For each tree fragment, we apply semantic knowledge according to its syntax property. Phrase-level tree fragments are informative to represent a subtopic of the microblogging message. In this way, we can retrieve accurate Wikipedia pages for these tree fragments. The word-level tree fragments are too general to map accurately to concepts in Wikipedia. We thus utilize WordNet as complement to deal with the word-level tree fragments.

Particularly, if a tree fragment is from the phrase-level, we build an "AND" query[4] which requires the retrieved pages to contain every term in the phrase. We utilize the title and bold terms (links) of the retrieved articles, combined with key phrases as semantic features. 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 onto the same semantic concept.

2.4 Clustering & Labeling

2.4.1 Feature Selection

We conduct feature selection to avoid aggravating the "curse of dimensionality". A single 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) contained in the message.

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

$$n = \frac{m \times \theta}{1 - \theta}, \qquad (1)$$

where θ is the ratio of semantic concepts to the feature space for clustering. In the experiments, we empirically set $\theta = 0.5$.

2.4.2 Text Representation for Clustering

To normalize the weight of each feature, we reformulate the weighting policy proposed by Zhang and Lee [7]. For

[2] http://incubator.apache.org/opennlp/

[3] Full list of the abbreviations can be found in http://en.wikipedia.org/wiki/Parse_tree/

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

Table 1: Clustering results using different text representation methods on Twitter Dataset

	F_1measure (Impr)	Accuracy (Impr)
BOW	0.493 (N.A.)	0.543 (N.A.)
BOT	0.504 (+2.27%)	0.556 (+2.29%)
WN_Method	0.499 (+1.28%)	0.553 (+1.85%)
Wiki_Method	0.525 (+6.37%)	0.576 (+5.97%)
WikiWN_Method	0.513 (+4.08%)	0.569 (+4.70%)
SemKnow	0.529 (+7.36%)	0.578 (+6.46%)
M^3	**0.554 (+12.27%)**	**0.628 (+15.55%)**

tree fragments f_i extracted from original parse tree, f_i is weighted according to the size and depth of a tree fragment:

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

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 Figure 1 (b) has $s(i) = 3$ and $d(i) = 3$. With this weighting scheme, the focus of the message can be measured according to its depth. 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. The result is that messages are represented in a refined feature space.

2.4.3 Labeling

Traditional labeling methods do not guarantee readability of the extracted labels. It is natural and effective to generate textual labels from the generated Wikipedia concepts, which have wide knowledge coverage and stably high quality.

We can map each tree fragment f_i to several semantic concepts, which are extracted as label candidates $\{l_{i1}, l_{i2}, \ldots, l_{in}\}$. For each labeling candidate l_{ij}, the informativeness score is measured by:

$$Info_{l_{ij}} = W_{f_i} \times tf_{ij} \times idf_{ij}, \quad (3)$$

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.

3. EXPERIMENTS

In this section, we empirically evaluate the effectiveness of the proposed Microblogging Message Management (M^3) framework.

3.1 Datasets

We crawled the hot queries published by Google Trends[5] between Jan. 1st 2008 and Dec. 31st 2010, and chose hot queries of different lengths according to statistical results. Thirty hot queries of diverse topics are selected from Google Trends. Each hot query is considered to be a trending topic, and we crawl the top five query suggestions from Google as subtopics of this topic. 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 5 subtopics in each group). Based on the query suggestions (subtopics), we use Twitter Search API[6] to crawl 100 tweets for each query suggestion and construct a dataset containing 150 categories. As the API will not return exactly 100 tweets for each query, it leaves 11362 tweets after text preprocessing.

3.2 Evaluation of Clustering

3.2.1 Experimental Setup

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

- *BOW*: Traditional "bag of words" model.
- *BOT*: Modification of Tree Kernel model.
- *WN_Method*: *BOW* model integrated with additional features from WordNet as presented in [3].
- *Wiki_Method*: *BOW* model integrated with additional features from Wikipedia as presented in [1].
- *WikiWN_Method*: Semantic concepts from WordNet [3] and Wikipedia [1] as features.
- *SemKnow*: The "bag of phrases" model integrated with additional features from external knowledge [4].
- 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. Similarly, we can easily apply this text representation framework to many clustering methods, such as *K-means*, *LDA*, *NMF* etc. In the experiments, *K-means* is employed and we set number of clusters $k = 150$.

3.2.2 Clustering Results and Discussion

The experimental results of the different methods on the dataset are displayed in Table 1. Based on the results, we make the following observations:

(1) *BOT* augments the performance of *BOW* model on the dataset. We believe that this is because of the utilization of syntactic information from the original messages. We note that *WN_Method*, *Wiki_Method*, *SemKnow* also achieve better performance as compared to *BOW* model. It demonstrates that the integration of semantic concepts from external knowledge bases improved the quality of the representation of microblogging messages for clustering.

(2) An interesting finding is that *WikiWN_Method* achieves comparable results with other baselines, which is beyond the observations of previous work [2]. *WikiWN_Method* works well without the integration of features from the original message. It shows that the combination of semantic features complement each other and contribute to the overall result.

[5]http://www.google.com/intl/en/trends/about.html/

[6]http://search.twitter.com/api/

Table 2: Ranking Results (NDCG@10)

	NDCG@10
Kphrase	0.342 (N.A.)
WN	0.338 (-1.17%)
Wiki	0.436 (+27.49%)
M^3	**0.498 (+45.61%)**

Table 3: Lists of top-5 labels generated from M^3

Subtopics	Generated Top-5 Labels
Apple Store	<u>Apple Store</u>, Retail Store, Apple Inc., Steve Jobs, iPad
Apple TV	<u>Apple TV</u>, iTunes, Apple Inc., iTunes Store, Digital Media Receiver
Apple iPad	iPad 2, iPad†, Tablet Computer, Apple A5 Processor, Foxconn
Apple Trailers	Trailer†, QuickTime, Mac OS, Trailer Film, Apple Inc.
Apple Support	Apple Care, Apple Inc., iPod Customer Support, Apple Store

(3) Among all the methods, M^3 achieves the best performance. 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 methods with $p - value < 0.01$.

3.3 Evaluation of Labeling

3.3.1 Experimental Setup and Criteria

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. The subtopics used for crawling microblogging messages are considered to be ground truth for cluster labeling. We use NDCG as the evaluation metric. We compare the performance of following methods:

- *Kphrase*: Traditional "bag of phrases" model.
- *WN*: The concepts extracted from WordNet [3].
- *Wiki*: The concepts extracted from Wikipeida [1].
- M^3: Labeling module of the proposed framework.

3.3.2 Ranking Results

We compare the ranking performance of our proposed framework with the other three methods. Table 2 shows NDCG@10 score of the four methods on the dataset.

From Table 2, we can observe that M^3 outperforms all the baselines. It demonstrates that the generated labels from M^3 not only cover more potential topics hidden in the microblogging messages, but also give the most relevant labels a higher ranking. Among the three baselines, $Wiki$ achieves the best performance. We believe that the improvement stems from the structure and meaningful concepts providing by Wikipedia.

3.4 A Usability Case Study

To illustrate the usability of our proposed framework, we show an example of top-5 generated textual labels for a trending topic "Apple" in Table 3. In the table, subtopics listed in the left side are considered "correct labels". The underlined labels are "identical" to correct labels and those with daggers are "inflections" 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-5 labels, although most of labels are highly related to subtopic "Apple Support". The failure is mainly because that there is no corresponding Wikipedia page named "Apple Support".

4. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a novel framework to improve the performance 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 for clustering. In addition, with help of abundant structured features from Wikipedia, the task of cluster labeling was solved without introducing significant computational cost. Empirical evaluations demonstrated that our framework significantly outperformed existing state-of-the-art methods.

This work suggests some interesting directions for future work. For example, it is interesting to explore if integrating social network information can improve the quality of message clustering. Moreover, NLP and external knowledge bases can be valuable to help understand microblogging messages, if we can find effective ways to use them.

Acknowledgement

This work is, in part, supported by ONR (N000141010091), AFOSR (FA95500810132) and Yahoo! Faculty Research and Engagement program.

5. REFERENCES

[1] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using Wikipedia. In *Proceedings of the 30th ACM SIGIR*, pages 787–788, 2007.

[2] E. Gabrilovich and S. Markovitch. Feature generation for text categorization using world knowledge. In *Proceedings of the 20th AAAI*, pages 1048–1153, 2005.

[3] A. Hotho, S. Staab, and G. Stumme. Wordnet improves text document clustering. In *Proceedings of the SIGIR 2003 Semantic Web Workshop*, pages 541–544, 2003.

[4] 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 *Proceeding of the 18th ACM CIKM*, pages 919–928, 2009.

[5] P. Treeratpituk and J. Callan. Automatically labeling hierarchical clusters. In *Proceedings of the 2006 international conference on Digital government research*, pages 167–176, 2006.

[6] K. Wang, Z. Ming, and T. Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. In *Proceedings of the 32nd ACM SIGIR*, pages 187–194, 2009.

[7] D. Zhang and W. Lee. Question classification using support vector machines. In *Proceedings of the 26th ACM SIGIR*, pages 26–32, 2003.