A Semi-Supervised Bayesian Network Model for Microblog Topic Classification

Yan Chen[1,2] *Zhoujun Li*[1] *Liqiang Nie*[2] *Xia Hu*[3]
Xiangyu Wang[2] *Tat − Seng Chua*[2] *Xiaoming Zhang*[1]
(1) State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
(2) School of Computing, National University of Singapore, Singapore
(3) Arizona State University, United States
chenyan@cse.buaa.edu.cn, lizj@buaa.edu.cn, nieliqiang@gmail.com,
xia.hu@asu.edu, dcswangx@nus.edu.sg, dcscts@nus.edu.sg,
yolixs@buaa.edu.cn

Abstract

Microblogging services have brought users to a new era of knowledge dissemination and information seeking. However, the large volume and multi-aspect of messages hinder the ability of users to conveniently locate the specific messages that they are interested in. While many researchers wish to employ traditional text classification approaches to effectively understand messages on microblogging services, the limited length of the messages prevents these approaches from being employed to their full potential. To tackle this problem, we propose a novel semi-supervised learning scheme to seamlessly integrate the external web resources to compensate for the limited message length. Our approach first trains a classifier based on the available labeled data as well as some auxiliary cues mined from the web, and probabilistically predicts the categories for all unlabeled data. It then trains a new classifier using the labels for all messages and the auxiliary cues, and iterates the process to convergence. Our approach not only greatly reduces the time-consuming and labor-intensive labeling process, but also deeply exploits the hidden information from unlabeled data and related text resources. We conducted extensive experiments on two real-world microblogging datasets. The results demonstrate the effectiveness of the proposed approaches which produce promising performance as compared to state-of-the-art methods.

Title and abstract in Chinese

基于半监督贝叶斯网络模型的微博主题分类模型研究

微博作为一种新型的社会媒体，其海量数据及展现主题的多样性使得用户要找到感兴趣主题的微博非常困难。 当前的一些研究主要采用面向长文本的挖掘方法来分类微博的主题，但微博作为一种短文本，其稀疏性和用词不规范严重影响了这些方法的性能。 针对微博的特点，本文提出了一种半监督贝叶斯网络模型，其充分利用外部相关的网络资源来丰富微博的文本特征， 并利用少量的未标注数据以及辅助的外部资源来预测大量待标注微博数据的主题。 本文的方法不仅能减少繁琐的人工标注过程，而且能够从未标注的微博数据以及相关资源中挖掘出微博隐藏主题的相关语义信息。 我们的实验基于Twitter和新浪微博两个数据集，试验结果表明，与目前的方法相比，本文提出方法的性能有显著提高。

Keywords: Semi-supervised algorithm, microblog classification, probabilistic graph model.

Keywords in L_2: 半监督分类算法，微博，话题分类，概率图模型.

1 Introduction

Microblogging services are becoming immensely popular in breaking-news disseminating, information sharing, and events participation. This enables users to express their thoughts and intentions in short textual snippets on a daily and even hourly basis. The most well-known one is Twitter (`www.twitter.com`), which has more than 140 million active users with 1 billion Tweets every 3 days[1] as of March 2012. Over time, a tremendous number of messages have been accumulated in their repositories, which greatly facilitate general users seeking information by querying their interested topics using the corresponding hashtag.

However, users often have to browse through large amount of results in order to find the information of their interests. This is due to the ambiguous hashtag and the presentation style. The microblogging platforms mix search results in a ranked list, determined by their relevance to the corresponding hashtag and published time. Unfortunately, most hashtags are very short, ambiguous and even vague, leading to unsatisfactory search results. For example, the returned list for queried hashtag "#apple" is extremely messy and diversified, potentially covering several different sub-topics: smartphone, computer, fruit and so on. In this case, users can benefit from overviews of search results based on meaningful and structural categories, such as, grasping at a glance the spread of categories covered by a given search topic and quickly locating the information of their interests with the assistance of the labeled categories. This is especially important for mobile search through handheld devices such as smartphones.

Classifying microblogs into pre-defined subtopic-oriented classes poses new challenges due to the following reasons. First, unlike normal documents, these messages are typically short, consisting of no more than 140 characters. They thus do not provide sufficient word co-occurrences or shared contexts for effective similarity measure (Hu et al., 2009). The data sparseness hinders general machine learning methods to achieve desirable accuracy. Second, microblogging messages are not well conformed as standard structures of documents. Sometimes they do not even obey grammatical rules (Hu and Liu, 2012b). Third, microblogs lack label information. It is time and labor consuming to label the huge amounts of messages.

Intensive efforts have been made on the classification of short texts utilizing machine learning techniques (Nie et al., 2011). Some representative research efforts are based on topic model (Ramage et al., 2009) (Zhao et al., 2011). As these approaches heavily rely on the term co-occurrence information, the sparsity of short and informal messages unduly influence the significant improvement of the performance. Some others explore some traditional supervised learning methods to classify microblogging messages (Lee et al., 2011) (Zubiaga et al., 2011) (Sriram et al., 2010) (Tang et al., 2012). The sparsity problem again hinders the similarity measurement. Moreover, it is laborious and time consuming to obtain labeled data from microblogging. Consequently, new approaches towards microblog classification are highly desired.

In this paper, we propose a semi-supervised learning approach to the classification of microblogging messages. We aim to tackle three challenges in this paper. First, to handle the data sparseness problem, our approach submits a query that is related to hashtag and category to Google Search Engine; meanwhile it incorporates the external information provided by search engine results to enrich the short microblogs. Second, to alleviate negative effect brought by informal words in microblogging, we employ linguistic corpus to detect informal words in microblogging messages and correct them into formal expressions. Third, with the integration of hashtag related resources,

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

our model is robust with only a small amount of training data, which greatly reduces the manually labeling costs. Our algorithm alternates between performing an E-step and M-step. Specifically, it first trains a classifier based on the available labeled messages as well as some auxiliary cues mined from the web, and probabilistically predicts the class labels of the unlabeled messages. It then trains a new classifier using all messages and the auxiliary cues, and iterates to convergence. We conduct experiments on the real-world datasets, and demonstrate that our proposed scheme yields significant accuracy in microblogging messages categorization.

The main contributions of this research can be summarized as follows,

- To the best of our knowledge, this work is the first attempt towards microblogs categorization using semi-supervised learning approach, which requires less labeled data and can thus be practically extended to large-scale datasets.
- Our approach incorporates external statistical knowledge to enrich the short microblogs, which greatly remedies the data sparseness issue.
- Our approach adopts a category-word distribution analysis, which well addresses the broader phenomenon existed in microblogs: non-standard language presentation and abundant spelling errors.

The reminder of this paper is organized as follows: we introduce the details of our proposed approach and experimental results in Section 2 and Section 3 respectively. In section 4, we briefly reviews of the related work, followed by concluding remarks in Section 5.

2 Semi-Supervised Graphical Model for Microblogs Classification

Before formulating our approach, we first define some notations. A set of messages is collected by a given hashtag t, which are partitioned into two subsets: a labeled set $M^l = \{m_1, m_2, \ldots, m_L\}$ and an unlabeled set $M^u = \{m_{L+1}, m_{L+2}, \ldots, m_{L+N}\}$. M^l includes only the example messages provided through user interaction, where each instance is associated with a predefined category c_i with belonging to $C = \{c_1, c_2 \ldots c_K\}$; while M^u includes all the other messages. We aim to predict the category label for each data point in M^u. Here we assume that each tweet belongs to only one category. Similar idea of assigning a single topic or category to a short sequence of words has been used before in (Diao et al., 2012) (Gruber et al., 2007) (Zhao et al., 2011).

2.1 The General Framework

We now introduce the overview of the whole processing that aims to classify microblogging messages by exploiting the internal and external resources. The workflow consists of three phrases, as shown in Figure 1. It includes the preprocessing of external resources, preprocessing of microblogging messages, and construction of Semi-Supervised Bayesian Network (SSBN) model.

Phrase 1: Preprocessing of External Resources Due to their short length, microblogging messages do not provide sufficient word co-occurrence or context shared information for effective similarity measure. Thus we utilize the external Google Search snippets to enrich the original feature space of the microblogging messages. The procedure of enrichment is as follows: we mine the list of hot topics from Google such as Apple, Obama, NBA, Facebook, *etc*. For each hot topic, we search them as hashtags for microblog messages from Twitter. The results contain a list of proper sub-hashtags, such as stock, ipad, ipo, app, ticket, education, *etc*. These sub-hashtags are manually

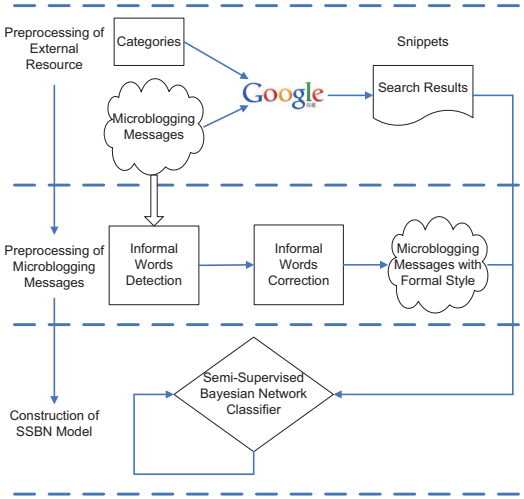


Figure 1: The General Framework

classified into K pre-defined categories. For a given hashtag t (for example, stock), we build K hashtag-category pairs (for example, stock Sports, stock Business, *etc*.), and consider each pair as a query to return 20 extended documents from Google Search Engine, denoted as S. Comparing with the way that only takes each hashtag as a query, the combination of hashtag and category can find more accurate documents. Next, we assign the tf.idf weight of each word for each category in S. We further use the google search results to estimate the category prior distribution.

Phrase 2: Preprocessing of Microblogging Messages It is worth noting that there is a large amount of misspelled and informal expressions in microblogging messages. This is different from the formal expressions and words used in Google Search results. To handle this mismatch problem, we first construct a microblog dictionary containing all the abbreviate forms of words used in Twitter from some dictionaries, such as Twitternary[2], twitterforteachers[3]. The dictionary contains 727 words. Giving a microblogging message, we first use this dictionary to detect the informal words, then correct them to the formal words. In this way, we are also able to collect more words related to the predefined categories from the labeled messages to tackle the sparseness problem in microblogging messages.

Phrase 3: Construction of SSBN model In order to fully integrate hashtag related resources and unlabeled data to a classifier, we propose a semi-supervised Bayesian network model. The semi-supervised classifier can offer robust solution to microblog topic classification for two reasons. First, it utilizes those labeled microblogging messages with hashtags by training a topic model based classifier, which is then used to find the category (label) distribution of unlabeled messages accurately. Second, it leverages the related external resources to provide a valuable context to microblogging messages. In this way, compared with supervised learning methods, we need only few labeled data for training. The details of SSBN model construction will be introduced in the next subsection.

[2]http://www.twittonary.com/
[3]http://twitterforteachers.wetpaint.com/page/Twitter+Dictionary

θ	The vector indicating category weights for message data collection.
ϕ	The vector indicating category weights for specific message.
θ', ϕ'	The $\|C\| \times \|N\|$ matrix indicating category-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 category vector.
c_j	The jth category.
M	The message collection in the original message data.
m	The message.
N	The word collection in the original message data.
t	The hashtag.
w	The word.
y	The category label of message.

Table 1: Important notations used in this paper and their descriptions.

2.2 Probabilistic Graph Model Construction

The above formulations intuitively reflect that the category prediction task comprises two estimations: coarse-grained category distribution and fined-grained category-word distribution. It is schematically illustrated in Figure 2, in which the corresponding notations are summarized in Table 1.

1. Category distribution: There are two kinds of category distribution in the data. Let θ denotes the category distribution obtained from the original message M, which is a weight vector representing the weight for each category. Similarly, let ϕ denotes the category distribution for external resources obtained from the search results S. The category distribution for the total data D is assumed to be a linear combination of θ and ϕ. Parameter α is employed as the weight to adjust the contributions of different sources. In addition, the original message data also consists of labeled and unlabeled data; and λ is used to denote the contribution of unlabeled data in generating the category distribution for M.

2. Category-word distribution: The category-word distribution also has two parts: θ' denotes the distribution of different words over different categories in the original messages, which is a $|C| \times |N|$ matrix. Here, $|C|$ is the number of categories, and $|N|$ is the number of words in the data. Similarly, ϕ' denotes the category-word distribution in the search results. The category-word distribution for data D is again assumed to be a linear combination of θ' and ϕ', where parameter β is employed as the weight to adjust the contributions of different sources.

Our semi-supervised Bayesian Network (SSBN) belongs to probabilistic graphical model, which formally denotes the probability of a message m falling into a category c as,

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

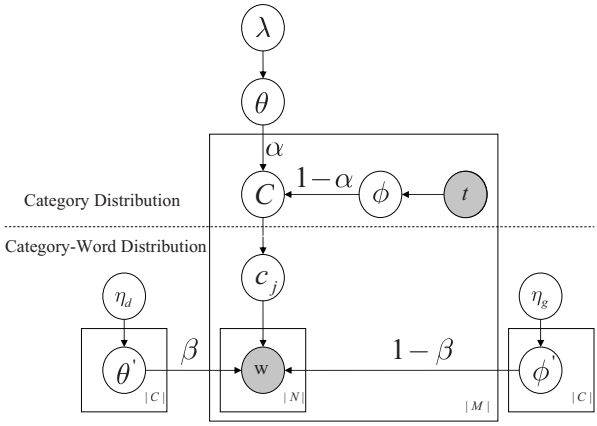


Figure 2: Probabilistic graphical representation of semi-supervised Bayesian network model.

where $P(c)$ is the prior probability of category in the message data collection. By assuming the presence of a word w is independent to the presence of any other word in m, we derive

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

2.3 Parameter Inference

In this section, we turn our attention to procedures for parameter inference with EM approach. In the expectation step, the distributions θ, ϕ, $\hat{\theta'}^{w_k}_{c_j}$ and $\hat{\phi'}^{w_k}_{c_j}$, will be estimated. Besides the labeled data and external resource, the parameter estimations also make use of the unlabeled data. Initially we assign category labels to unlabeled data with an uniform distribution, i.e., the probability is $\frac{1}{|C|}$ for each category. In the following iterations, labels of unlabeled data and SSBN model are alternatively updated and reinforced until convergence.

Estimating θ: θ represents the probability of each category in the original message data collection. It is proportional to the expected number of messages that was assigned to this category.

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

As aforementioned in section 2.2, the message data collection consists of labeled messages M^l and unlabeled messages M^u. They have different contribution to the category probability estimation. The function $\Lambda(i)$, defined as in equation (4), is employed to achieve that goal. The parameter $\lambda \in [0,1]$.

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

Estimating ϕ: ϕ denotes the prior category probability distributes over the Google Search results. In this paper, the prior probability of category c_j for a hashtag t completely depends on the

relationship between the corresponding hashtag t and the predefined category names,

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

where μ is a smoothing factor and $NGD(t, c_j)$ is the Normalized Google Distance[4], which is employed to calculate distance between the tag t and the category c_j. It can be observed that a smaller value of NGD leads to more contribution of c_j for the specific message.

Estimating θ' and ϕ': θ' and ϕ' respectively denote the category-word distributions over original message collection and Google Search results. Both of them are $|C| \times |N|$ matrices. They can be estimated using the following formulas:

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

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

where $n_{d\,c_j}^{w_k}$ and $n_{g\,c_j}^{w_k}$ are respectively the number of times that the word w_k has occurred in the category c_j in message data collection and Google Search results (retrieved by the combination of hashtag t and the name of the j-th category). η_d and η_g are hyperparameters with a small value for smoothing purpose to avoid the zero problem.

The maximum likelihood category label for a given message m_i is,

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

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

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

where the prior probability for category c_j is obtained by linearly fusing two estimations on two resources,

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

where α is a trade-off parameter to balance the contributions between two kinds of category distribution. The maximum likelihood probability for the each message m_i can be derived as:

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

Similar to α, β is tuned to control the contribution between the the category-word distribution over two different resources.

[4] `http://en.wikipedia.org/wiki/Normalized_Google_distance`, here in case of NGD not equal to zero, we add a small constant closing to zero.

3 Experiments

In this section, we first evaluate our proposed model on two real-world datasets, utilizing a range of popular metrics. We then compare our model with the state-of-the-art text classification approaches on microblogs. Also, we study the sensitivity of the training dataset size, convergence analysis followed by the impact analysis on the parameters.

3.1 Experimental Settings

In our experiments, two large-scale real-world datasets were constructed:

- **Twitter**: The Twitter dataset was generated from Trec-Twitter2011[5]. First, we collected 10 hot topics from Google Trends[6], including NBA, Apple, facebook, *etc*. For each topic, we manually selected several low-level sub-topics and combined each of them with the high-level topic. Take the topic "Apple" as an example. We extended it with "Apple stock", "Apple ipad", etc. We manually determine which category the sub-topics belong to. For example, "stock" is classified to Business, while "ipad" is assigned to science. These pairs are naturally viewed as queries. Then the Twitter dataset was constructed by retrieving all the related messages from Trec-Twitter2011 based on these queries. To validate the robustness of our proposed model on partially noisy data, we deliberately did not provide ground truth for this dataset. Instead, the returned messages under a query are directly considered as belongings to the same category as the sub-topic. The Twitter dataset is in this way labeled semi-automatically based on sub-topics. The ground truth is so-called pseudo ground truth. For example, all the messages searched by "Apple stock" are regarded as business category.
- **Sina Weibo**: Based on selected trending topics of Sina Weibo, we crawled a collection of messages. And then manually assigned each messages into one of 7 predefined categories: sports, politics, science&tech, game, movie, music and others. The messages fallen into "others" are removed; and up to $15,811$ unique messages were remained. To build the ground truth, we adopted a manual labeling procedure. We divided 15 people with different background into 3 teams to manually label these messages. Every team labeled the complete dataset. The voting method was employed to combine the label results from different teams. For each message, only one category label with the majority voting was selected as the ground truth label. For the cases that a message received three different categories, a discussion was carried out among the labelers to decide the final ground truths.

The distributions of different categories over two datasets are displayed in Table 2. For each dataset, we devise 4 test configurations with different amount of training data: 5%, 20%, 50% and 90% for training respectively, and use the corresponding reminders for testing. The training data is randomly selected.

In this work, we utilize several widely-used performance metrics to evaluate our classification task: average accuracy, precision, recall, and $F1$ score (Sokolova and Lapalme, 2009) (Rosa et al., 2011). Average accuracy evaluates the average effectiveness for each category of a classifier. Precision is the fraction of retrieved messages that are relevant to the search, while recall is the percentage of the relevant messages that are successfully retrieved, and F1 measure combines both of recall and precision. For some cases, we also provide the $macro-$ and $micro-$ values. The $micro-$ assigns equal weight to each message, while $macro-$ treats each category equally.

[5] http://trec.nist.gov/data/tweets/
[6] http://www.google.com/trends/

Twitter		Sina Weibo	
Total	16935	Total	15811
Sports	2720	Sports	2602
Entertainment	2816	Movies	2694
Business	2912	Games	2605
Science&Tech	2827	Science&Tech	2647
Politics	2937	Politics	2654
Education	2723	Music	2609

Table 2: The distribution of different categories over two datasets.

Twitter				Sina Weibo			
Category	*Precision*	*Recall*	$F1$	*Category*	*Precision*	*Recall*	$F1$
Sports	**0.9322**	0.9483	**0.9402**	Sports	**0.9318**	0.8747	**0.9023**
Entertainment	0.9000	0.5625	0.6923	Movies	0.8848	0.8207	0.8515
Business	0.8043	0.5323	0.6382	Games	0.8090	0.9283	0.8646
Science&Tech	0.6937	**0.9801**	0.8124	Science&Tech	0.8688	0.8323	0.8502
Politics	0.9096	0.9640	0.9360	Politics	0.8661	**0.9324**	0.8980
Education	0.5000	0.5519	0.5165	Music	0.8819	0.8699	0.8759
Micro-average	0.7979	0.7979	0.7979	Micro-average	0.8798	0.8798	0.8798
Macro-average	0.7934	0.6043	0.6128	Macro-average	0.8737	0.8764	0.8738

Table 3: Performance of SSBN model on two datasets with 5% training data and 95% testing data, respectively.

3.2 On Classification Performance Analysis

We first conducted experiment to evaluate the effectiveness of our proposed SSBN model on two datasets. Table 3 displays the average performance in terms of different metrics. Here the parameters are set as $\alpha = 0.5$, $\beta = 0.9$, $\lambda = 0.4$ for Twitter and $\alpha = 0.9$, $\beta = 0.9$, $\lambda = 0.3$ for Sina Weibo, respectively. The parameters selection will be introduced later.

It is observed that our proposed scheme achieves promising precision, recall and $F1$ scores despite of limited availability of labeled data. For twitter dataset, most of the categories achieve precision score higher than 0.85, and the best precision score is up to 0.93 (sports). Half of the categories obtain good results in terms of recall and F1, higher than 0.94 and 0.83, respectively. Our approach yields significant performance over the dataset with pseudo ground truths. This demonstrates the robustness of our method to noisy data. When it comes to Sina Weibo, all the categories achieve remarkable performance of greater than 0.80 across all evaluating metrics. This observation verifies that our method is more stable in less training data. However, our method fails for certain categories such as the Business and Education categories in Twitter dataset. This poor performance mainly comes from the unreliable pseudo ground truths. "Business" and "Education" frequently broaden to various sub-topics. Therefore, the messages retrieved by these types of queries are not internal coherent, at least not as strong as others' categories, even they are assumed to belong to the same category. The unreliable pseudo ground truths bring unpredictable noise to our model.

3.3 On Classification Performance Comparison

To demonstrate the effectiveness of our proposed approach, we compare it against the following the state-of-the-art classifying methods (Phyu, 2009) (Kotsiantis, 2007):

Classifier	*Accuracy*	*MicroP*	*MicroR*	*MicroF*1	*MacroP*	*MacroR*	*MacroF*1
SSBN	**0.8875**	**0.8875**	**0.8875**	**0.8875**	0.8282	0.7627	0.7845
SVM	0.8670	0.8670	0.8670	0.8670	0.8768	0.7611	**0.7860**
NB	0.8722	0.8696	0.8722	0.8722	**0.8879**	0.7329	0.7587
KNN	0.7268	0.7268	0.7268	0.7268	0.6721	0.6471	0.6516
Rocchio	0.8180	0.8204	0.8180	0.8192	0.7361	**0.8384**	0.7605
L-LDA	0.8605	0.8605	0.8605	0.8605	0.8467	0.7223	0.7532

Table 4: Performance comparison among SSBN and other supervised baseline methods on twitter with 90% training data.

Classifier	*Accuracy*	*MicroP*	*MicroR*	*MicroF*1	*MacroP*	*MacroR*	*MacroF*1
SSBN	**0.9020**	**0.9020**	**0.9020**	**0.9020**	0.8976	**0.9045**	**0.9004**
SVM	0.8991	0.8991	0.8991	0.8991	**0.9017**	0.8971	0.8991
NB	0.9015	0.9015	0.9015	0.9015	0.8990	0.9024	0.9003
KNN	0.8565	0.8565	0.8565	0.8565	0.8589	0.8486	0.8526
Rocchio	0.8802	0.8803	0.8802	0.8802	0.8769	0.8832	0.8781
L-LDA	0.8905	0.8905	0.8905	0.8905	0.8876	0.8989	0.8932

Table 5: Performance comparison among SSBN and other supervised baseline methods on Sina Weibo with 90% training data.

- **SVM** (Cortes and Vapnik, 1995) is a supervised learning method. In our experiment, we use an open source package LIBSVM[7] with linear kernel function as baseline.
- **Naive Bayesian** (NB) is a simple probabilistic classifier by applying Bayesian theorem with strong independence assumptions. We use a multi-nomial naive bayesian classifier in our experiment (Yang and Pederson, 1997).
- **K Nearest Neighbors** (KNN) clusters objects based on the closest training examples in the feature space (Creecy et al., 1992). An unlabeled message is assigning the label which is most frequent among the K training samples nearest to the message.
- **Rocchio** (Schapire et al., 1998) is a variant of the Vector Space Model. The average of the relevant documents is viewed as the centroid of the "class".
- **Labeled LDA** (L-LDA) incorporates supervision by constraining LDA model to use only those topics that correspond to an observed label set (Ramage et al., 2009).
- **Transductive SVM** (Trans-SVM) is a semi-supervised SVM method. We extend the binary Transductive SVM in svm-light (Joachims, 1999) to multi-class classifier by incorporating one-against-all strategy.
- **Semi-Naive Bayesian classifiers** (Semi-NB) is a famous semi-supervised text classification method (Nigam et al., 2000). We employ it by using only unlabeled microblogging messages as a prior.

For each aforementioned approaches, the involved parameters are carefully tuned, and the parameters with best performance are used to report the final comparison results. In addition, the same underlying features are utilized for approaches learning. To be fair, our proposed SSBN model was trained with up to 90% data compared with supervised methods, while only 5% training data when compared with semi-supervised approaches. Here, the values of the parameters in SSBN model are set as $\alpha = 0.5, \beta = 0.9$, $\lambda = 0.4$ for Twitter dataset and $\alpha = 0.9$, $\beta = 0.9$, $\lambda = 0.3$ for Sina Weibo dataset.

[7] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Classifier	*Accuracy*	*MicroP*	*MicroR*	*MicroF*1	*MacroP*	*MacroR*	*MacroF*1
SSBN	**0.7979**	**0.7979**	**0.7979**	**0.7979**	**0.7934**	**0.6043**	**0.6128**
Trans-SVM	0.6707	0.6707	0.6707	0.6707	0.6602	0.5108	0.4491
Semi-NB	0.7156	0.7156	0.7156	0.7156	0.7308	0.5653	0.549

Table 6: Performance comparison among SSBN and other semi-supervised baseline methods on Twitter with 5% training data.

Classifier	*Accuracy*	*MicroP*	*MicroR*	*MicroF*1	*MacroP*	*MacroR*	*MacroF*1
SSBN	**0.8798**	**0.8798**	**0.8798**	**0.8798**	**0.8737**	**0.8764**	**0.8738**
Trans-SVM	0.8084	0.8084	0.8084	0.8084	0.8049	0.8085	0.8052
Semi-NB	0.8198	0.8198	0.8198	0.8198	0.8225	0.8217	0.8204

Table 7: Performance comparison among SSBN and other semi-supervised baseline methods on Sina Weibo with 5% training data.

The comparison results with supervised methods on two datasets are illustrated in Table 4 and Table 5, respectively. It is observed from the tables that our proposed model in general performs better than SVM, NB and L-LDA, and remarkably better than KNN and Rocchio. Even the performance of our method for $MacroP$, $MacroR$ and $MacroF1$ on Twitter and $MacroP$ on Sina Weibo does not achieve the best results, they are still comparable and convincing. Table 6 and Table 7 respectively display the comparison results with semi-supervised methods on two datasets, using 5% as training data. It can be observed that our proposed approach are consistently and significantly better than the current publicly disclosed the state-of-the-art semi-supervised algorithms, across various evaluating metrics. This comprehensive improvements are due to the facts that the integrated external knowledge enriches the message representation and the leveraging intrinsic information detected from abundant unlabeled data enhances the prediction accuracy.

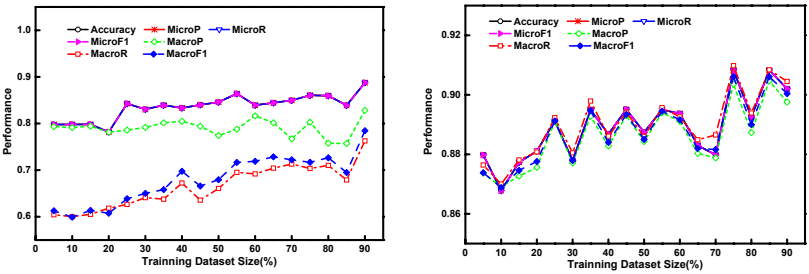
3.4 On the Sensitivity of Training Data Size and Convergence Analysis

In this section, we conduct experiments to investigate the influence of training data size on the overall performance. We progressively increase the size of training corpus at step size of 10%. The experimental results on Twitter and Sina Weibo are respectively illustrated in Figures 3a and 3b. It is observed that the overall trend is upwards along with increasing training set. This is coherent and consistent with our common sense. Also, it is observed that a smaller training set size still produces a robust model on less noisy dataset, with greater than 87% on Sina Weibo.

Perplexity, which is widely used in the topic modeling fields to analyze the convergence of a model (Blei et al., 2003) (Zhao et al., 2010). We do perplexity comparison of SSBN and L-LDA on the testing data when parameters in SSBN model are set as $\alpha = 0.5, \beta = 0.9$, $\lambda = 0.4$ for Twitter and $\alpha = 0.9$, $\beta = 0.9$, $\lambda = 0.3$ for Sina Weibo dataset. Compared with L-LDA model, SSBN model has a lower perplexity value, which means that the words are less surprising to SSBN model, and SSBN model has a powerful predication than L-LDA model.

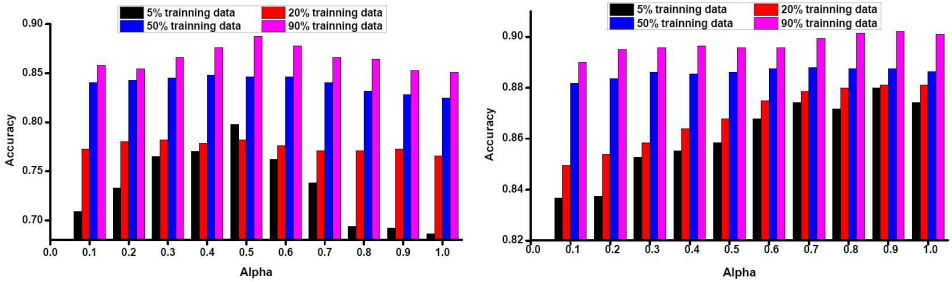
3.5 On the Sensitivity of Parameters

Parameters of α, β and λ are important in our method. In this subsection, we further conduct experiments to study the effect of these parameters. A grid search is performed to select the optimal parameter values.



(a) Twitter (b) Sina Weibo

Figure 3: Performance sensitivity of training set size on Twitter and Sina Weibo.



(a) Twitter (b) Sina Weibo

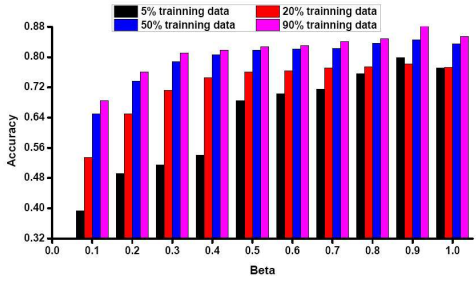
Figure 4: The Performance with varying α and training data size when other parameters are fixed.

3.5.1 Effect of Parameter α

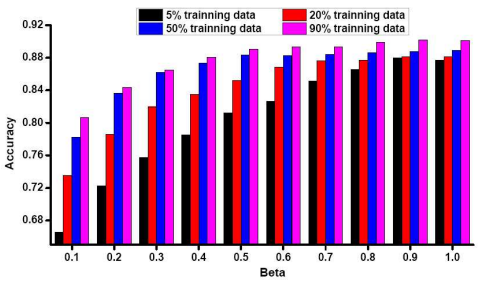
The trade-off parameter α is used to balance the effects of two kinds of prior knowledge at category level: microblogging data collection and external resources. A larger α indicates that more information is preserved from our data collection into the category distribution. A smaller α means that the cues mined from external resources play a dominant role in our model. Figure 4 illustrates the average performance with various α and training collection size on two different datasets. It is observed that the performance increases with the gradual increase of α, and arrives at a peak at certain α, then the performance decreases. This result reflects that an optimal performance comes from an appropriate combination of external and internal resources, rather than pure individual knowledge. Also it verifies that the incorporation of Google resources has been proven useful. Empirical optimal value of α is within $[0.5, 1]$.

3.5.2 Effect of Parameter β

There are two category-word distributions, θ' and ϕ', which are respectively generated from our data collection and google search results; and parameter β is utilized to adjust the contribution between these two different resources in category-word level. Larger β implies larger likelihood a word is generated from θ'. The effects of parameter β on Twitter and Sina Weibo are shown in Figure 5. It is clearly observed that larger values of β frequently lead to higher accuracies with different training set sizes, and the accuracy reaches peak value when β locates at 0.9. However, when β trends to 1, the performance slightly decreases. Empirical optimal value of β is within $[0.5, 1]$.

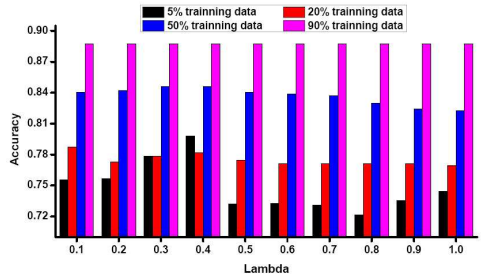


(a) Twitter

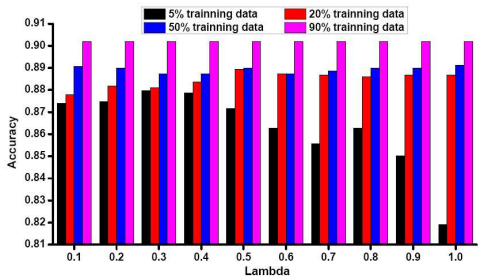


(b) Sina Weibo

Figure 5: The Performance with varying β and training data size when other parameters are fixed.



(a) Twitter



(b) Sina Weibo

Figure 6: The Performance with varying λ and training data size when other parameters are fixed.

3.5.3 Effect of Parameter λ

λ indicates the contribution from unlabeled data points, between 0 and 1. When λ is close to 1, knowledge from unlabeled data is considered as important as labeled data. On the other hand, when λ at near-zero value, our model approaches a supervised learning algorithm. The results are illustrated in Figure 6, from which we observe some insights: (1) varying λ has little impact on average accuracy for a large training set, such as 50 percent as training set, especially for 90 percent as training set; (2) the best accuracy occurs at $\lambda = 0.4$ and $\lambda = 0.3$ respectively for Twitter and Sina Weibo, and then drops down quickly, which illustrates unlabeled data could give some feedback to improve classification performance. Empirical optimal value of λ is within $[0.3, 0.5]$.

4 Related Work

The task of topic classification of microblogging messages is to assign the pre-defined class labels to unlabeled messages given a collection of messages. It has been demonstrated to be a fundamental task for many applications, such as query disambiguation (Teevan et al., 2011), location prediction (Gao et al., 2012) and hot topic tracking (Weng and Lee, 2011), *etc*. To the best of our knowledge, our work is the first attempt to utilize semi-supervised learning methods to classify microblogging messages. There are, however, several lines of related work.

The significance of topic models has been exploited in microblog clustering and classification. A representative work was proposed in 2010 (Hong and Davison, 2010), where latent dirichlet allocation (LDA) (Blei et al., 2003) and author-topic model (Rosen-Zvi et al., 2010) were deeply investigated to automatically find hidden topic structures on Twitter. Following that, Zhao et al. (2011) performed content analysis through Twitter-LDA modeling on a Twitter corpus collected within a three month span. Several variants of LDA to incorporate supervision have been proposed

by Ramage et al. (2009, 2010), and have been shown to be competitive with strong baselines in the microblogging environment. Although these LDA-based topic model greatly save cognitive and physical effort required from user interaction, their performances are usually not very satisfactory. The main reason is due to the sparsity of short informal messages that makes similarity comparison difficult. Different from previous models, we employed a two-step pre-processing: detecting informal words using dictionary and correcting the words into formal ones. This helps to alleviate the negative effects brought by short message sparsity to some extent.

Lee et al. (2011) classified tweets into pre-defined categories such as sports, technology, politics, *etc*. Instead of topic models, they constructed word vectors with tf-idf weights and utilized a Naive Bayesian Multinomial classifier to classify tweets. Further, Support Vector Machines achieved good performance to classify Twitter messages, as reported by Zubiaga et al. (2011). Sriram et al. (2010) proposed to use a small set of domain-specific features extracted from the author's profile and text to represent short messages. Their method, however, requires extensive pre-processing to conduct effectively feature analysis, which was impractical to as a general solution for classification of microblogging messages. The performance improvement of the supervised methods mainly depend on a large scale of labeled training data, which is laborious and time consuming. Further, the sparsity problem hinders significant performance improvement. To break the current impasse between annotation cost and effectiveness, we proposed to utilize semi-supervised learning methods. We trained a semi-supervised classifier by using the large amount of unlabeled data, together with labeled data. In addition, our work is novel in that we mined the information cues from Google Search Engine and seamlessly fused them with informal microblogging messages.

5 Conclusion and Future Work

In this paper, we proposed a novel scheme to classify microblogging messages, which addresses three concerns in microblog classifications. First, the incorporation of external resources to supplement the short microblogs well compensates the data sparseness issue. Second, the semi-supervised classifier seamlessly fuse labeled data structure and external resources into the training process, which reduced the requirement for manually labeling to a certain degree. Third, we model the category probability of a given message based on the category-word distribution, and this successfully avoided the difficulty brought about by the spelling errors that are common in microblogging messages. We proposed a semi-supervised learning approach to classify microblogging messages, and the experimental results demonstrated its effectiveness as compared to existing the state-of-the-art methods, as well as practically extension to large-scale dataset.

This work suggests some interesting directions for further exploration. It is interesting to explore whether: (1) the incorporation of social network structure can improve the performance of microblogging classification (Hu and Liu, 2012a); (2) the use of external resources such as Wikipedia and WordNet might be valuable for understanding microblogging messages; and (3) the provision of category summarization can help to organize microblogging messages.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (grant number 61170189, 60973105), the National Natural Science Fund for Young Scholar (grant number 61202239), the Research Fund for the Doctoral Program of Higher Education (grant number 20111102130003), and the Fund of the State Key Laboratory of Software Development Environment (grant number SKLSDE-2011ZX-03),

References

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

Creecy, R. H., Masand, B. M., Smith, S. J., and Waltz, D. L. (1992). Trading mips and memory for knowledge engineering. In *Communication of the ACM*, volume 35, pages 48–64.

Diao, Q., Jiang, J., Zhu, F., and Lim, E.-P. (2012). Finding bursty topics from microblogs. In *Proceedings of Association for Computational Linguistics*.

Gao, H., Tang, J., and Liu, H. (2012). Exploring social-historical ties on location-based social networks. In *Proceedings of International AAAI Conference on Weblogs and Social Media*.

Gruber, A., Rosen-Zvi, M., and Weiss, Y. (2007). Hidden topic markov model. In *Proceedings of International Conference on Artificial Intelligence and Statistics*.

Hong, L. and Davison, B. D. (2010). Empirical study of topic modeling in twitter. In *Proceedings of KDD Workshop on Social Media Analytics*.

Hu, X. and Liu, H. (2012a). Social status and role analysis of palin's email network. In *Proceedings of the international conference companion on World Wide Web*.

Hu, X. and Liu, H. (2012b). Text analytics in social media. *Mining Text Data*, pages 385–414.

Hu, X., Sun, N., Zhang, C., and Chua, T.-S. (2009). 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*.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of International Conference on Machine Learning*.

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, 31:249–268.

Lee, K., Palsetia, D., Narayanan, R., Patwary, M. M. A., Agrawal, A., and Choudhary, A. (2011). Twitter trending topic classification. In *Proceedings of ICDM Workshop on Optimization Based Methods for Emerging Data Mining Problems*.

Nie, L., Wang, M., Zha, Z.-j., Li, G., and Chua, T.-S. (2011). Multimedia answering: enriching text qa with media information. In *Proceedings of Annual ACM Conference on Special Interest Group on Information Retrieval*.

Nigam, K., Mccallum, A. K., Thrun, S., and Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. In *Machine Learning - Special issue on information retrieval*, volume 39, pages 103–134.

Phyu, N. P. (2009). Survey of classification techniques in data mining. In *Proceedings of International MultiConference of Engineers and Computer Scientists*.

Ramage, D., Dumais, S., and Liebling, D. (2010). Charaterizing microblog with topic models. In *Proceedings of International AAAI Conference on Weblogs and Social Media*.

Ramage, D., Hall, D., Nallapati, R., and Manning, C. D. (2009). 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*.

Rosa, K. D., Shah, R., Lin, B., Gershman, A., and Frederking, R. (2011). Topical clustering of tweets. In *Proceedings of SIGIR Workshop on Social Web Search and Mining*.

Rosen-Zvi, M., Chemudugunta, C., Griffiths, T., Smyth, P., and Steyvers, M. (2010). Learning author-topic models from text corpora. *ACM Transactions on Information Systems*, 28:1–38.

Schapire, R. E., Singer, Y., and Singhal, A. (1998). Boosting and Rocchio applied to text filtering. In *Proceedings of Annual ACM Conference on Research and Development in Information Retrieval*, pages 215–223.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45:427–437.

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., and Demirbas, M. (2010). Short text classification in twitter to improve information filtering. In *Proceedings of Annual ACM Conference on Research and Development in Information Retrieval*.

Tang, J., Wang, X., Gao, H., Hu, X., and Liu, H. (2012). Enriching short text representation in microblog for clustering. *Frontiers of Computer Science in China*, 6(1):88–101.

Teevan, J., Ramage, D., and Morris, M. R. (2011). #twittersearch: a comparison of microblog search and web search. In *Proceedings of ACM Conference on Web Search and Data Mining*.

Weng, J. and Lee, B.-S. (2011). Event detection in twitter. In *Proceedings of Association for the Advancement of Artificial Intelligence*.

Yang, Y. and Pederson, J. (1997). Feature selection in statistical learning of text categorization. In *Proceedings of International Conference on Machine Learning*.

Zhao, T., Li, C., Ding, Q., and Li, L. (2010). User-sentiment topic model: refining user's topics with sentiment information. In *Proceedings of ACM SIGKDD Workshop on Mining Data Semantics*.

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., and Li, X. (2011). Comparing twitter and traditional media using topic models. In *Proceedings of European Conference on IR Research*.

Zubiaga, A., Spina, D., Fresno, V., and Martinez, R. (2011). Classfying trending topics: A typology of conversation triggers on twitter. In *Proceedings of ACM Conference on Information and Knowledge Management*.