Partially Supervised Text Classification with Multi-Level Examples

Tao Liu, Xiaoyong Du
Key Laboratory of Data Engineering and Knowledge Engineering (Renmin University of China), MOE, School of Information, Renmin University of China, 100872, Beijing, China
{tliuruc, duyongruc}@google.com

Yongdong Xu
School of Computer Science and Technology, Harbin Institute of Technology (Weihai), 264209, Weihai, China
ydxu@insun.hit.edu.cn

Minghui Li
Microsoft Asian Research and Development Group, 100080, Beijing, China,
mhuil@microsoft.com

Xiaolong Wang
School of Computer Science and Technology, Harbin Institute of Technology, 150001, Harbin, China
wangxl@insun.hit.edu.cn

Abstract

Partially supervised text classification has received great research attention since it only uses positive and unlabeled examples as training data. This problem can be solved by automatically labeling some negative (and more positive) examples from unlabeled examples before training a text classifier. But it is difficult to guarantee both high quality and quantity of the new labeled examples. In this paper, a multi-level example based learning method for partially supervised text classification is proposed, which can make full use of all unlabeled examples. A heuristic method is proposed to assign possible labels to unlabeled examples and partition them into multiple levels according to their labeling confidence. A text classifier is trained on these multi-level examples using weighted support vector machines. Experiments show that the multi-level example based learning method is effective for partially supervised text classification, and outperforms the existing popular methods such as Biased-SVM, ROC-SVM, S-EM and WL.

1. Introduction

With an increasing number of documents on the web, it is very important to build a text classifier which can identify a class of documents a particular user prefers. For example, it is desirable to have a system which can recommend news for a particular user based on his previous reading preference. This task is, given some seed documents of a certain class, building a text classifier which identifies more documents of the given class from new data. We name the class of documents a particular user concerned as positive examples (*P*), and the documents which do not fit the user interest as negative examples (*N*). For a news reader, *P* can be obtained from user reading history, but *N* is unknown. Additional resources which can be used to build the classifier are a large number of unlabeled examples (*U*), such as all kinds of news from the internet. This problem is learning from positive and unlabeled examples, and it is also called partially supervised text classification (TC) (Liu et al. 2002).

Partially supervised TC is an extension of semi-supervised TC (Nigam, McCallum and Thrun 1998; Nigam et al. 2000). As we know, supervised TC (Sebastiani 2002) needs a large number of manually labeled positive and negative examples to build a classifier. Semi-supervised TC makes use of unlabeled data to alleviate the intensive effort of manually labeling. Compared with semi-supervised TC, no pre-given negative training examples is required for partially supervised TC. In this paper, we concentrate on this problem because of its great importance.

Most current methods such as S-EM (Liu et al. 2002), ROC-SVM (Li and Liu 2003), PEBL (Yu, Han, and Chang 2004), CR_SVM (Li and Liu 2010) solve the problem in two steps: 1) automatically label some examples from unlabeled data to enlarge the original training set, 2) train text classifiers using original positive examples and newly labeled examples. All training examples are equally used to build the classifier in these methods, i.e., treat the pre-given positive examples and other examples obtained in the first step equally. In fact, the confidence of these two types of examples is different. The pre-given positive examples can be considered as golden data because they are labeled manually, while automatically acquired training examples have lower confidence because there are inevitable labeling errors in them. Biased-SVM (Liu et al.

2003) solves the problem by minimizing the number of unlabeled examples classified as positives and constraining golden positive examples to be correctly classified. Biased-SVM is a one-step method, which does not select additional training examples. Its performance is not very good when the given positive example set is small.

A novel multi-level example based learning method is therefore proposed in this paper for partially supervised TC. A heuristic method is firstly used to generate multi-level examples according to their confidence. Both the quality and quantity of training examples are important for the training of a high quality classifier. It is difficult to guarantee both high precision and recall for labeling new training examples, so our multi-level example generation method needs a trade-off between precision and recall by partitioning training examples according to their confidence into multiple levels. Secondly weighted support vector machine is used to discriminatingly treat multi-level training examples. Experimental results indicate that the proposed method outperforms traditional methods.

2. Method

The purpose of partially supervised TC (Liu et al. 2002) is to find function *f* which maps *X* to *Y*, where *X* is documents set, and *Y* is labels set which is {-1, +1}. The training examples include a small number of positive examples (*P*), and a large number of unlabeled examples (*U*). Unlabeled examples are mixed with other positive examples and negative examples (*N*). The assumption is: examples in *P* are randomly selected from all positive examples. i.e. the feature distribution of positive examples in *P* is the same as that of positive examples in *U*.

A multi-level example based learning (MLEL) method is proposed for partially supervised TC. A heuristic method is firstly used to label additional training examples from *U*. Multi-level training examples are generated including golden positives (*GP*), potential positives (*PP*), strong negatives (*SN*), reliable negatives (*RN*) and potential negatives (*PN*). A learning method based on weighted support vector machine (WSVM) is used to train the text classifier on these multi-level examples. Algorithm 1 shows the general framework of the MLEL method.

Algorithm 1: MLEL (*P*, *U*)

Input: positive documents *P*, unlabeled documents *U*

Output: a text classifier

Obtain positive feature set (*PF*) and word positive degree (PD_{word}) for each feature using Positive Feature Selection algorithm.

Use Multi-level Example Generation algorithm to obtain *GP*, *PP*, *SN*, *RN* and *PN*.

Train text classifier using WSVM.

Before probable positives and negatives are selected from *U*, some feature words which can differentiate positive and negative examples are identified (Blum and Langley 1997). Since there are no pre-given *N*, beside *P* and *U*, it is better to identify positive feature words which can reflect and represent the characteristic of *P*. Positive Degree (*PD*) is used to judge if an unlabeled document is a positive example. The document positive degree (PD_{doc}) is defined using the positive feature set in section 2.1.

2.1 Positive Feature Selection

Positive features are words which can reflect and represent the characteristics of positive examples and distinguish it from that of negative examples. Two statistical criteria named *Specialty* and *Popularity* are used to judge whether a word is a positive feature or not. These two criteria make use of the statistical information of words among *P* and *U* to identify positive features.

The Specialty criterion depicts a positive feature specially used in *P*. As for word occurrence frequency, a word tends to be a positive feature if it occurs more frequently in *P* than in *U*. For word *w*, its *Specialty* is as:

$$Specialty(w) = f(w,P)/(f(w,P) + f(w,U)) \quad (1)$$

where $f(w, P)$ and $f(w, U)$ denote the frequency of word *w* occurring in *P* and *U* respectively.

The Popularity criterion depicts a positive feature popularly used in *P*. Supposing two words have the same occurrence frequency in *P*, the one which occurs in more positive examples is more likely a positive feature than another. The potential hypothesis is that the word with more uniform occurrence distribution in a certain domain is more likely to be a feature of this domain (Navigli and Velardi 2004). Information entropy is used to measure the distribution of word *w* in *P* as shown below:

$$Ent(w,P) = -\sum_{i=1}^{n_P} NProb(d_i|w)\log(NProb(d_i|w)) \quad (2)$$

where $NProb(d_i|w)$ denotes the normalized probability of word *w* occurring in document d_i, and n_P denotes the number of documents of *P*. By normalizing the above entropy into range [0, 1], popularity of word *w* can be expressed as:

$$Popularity(w) = Ent(w,P)/Z \quad (3)$$

where *Z* is the normalization factor, which is the maximal value of $Ent(w,P)$, i.e. $log(n_p)$.

Normalized probability $NProb(d_i|w)$ is used to take into account the influence of different document lengths on the word occurrence probability, and expressed as:

$$NProb(d_i|w) = \frac{Prob(d_i|w)/l_i}{\sum_{j=1}^{n_P}(Prob(d_j|w)/l_j)} \quad (4)$$

where $Prob(d_i|w) = f(w,d_i)/f(w,P)$, $l_i = \sum_{w\in d_i} f(w,d_i)$.

PD_{word} is defined on the basis of *Specialty* and *Popularity* as:

$$PD_{word}(w) = Specialty(w) + Popularity(w) \quad (5)$$

Both *Specialty* and *Popularity* are important for selecting high quality positive features. We take word w as a positive feature if it satisfies $Specialty(w) > \alpha$, $Specialty(w) > \beta$ and $PD_{word}(w) > \gamma$. α, β and γ are thresholds for *Popularity*, *Specialty* and PD_{word} respectively. They are determined as the average values of *Popularity*, *Specialty* and PD_{word} according to the experiments.

2.2 Multi-Level Example Generation

Both the quality and quantity of training examples are important for the training of a high quality classifier, but it is difficult to obtain both high precision and recall for newly labeled training examples. Here, our method is used to select as many new training examples as possible from unlabeled examples to make the best use of them. Multi-level examples are generated based on document positive degree which reflects the example labeling confidence. Then, a text classifier is discriminatively trained using these multi-level examples.

2.2.1 Document Positive Degree

For each unlabeled example, we use the document positive degree (PD_{doc}) to describe its possibility of being a positive example. The positive degree of document d_i is computed based on the positive features selected using the following formula:

$$PD_{doc}(d_i) = (\sum_{w \in PF, w \in d_i} PD_{word}(w))/\log(l_i) \quad (6)$$

where $\log(l_i)$ is the normalization factor and $l_i = \sum_{w \in d_i} f(w, d_i)$. The positive degree of a document increases with the possibility of the document as a positive example.

2.2.2 Multi-Level Positives Acquisition

In the problem of partially supervised TC, the set of pre-given positive training examples with highest labeling confidence is a first-level positive named as golden positives (*GP*).

It can be seen from formula (6) that the possibility of an unlabeled document to be a positive example is determined by the number of positive features it contains and the positive degree of all included positive features. $PD_{doc}(d_i)$ increases with the probability of d_i as a positive example. Supposing $\overline{PD(P)}$ and $\overline{PD(U)}$ denote the average document positive degree on examples of P and U respectively. Document d_x in U which satisfies the formula (7) is taken as second-level positives, named as potential positives (*PP*). It is very difficult to extract positive examples from U, so we will not extract positive examples besides potential positives.

$$PD_{doc}(d_x) > \overline{PD(P)} \quad (7)$$

2.2.3 Multi-Level Negatives Acquisition

Document d_x whose $PD_{doc}(d_x)$ equals zero is taken as a first level negative example, i.e. those unlabeled examples which do not contain any positive features are taken as negative examples. Since it is normal for a negative example to contain a small number of positive features, this labeling criterion is rigorous and yields high confident negatives called strong negatives (*SN*).

To select the second level negative examples with much looser criterion than strong negatives acquisition, we use average positive degree of all unlabeled documents as the empirical threshold. Document d_x is taken as a medium confident negative example if it satisfies the formula (8). This set of negatives is called reliable negatives (*RN*).

$$0 < PD_{doc}(d_x) \leq \overline{PD(U)} \quad (8)$$

Remaining unlabeled examples after selecting *PP*, *SN* and *RN* are taken as the third level negative examples with low confidence and called potential negatives. We discover through experiments these remaining unlabeled examples are also useful for training classifier.

2.3 Multi-Level Example Based Learning

Weighted Support Vector Machine (Vapnik 1995) is used to train the classifier on multi-level examples, and assign different weights to the examples with different confidence. The optimizing goal is:

$$\begin{aligned} &\text{minimize: } \frac{1}{2}\|w\|^2 + c'_+ \sum_{i \in GP} \xi_i + c''_+ \sum_{i \in PP} \xi_i + \\ &c'_- \sum_{i \in SN} \xi_i + c''_- \sum_{i \in RN} \xi_i + c'''_- \sum_{i \in PN} \xi_i \\ &\text{subject to: } y_i(w^T x_i + b) \geq 1 - \xi_i (i = 1,2, \ldots, n) \end{aligned} \quad (9)$$

where ξ_i is a slack variable which allows the misclassification of some training examples, c'_+, c''_+, c'_-, c''_- and c'''_- represent the penalty factors of misclassification for *GP*, *PP*, *SN*, *RN* and *PN* example sets respectively. Each of these different parameters gets its own value. Different parameters c'_+ and c''_+ are used for *GP* and *PP* because the confidence of *GP* and *PP* is different, and the same condition holds for c'_-, c''_- and c'''_-. Different parameters c'_+ and c'_-, are chosen for *GP* and *SN* not only because their confidence is different, but also because the dataset in the problem of partially supervised TC is always unbalanced. The total number of positives is far less than that of negatives among the unlabeled example set.

3. Related Work

A number of methods have been proposed for this problem. The main difference for these methods is how to use unlabeled examples, and there are five types: 1) Select possible negative examples from U as N, and then build

classifiers using P and N (Liu et al. 2002; Li and Liu 2003; Yu, Han, and Chang 2004; Li and Liu 2010); 2) Select possible positive examples P' and negative examples N from U, and then build classifiers using $P \cup P'$ and N (Fung et al. 2006; Li, Liu, and Ng 2007); 3) Treat all examples of U as possible negative examples N, and take the problem as learning with noise, i.e. assign different class weights (Lee and Liu 2003; Liu et al. 2003); 4) Take each example of U as both possible positive and negative example with certain probability, and train the classifiers with these examples (Elkan and Noto 2008); 5) Discard U and only use P to build classifier (Manevitz and Yousef 2001).

The first two methods are very similar. The popular used techniques for extracting N or P' include spy (Liu et al. 2002), Rocchio (Li and Liu 2003), 1-DNF (Yu, Han, and Chang 2004) and PNLH (Fung et al. 2006). Positive feature selection is not required for Spy and Rocchio, and it is required for 1-DNF and PNLH. After extracting N or P', standard machine learning methods such as Naive Bayes and SVM are used to train classifiers.

The third type of method includes Biased-SVM (Liu et al. 2003) and WL (Lee and Liu 2003). Biased-SVM assigns different class weights for the positive and negative class of SVM classifier, which is the most related work with ours. WL uses Logistic Regression after weighting the negative class. Our method differs from them because we generate multi-level examples, and we do not simply take all unlabeled examples as negatives.

The fourth type of method (Elkan and Noto 2008) is used for protein record identification. It takes each unlabeled example as both positive and negative example with weights pre-computed by an additional classifier trained on P and U. In our method, each example has only one label, and parameters are selected on the validation set.

The fifth type of method such as one-class SVM (Manevitz and Yousef 2001) estimate the distribution of positive examples without using unlabeled examples. This method is sensitive to the input representation.

There are also some other methods (Denis, Gilleron, and Letouzey 2005) which need information about the ratio of positives in U to solve the problem.

4. Experiment

4.1 Experimental Setup

4.1.1 Datasets and Preprocessing

Newsgroup [1] and Reuters [2] corpus are used to construct datasets as detailed below. For Newsgroup corpus, 80 percent examples are randomly selected from each category as training data, and 20 percent examples are taken as testing data. For Reuters corpus, the top ten popular categories are used. Modified-Apte split method of this corpus is used for creating training and testing dataset.

Suppose each corpus include n categories, taking examples of each category as positives by turns, and taking examples of other corresponding n-1 categories as negatives, n datasets are obtained in this way. For training data of each dataset, randomly select $100\times k$ percent positives to form positive set P, and blend other $100\times(1-k)$ percent positives with negatives to form unlabeled set U. Different k (0.1, 0.2, …, 0.9) is chosen to create different scenario. For each training dataset, 30 percent of examples are taken as the validation set.

Stop words are filtered in the data preprocessing. Each document is represented as a vector of TFIDF value of all occurred words except stop words.

LIBSVM [3] package is used for the implementation of SVM for both MLEL and Biased-SVM, and the popularly used linear function is chosen as kernel. LPU package [4] is used for the implementation of S-EM, ROC-SVM and WL.

Penalty factors of MLEL are optimized on validation sets. The range of values for c is from the set: $\{2^{-7}, 2^{-6}, \ldots, 2^{5}\}$ and final used values are auto-selected.

4.1.2 Evaluation Criteria

F score on positive class is used to evaluate the performance of partially TC on the testing set. F score is computed by precision (p) and recall (r) as: $F=2pr/(p+r)$.

F score cannot be computed on the validation dataset during the training process because there is no golden negative example. An approximate computing method (Lee and Liu 2003) is used to evaluate the performance by $\text{pseudoF}=r_P^2/Prob(f(X)=1)$, where X is the random variable representing the input vector, $Prob(f(X)=1)$ is the probability of an input example classified as positive, r_P is the recall for positive set P in the validation set.

4.2 Positive Feature Selection Method Comparison

Traditional feature selection methods (Yang and Pedersen 1997) for supervised TC cannot be directly used in the positive feature selection of this problem. Though we can use them by taking unlabeled example as negative, the results are not satisfying. Here we only compare our positive feature selection technique of MLEL with that of 1-DNF (Yu, Han, and Chang 2004) and PNLH (Fung et al. 2006) which have been used in the same problem.

By substituting the positive feature selection technique of MLEL with that of 1-DNF and PNLH, we get system A and B. As shown in Table 1, the system with our feature selection method (MLEL) obtains best average F score than the system with other two feature selection methods.

[1] http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html
[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[3] LIBSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm
[4] http://www.cs.uic.edu/~liub/LPU/LPU-download.html

Table 1: Average F Score Obtained by Different Positive Feature Selection Methods

Corpus	Sys A	Sys B	MLEL
Newsgroup	0.862	0.851	0.886
Reuters	0.789	0.787	0.823

4.3 Effect of Weighting Multi-Level Examples

4.3.1. Labeling Precision of Multi-Level Examples

This experiment is made to show that different levels of examples have different labeling precision. We should point out the actual labels of unlabeled examples in *U* are used to compute labeling precision, but these actual labels of unlabeled examples are not used in training classifier. Each row of Table 2 shows the average labeling precision of all datasets generated by one corpus. Precision of *GP* is regarded as 1 since *GP* are obtained manually. It is very difficult to extract additional positive examples from *U*. The precision of *PP* is much lower than that of *GP*, therefore it is essential to assign different weights for *GP* and *PP* for training the classifier. The precision of *PN* equals to the ratio of real negatives in remaining unlabeled examples after heuristic labeling. It can be seen the labeling precision of multi-level negatives decreases with the example level increase.

Table 2: Average Labeling Precision of Multi-Level Examples

Corpus	GP	PP	SN	RN	PN
Newsgroup	1.0	0.79	0.98	0.96	0.83
Reuters	1.0	0.71	0.99	0.96	0.75

4.3.2 Comparison with Biased-SVM

In this section we compare the MLEL method with a baseline method Biased-SVM (Liu et al. 2003), which takes all the examples in *U* as negatives, and trains SVM using *P* and *U*. It uses the following formula as optimizing goal.

$$\text{Minimize: } \frac{1}{2}\|w\|^2 + c_+ \sum_{i\in P} \xi_i + c_- \sum_{i\in U} \xi_i \qquad (10)$$

As shown in Figure 1, MLEL outperforms Biased-SVM in most cases (*k* from 0.1 to 0.8) on both corpora. The improvement is much larger for smaller *k*. When *k* equals 0.9, Biased-SVM and MLEL obtains very similar performance (The difference of average F score is within 1 percent). Because the number of positives in *U* is very small when *k*=0.9, Biased-SVM obtains good performance by using all examples in *U* as negatives in this scenario.

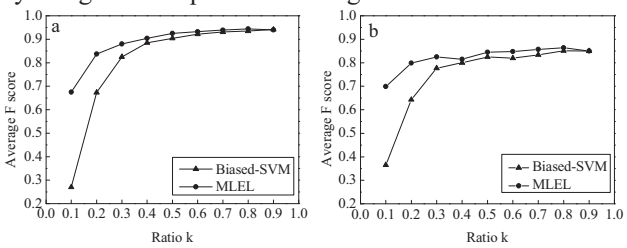

Figure 1: Average F Score Comparison between MLEL and Baseline System on (a) News-group and (b) Reuters Corpus

4.3.3 Effect of Weighted Potential Positives

The labeling confidence of potential positives (*PP*) is far less than that of golden positives. To show the effect of weighted *PP* for the classifier, comparisons are made among the three cases: use *PP* without independent weight; discard *PP*; use *PP* with independent weight. It can be seen from Figure 2: using *PP* with independent weight obtains the best performance on both corpora. Properly weighting *PP* is important to improve the system performance. Discarding *PP* leads to the worst average *F* score on Newsgroup corpus. The result is a little bit different on Reuters corpus: discarding *PP* obtains better average *F* score than using *PP* without independent weight for *k* from 0.3 to 0.9. But for very small *k* (0.1), discarding *PP* leads to poor average *F* score. Using *PP* with independent weight greatly enhances the average *F* score for *k*=0.1 on Reuters corpus.

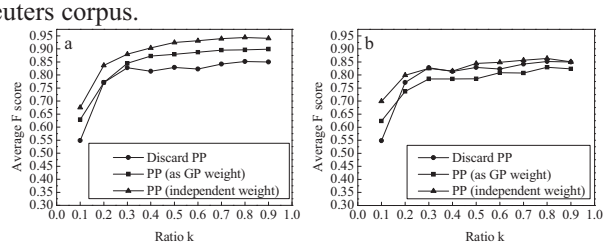

Figure 2: Effect of Using Independently Weighted PP for (a) Newsgroup and (b) Reuters Corpus

4.3.4 Effect of Weighted Potential Negatives

Potential negatives, which are examples remaining unlabeled, are useful to the training of classifier when assigned lower weights. It can be seen from Figure 3 for most *k* (from 0.2 to 0.9), "with *PN*" improves the system performance, and for very small k (0.1), "with *PN*" destroys the system performance since the current positive ratio in *PN* is comparatively high.

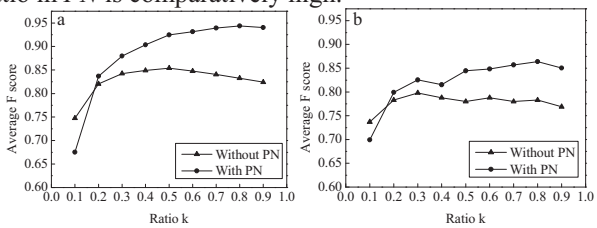

Figure 3: Effect of Potential Negatives for (a) Newsgroup and (b) Reuters Corpus

4.4 Comparison with Other Methods

MLEL is compared with other popular methods including S-EM (Liu et al. 2002), ROC-SVM (Li and Liu 2003) and WL (Lee and Liu 2003). It can be seen from Figure 4 that MLEL outperforms these methods in most *k* on Newsgroup and Reuters corpora. Averagely, MLEL outperforms S-EM, ROC-SVM and WL by 10, 4.5, and 4.4 percent respectively on the average *F* score of all *k* on Newsgroup corpus, and by 6.8, 2.0, and 3.7 percent respectively on the average *F* score of all *k* on Reuters corpus.

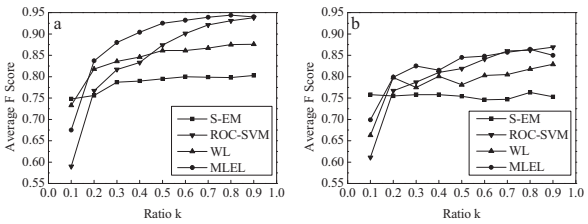

Figure 4: Average F Score Comparison between MLEL and Other Methods on (a) News-group and (b) Reuters Corpus

5. Conclusion

Both the quality and quantity of training examples are important for the performance of text classification. For partially supervised text classification with only positive examples and unlabeled examples, it is difficult to obtain negative and more positive examples with high quality and quantity from unlabeled examples. In order to take full use of the large scale unlabeled examples, we propose a new heuristic method to generate multi-level training examples according to their labeling confidence. Different weight is assigned to each level of examples to make fully use of all examples in a discriminative way. Experiments show that the multi-level examples based weighting method outperforms the traditional class-based weighting method on the performance of partially supervised text classification, especially in the scenario that the number of pre-given positive training examples is small. Furthermore, our proposed method obtains better performance than state-of-the-art methods, such as ROC-SVM, S-EM and WL, in most cases of positive ratio *k*.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China with grant No. 61003204, 60873017 and 60803092. The authors would like to thank all reviewers for their detailed evaluation and kind suggestions.

7. References

Blum A. L. and Langley P. 1997. Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence* 97(1):245-271.

Denis F., Gilleron R., and Letouzey F. 2005. Learning from Positive and Unlabeled Examples. *Theoretical Computer Science* 348(1):70-83.

Elkan C. and Noto K. 2008. Learning Classifiers from Only Positive and Unlabeled Data. In *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining*, 213-220. Las Vegas, Nevada, USA.

Fung G. P. C., Yu J. X., Lu H., and Yu P. S. 2006. Text Classification without Negative Examples Revisit. *IEEE Transactions on Knowledge and Data Engineering* 18(1):6-20.

Joachims T. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the 10th European Conference on Machine Learning*, 137-142. Chemnitz, Germany.

Lee W. S. and Liu B. 2003. Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression. In *Proceedings of the 20th International Conference on Machine Learning*, 448-455. Washington DC, United States.

Li X. and Liu B. 2003. Learning to Classify Text Using Positive and Unlabeled Data. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 587-594. Acapulco, Mexico.

Li X., Liu B., and Ng S. 2007. Learning to Classify Documents with Only a Small Positive Training Set. In *Proceedings of the 18th European Conference on Machine Learning*, 201-213. Warsaw, Poland.

Li X., Liu B., and Ng S. 2010. Negative Training Data can be Harmful to Text Classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 218-228. Massachusetts, USA.

Liu B., Dai Y., Li X., Lee W. S., and Yu P. S. 2003. Building Text Classifiers Using Positive and Unlabeled Examples. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, 179-188. Melbourne, Florida, United States.

Liu B., Lee W. S., Yu P. S., and Li X. 2002. Partially Supervised Classification of Text Documents. In *Proceedings of the 19th International Conference on Machine Learning*, 387-394. Sydney, Australia.

Manevitz L. M. and Yousef M. 2001. One-Class SVMs for Document Classification. *Journal of Machine Learning Research* 2:139-154.

Navigli R. and Velardi P. 2004. Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Computational Linguistics* 30(2):151-179.

Nigam K., McCallum A. K., and Thrun S. 1998. Learning to Classify Text from Labeled and Unlabeled Documents. In *Proceedings of the 15th National Conference on Artificial Intelligence*, 792-799. Madison , Wisconsin, United States: AAAI Press.

Nigam K., McCallum A. K., Thrun S., and Mitchell T. 2000. Text Classification from Labeled and Unlabeled Documents Using EM. *Machine Learning* 39(2/3):103-134.

Sebastiani F. 2002. Machine Learning in Automated Text Categorization. *ACM Computer Surveys* 34(1):1-47.

Vapnik V. N. eds. 1995. *The Nature of Statistical Learning Theory*. New York: Springer-Verlag.

Yang Y. and Pedersen J. O. 1997. A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the 14th International Conference on Machine Learning*, 412-420. Nashville, Tennessee, United States.

Yu H., Han J., and Chang K. C. C. 2004. PEBL: Web Page Classification without Negative Examples. *IEEE Transactions on Knowledge and Data Engineering* 16(1):70-81.