Short Text Classification by Detecting Information Path

Shitao Zhang[1] Xiaoming Jin[1] Dou Shen[2] Bin Cao[3] Xuetao Ding[1] Xiaochen Zhang[1]
[1]Key Laboratory for Information System Security, Ministry of Education
Tsinghua National Laboratory for Information Science and Technology
School of Software, Tsinghua University, Beijing, China
[2]Baidu Corporation, Beijing, China [3]Microsoft Research Asia, Beijing, China
shitaozhang@gmail.com xmjin@tsinghua.edu.cn doushen@gmail.com
bincao@microsoft.com helius.ylabs@yahoo.com jasiazhang@gmail.com

ABSTRACT

Short text is becoming ubiquitous in many modern information systems. Due to the shortness and sparseness of short texts, there are less informative word co-occurrences among them, which naturally pose great difficulty for classification tasks on such data. To overcome this difficulty, this paper proposes a new way for effectively classifying the short texts. Our method is based on a key observation that there usually exists ordered subsets in short texts, which is termed "information path" in this work, and classification on each subset based on the classification results of some pervious subsets can yield higher overall accuracy than classifying the entire data set directly. We propose a method to detect the *information path* and employ it in short text classification. Different from the state-of-art methods, our method does not require any external knowledge or corpus that usually need careful fine-tuning, which makes our method easier and more robust on different data sets. Experiments on two real world data sets show the effectiveness of the proposed method and its superiority over the existing methods.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology—*Feature evaluation and selection*

Keywords

Short text classification; Information path; Anchor shackle term

1. INTRODUCTION

With the explosion of online social network applications and e-commerce, short texts such as microblogging, product reviews and search snippets are becoming more popular on the Internet. Automatic classification serves as a useful way to explore short texts. Traditional text classification techniques are mainly based on common words among both labeled and unlabeled data that belong to the same category to measure text similarities. However, due to the shortness and sparseness of short text, such common words are insufficient. This gap makes short text classification a challenging problem [10]. To alleviate the sparseness of short text, state-of-art works mainly focus on expanding short texts with knowledge extracted from auxiliary long text corpus [5]. Nevertheless, this process is usually domain dependent and thus requires remarkable human efforts in collecting and tuning the data. On the other hand, utilizing inherent characteristics of short text other than the common consensus of shortness and sparseness, which might benefit the construction of more accurate classifier for short text data, however remains unexplored in the literature.

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505638 .

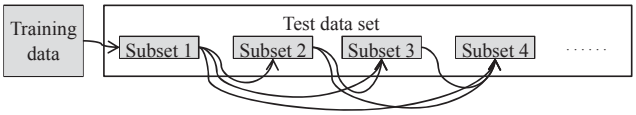


Figure 1: Illustration of information path

Based on the observation of several inherent characteristics of short text data sets, this paper proposes an effective and auxiliary-resource-free method for short text classification, which saves human efforts for collecting the auxiliary corpus remarkably. Specifically, an inherent characteristic of real world short text data is observed: usually, ordered subsets of short text (termed "*information path*") can be found in test dataset and if we classify each subset using classification results of previous subsets, it would achieve better classification results than classifying the entire dataset directly. Figure 1 illustrates the main concept of information path. Essentially, it is a path that consists of sequential subsets in the test dataset. And according to this path, instances of former classified subsets can assist classification of later subsets.

The intuition behind *information path* can be explained as follows: though data shortness may yield less common words between training and test data that belong to the same category, *some* instances of test data are likely to share common discriminative terms with training data, and thus these instances can be labeled more accurately. In addition, these newly labeled texts bring in new common words. Thus a natural idea is to reuse these correctly labeled texts to fill the gap between training data and the other test data.

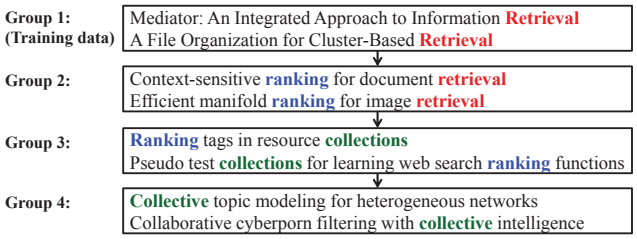


Figure 2: Example of information path on paper titles

Figure 2 shows the insight intuition of information path by a more concrete example of short text classification on paper titles. Suppose the task is to predict the conference in which a paper is published given the title of the paper. Paper titles shown in this figure are all from one conference (SIGIR) and thus they should share the same label. Instances of Group 1 are training data. If we train a classifier using instances in Group 1 and then classify the other three groups directly, the accuracy might be low. The reason is that the data in Group 1 has few if any common words with instances in Group 3 and 4. However, data in Group 2 can serve as a bridge to connect instances of training data and other test data in Group 3. Specifically, Group 2 shares a common word "retrieval" with data in Group 1, and "rank" with Group 3. Thus we can build a classifier on Group 1 and classify Group 2 firstly, and then utilize both Group 1 and 2 to train a new classifier to label instances in Group 3. This method can achieve better classification result as Group 2 brings in common word with Group 3. In a similar way, data in Group 4 can be labeled using a classifier built on previous labeled instances. In this way, Group 2, 3 and 4, which are all subsets of test data, are connected together to form the information path. And according to this information path, classification result of former subsets can help the classification of later subsets.

In this paper, we incorporate inherent characteristics of short text in semi-supervised learning framework. In general semi-supervised learning frameworks, instances with high classification confidence are moved from test data to training data, and then build a new classifier on the whole training data to classify other test data in each iteration. Based on this framework, our method considers the inherent characteristics of short text, which are as follows.

Firstly, common word that has good categorical discriminability can be found in selected groups of short texts, while that can be hardly found for the entire dataset. Such word is termed *anchor shackle term* in our paper (e.g., retrieval, rank, collect, in Figure 2). This is reasonable because when people talk about a specific topic, they tend to adopt the commonly used words in short texts for being understood easily. Paper titles serve as a good example to illustrate this observation.

Secondly, for two groups of short texts A and B that are from the same category but with seldom common discriminative terms, it is possible to find another group of short texts C from the same category that can bridge the gap between A and B in the information path. That is, A can be used to classify C, and then the newly labeled C together with A in turn can be used to classify B.

Based on information path detection, the short texts can be classified sequentially where classification of each subset can get prearranged help from some previously labeled texts. This fills the gap of short texts without demanding auxiliary resources and human efforts. The main contributions of this paper are summarized below:

- We explore some inherent characteristics that are very common in real-world short texts and propose the concept of *information path* in this paper. As far as we know, this is the first work to exploit these inherent characteristics for short text classification.
- We propose a method to extract the information path in short text dataset and utilize it to improve the accuracy of short text classification under the framework of semi-supervised learning.
- Experimental results on two real short text dataset indicate that the proposed schema remarkably outperforms state-of-art methods for short text classification and common semi-supervised learning techniques.

2. RELATED WORK

Existing techniques for short text classification are mainly based on feature expanding. The major difference between these techniques is how to obtain the extra features. One is based on search engine that integrates the search results into the short text [12, 13, 18, 2]. The other is to exploit an external corpus and utilize the topics as new features [10, 4, 11, 3, 16]. There is also a work[7] that uses the transfer learning methods to exploit the external data.

Although these methods can improve the accuracy of short text classification to some extent, they all need some external auxiliary data obtained through a time-consuming collection. These methods do not pay enough attention to the inherent characteristics of the short text itself. The classification accuracy is improved by doing the topic correlation analysis of training and test data in [6]. There is also another work[14] improving the accuracy significantly by selecting some useful features. In [15], the proposed method yields similar accuracy compared to the baseline by using the representative query words of the short text to search the categories from the labeled short texts. These two works indicate that some inherent features of short text can play a more important role for classification.

In addition to the techniques for short text classification, our method also relates to some other existing machine learning techniques including: (1) semi-supervised learning [8, 19] that improves the learning process by exploiting the unlabeled data in addition to the labeled training data, (2) transfer learning [9] that applies the knowledge learned from one domain to another domain, (3) curriculum learning [1] that learns the knowledge from easy to difficult gradually. Our method is different from these techniques. It is under the framework of self-training, but we reuse the data for tracking the inherent anchor shackle terms changes within the short texts and detecting the information path according to the characteristics of short texts. However, other techniques of semi-supervised learning aim to get more correctly labeled data for a better classifier in each iteration. Our method can be seen as a knowledge transfer process

through the information path, but there is no definite domain information. The curriculum learning is a strategy to determine the sequence of instances to be learned when the classifier is trained (when the knowledge is learned), but our method can be seen as a strategy to determine the sequence of instances when they are classified (when the problem is solved). In our method, the sequence of instance is not determined by the difficulty of the classification, but by the changing of the detected anchor shackle terms. And there is not a definite concept of difficult knowledge or easy knowledge used in our work.

3. INFORMATION PATH

In this section, we formally define the problem of short text classification and denote the concepts of information path and anchor shackle term.

The problem of classifying short text is: given a training dataset $\mathcal{D}_{train} = \{< x_n, y_n >\}$ where x_n and y_n denotes the feature vector and label of the n-th training sample in $\mathcal{D}_{train}$ respectively and a testing dataset $\mathcal{D}_{test} = \{x'_n\}$ where x'_n is the n-th feature vector in $\mathcal{D}_{test}$, the short text classification problem is to predict a class label y'_n for each $x'_n \in \mathcal{D}_{test}$. Due to the sparseness of short text, specifically most dimensions of vectors x_n and x'_n are zero values.

Information path is constituted by a sequence of mutually inclusive data subset $\mathcal{D}_n \in \mathcal{D}_{test}$ of the test data $\mathcal{D}_{test}$, where classification on each subset $\mathcal{D}_n$ based on previous subsets $\mathcal{D}_1, \ldots, \mathcal{D}_{n-1}$ can yield an higher overall accuracy than classification on the entire test data directly. The concept of information path is illustrated in Figure 1 intuitively. The key point to detect the information path is to construct all subsets $\mathcal{D}_n$ of test data and determine the classified order of them.

An Anchor shackle term is the word that is commonly used in short text when talking about a specific topic. Generally, these anchor shackle terms are presented as common words that have good categorical discrimination in the context of short texts.

Information path describes the subset that each instance belongs to and the order in which the subsets are classified. Ideally, each subset on information path should have small gap with the subsets that it is directly connected to. Specifically, in short texts, if two texts have common anchor shackle term, they have large probability of talking about the same topic, and thus the gap between them is small. So the anchor shackle terms are the connections between subsets of the information path.

4. PROPOSED METHOD

This section proposes our method for selecting the anchor shackle term and classifying the instances according to the detected information path.

4.1 Anchor Shackle Term Selection

The anchor shackle term is the core of information path detection. In addition to its strong categorical discriminability, it should also be commonly used. We define a precise process to decide which term is anchor shackle term. First, as a widely used word, it should be the common word between $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$. So the anchor shackle term w satisfies the following criteria: $w \in W, W = \{w | d_{\mathcal{D}_{train}}(w) \geq 1 \wedge d_{\mathcal{D}_{test}}(w) \geq 1\}$, here $d_{\mathcal{D}}(w)$ means the number of documents in dataset $\mathcal{D}$ containing the anchor shackle term w. Next, we adopt *entropy* to measure its categorical discriminability since entropy is a very useful indicator for feature selection. We just use training set $\mathcal{D}_{train}$ since instance labels are required when calculating the entropy. The formula of calculating the categorical discriminability of data is shown as follows:

$$Entropy(w) = -\sum_{c_i} \frac{d(w, c_i)}{d_{\mathcal{D}_{train}}(w)} \cdot \log \frac{d(w, c_i)}{d_{\mathcal{D}_{train}}(w)} \quad (1)$$

Here $d(w, c_i)$ denotes the number of documents containing the word w with the label c_i. We do not consider documents that do not contain the word w like the work [17]. After the words having been selected, we use those documents containing the selected word. So the categorical mixing degree of documents containing the word is more important to us. We consider the characteristic of wide usage of the anchor shackle term by using appearing time in short texts. Assuming there are two common words, if the numbers of documents containing the words are either both very large or both very small, the appearing time cannot be a very important factor to determine which is to be selected as both of them are widely or rarely used. At this time, we should care more about the entropy. Only when the appearing times of two words are neither too large nor too small, the effect of appearing time and entropy should be considered equally. The sigmoid function can meet this requirement.

Based on this intuition, we design the following formula to score each word $w \in W$ and then select the word with the highest score as the anchor shackle term in each iteration.

$$score(w) = \frac{f(d_{Tr}(w) - \bar{d}_{Tr}(W_{Tr})) \cdot f(d_{Te}(w) - \bar{d}_{Te}(W_{Te}))}{f(Entropy(w))} \quad (2)$$

Here $f(x)$ is the sigmoid function that $f(x) = \frac{1}{1+e^{-x}}$. The subscript Tr or Te represent $\mathcal{D}_{train}$ or $\mathcal{D}_{test}$ respectively. $d_{Tr}(w) = d_{\mathcal{D}_{train}}(w)$. $\bar{d}_{Tr}(W_{Tr})$ is the average appearing time of words in $\mathcal{D}_{train}$, $W_{Tr} = \{w | d_{\mathcal{D}_{train}}(w) \geq 1\}$, $\bar{d}_{Tr}(W_{Tr}) = \frac{\sum_w d_{\mathcal{D}_{train}}(w)}{|W_{Tr}|}, w \in W_{Tr}$ and $|W_{Tr}|$ is the number of elements in set W_{Tr}. $d_{Te}(w), \bar{d}_{Te}(W_{Te})$ are the same terms calculated on $\mathcal{D}_{test}$. The $Entropy(w)$ can be calculated according to the formula 1.

As the formula 2 shown, one common word is selected, which meets two criterias: 1) it appears enough times in the short texts; 2) it has strong categorical discriminability. Detecting two or more anchor shackle terms in one iteration is also available in our method, but the quality of them might not be good enough.

4.2 Main Framework

The main framework of our method is shown in Figure 3. Our method is executed by iterations of detecting following subsets on information path. The method is formally illustrated as follows:

Step 1. Select the anchor shackle term w according to the data of $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$.

Step 2. Generate the subsets d_{test} from $\mathcal{D}_{test}$ as the following subset on information path. According to the first characteristic of short text, the instances in training data that contain the selected anchor shackle term have small gap with test data. So, the generated subsets should satisfy:

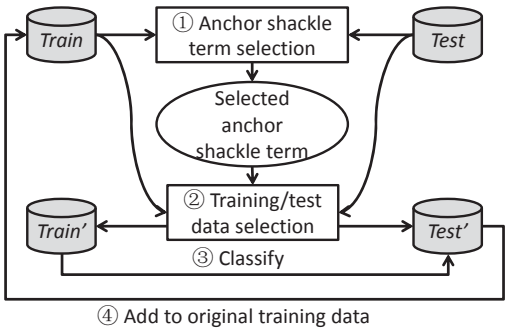


Figure 3: The framework of our method

$d_{test} \subseteq \mathcal{D}_{test} \wedge d_{train} \subseteq \mathcal{D}_{train}$ for $\forall x'_n \in d_{test}$, satisfying $w \in x'_n \wedge w \in x_n, < x_n, y_n > \in d_{train}$.

Step 3. Label the subset of test data d_{test} on information path. Train a classifier $\mathcal{C}$ based on selected subset of training data d_{train}, then yield $\widehat{y'_n}$ for each $x'_n \in d_{test}$ by $\mathcal{C}$. Now, $d_{test} = \{< x'_n, \widehat{y'_n} >\}$.

Step 4. Move the data d_{test} to the training data. $\mathcal{D}_{train} = \mathcal{D}_{train} \cup d_{test}$ and $\mathcal{D}_{test} = \mathcal{D}_{test} \setminus d_{test}$.

Step 5. If $\mathcal{D}_{test} = \emptyset$, the algorithm is terminated. Otherwise, go back to Step 1.

Note that those documents that have no common words with all other documents will be classified at last by the classifier trained on all the labeled data.

5. EXPERIMENTAL RESULTS

We evaluate the effectiveness of our method on two real-world short text dataset. Our experiments mainly focus on three aspects of our method: (1) To evaluate the performance by comparing with five baseline methods. (2) To offer the attributes by showing the anchor shackle terms. (3) To offer the insight of efficient-wise by giving the computational complexity and consuming time.

5.1 Dataset

Two real-world short text dataset are used. One is the "Search snippets"[1], which had been commonly used in other works on short text classification. The other is "Paper titles" extracted from DBLP[2], which contains thousands of paper titles collected by ourselves.

The Search snippets dataset is introduced in [10]. It consists of search snippets retrieved from Google. They used various phrases belonging to 8 different categories to do the Web search and selected top 20 (for training data) or 30 (for test data) snippets from the retrieval results to construct the dataset. Details of this dataset is summarized in Table 1.

The Paper titles dataset consists of paper titles belonging to two different categories. One is **multimedia**, the other is **network**. The details of the conferences and transactions used to consist of the dataset are illustrated in Table 2. Since the amount of paper titles for different categories are so unbalance, we randomly select 450 paper titles

[1]Collected by Xuan-Hieu Phan, using JWebPro: http://jwebpro.sourceforge.net.

[2]http://www.informatik.uni-trier.de/~ley/db/.

Table 1: Google search snippets as training & test data

Category	#Training data	#Test data
Business	1200	300
Computers	1200	300
Culture-Arts-Ent	1880	330
Education-Science	2360	300
Engineering	220	150
Health	880	300
Politics-Society	1200	300
Sports	1120	300
Total	10060	2280

Table 2: Research paper titles as training & test data

Data	Category / Item	Multimedia	Network
Train	Conf. or Trans.	TIP	MOBICOM
	Year	1995–2009	1995–2011
	Amount	2888	480
	Conf. or Trans.	TMM	SECON
	Year	1999–2013	2004–2012
	Amount	1308	740
Test	Conf. or Trans.	TVCG	JSAC
	Year	1995–2009	1993–2011
	Amount	1104	2614

(uniform in different years) of each transaction or conference in each time of experiment and do the experiment for 20 times.

5.2 Baseline Methods

In the experiments, we compare our method with five baseline methods. (1) **PH** is an enriching method introduced in [10], it extends short texts with single granularity topics. (2) **CH** is an enriching method proposed in [3], which extends short texts with multi-granularity topics. (3) **Conf** is a semi-supervised learning method which moves the instances with high classification confidence from test data to training data in one iteration. (4) **NI** is a semi-supervised learning method introduced in [8] which updates the label of texts by rebuilding the classifier according to both training data and test data in each iteration. (5) **Conv** is a conventional method which simply classifies the unlabeled test data by a classifier trained by all of the given training data.

Moreover, notice that two enriching methods need external text resources. For experiments on the Search snippets, an external dataset[3] is available. For experiments on the Paper titles, we collect an external corpus related to two fields (Multimedia and Network) from Wikipedia by using JWikiDocs following the same way as illustrated in [10].

5.3 Results

Logistic Regression[4] is used as the classifier. Default parameter settings of it are used. Note that each result on Paper titles dataset in the following figures (Figures 4(b) and

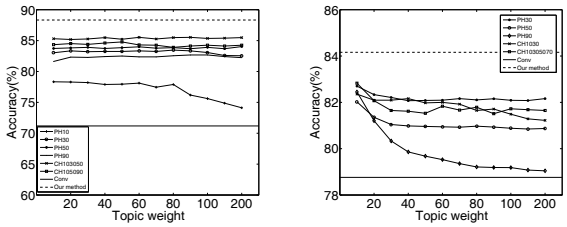
[3]Collected by Xuan-Hieu Phan, using JWikiDocs: http://jwebpro.sourceforge.net

[4]Liblinear: http://www.csie.ntu.edu.tw/~cjlin/liblinear, L2-regularized logistic regression.

5(b)) is the average results of 20 times experiments with the corresponding parameter settings.

5.3.1 Comparison with Enriching Methods

For both enriching methods PH and CH, two parameters need to be set: (1) the number of topics learned from external dataset, (2) the weight of topics. In addition, the method CH has another parameter, i.e. the number of granularity of topics, which is set to 3 on Search Snippets as in [3], and to 2 and 4 respectively on Paper Titles.



(a) Search Snippets (b) Paper Titles

Figure 4: Accuracy comparison with the enriching methods

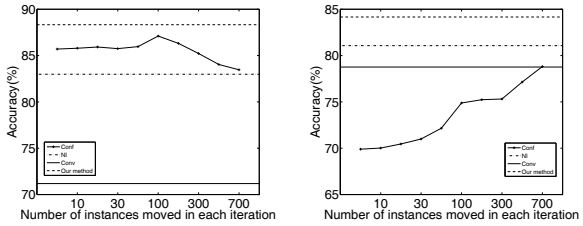
The results are illustrated in Figure 4. The numbers after the PH and CH of the legend represent the number of topics mined for external corpus. We can see that our method outperforms the baseline methods on both dataset.

In addition, some details can explain the reason why our method works well. For example, there are texts talking about "bank's transactions processed by computer program" in test data of Search Snippets, which belong to the "business" category. However, no corresponding labeled text talking about similar topics can be found in the same category. Thus the enriching methods simply extend the topics about computer programming and bank to the unlabeled texts. As the training dataset has a category "computers", which contains large amount of texts talking about computer programming, the texts are classified incorrectly to the category "computers". However, before classifying these texts, our method has automatically found and classified some unlabeled texts talking about "bank" in advance. As these texts also contain a lot of words such as stock and loans, which are widely existed in the labeled training data, these texts can be classified correctly. And then with the help of these correctly classified texts, the texts talking about "bank's transactions processed by computer program" can be labeled correctly.

5.3.2 Comparison with Semi-supervised Methods

In method Conf, the amount of instances moved from training dataset to test dataset in each iteration is a parameter that needs to be tuned. For the method NI, the parameter of iteration times is set to 50.

The results are illustrated in Figure 5. We discover that our method performs better than the baseline methods. For Search snippets, it consists of search results of synonyms obtained from Google. As the search words of same category share similar meanings, the search snippets are also related to each other in content for same category. Thus, the information path can be detected well by anchor shackle terms. For Paper Titles, though the texts in the same category are talking about similar topics, there are also some gaps between paper titles because each paper can only talk about a sub-field of the category it belongs to. However, paper titles of the same conference are more likely to be related with each other as each conference always has a specific theme, which can be viewed as a sub-field of the category. Moreover, the Conf method performs even worse than the Conv method mainly because of this gap. The label got with high confidence is not correct.



(a) Search Snippets (b) Paper Titles

Figure 5: Accuracy comparison with the semi-supervised learning methods

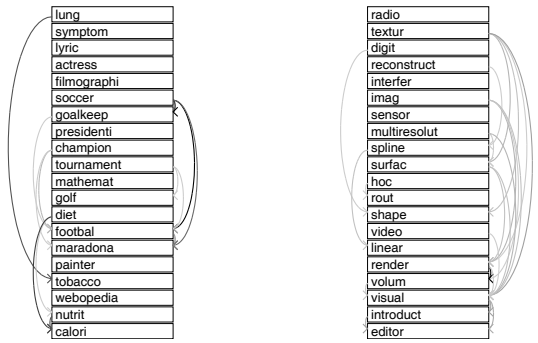
On Paper titles dataset, though the average performance of our method outperforms other methods, we provide some details here. Comparing with baseline methods in different parameter settings, our method gets the best accuracy in 10 out of 20 experiments. The second best method "CH10305070" only achieves best accuracy in 6 out of 20 experiments (with its carefully parameter setting). Our method has dominant advantages on this dataset.

5.3.3 Illustration of Information Path

In order to offer the insights of our method, the relations of anchor shackle terms are given. Here, the relations of anchor shackle terms in the first 20 iterations on two dataset are illustrated in Figure 6. We remove stop words and stem the words using TMG[5] for preprocessing, so some words in the two figures are different from their formal spellings. The anchor shackle terms chosen to be shown on the Paper titles are results of the 11th experiments as it achieves similar accuracy to the average level of our method on that dataset. The directed lines connecting two anchor shackle terms represent the texts of test data containing the two terms. These texts are labeled when the term of starting point is detected and help classify the texts containing the term of terminal point. The more texts like this, the blacker the line is and closer the relation of the two anchor shackle terms, since the two terms are more likely to prompt in the same text. The terms with no directed lines pointed to indicate that the test data containing that term are classified without help of the former classified text in test data. The texts containing those terms have small gap with training data.

The anchor shackle terms that connect the subsets of information path are quite reasonable on the two dataset. For example, there is an information path on Search snippets connected by a chain of words: lung → tobacco → prevent → epidem → infect → ... (The words after "tobacco" is not illustrated in Figure 6(a) because they are detected in later iterations). Actually, most test documents containing the word "infect" should be classified to the category "Health". However, in the training data, texts containing "infect" are

[5]http://scgroup20.ceid.upatras.gr:8000/tmg/.



(a) Search Snippets (b) Paper Titles

Figure 6: The relations of anchor shackle terms selected on two dataset in first 20 iterations

in both the "Health" and "Computer" categories, since some texts are talking about "computer virus" in training data. So, if we classify the test data directly, texts those belong to the category "Computers" in training data would bring negative effect because such texts have word co-occurrences with texts in "Health" category. If we use all the training data to build the classifier as the method Conf, texts of category "Computer" talking about "computer virus" might also bring in negative effects. The classifier can classify the texts talking about "Health" as "Computers" with a high confidence, while the given label is totally wrong. Our method detects the texts talking about "lung" first, which are more certain to the category "Health" according to the training data. Then we distinguish the confusing texts later when their related texts are labeled.

5.3.4 *Efficiency of Our Method*

The computational complexity of our algorithm for each iteration is $O(|V|+(|\hat{L}|+1)\cdot|D|+Tr(\mathcal{C})+Te(\mathcal{C}))$, where $|V|$ is the size of the vocabulary, $|\hat{L}|$ is the average amounts of common words appearing in each document, $|D|$ is the size of instances of the whole dataset, $Tr(\mathcal{C})$ and $Te(\mathcal{C})$ are the training and predicting complexity of the chosen classifier $\mathcal{C}$ respectively. Since the algorithm is terminated automatically when all the test data is labeled, iteration number is determined by the data itself. The more word co-occurrence of the dataset, the fewer the iteration times. With respect to quantitative analysis, average running time (20 experiments) on the real dataset is 135.58 seconds (on Search snippets) and 29.24 seconds (on Paper titles) respectively.

6. CONCLUSION

This paper proposes an auxiliary-resource-free method for short text classification by incorporating the inherent *information path* that is very common in various real world applications. Since our method is based on the essential characteristics of the short text data, it is reasonable to anticipate better classification accuracy. But if the *information path* is not existed in the given dataset, our method may not perform well.

In future, we intend to involve some existing methods, including those leveraging auxiliary resource into our schemas to further improve its classification accuracy.

7. ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (60973103, 90924003).

8. REFERENCES

[1] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proc. ICML*, 2009.

[2] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. In *Proc. WWW*, 2007.

[3] M. Chen, X. Jin, and D. Shen. Short text classification improved by learning multi-granularity topics. In *Proc. IJCAI*, 2011.

[4] X. Hu, X. Zhang, C. Lu, E. Park, and X. Zhou. Exploiting wikipedia as external knowledge for document clustering. In *Proc. KDD*, 2009.

[5] O. Jin, N. Liu, K. Zhao, Y. Yu, and Q. Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proc. CIKM*, 2011.

[6] L. Li, X. Jin, and M. Long. Topic correlation analysis for cross-domain text classification. In *Proc. AAAI*, 2012.

[7] G. Long, L. Chen, X. Zhu, and C. Zhang. Tcsst: transfer classification of short & sparse text using external data. In *Proc. CIKM*, 2012.

[8] K. Nigam. *Using unlabeled data to improve text classification*. PhD thesis, CMU, 2001.

[9] S. Pan and Q. Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[10] X. Phan, L. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proc. WWW*, 2008.

[11] X. Quan, G. Liu, Z. Lu, X. Ni, and L. Wenyin. Short text similarity based on probabilistic topics. *KAIS*, 25(3):473–491, 2010.

[12] M. Sahami and T. Heilman. A web-based kernel function for measuring the similarity of short text snippets. In *Proc. WWW*, 2006.

[13] D. Shen, R. Pan, J. Sun, J. Pan, K. Wu, J. Yin, and Q. Yang. Query enrichment for web-query classification. *TOIS*, 24(3):320–352, 2006.

[14] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in twitter to improve information filtering. In *Proc. SIGIR*, 2010.

[15] A. Sun. Short text classification using very few words. In *Proc. SIGIR*, 2012.

[16] D. Vitale, P. Ferragina, and U. Scaiella. Classification of short texts by deploying topical annotations. In *Proc. ECIR*, 2012.

[17] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Proc. ICML*, 1997.

[18] W. Yih and C. Meek. Improving similarity measures for short segments of text. In *Proc. AAAI*, 2007.

[19] X. Zhu. Semi-supervised learning literature survey. *Computer Sciences*, 2005.