Chinese Short Text Classification Based on Domain Knowledge

Xiao Feng
Institute of Automation
Chinese Academy of Science
xiao.feng@ia.ac.cn

Yang Shen
State Administration for Industry & Commerce of the People's Republic of China
shenyang@saic.gov.cn

Chengyong Liu
Information Center of General Administration of Press and Publication of PR China
liucy_gapp@sina.com

Wei Liang
Institute of Automation
Chinese Academy of Science
wei.liang@ia.ac.cn

Shuwu Zhang
Institute of Automation
Chinese Academy of Science
shuwu.zhang@ia.ac.cn

Abstract

People are generating more and more short texts. There is an urgent demand to classify short texts into different domains. Due to the shortness and sparseness of short texts, conventional methods based on Vector Space Model (VSM) have limitations. To tackle the data scarcity problem, we propose a new model to directly measure the correlation between a short text instance and a domain instead of representing short texts as vectors of weights. We firstly draw domain knowledge for each user-defined domain using an external corpus of longer documents. Secondly, the correlation is calculated by measuring the proportion of the overlapping part of the instance and the domain knowledge. Finally, if the correlation is greater than a threshold, the instance will be classified into the domain. Experimental results show that the classifier based on the proposed model outperforms the state-of-the-art baselines based on VSM.

1 Introduction

In recent years, web services are generating more and more short texts including micro-blogs, customer reviews, chat messages and so on. However, a user is often only interested in very small part of these data. There is an urgent demand to classify incoming short texts into different domains, so that users are not overwhelmed by the raw data. As short texts do not provide sufficient word occurrences (i.e., the length of a micro-blog is limited to 140 characters), conventional text classifiers often cannot achieve high accuracy, especially when the number of training examples is small.

Vector Space Model (VSM) is a very popular document representation model, where each document is represented as a vector of weights. Text classification methods based on VSM perform well when processing documents in regular length (Berry and Michael, 2004). But, the sparsity of VSM will reduce the classification accuracy when processing short texts.

There have been several studies that attempted to solve the problem of data sparseness in VSM. One way is to select more useful features using additional semantics from Wikipedia (Banerjee *et al.* ,2007), WordNet (Hu *et al* , 2009) or HowNet (Liu *et al.* , 2010). Another way is to expand the coverage of classifier by using background knowledge drawn from much longer external data sources. Zelikovitz and Hirsh (2000) utilized a corpus of unlabeled longer documents as a “bridge”, to connect the test example with training examples. Phan *et al.* (2008) and Chen *et al.* (2011) integrated the original short text with hidden topics discovered from external large-scale data collections to add more meta-information. These researches have shown positive improvement by enriching the representation of feature vectors, but a disadvantage is the high computational complexity.

In this paper, we try to solve the sparse problem from another direction with lower computational complexity. We propose a new model to directly measure the correlation between a short text instance and a domain, using domain knowledge drawn from a labeled external corpus of related longer documents. We performed a careful evaluation for our model on micro-blog

International Joint Conference on Natural Language Processing, pages 859–863,
Nagoya, Japan, 14-18 October 2013.

classification task, and achieved consistent improvements over two baselines.

The overall framework of our approach is shown in Figure 1. We firstly draw domain knowledge for each user-defined domain using the external corpus. Secondly, the correlation between a short text instance and a domain is calculated by measuring the proportion of the overlapping part of this instance and the domain knowledge of this domain. Finally, if the correlation is greater than a threshold, the instance will be classified into the domain. The main advantages of our approach include the following points:

- Good generalization performance: domain knowledge learned from longer documents can cover lots of terms that do not exist in a small labeled training set.
- Easy to implement: No need to construct VSM to train classifiers. All we need to prepare is the domain knowledge.

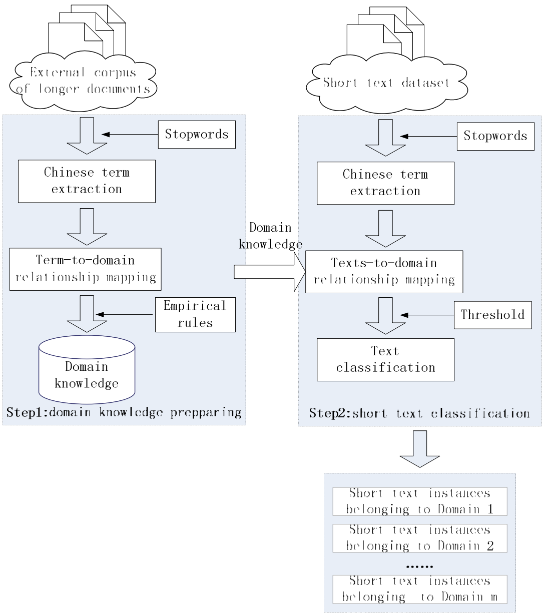
External corpus of longer documents
Short text dataset
Stopwords
Chinese term extraction
Term-to-domain relationship mapping
Empirical rules
Domain knowledge
Step1:domain knowledge prepparing
Stopwords
Chinese term extraction
Domain knowledge
Texts-to-domain relationship mapping
Threshold
Text classification
Step2:short text classification
Short text instances belonging to Domain 1
Short text instances belonging to Domain 2
......
Short text instances belonging to Domain m

Figure 1. The framework of our approach

2 Our Approach

2.1 Domain knowledge preparing

To expand the coverage of our model, we utilize a labeled external corpus of longer documents to draw domain knowledge. Text documents in this corpus have been classified into user-defined domains. There are two main conditions that should be followed to choose an appropriate external corpus. First, the coverage of vocabulary should be sufficient. Second, the user-defined domains should be consistent with the classification problem. In fact, the second condition will not be a problem, because a large number of web documents belonging to different domains can be crawled from portal sites such as Sina[1] and Sohu[2].

After Chinese term extraction and removing stop words, we obtain an initial term list appearing in the corpus, denoted by $T = \{t_1, t_2, \cdots, t_n\}$.

The aim of term-to-domain relationship mapping is to select K -number of most related terms for each domain from T . The set of selected terms is regarded as the domain knowledge of a domain. How terms are related to each domain is measured by applying Chi-square statistical term-to-domain independency measurement. The measurement is based on the co-occurrence frequencies of a term and a domain. We firstly assume that the term and the domain are statistically independent, and then compare the observed frequency and the expected frequency.

Let $t_i (i = 1, 2, \cdots, n)$ be a term in the initial term list T , and $d_j (j = 1, 2, \cdots, m)$ be a domain in the user-defined domain list $D = \{d_1, d_2, \cdots, d_m\}$. The expected frequency is defined as:

$$E_{e_t e_c} = \frac{\sum_{p \in \{0,1\}} O_{p e_c} \sum_{q \in \{0,1\}} O_{e_t q}}{N}, e_t \in \{0,1\}, e_c \in \{0,1\} \quad (1)$$

where $N = O_{11} + O_{01} + O_{10} + O_{00}$, O_{11} denotes the observed frequency of documents which contain t_i and belong to d_j , O_{01} denotes the observed frequency of documents which do not contain t_i but belong to d_j , O_{10} denotes the observed frequency of documents which contain t_i but not belong to d_j , and O_{00} denotes the observed frequency of documents that neither contain t_i nor belong to d_j .

The Chi value for t_i and d_j is defined as:

$$\chi^2(t_i, d_j) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(O_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \quad (2)$$

Note that the greater the Chi value is, the closer the relationship between t_i and d_j is.

Let $\overrightarrow{DK} = \{dk_1, dk_2, \cdots, dk_m\}$ be the domain knowledge vector, dk_j be the domain knowledge of domain d_j , $\overrightarrow{tr_i}$ be the Chi value vector of term t_i , and $\overrightarrow{dr_j}$ be the Chi value vector of do-

[1] http://www.sina.com.cn/
[2] http://www.sohu.com/

main d_j. The algorithm of term-to-domain relationship mapping includes three main steps:

Step1: For each term t_i, construct its Chi value vector $\overrightarrow{tr_i} = \{\chi^2(t_i,d_1), \chi^2(t_i,d_2), \cdots, \chi^2(t_i,d_m)\}$.

Step2: For each $\overrightarrow{tr_i}$, find its largest item $\chi^2(t_i,d_j)$ and put it into $\overrightarrow{dr_j}$.

Step3: Sort items in $\overrightarrow{dr_j}$ in descending order. Select the corresponding terms of the first K-number of items, and put them in dk_j. All terms in dk_j are arranged in descending order under its Chi value.

2.2 Short text classification

In this section, we introduce an intuitive model to directly relate each short text instance to one or more specific domains. How a short text instance, denoted by g, is related to a domain d_j can be measured based on the correlation between them. The correlation is calculated by measuring the proportion of the overlapping part of g and the domain knowledge dk_j (Liu *et al*, 2012), see Figure 2.

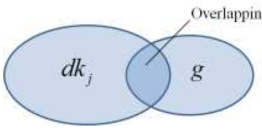

Figure 2. The overlapping part of a short text instance and the domain knowledge

To measure the proportion of the overlapping part, we need to compute the score of domain knowledge dk_j and the score of the overlapping part of g and dk_j, and then normalize the latter by the former. Moreover, we introduce a weight, denoted by w_{jk}, to indicate the importance of a term in dk_j, when calculating the scores. As the Chi values of terms in different domain knowledge vary greatly, we define the weight of a term based on its ordering position in the domain knowledge. The weight is defined as:

$$w_{jk} = \frac{K+1-k}{K}, k = 1,2,\cdots,K \quad (3)$$

where k is the order of the term in dk_j, and K is the number of terms in dk_j.

Finally, the correlation between g and d_j is calculated based on scores as:

$$score_{dk_j} = \frac{1}{L}\sum_{k=1}^{K} w_{jk} \quad (4)$$

$$score_{overlapping} = \sum_{k=1}^{K} w_{jk} \times \frac{tf(t_{jk})}{len(g)} \quad (5)$$

$$correl(g,d_j) = \frac{score_{overlapping}}{score_{dk_j}} \quad (6)$$

where L is the average length of documents in the external corpus (i.e. the average size of the term lists of documents), $tf(t_{jk})$ is the frequency of term t_{jk} appearing in g, and $len(g)$ is the length of g.

If $correl(g,d_j)$ is greater than a threshold δ, g will be classified into d_j. The optimized value of δ can be obtained by cross-validation.

3 Experiments and results

3.1 Data Sets

We collect short texts from Sina micro-blog[3], and use an open corpus[4] collected by Sogou Lab from the Internet as the external corpus.

External Corpus Documents belong to 8 domains: Finance, IT, Health, Sports, Tour, Education, Film&TV, and Military. Each domain contains 600 documents. The vocabulary is 69909 terms. The average length of documents is 403 terms.

Micro-blog Dataset We manually choose training samples and test samples for each user-defined domain. Samples in the training set and the test set are totally exclusive. The average length of micro-blogs is 31 terms. There is a noise set in the test set containing micro-blogs which do not belong to any user-defined domain, see Table 1.

Domain	**#Train data**	**#Test data**
Finance	600	300
IT	600	300
Health	600	300
Sports	600	300
Tour	600	300
Education	300	150
Film&TV	600	300
Military	300	150
Noise set	0	10,000
Total	4200	12100

Table 1. Description of the micro-blog dataset

[3] http://t.sina.com.cn

[4] http://www.sogou.com/labs/dl/c.html

3.2 Measurement

We adopt the F1-measure as our performance criterion to balance the influence between precision and recall.

$$precision = \frac{TP}{TP + FP} \tag{7}$$

$$recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{9}$$

where *TP* denotes the numbers of relevant samples classified as relevant, *FN* denotes the numbers of relevant samples classified as irrelevant, and *FP* denotes the number of irrelevant samples classified as relevant.

3.3 Experiment results and analysis

In order to obtain the optimized value of δ, we randomly divided the training set into five equal partitions and performed 5-fold cross-validation. In Table 2, we can find that our classifier achieves the highest F1-measure when $\delta = 0.2$ and $K = 500$. Thus in all following experiments, we employ $\delta = 0.2$.

δ K	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
100	0.8201	0.8847	0.9110	0.9189	0.9125
200	0.8635	0.9274	0.9476	0.9421	0.8821
300	0.8934	0.9506	0.9447	0.9235	0.7203
400	0.9187	0.9519	0.9417	0.7286	0.4537
500	0.9388	**0.9554**	0.8494	0.5463	0.2293
600	0.9453	0.9523	0.7545	0.4372	0.1120
700	0.9482	0.8938	0.6842	0.3213	0.0503

Table 2. 5-fold cross-validation on training set

The next experiment is to compare our method with two baselines based on VSM of TFIDF weights on the test set. Both the baselines are composed of 8 SVM (Support Vector Machine) classifiers (one for each domain to decide whether a test sample belongs to this domain). One of them uses terms in domain knowledge drawn from the external corpus as features to enrich the representation of VSM ("VSM with E" for short). The other one only uses terms in domain knowledge drawn from the training set as features to construct VSM ("VSM without E" for short). We use RBF kernel and optimized parameters which are chosen by grid-search in LIBSVM[5] to train SVM classifiers.

	VSM without E	**VSM with E**	**Our Method**
Optimized K	700	500	500
Optimized F1-measure	0.8965	0.9285	**0.9520**

Table 3. The overall optimized results of our method and VSM-based methods

Table 3 shows the overall optimized result of each method with its optimized K on the test set, and Figure 3 shows the optimized result of each domain in more details. We can find that our method achieves the highest overall F1-measure, and achieves 5.7%, 1.7%, 3.0%, 1.9%, 3.5%, 0.06%, 2.9% and 1.4% improvements over VSM with E for Finance, IT, Health, Sports, Tour, Education, Film&TV, and Military respectively. This means our method could improve the performance of short text classification by solving data sparsity problem effectively.

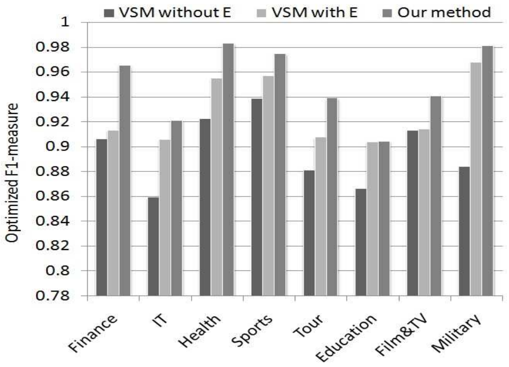

Figure 3. Optimized results of each domain in more details

4 Conclusion

In this paper, we propose a new model based on domain knowledge to solve the data sparsity problem in short text classification. We validate through experiments that classifier based on our model outperforms classifiers based on VSM. In the future work, we will try to combine the external knowledge and the training set to further improve the performance of short text classification.

Acknowledgments

The wok has been supported by the National Key Technology R&D Program of China under Grant No. 2011BAH16B02 ,2013BAH61F01 and 2012BAH88F03.

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

References

Banerjee S, Ramanathan K, and Gupta A. 2007. *Clustering Short Texts Using Wikipedia*. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007: 787-788.

Hu X, Sun N, Zhang C, and Chua T. 2009. *Exploiting internal and external semantics for the clustering of short texts using world knowledge*. Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009: 919-928.

Zelikovitz S and Hirsh H. 2000. *Improving short text classification using unlabeled background knowledge to assess document similarity*. Proceedings of the Seventeenth International Conference on Machine Learning. 2000: 1183-1190.

Phan X, Nguyen L, and Horiguchi S. 2008. *Learning to classify short and sparse text & web with hidden topics from large-scale data collections*. Proceedings of the 17th international conference on World Wide Web. ACM, 2008: 91-100.

Chen M, Jin X, and Shen D. 2011. *Short text classification improved by learning multi-granularity topics*. Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Three. AAAI Press, 2011: 1776-1781.

Liu Z, Yu W, Chen W, et al. 2010. *Short text feature selection for micro-blog mining*. Computational Intelligence and Software Engineering (CiSE), 2010 International Conference. IEEE, 2010: 1-4.

Berry, and Michael. 2004. *Survey of Text Mining I: Clustering, Classification, and Retrieval*. volume 1. Springer-Verlag New York Incorporated, 2004.

Liu J N K, He Y, Lim E H Y, et al. 2012. *Domain ontology graph model and its application in Chinese text classification*. Neural Computing and Applications, 2012: 1-20.