Semi-supervised Text Categorization with Only a Few Positive and Unlabeled Documents

Fang Lu
College of Mathematics and Computer Science
Fuzhou University
Fuzhou, China

Qingyuan Bai *
College of Mathematics and Computer Science
Fuzhou University
Fuzhou, China

***Abstract*—This paper studies a special case of semi-supervised text categorization. We want to build a text classifier with only a set *P* of labeled positive documents from one class (called positive class) and a set *U* of a large number of unlabeled documents from both positive class and other diverse classes (called negative class). This kind of semi-supervised text classification is called positive and unlabeled learning (PU-Learning). Although there are some effective methods for PU-Learning, they do not perform very well when the labeled positive documents are very few. In this paper, we propose a refined method to do the PU-Learning with the known technique combining Rocchio and K-means algorithm. Considering the set *P* may be very small (≤5%), not only we extract more reliable negative documents from *U* but also enlarge the size of *P* with extracting some most reliable positive documents from *U*. Our experimental results show that the refined method can perform better when the set *P* is very small.**

Keywords-semi-supervised learning; text categorization; cluster

I. INTRODUCTION

Text categorization is one important task of text mining, aiming to solve the classification of large numbers of documents. There are some famous traditional text classification methods, such as Support Vector Machine, Naïve Bayes, K-Nearest Neighbor and so on. All these methods need lots of documents manually labeled from every class to train text classifiers, so they are also called supervised learning methods. However, getting lots of documents manually labeled is so time-consuming and unreasonable. Then a new learning method namely semi-supervised learning has been introduced into text categorization field. Semi-supervised text categorization uses only a few labeled documents and lots of unlabeled documents to train the classifier, so it doesn't need lots of labeled documents.

While this paper focuses on a special case of semi-supervised text classification which is with only a few labeled positive documents from one class and lots of unlabeled documents from both positive and many other classes. This problem is also called positive and unlabeled learning (PU-Learning)[1]. *U* is the unlabeled documents set, *P* represents the positive documents set from one special class which users are interested in. PU-Learning only has the knowledge about the positive and unlabeled documents set and build a classifier to classify the unlabeled documents into positive class or not positive class (namely negative class). PU-Learning can often be seen in real problems, for example, if at present we have only small positive web pages of sports news and lots of unlabeled web pages from both sports news and other kinds of news like entertainment, education, economic and so on, then how can we sort out the web pages of sports news with these positive and unlabeled web pages we have now? So this can be called a PU-Learning problem.

This paper is organized as follows: Section II introduces the related works, and our work is briefly presented in Section III. In Section IV, we introduce our method in detail. Then Section V gives the experimental methods and results. At last, the conclusion is given in Section VI.

II. RELATED WORKS

First, For PU-Learning, many researchers have proposed some effective methods. Liu et al[1] put forward S-EM algorithm, which uses spy technique to find out the reliable negative documents from *U* and then trains the classifier with EM algorithm. Yu et al[2] proposed the PEBL frame based on SVM algorithm to solve PU-Learning. Li et al[3] bring forward Rocchio-SVM algorithm, which firstly uses Rocchio algorithm to get the initial reliable negative documents from U and then adopts SVM algorithm to learn the classifier. Its experimental results show that Rocchio-SVM can achieve better performance than S-EM and PEBL. All these methods can be called as two-step algorithm, which firstly extracts the reliable negative examples from *U* and secondly adopts some text classification method to train text classifier. Liu et al[4] introduced the all two-step algorithms in their paper. The techniques of first step can be Naïve Bayes algorithm, Rocchio algorithm, the spy technique in S-EM, 1-DNF in PEBL. And the second step can use these four methods: running SVM only once, running EM, running SVM iteratively, running SVM iteratively and then select a final good classifier. So there are 16 methods combining the techniques of the above two steps. From experimental results, Liu et al found that the best technique of the first step is Rocchio algorithm.

Because the two-step algorithms only extract the reliable negative examples from *U*, when the positive documents set *P* is very small, they would get worse performance. Some researchers begin to consider extracting the reliable positive documents from *U* to expand *P*. These relative works can be seen in[5][6][7]. In this paper we also focus on the condition when P is very small (≤5%), and then give a new technique to improve the PU-Learning.

* Corresponding author.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on September 01,2021 at 09:27:12 UTC from IEEE Xplore. Restrictions apply.

III. OUR WORK

Because the two-step algorithms for PU-Learning perform well on the supposition that there are sufficient positive documents, they just mainly consider how to get more reliable negative examples from *U* for training text classifier. However, the two-step algorithms would not perform well when *P* is very small. This is due to the uniqueness of the PU-Learning problem. Firstly, there are no labeled negative documents; secondly, the positive documents are from one special class and the negative documents from other diverse classes. When the positive set *P* is small, it can't reflect the true feature distribution of the positive class well. The small positive set and high diversity of negative classes will make training a good classifier more difficult.

On this condition, we propose a new technique which not only extracts more reliable negative examples from *U* but also extracts most reliable positive examples from *U* to increase the size of *P*. Since Rocchio algorithm has been proved a good classifier which can extract more reliable negative examples from *U*[3][4], in this paper we adopt the Rocchio algorithm and k-means clustering method to extract more pure negative examples and get most reliable positive examples to increase the size of *P*. The experimental results show that when *P* is very small (≤5%), our method can get a significant improvement on the performance of text classification.

IV. PROPOSED METHOD

For PU-Learning, we only have the positive documents set *P* and the unlabeled documents set *U*. Since the unlabeled documents are from both positive class and other diverse classes and moreover *P* is very small, we can't directly treat *U* as *N*. In first step of our method, we also use Rocchio algorithm to find initial reliable negative documents from *U*, and then we adopt Rocchio and K-means algorithm to extract not only more reliable negative documents but also most reliable positive documents from *U*. In second step, we use Naïve Bayes algorithm to train the text classifier for Naïve Bayes is an effective text classifier.

A. Techniques of Step 1

1) Finding Reliable Negtive Documents with Rocchio Algorithm

Firstly, the entire unlabeled set *U* is treated as negative documents and then the set *P* and *U* are used to train the Rocchio classifier. Finally Rocchio classifier is used to classify the unlabeled set *U* and those documents classified as negative are initial reliable negative data, denoted by *N*. The algorithm is described as follows[3]:

a) The positive set *P* is assigned the positive class, and the unlabeled set *U* is the negative class.

b) The positive prototype vector is calculated as:

$$\vec{c}_+ = \alpha \frac{1}{|P|} \sum_{\vec{d} \in P} \frac{\vec{d}}{\left\|\vec{d}\right\|} - \beta \frac{1}{|U|} \sum_{\vec{d} \in U} \frac{\vec{d}}{\left\|\vec{d}\right\|} . \quad (1)$$

The negative prototype vector is calculated as:

$$\vec{c}_- = \alpha \frac{1}{|U|} \sum_{\vec{d} \in U} \frac{\vec{d}}{\left\|\vec{d}\right\|} - \beta \frac{1}{|P|} \sum_{\vec{d} \in P} \frac{\vec{d}}{\left\|\vec{d}\right\|} . \quad (2)$$

c) N = { };

d) for each $\vec{d}$ in *U* do

e) if $sim(\vec{d}, \vec{c}_+) \le sim(\vec{d}, \vec{c}_-)$ then

f) $N = N \cup \{\vec{d}\}$;

Here, $|P|$ is the total number of the positive documents, and $|U|$ is the total number of the unlabeled documents. Each document is represented as a vector $\vec{d}$, $\vec{d} = (d_1, d_2, ..., d_n)$. Each element d_i in $\vec{d}$ represents a word t_i and can be calculated as the combination of *term frequency (tf)* and *inverse document frequency (idf)*, i.e., $d_i = tf(\vec{d}, t_i) * idf(t_i)$. In this paper, we adopt the formula of *tf* and *idf* used in the Cornell SMART system as follows[8]:

$$tf(\vec{d}, t_i) = \begin{cases} 0, & if \quad freq(\vec{d}, t_i) = 0 \\ 1 + \log(1 + \log(freq(\vec{d}, t_i))), & others \end{cases} . \quad (3)$$

Where $freq(\vec{d}, t_i)$ is the number of times that word t_i occurs in document $\vec{d}$.

$$idf(t_i) = \log \frac{1 + |D|}{|df(t_i)|}, \quad (4)$$

Here $|D|$ is the total number of documents and $df(t_i)$ is the number of the documents where word t_i occurs at least once.

Rocchio classifier is constructed by positive and negative prototype vectors $\vec{c}_+$ and $\vec{c}_-$. α and β parameters are factors that adjust the relative impact of positive and negative training examples. Setting α = 16, β = 4 is recommended in paper[9]. Rocchio classifier uses the cosine measure to compute the similarity of each $\vec{d}$ with each prototype vector, i.e., $sim(\vec{d}, \vec{c}_+)$ and $sim(\vec{d}, \vec{c}_-)$. The documents classified as negative class form the initial negative set *N*.

Although Rocchio classifier is a good text classifier and can extract initial negative documents from *U*, there are still some positive documents or more in the initial negative set *N* for the weakness of directly treating the unlabeled set *U* as negative documents in training Rocchio classifier. So the negative set *N* is still not pure especially when *P* is very small (≤5%). Considering the high diversity of the initial negative set *N*, Li et al[3] proposed the approach using clustering to partition the negative set *N* into many clusters and then combining the Rocchio algorithm to extract more pure negative documents. But Li et al[3] just want to purify the initial negative set *N* and the purifying performance is not very significant when *P* is very small (≤5%). In our paper, we also use the Rocchio with clustering method, but we give a better purifying method and moreover try to increase the size of *P* by extracting some most reliable positive documents from *U*.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on September 01,2021 at 09:27:12 UTC from IEEE Xplore. Restrictions apply.

2) Purifying the Negative Set N and Expanding the Small Positive Set P with Rocchio and Clustering

We use Q to denote the documents set which is the rest part of U after extracting N from U, i.e., $Q = U - N$, and we can derive that most documents in Q are possible positive documents.

Firstly, in step 1 of the algorithm, we still perform the Rocchio algorithm to extract the initial negative set N as shown in last section. Thus the unlabeled set U is partitioned into the negative set N and the other set Q as discussed above. Then in step 2, using the K-means method the negative set N is clustered into k clusters, $N_1, N_2, ..., N_k$. In step 3, as in Li et al[3] we produce a positive prototype vector $\vec{p}_i$ and a negative prototype vector $\vec{n}_i$ for the positive set P and each cluster N_i, $i \in \{1,2,...,k\}$. In step 4, we introduce two extraction methods of purifying the negative set N. Method 1 is the approach proposed by Li et al[3], and it holds that for each document $\vec{d}$ in N the $\vec{d}$ can be put into the final negative set RN if the maximum similarity between $\vec{d}$ and all the positive prototype vector $\vec{p}_i$ is less than the similarity between $\vec{d}$ and any negative prototype vector $\vec{n}_i$. However, Method 1 is conservative in removing the likely positive documents from N, so the purifying performance is not significant when P is very small. Method 2 is another extraction approach proposed by us, and we put the document $\vec{d}$ into the final negative set RN if for each $\vec{d}$ in N the average similarity between $\vec{d}$ and all the positive prototype vectors $\vec{p}_i$ is less than that between $\vec{d}$ and all the negative prototype vectors $\vec{n}_i$. Our extraction method is less conservation than Method 1 and experimental results show that when P is very small our extraction method can perform better. In step 5, we want to increase the size of P by extracting some most reliable positive documents from the documents set Q ($Q = U - N$). For each $\vec{d}$ in Q, if the minimum similarity between $\vec{d}$ and all the positive prototype vectors $\vec{p}_i$ is more than the maximum similarity between $\vec{d}$ and all the negative prototype vectors $\vec{n}_i$, we think that this document $\vec{d}$ can be the initial reliable positive document and the set of initial reliable positive documents is denoted by RP. But in order not to bring some noisy (negative) data into P, we only extract $a\%$ most reliable positive documents from the set Q and these documents form the final reliable positive documents set LP. (In our experiments, we are conservative to set $a\% = 10\%$.)

The detailed algorithm is shown as follows:

Step 1: Perform the Rocchio algorithm in last section and generate the initial negative set N, and the other set Q ($Q = U - N$);

Step 2: Perform K-means clustering of the initial negative set N to produce k clusters, $N_1, N_2, ..., N_k$.

Step 3: for $i = 1$ to k do

$$\vec{n}_i = \alpha \frac{1}{|N_i|} \sum_{\vec{d} \in N_i} \frac{\vec{d}}{\|\vec{d}\|} - \beta \frac{1}{|P|} \sum_{\vec{d} \in P} \frac{\vec{d}}{\|\vec{d}\|}, \quad (5)$$

$$\vec{p}_i = \alpha \frac{1}{|P|} \sum_{\vec{d} \in P} \frac{\vec{d}}{\|\vec{d}\|} - \beta \frac{1}{|N_i|} \sum_{\vec{d} \in N_i} \frac{\vec{d}}{\|\vec{d}\|}. \quad (6)$$

Step 4: Purifying the negative set N, (two methods):

(Method 1:)

a) $RN = \{\}$;

b) for each $\vec{d}$ in N do

if there exists a $\vec{n}_i$, $i \in \{1,2,...,k\}$, *s.t.*

$\max sim(\vec{d}, \vec{p}_i) \le sim(\vec{d}, \vec{n}_i)$ then

$RN = RN \cup \{\vec{d}\}$;

(Method 2:)

a) $RN = \{\}$;

b) for each $\vec{d}$ in N do

if $\frac{1}{k} \sum_{i \in \{1,2,...,k\}} sim(\vec{d}, \vec{p}_i) \le \frac{1}{k} \sum_{i \in \{1,2,...,k\}} sim(\vec{d}, \vec{n}_i)$

then $RN = RN \cup \{\vec{d}\}$;

Step 5: Expanding the small positive set P:

a) $RP = \{\}$;

b) for each $\vec{d}$ in Q do

if $i \in \{1,2,...,k\}$, *s.t.*

$\min sim(\vec{d}, \vec{p}_i) > \max sim(\vec{d}, \vec{n}_i)$ then

$RP = RP \cup \{\vec{d}\}$;

c) Rank the set RP according to the similarity between each $\vec{d}$ in RP and center vector $\vec{p}$ of the original positive set P. Then choose the $a\%$ nearest $\vec{d}$ in RP to $\vec{p}$, and these documents form the final reliable positive set LP.

d) At last, we can add the set LP into the original positive set P, i.e., $P = P \cup LP$.

Output: the final negative set RN and the final positive set P.

Then we can build a text classifier using the final negative set RN and positive set P as the training data.

B. Technique for Step 2

1) Building Text Classifier

For text categorization, Naïve Bayes is an effective classifier. So in our experiment, we adopt Naïve Bayes to build text classifier. Given the training data, each document is considered as an ordered list of words. And all words

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on September 01,2021 at 09:27:12 UTC from IEEE Xplore. Restrictions apply.

considered for classification form the vocabulary $V = \{t_1, t_2, ..., t_{|V|}\}$. Then in our paper, we just consider two classes, i.e., $C = \{C_1, C_2\}$. C_1 is the positive class and C_2 is negative.

Firstly, we need to compute the probability of each word given a class, i.e., $\Pr(t_k|C_j)$ and the probability of a class, i.e., $\Pr(C_j)$. The formula of $\Pr(t_k|C_j)$ is as follows:

$$\Pr(t_k \mid C_j) = \frac{1 + \sum_{i=1}^{|D|} N(t_k, \vec{d}_i) \Pr(C_j \mid \vec{d}_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N(t_s, \vec{d}_i) \Pr(C_j \mid \vec{d}_i)}. \quad (7)$$

Here $|D|$ is the total number of documents in training set. $N(t_k, \vec{d}_i)$ is the times the word t_k occurs in the document $\vec{d}_i$. If the class of the document $\vec{d}_i$ is C_j, then $\Pr(C_j|\vec{d}_i) = 1$; if not then $\Pr(C_j|\vec{d}_i) = 0$. $|V|$ is the total number of the all words in vocabulary V and this formula adopts smoothing strategy for preventing zero probabilities for infrequently occurring words.

The computation of $\Pr(C_j)$ is as follows:

$$\Pr(C_j) = \frac{1 + \sum_{i=1}^{|D|} \Pr(C_j \mid \vec{d}_i)}{|C| + |D|}, \quad (8)$$

Where $|C|$ is the total number of classes, in our paper this value is 2.

Then based on the Bayesian probability and multinomial model, for one test document $\vec{d}$, we compute the posterior probability for each class C_j, $\Pr(C_j|\vec{d})$. The formula is as follows:

$$\Pr(C_j \mid d) = \frac{\Pr(C_j) \prod_{k=1}^{|V|} \Pr(t_k \mid C_j)}{\sum_{r=1}^{|C|} \Pr(C_r) \prod_{k=1}^{|V|} \Pr(t_k \mid C_r)}. \quad (9)$$

Finally, the class with the highest posterior probability is assigned to the final class of the test document $\vec{d}$. In our paper, there are only two classes: positive and negative, so one test document will be classified as either positive or negative class.

V. Experiment and Evaluation

A. The Performance Measure

In our experiment, we use the popular *F1* measure. *F1* measure takes into account both recall and precision to evaluate a text classification system. They are defined as follows[10]:

$$\mathrm{Re}\,call = \frac{number\ of\ correct\ positive\ predictions}{number\ of\ positive\ examples}, \quad (10)$$

$$\mathrm{Pr}\,ecision = \frac{number\ of\ correct\ positive\ predictions}{number\ of\ positive\ predictions}, \quad (11)$$

$$F1 = \frac{2 \times \mathrm{Pr}\,ecision \times \mathrm{Re}\,call}{\mathrm{Pr}\,ecision + \mathrm{Re}\,call}. \quad (12)$$

For evaluating performance average across categories, there are two forms of *F1* average measure: Macro-average and Micro-average of *F1* measure.

Macro-F1 = average of within-category *F1* values. (13)

Micro-F1 = *F1* over categories and documents. (14)

Then in our experiment, we use *Macro-F1* as the performance measure.

B. Experimental Data Set

In our experiment, we use TanCorp-12 dataset of TanCorp corpus which is a popular data set for Chinese text classification experiment[10][11]. The dataset contains 12 categories and the total documents amount to 14150. Then in our experiment, each category is employed as the positive class, and the rest as the negative class. This gives us 12 datasets.

In our experiment, for each dataset we randomly select 30% of the documents as test set. The rest (70%) are used to produce training sets as follows: *b%* of the documents from the positive class is selected as the positive set *P*, but *b%* of the documents from negative class is not used; and the remaining (1-*b%*) documents from both positive and negative class are used for the unlabeled set *U*. For we just focus on the situation when the positive set *P* is very small, we range *b%* from 1%~5% to test our method.

C. Experimental Setup and Results

In our experiment, we use the Correlation Coefficient method for feature selection in the data preprocessing and we select 3000 words as the final feature items for text classification.

For the value *k* of K-means clustering, experimental results in Li et al[3] have shown that if *k* is not very small the influence of the value *k* in our experiments would not be significant. So in our experiments, we set *k* = 11 or 12.

Then in order to comparison we implemented 6 different algorithms as follows:

A1: In this algorithm, we just treat the unlabeled set *U* as the negative set *N* and with the positive set *P* as the training data a Naïve Bayes text classifier is built. The motivation in doing this experiment we just want to prove that whether directly treating *U* as *N* would build a bad text classifier.

A2: Firstly we use the Rocchio algorithm to extract the initial reliable negative documents set *N* from *U*. Then we employ the positive set *P* and the initial reliable negative set *N* as the training set to produce a Naïve Bayes text classifier.

A3: Here we just adopt the technique of purifying the initial negative set *N*, but don't expand the original positive set *P*. And the final reliable negative set *RN* is output. The final reliable negative set *RN* and the original positive set *P* are used to train the Naïve Bayes text classifier. According to the two different extraction methods of purifying the initial negative set *N*, this algorithm has two different experiments as follows:

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on September 01,2021 at 09:27:12 UTC from IEEE Xplore. Restrictions apply.

A3-1: Employ Method 1 for extraction method.

A3-2: Adopt Method 2 for extraction method.

A4: Here we not only purify the initial negative set *N* but also expand the original positive set *P*. At last, we get the final negative set *RN* and the new positive set *P* which is made up of the original positive set *P* and the finial reliable positive set *LP*. Then the final negative set *RN* and the new positive set *P* are employed to build a Naïve Bayes text classifier. For the two different extraction methods of purifying the initial negative set *N*, this algorithm also has two corresponding experiments as follows:

A4-1: Use Method 1 for extraction method and expand the original positive set *P*.

A4-2: Adopt Method 2 for extraction method and expand the original positive set *P*.

Figure 1 shows the experimental results as follows:

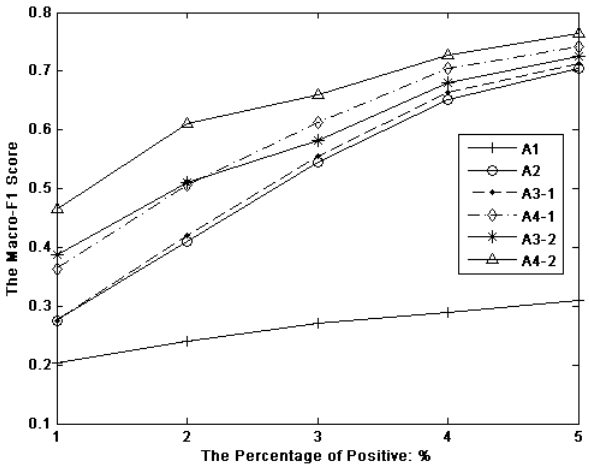

Figure 1. The Macro-F1 scores for all b% settings

From Figure 1, we can derive that A1 has the worst performance because it just treats the unlabeled set *U* as the negative set *N* and the unlabeled set *U* may include many positive documents. When *P* is very small, treating *U* as *N* produces many noisy examples in building text classifier.

While A2 which uses the Rocchio algorithm to extract the initial negative set *N* from *U* performs better than A1.

Then for A3 which adopts the Rocchio with clustering technique to purify the initial negative set *N*, we can find that A3-2 performs better than A3-1. For A3-2 using Method 2 for extraction method can get more pure negative documents than A3-1 with Method 1 when *P* is very small.

For A4 which not only purifies the initial negative set *N* but also expands the size of *P*, from Figure 1 we can find that A4 performs better than the corresponding A3. So we believe that increasing the size of *P* can improve the performance further when *P* is very small in PU-Learning.

In all, A3-2, A4-1, A4-2 which use the technique proposed by us perform better in PU-Learning when the positive set *P* is very small.

VI. CONCLUSION

For PU-Learning, we focus on the situation when the positive set *P* is very small (≤5%). As the two-step algorithm, in Step 1, we propose a more effective extraction method of purifying the initial negative set *N* and expanding the original small positive set *P*. Then in step 2, we adopt Naïve Bayes which is a popular text classifier. The experimental results show that the technique proposed by us have better performance in PU-Learning when *P* is very small.

However, in our experiments when we extract *a%* most reliable positive documents from the set *Q*, in order not to introduce more noisy data, we are conservative to set *a%* = 10%. Thus in further study, we want to find out the influence of parameter *a%* in positive documents expansion.

ACKNOWLEDGMENT

The work was supported by a grant from the State Key Laboratory of Computer Science of the Institute of Software of Chinese Academy Sciences and by a grant from Scientific Research Foundation for Returned Scholars, Ministry of Education of China. We would like to thank the anonymous reviewers for their comments and suggestions.

REFERENCES

[1] B. Liu, W. S. Lee, P. S. Yu, and X.L. Li, "Partially supervised classification of text documents," Proceedings of 19th International Conference on Machine Learning, 2002, pp.387-394.

[2] H. Yu, J. Han, and K. C.-C. Chang, "PEBL: positive example based learning for web page classification using SVM," Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, 2002, pp.239-248.

[3] X.L. Li and B. Liu, "Learning to classify texts using positive and unlabeled data," Proceedings of 18th International Joint Conference on Artificial Intelligence, 2003, pp.587-592.

[4] B. Liu, Y. Dai, X.L. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," Proceedings of the Third IEEE International Conference on Data Mining, 2003, pp.179.

[5] S. Yu, X. Zhou, and C. Li, "Semi-supervised text classification using positive and unlabeled data," Proceedings of the 2006 conference on Advances in Intelligent IT, 2006, pp.249-254.

[6] X. L. Li, B. Liu, and S.-K. Ng, "Learning to classify documents with only a small positive training set," Proceedings of the 18th European conference on Machine Learning, 2007, pp.201-213.

[7] P. Garg and S. Sundararajan, "Active learning in partially supervised classification," Proceedings of the 18th ACM conference on Information and knowledge management, 2009, pp.1783-1786.

[8] J. Han and M. Kamber, Data Ming:Concepts and Techniques, 2nd ed. Beijing: China Machine Press, 2007, pp.401-407.

[9] J. J. Rocchio, "Relevance feedback in information retrieval," in The Smart retrieval system experiments in automatic document processing, G. Salton. Englewood, Cliffs, New Jersey: Prentice Hall, 1971, pp.313-323.

[10] S. Tan, X. Cheng, M. M. Ghanem, B. Wang, and H. Xu, "A Novel Refinement Approach for Text Categorization," Proceedings of the 14th ACM international conference on Information and knowledge management, 2005, pp.469-476.

[11] S. Tan and Y. Wang, Chinese corpus of text categorization-TanCorpV1.0, http://www.searchforum.org.cn/tansongbo/corpus.htm.

Authorized licensed use limited to: Huazhong University of Science and Technology. Downloaded on September 01,2021 at 09:27:12 UTC from IEEE Xplore. Restrictions apply.