Improving Text Categorization Using Domain Knowledge

Jingbo Zhu and Wenliang Chen

Natural Language Processing Lab
Institute of Computer Software and Theory
Northeastern University, Shenyang, P.R. China, 110004
{zhujingbo,chenwl}@mail.neu.edu.cn

Abstract. In this paper, we mainly study and propose an approach to improve document classification using domain knowledge. First we introduce a domain knowledge dictionary NEUKD, and propose two models which use domain knowledge as textual features for text categorization. The first one is BOTW model which uses domain associated terms and conventional words as textual features. The other one is BOF model which uses domain features as textual features. But due to limitation of size of domain knowledge dictionary, we study and use a machine learning technique to solve the problem, and propose a BOL model which could be considered as the extended version of BOF model. In the comparison experiments, we consider naïve Bayes system based on BOW model as baseline system. Comparison experimental results of naïve Bayes systems based on those four models (BOW, BOTW, BOF and BOL) show that domain knowledge is very useful for improving text categorization. BOTW model performs better than BOW model, and BOL and BOF models perform better than BOW model in small number of features cases. Through learning new features using machine learning technique, BOL model performs better than BOF model.

1 Introduction

Text categorization (TC) is the problem of automatically assigning one or more predefined categories to free text documents. TC is a hard and very useful operation frequently applied to the assignment of subject categories to documents, to route and filter texts, or as a part of natural language processing systems. A growing number of statistical classification methods and machine learning techniques have been applied to text categorization in recent years, such as Rocchio[1][2], SVM[3], Decision Tree[4], Maximum Entropy model[5], naïve Bayes[6]. In those models, typically the document vectors are formed using bag-of-words model. Each document text is represented by a vector of weighted terms. The terms attached to documents for content representation purposes may be words or phrases derived from the document texts by an automatic indexing procedure.

As we know, it is natural for people to know the topic of the document when they see specific words in the document. For example, when we read a news, if title of the news includes a word "姚明(Yao Ming)", as we know, "姚明(Yao Ming)" is a famous China basketball athlete in US NBA game, so we could recognize the topic of the document is about "篮球, 体育(Basketball, Sports)" with our domain knowledge. In this paper, we call the specific word "姚明(Yao Ming)" as a *Domain Associated Term* (DAT). A DAT is a word or a phrase (compound words) that enable humans to

A. Montoyo et al. (Eds.): NLDB 2005, LNCS 3513, pp. 103–113, 2005.

recognize intuitively a topic of text with their domain knowledge. As we know, domain knowledge is a kind of commonsense knowledge. We think that domain knowledge is very useful for text understanding tasks, such as information retrieval, text categorization, and document summarization.

Some researchers used knowledge bases to knowledge-based text categorization[7]. First they group words into special semantic clusters according to their definition of knowledge bases such as WordNet or HowNet. Then they use these clusters as features for text categorization. As we know, WordNet and HowNet are lexical and semantic knowledge dictionaries. Sangkon Lee *et. al.* [8] proposed a new passage retrieval method which divides a text into several passages by using field-associated terms like our DATs. But their knowledge bases are generated by hand, and in particular, due to limitation of size of knowledge bases, they can't include enough words or features for text categorization. In this paper, we study and propose two models which use domain knowledge as textual features to improve text categorization. And we use a machine learning technique to solve problem of knowledge-based text categorization caused by limitation of size of domain knowledge dictionary.

The following paper is organized as follows. In section 2, our domain knowledge dictionary is introduced. The baseline NB system based on BOW model is given in section 3. In section 4 we propose two new models using domain knowledge as textual features for text categorization. In section 5, we propose a machine learning technique to improve knowledge- based text categorization. Comparison experimental results of four models are given in section 6. At last, we address conclusions and future work in section 7.

2 Domain Knowledge Dictionary

First, we introduce briefly the domain knowledge hierarchy description framework (DKF) which can be divided into three levels shown in Figure 1: *Domain Level (DL)*, *Domain Feature Level (DFL)* and *Domain Associated Term Level (DATL)*. The DL is the top level which includes many domains, such as "体育(Sports)", "军事(Military Affairs)". The DFL is the second level which includes many domain features. A domain has one or more domain features. For example, domain "军事(Military Affairs)" has many domain features, such as "军队(Army Feature)", "武器(Weapon Feature)" and "战争(War Feature)". The DATL is the third level which includes many domain associated terms. As we know, many domain associated terms could indicate a same domain feature. For example, for domain feature "战争(War)", it includes many domain associated terms such as "中东战争(Mid-East War)", "伊拉克战争(Iraq War)" and "阿富汗战争(Afghanistan War)".

Since 1996 we employed a semi-automatic machine learning technique to acquire domain knowledge from a large amount of labeled and unlabeled corpus, and built a domain knowledge dictionary named NEUKD[9][10]. Items defined in the NEUKD include domain associated term, domain feature and domain. Currently 40 domains, 982 domain features and 413,534 domain associated terms are defined in NEUKD. Some instances defined in NEUKD are shown in Table 1. For example, term "三峡工程(The Sanxia Project)" indicates domain feature "水利工程(Irrigation Project)" of domain "水利(Irrigation Works)".

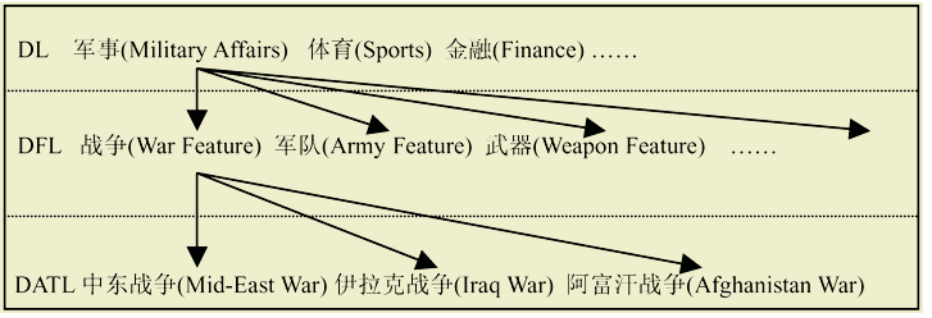

Fig. 1. Parts of domain knowledge hierarchy description framework (DKF).

Table 1. Some instances defined in NEUKD.

Domain Associated Terms	Domain Features	Domain
姚明 (Yao Ming)	篮球，运动员 (Basketball, Athlete)	体育 (Sports)
三峡工程 (The Sanxia project)	水利工程 (Irrigation Project)	水利 (Irrigation Works)
赛季 (Match Season)	比赛 (Match)	体育 (Sports)
阿森纳队 (Arsenal Team)	足球 (Football)	体育 (Sports)
中国工商银行 (Industrial and commercial bank of China)	银行 (Bank)	金融 (Finance)

3 Baseline NB System

In recent years Naïve Bayes (NB) approaches has been applied for document classification, and found to perform well. The basic idea in naïve Bayes approaches is to use the joint probabilities of words and categories to estimate the probabilities of categories when a document is given. The naïve part of NB method is the assumption of word independency, i.e., the conditional probability of a word given a category is assumed to be independent from the conditional probabilities of other words given that category. This assumption makes the computation of the NB classifiers far more efficient than the exponential complexity of non-naïve Bayes approaches because it does not use word combinations as predictors[11]. There are several versions of the NB classifiers. Recent studies on a multinomial mixture model have reported improved performance scores for this version over some other commonly used versions of NB on several data collections[6]. There are several versions of the NB classifiers. McCallum and Nigam gave comparative analysis between two different NB models: multivariate Bernoulli model and multinomial model. They found that the multivariate Bernoulli performs well with small vocabulary sizes, but that the multinomial performs better at larger vocabulary size.

In this paper we use the multinomial mixture model of NB by to classify documents. We only describe multinomial NB model briefly since full details have been presented in [6]. The basic idea in naïve Bayes approaches is to use the joint prob-

abilities of words and categories to estimate the probabilities of categories when a document is given. Given a document d for classification, we calculate the probabilities of each category c as

$$P(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)}$$
$$= P(c)\prod_{i=1}^{|T|} \frac{P(t_i \mid c)^{N(t_i|d)}}{N(t_i \mid d)!}$$

Where P(c) is the class prior probabilities, $N(t_i|d)$ is the frequency of word t_i in document d, T is the vocabulary and $|T|$ is the size of T, t_i is the i^{th} word in the vocabulary, and $P(t_i|c)$ thus represents the probability that a randomly drawn word from a randomly drawn document in category c will be the word t_i. We can calculate Bayes-optimal estimates for these parameters from a set of labeled training data.

In this paper, we use NEU_TC data set[12] to evaluate the performance of NB classifier and our classifiers. The NEU_TC data set contains Chinese web pages collected from web sites. The pages are divided into 37 classes according to "Chinese Library Categorization"[13]. It consists of 14,459 documents. We do not use tag information of pages. We use the toolkit CipSegSDK[14] for word segmentation. We removed all words that had less than two occurrences. The resulting vocabulary has about 60000 words.

In the experiments, we use 5-fold cross validation where we randomly and uniformly split each class into 5 folds and we take four folds for training and one fold for testing. In the cross-validated experiments we report on the average performance. For evaluating the effectiveness of category assignments by classifiers to documents, we use the conventional recall, precision and F_1 measures. Recall is defined to be the ratio of correct assignments by the system divided by the total number of correct assignments. Precision is the ratio of correct assignments by the system divided by the total number of the system's assignments. The F_1 measure combines recall (r) and precision (p) with an equal weight in the following form:

$$F_1(r, p) = \frac{2rp}{r + p}$$

In fact, these scores can be computed for the binary decisions on each individual category first and then be averaged over categories. The way is called macro-averaging method. For evaluating performance average across class, we use the former way called micro averaging method in this paper which balances recall and precision in a way that gives them equal weight. The micro- averaged F_1 measure has been widely used in cross-method comparisons.

The most commonly used document representation is the so called vector space model[15]. In the vector space model, documents are represented by vectors of terms (textual features, e.g. words, phases, etc.). A document D can be represented by a description vector dv as: $dv = <c_1, c_2, \ldots\ldots c_n>$. Where n is the total number of the selected terms and c_i denotes the term weight of a term t_i in the document D. Conventional bag-of-words model (BOW) uses conventional words as textual features, so each document text can be represented by a vector of weighted words.

In this paper, we use the BOW model as baseline NB system. Given above experimental settings, CHI measure is used for feature selection which is the best features

selection methods according to our experiments, the best performance of baseline NB system is 74.6% F1.

4 Domain Knowledge as Textual Features

4.1 BOTW Model

In this paper, we wish to use domain knowledge dictionary (NEUKD) to improve text categorization. In BOW model, conventional words are used as textual features for text categorization. As above mentioned, more than 400000 domain associated terms (DATs) are defined in the NEUKD, such as "姚明(Yao Ming) ", "三峡工程(The Sanxia project)", and "中国工商银行(Industrial and commercial bank of China)" shown in table 1. In this paper, we use both those domain associated terms and conventional words as textual features, called BOTW models (short for bag-of-terms and words model).

Now we give an example to explain simply the differences between BOW and BOTW models. For example, in the previous examples, a DAT "三峡工程(The Sanxia project, Sanxia is a LOCATION name of China)" can be used as a textual feature in BOTW model. But in BOW model it is not used as a textual feature, we consider two words "三峡(The Sanxia)" and "工程(project)" as two different textual features, respectively. From above examples, it is natural for us to understand those domains associated terms are a richer and more precise representation of meaning than keywords (conventional words).

In fact, the classification computation procedure based on BOTW model is same as BOW model. CHI also is used to feature selection in our experiments. According to experimental results, the best performance of BOTW-based classifier is 76.7% F1 which is higher 2.1% than baseline system (BOW model).

4.2 BOF Model

As above mentioned, in our NEUKD, each DAT is associated with one or more domain features which the DAT indicates. Such as the DAT "三峡工程(The Sanxia Project)" indicates domain feature "水利工程(Irrigation Project)" of domain "水利(Irrigation Works)". Similar to BOTW model, we want to use those domain features as textual features in NB classifier, called BOF model (short for bag-of-features model). In other words, we do not use "三峡工程(The Sanxia Project)" as a textual feature, but its domain feature "水利工程(Irrigation Project)" as a textual feature in the BOF model.

In BOF model, we firstly transform all DATs into domain features according to definitions in NEUKD, and group DATs of same domain features as a cluster, called Topic Cluster.

Let T denote set of domain feature, t is a domain feature of T, F denote set of Topic Cluster, DF is set of DATs, df_i is i^{th} DAT in DF.

If a DAT df_i has a domain feature t_j, then df_i can been added into the Topic Cluster $F(t_j)$. We group all domain features in NEUKD into the Topic Clusters. For Examples,

Topic Cluster named "体育(sports)" includes some DATs, such as "赛季(match season)", "阿森纳队(Arsenal)", "奥运会(Olympic Games)", "乒乓球(Table Tennis)", "姚明(Yao Ming)".

In BOF model, we use topic clusters as textual features for text categorization. Also the classification computation procedure based on BOF model is same as BOW model. According to experimental results, the best performance of BOF-based classifier is 68% F1 which is less than BOW and BOTW models. The main reason is that due to the limitation of size of our NEUKD, a large amount of words are removed in training procedure, because no domain features of those removed words are defined in our NEUKD. According to statistical analysis of words occurring in training corpus, we find that 65.01% words occurring in training corpus are not included in the NEUKD. In fact, many of those removed words are useful for text categorization. As denoted in section 2, about 1000 domain features are defined in our NEUKD, so for BOF model, the maximum number of textual features is the total number of domain features defined in NEUKD. But it is very significant that when BOF model performs better than BOW and BOTW model in small number of textual features cases. Detailed analysis will be given in following sections.

5 BOL Model

To solve the above problem of the limitation of NEUKD, in this paper, we propose a machine learning technique to improve BOF model. The basic ideas are that we wish to learn new words from labeled documents, group them into the predefined topic clusters based on NEUKD which are formed and used as textual features in BOF model discussed in section 4.2, and use new topic clusters as textual features for text categorization. We call the new model as BOL model which is extended version of BOF model. First we group all DATs originally defined in NEUKD into topic clusters as described in BOF model, which are used as seeds in following learning procedure. Then we want to group other words (not be defined in NEUKD) into these topic clusters.

In this section, we introduce how to learn some words from labeled documents using topic clusters and class distribution of words. We are focus on two topics:

- How to measure the similarity between a word and a topic cluster;
- Learning algorithm.

The first question of such procedures is how to measure the similarity between a word and a cluster. Class distribution of words has showed good performance in words clustering [16][17]. We use a form of "Kullback-Leibler divergence to the mean." Unlike previous works, we propose a new similarity measure for learning algorithm. In our algorithm, the word can only be grouped into one cluster.

5.1 How to Measure the Similarity

We should define a similarity measure between a word and a topic cluster, and add the word into the most similar cluster that no longer distinguishes among the word and other members (words or DATs) of the topic cluster. Then, the parameters of the cluster become the weighted average of the parameters of its members.

Firstly, we define the distribution $P(C|w_t)$ as the random variable over classes C, and its distribution given a particular word w_t. When we have two words w_t and w_s, they will be put into the same cluster f. The new distribution of the cluster is defined

$$P(C|f)=P(C|w_t \vee w_s)$$
$$=\frac{P(w_t)}{P(w_t)+P(w_s)}P(C|w_t)+\frac{P(w_s)}{P(w_t)+P(w_s)}P(C|w_s) \quad (1)$$

Now we consider the case that a word w_t and a topic cluster f will be put into a new cluster f_{new}. The distribution of f_{new} is defined as

$$P(C|f_{new})=P(C|w_t \vee f)$$
$$=\frac{P(w_t)}{P(w_t)+P(f)}P(C|w_t)+\frac{P(f)}{P(w_t)+P(f)}P(C|f) \quad (2)$$

Secondly, we turn back the above question of how to measure the difference between two probability distributions. Kullback-Leibler divergence is used to do this. The KL divergence between the class distributions induced by w_t and w_s is written as $D(P(C \mid w_t) \,\|\, P(C \mid w_s))$, and is defined as

$$-\sum_{j=1}^{|C|} P(c_j \mid w_t)\log(\frac{P(c_j \mid w_t)}{P(c_j \mid w_s)}) \quad (3)$$

But KL divergence has some odd properties. In order to cover its problems, Baker and McCallum[16] proposed a measure named "KL divergence to the mean" to measure the similarity of two distributions. It is defined

$$S_{Baker}=\frac{P(w_t)}{P(w_t)+P(w_s)}D(P(C|w_t)\|P(C|w_t \vee w_s))$$
$$+\frac{P(w_s)}{P(w_t)+P(w_s)}D(P(C|w_s)\|P(C|w_t \vee w_s))$$

In this paper, we usually measure the similarity of a word and a topic cluster. The cluster has included many words that defined in NEUKD. "KL divergence to the mean" has some problems when it measures the similarity between a word and a cluster. In most cases, if the cluster includes more words, then the result is more similar. Experimental results show that several clusters include so many words while most clusters include only few words. The reason is that Baker and McCallum's "KL divergence to the mean" doesn't account for global information. It can't work well if the numbers of features in the clusters are very different at beginning.

Thus, in learning algorithm we use a new measure that does not have this problem. We add a factor according to the number of words in the cluster. The new similarity of a word w_t and a cluster f_j is defined

$$S=\frac{N(w_t)+N(f_j)}{\sum_{i=1}^{|L|}N(f_i)+|W|}S_{Baker}(w_t,f_j) \quad (4)$$

$$S_{Bucket}(w_t, f_j) = \frac{P(w_t)}{P(w_t)+P(f_j)} D(P(C|w_t) \| P(C|w_t \vee f_j)) + \frac{P(f_j)}{P(w_t)+P(f_j)} D(P(C|f_j) \| P(C|w_t \vee f_j))$$

Where $N(f_i)$ denote the number of words in the cluster f_i, W is the list of candidate words. Equation 4 can be understood as the balance of all clusters according to the number of words in them. Our experimental results show that it can work well even if the numbers of features in them are very different at the beginning.

5.2 Learning Algorithm

Table 2. The Learning Algorithm.

- Preprocessing: Text segmentation, extracting candidate words, and sort the candidate words by CHI method. As above mentioned, all candidate words which are not defined in NEUKD will be grouped into topic clusters in this process.
- Initialization: The words, which are defined in NEUKD, are first added to corresponding topic clusters according to their associated domain features, respectively.
- Loop until all candidate words have been put into topic clusters:
  - Measure similarity of a candidate word and each topic cluster, respectively.
  - Put the candidate word into the most similar topic cluster.

6 Experimental Results

6.1 Experiment 1: Comparison of BOW, BOTW, BOF, and BOL Classifiers

Using experimental settings discussed in section 2 to evaluate the performance of these four models based on NB classifier, we construct four systems in the experiments, including BOW, BOTW, BOF and BOL classifier. CHI measure is used to feature selection in all system. Detailed comparison results are shown in figure 2.

In figure 2, we could find that BOTW classifier always performs better than BOW classifier when the number of features is larger than about 500. As above mentioned, BOTW classifier considers domain associated items (DAIs) as textual features. From comparative experimental results of BOTW and BOW classifiers, we think that domain associated items are a richer and more precise representation of meaning than conventional words.

Because the total number of domain features in NEUKD is only 982, in figure 2 we find the maximum number of features (domain features) for BOF and BOL classifier is less than 1000. When the number of features is between 200 and 1000, BOF classifier performs better than BOW and BOTW classifiers. It is also obvious that BOL classifier always performs better than other three classifiers when the number of features is less than 1000. As above mentioned, in BOL model, we use a machine learning technique to solve the problem of limitation of size of NEUKD, and group rest 65.01% words into predefined topic clusters as textual features in BOL model. So the classifier based on BOL model can yield better performance than BOF model.

6.2 Experiment 2: Performance Analysis Based on Different Size of Corpus

In this experiment, we study the performance of BOW and BOL models when varying number of features and size of training corpus. In Figure 3, T10, T30 and T50 denote the different number of training corpus as 10, 30 and 50 training documents for each category. Naturally, the more documents for training procedure are used, the better the performance of classifier is.

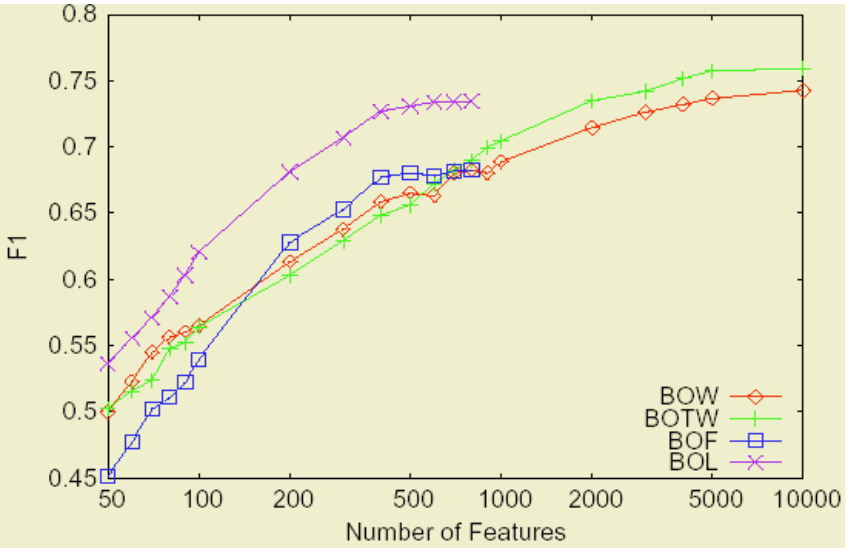

Fig. 2. Experimental results of BOW, BOTW, BOF, BOL classifiers.

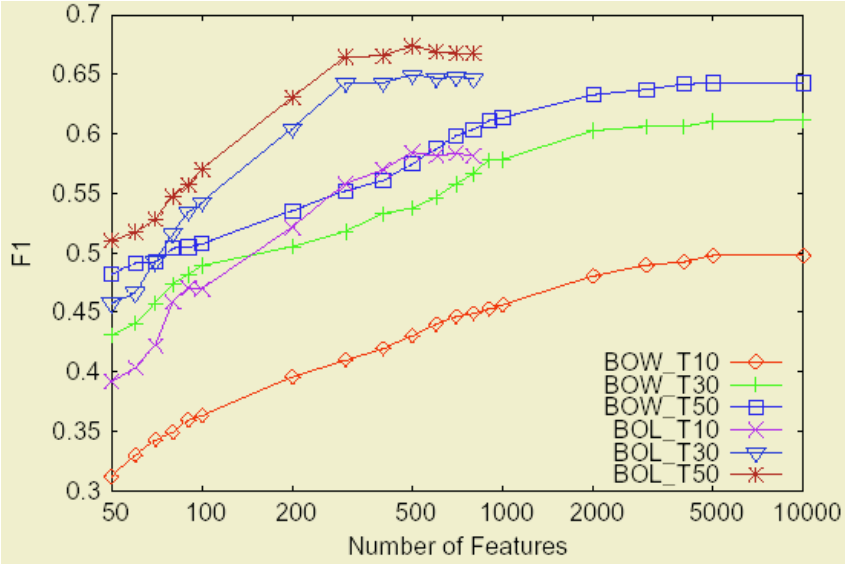

Fig. 3. Performance Analysis Based on Different Size of Training Corpus.

In Figure 3, BOL_T10 classifier yields 58.4% F1 with 500 features and BOW_T10 only yields 49.7% F1 with same number of features. And when the number of features is 500, BOW_T50 classifier provides only 57.5% F1, which is lese 0.9% than BOL_T10 classifier. It is obvious that BOL performs better than BOW in small number of features cases.

The best result of BOL_T50 classifier is 67.4% F1, which is higher 9% than BOL_T10 classifier. And the best result of BOW_T50 is 64.2% F1, which is higher 14.5% than BOW_T10 classifier. And the best performance of BOL_T30 classifier is 65.0% F1, which is higher 0.8% than the best performance of BOW_T30 classifier. The best performance of BOL_T50 is 67.4% F1, which is higher 3.2% F1 than the best performance of BOW_T50 classifier. When given small size of training corpus, BOL performs better than BOW. As we know, small size of training corpus would cause serious data sparseness problem. From above comparative experimental results we find that domain knowledge is beneficial to solve data sparseness problem.

7 Conclusions and Future Work

In this paper, we study and propose an approach to improve text categorization by using domain knowledge dictionary (NEUKD). We propose two models using domain knowledge as textual features. The first one is BOTW model which uses domain associated terms and conventional words as textual features. The other one is BOF model which uses domain features as textual features. But due to limitation of size of domain knowledge dictionary, many useful words are removed in training procedure. We study and use a machine learning technique to solve the problem to improve knowledge-based text categorization, and propose a BOL model which could be considered as the extension version of BOF model. We use NB system based on BOW model as baseline system. Comparison experimental results of those four models (BOW, BOTW, BOF and BOL) denote that domain knowledge is very useful for improving text categorization. In fact, a lot of knowledge-based NLP application systems also face the problem of limitation of size of knowledge bases. Like our work discussed in this paper, we think using machine learning techniques is a good way to solve such problem. In the future work, we will study how to apply the domain knowledge to improve information retrieval, information extraction, topic detection and tracking (TDT) etc.

Acknowledgements

This research was supported in part by the National Natural Science Foundation of China & Microsoft Asia Research Centre(No. 60203019), the Key Project of Chinese Ministry of Education (No. 104065), and the National Natural Science Foundation of China (No. 60473140).

References

1. David J. Ittner, David D. Lewis, and David D. Ahn, Text categorization of low quality images. In Symposium on Document Analysis and Information Retrieval, Las Vegas, , Las Vegas. 1995.

2. D. Lewis, R. Schapire, J. Callan, and R. Papka, Training Algorithms for Linear Text Classifiers, Proceedings of ACM SIGIR, pp.298-306, 1996.
3. T. Joachims, Text categorization with Support Vector Machines: Learning with many relevant features. In Machine Learning: ECML-98, Tenth European Conference on Machine Learning, pp. 137--142. 1998
4. D. Lewis, A Comparison of Two Learning Algorithms for Text Categorization, Symposium on Document Analysis and IR, 1994
5. K. Nigam, John Lafferty, and Andrew McCallum, Using maximum entropy for text classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61--67, 1999
6. McCallum and K.Nigam, A Comparison of Event Models for naïve Bayes Text Classification, In AAAI-98 Workshop on Learning for Text Categorization,1998
7. Scott, Sam Stan Matwin. Text classification using WordNet hypernyms. Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems, Montreal, 1998.
8. Sangkon Lee and Masami Shishibori. Passage Segmentation Based on Topic Matter, Computer Processing of Oriental Languages, 2002. V15, No 3, P305-340
9. Zhu Jingbo and Yao Tianshun. FIFA-based Text Classification, Journal of Chinese Information Processing, V16, No3, 2002.(In Chinese)
10. Chen Wenliang, Zhu Jingbo, Yao Tianshun, Automatic Learning Field Words by Bootstrapping, Proceedings of the 7th national conference on computational linguistics (JSCL 2003), 2003, (In Chinese)
11. Yiming Yang and Xin Liu, A re-examination of text categorization methods. In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval, 1999
12. Chen Wenliang, Chang Xingzhi, Wang Huizhen, Zhu Jingbo, and Yao Tianshun, Automatic Word Clustering for Text Categorization Using Global Information. AIRS2004, Beijing, 2004
13. China Library Categorization Editorial Board. China Library Categorization (The 4th ed.) (In Chinese), Beijing, Beijing Library Press 1999
14. Yao Tianshun, Zhu Jingbo, Zhang li, and Yang Ying, Natural Language Processing- research on making computers understand human languages, Tsinghua University Press, 2002, (In Chinese).
15. G.Salton and M.J.McGill, An introduction to modern information retrieval, McGraw-Hill, 1983
16. L.D.Baker and A.K.MCallum. Distributional clustering of words for text classification. In Proc. 21st Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pages 96--103, 1998.
17. F. Pereira. N. Tishby. L. Lee. Distributional clustering of English words. In 30th Annual Meeting of the ACL, p183-190, 1993.