Protein-Protein Interactions Classification from Text via Local Learning with Class Priors

Yulan He and Chenghua Lin

School of Engineering, Computing and Mathematics
University of Exeter, North Park Road, Exeter EX4 4QF
{y.he,cl322}@exeter.ac.uk

Abstract. Text classification is essential for narrowing down the number of documents relevant to a particular topic for further pursual, especially when searching through large biomedical databases. Protein-protein interactions are an example of such a topic with databases being devoted specifically to them. This paper proposed a semi-supervised learning algorithm via local learning with class priors (LL-CP) for biomedical text classification where unlabeled data points are classified in a vector space based on their proximity to labeled nodes. The algorithm has been evaluated on a corpus of biomedical documents to identify abstracts containing information about protein-protein interactions with promising results. Experimental results show that LL-CP outperforms the traditional semi-supervised learning algorithms such as SVM and it also performs better than local learning without incorporating class priors.

Keywords: Text classification, Protein-protein interactions, Semi-supervised learning, Local learning.

1 Introduction

Text classification is the process of categorizing documents into different classes using predefined category labels. It is a difficult task because of the complexity and ambiguity of natural language, where a word may have different meanings or multiple phrases can be used to express the same idea. The task of classifying biomedical literature is made more complex than even standard text classification by the fact that such papers use a varied and specialized vocabulary. The corpus also tends to be very large because of the vast number of papers available, with online repositories such as PubMed[1] containing over 16 million citations alone. This makes it an uphill task to get the data that one needs.

One area where biomedical text classification would be useful is in the study of protein-protein interactions (PPI). Analyzing these interactions is invaluable in learning more about cellular function, which in turn paves the way for breakthroughs in medicine and biochemistry. As such, the cataloguing of interactions between proteins is an essential facet of data mining in biomedical literature, which has led to the creation of online databases specifically devoted to this task [1,2]. In order to narrow down the search for human and automated curators alike, text classification could be performed to separate the documents that describe protein-protein interactions from those that do not. It

[1] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?DB=pubmed

H. Horacek et al. (Eds.): NLDB 2009, LNCS 5723, pp. 182–191, 2010.

has been shown to be possible to identify Medline papers containing such interactions by examining the word frequencies in their abstracts instead of the entire documents [3]. This would speed up the rate at which classification is performed, making it more cost-effective to use before searching for interactions within the documents.

So far, there has not been much work done on text classification of biomedical literature, or at least that focus on protein interactions. One popular method of classification in use appears to be support vector machines. It has been used in general classification of biomedical literature along with clustering [4], and has also been applied specifically to classifying PPI-related documents [5]. Another approach uses a Bayesian method to calculate the relevance of documents to PPI [3]. In this method, documents containing PPI were used as a training set and their frequencies compared with a dictionary of the most common words in the corpus. From this, a list of discriminating words were obtained that might differentiate other relevant documents as well. Each abstract was then ranked using Bayesian probabilities that were converted into log likelihood scores.

In the area of identifying significant terms before classification, substring matching has been proposed [6]. This method involves indexing all substrings of the words in the corpus and ranking them based on relevance to classification. This results in a greatly enlarged vocabulary, but is able to identify word parts like *acety* and *peptide* that are meaningful in biology, that traditional stemming algorithms are unable to find.

Traditionally, classification problems have been handled by supervised learning, in which the entire training set consists of labeled data. However, such data is difficult and tedious to obtain, making it impractical for real-life situations. This has resulted in a growing interest in semi-supervised learning, where the training set only has a small proportion of labeled data in comparison with the large amount of unlabeled data in the set. The fact that training a classifier using both labeled and easily obtained unlabeled data makes semi-supervised learning much more flexible.

This paper proposed a semi-supervised learning algorithm based on local learning with class priors (LL-CP) for biomedical text classification. The LL-CP algorithm represents labeled and unlabeled examples as vertices in a connected graph. The label information from the labeled vertices is then propagated to the whole dataset using the linear neighborhoods with sufficient smoothness. The class prior has been incorporated to force the class distribution of the unlabeled set to be similar to that of the labeled set. Experiments have been extensively studied to identify text documents containing protein-protein interactions with only a limited number of label documents.

The rest of the paper is organized as follows. Section 2 presents the semi-supervised learning algorithm based on local learning with class priors for protein-protein interactions classification. Experimental setup and results are discussed in Section 3 and Section 4 respectively. Finally, Section 5 concludes the paper and outlines the possible future work.

2 Local Learning with Class Priors (LL-CP) for PPI Classification

In the local learning framework, data objects are represented as vertices in a fully connected graph with weighted edges. Each vertex has soft labels (i.e. the value of the label can be continuous) associated with it, which stand for the distribution over the various classes for that vertex. The larger the weight is on an edge, the closer the vertices

connected by that edge are to each other, and the easier it is for labels to propagate through that edge. Most graph-based semi-supervised learning methods [7,8,9] adopted a Gaussian function to calculate the edge weights of the graph and as a result, they are sensitive to the setting of the variance σ of the Gaussian function. A small variation of σ could affect the classification accuracy dramatically. More recently, several algorithms [10,11] have been proposed to overcome this problem where the predicted label at an unlabeled point x_i is the weighted average of its neighbors' solutions.

We propose a semi-supervised learning algorithm based on local learning with class priors (LL-CP). Assume that a class prior conditional probability is given in the form of $\tilde{P}(y|x_i)$ where $y \in \{-1, 1\}$ is the binary class label and x_i is a input instance. This prior knowledge essentially expresses our belief about the conditional distribution of the labels given the input features. It could be obtained in various ways. In the simplest case, it could be obtained from human prior knowledge. If such knowledge is not available, it could either be the maximum entropy prior $\forall x, y : \tilde{P}(y|x) = 0.5$ or the class prior estimated from the labeled data only. In this paper, we are particularly interested in obtaining the class prior in the later case. This essentially enforce that all unlabeled data are not put in the same class [12].

Thus, our goal is to find a model which minimizes the prediction error for each document as much as possible while at the same time its probabilistic predictions over the unlabeled data resembles the given class prior. Let $D = \{d_1, d_2, ..., d_{|D|}\}$ be a set of $|D|$ cosine-normalized document vectors with N dimensions each, with the first l documents being labeled and the remaining u documents left unlabeled. We have two classes here, either positive or negative. The document space is represented as a fully-connected graph where each node represents a document, and the edge between any two nodes represent a relationship between them.

Assume each document can be optimally reconstructed using a linear combination of its neighbors. Thus, for each document d_i, the objective is to minimize the least square error

$$\varepsilon_i = \frac{1}{n_i} \sum_{d_j \in \mathcal{N}_i} \|\mathbf{w}_j^T d_j - f_j\|^2 + \lambda_i \|\mathbf{w}_i\|^2 \tag{1}$$

$$s.t. \sum_j w_{ij} = 1, w_{ij} \geqslant 0 \tag{2}$$

where $\mathcal{N}_i$ represents the neighborhood of d_i, $n_i = |\mathcal{N}_i|$ is the cardinality of $\mathcal{N}_i$, f_j indicates whether d_j belongs to a positive or negative class, and w_{ij} is the contribution of d_j to d_i with larger w_{ij} indicating closeness of the documents, and λ_i is a regularization parameter.

It has been shown in [13] that the optimal solution is

$$\mathbf{w}_i^* = \mathbf{D}_i(\mathbf{D}_i^T \mathbf{D}_i + \lambda_i n_i \mathbf{I}_i)^{-1} \mathbf{f}_i \tag{3}$$

where $\mathbf{D}_i = [d_i^1, d_i^2, ..., d_i^{n_i}]$ in which d_i^k denotes the k-th nearest neighbor of d_i, $\mathbf{f}_i \in \mathcal{R}^{n_i}$ is the vector $[f_j]^T$ for $d_j \in \mathcal{N}_i$, and $\mathbf{I}$ is an $n_i \times n_i$ identity matrix.

An iterative procedure is then performed to propagate labels of the labeled data to the remaining unlabeled data using the graph constructed in the above step. In each

iteration, the label information of a document object is updated by the label information from its neighborhood. At time $t+1$, the label of d_i becomes

$$f_i^{t+1} = \alpha \sum_{j:d_j \in \mathcal{N}_i} w_{ij} f_j^t + (1-\alpha) y_i \tag{4}$$

where $0 < \alpha < 1$ determines the amount of the label information that d_i receives from its neighbors. y_i is the label of d_i at the initial state. That is, if d_i is initially labeled, then y_i is its original label; if d_i is initially unlabeled, then $y_i = 0$. f_i^t is the predicted label at iteration t.

The label of each document object is updated iteratively until the predicted labels of the data do not change in several successive iterations.

There are several ways to incorporate the class prior knowledge into the local learning process. First, the class prior information can be added as an additional constraint into the objective function. Let P_l be the multinomial distribution of class proportion in the labeled set, and $\tilde{P}_{\mathbf{W}}$ be the class proportion produced by the current model parameterized by $\mathbf{W}$, P_l and $\tilde{P}_{\mathbf{W}}$ are defined as:

$$P_l = \frac{1}{l} \sum_{j=1}^{l} f_j \tag{5}$$

$$\tilde{P}_{\mathbf{W}} = \frac{1}{u} \sum_{i=l+1}^{l+u} f_i \tag{6}$$

where l and u are the number of documents in the labeled set and unlabeled set respectively. An additional constraint $P_l = \tilde{P}_{\mathbf{W}}$ could be added.

It is also possible to add the class prior itself as a regularizer to the objective function by minimizing the KL-divergence of P_l and $P_{\mathbf{W}}$ [14]. We leave this as future work for further exploration.

We follow a simple procedure called class mass normalization (CMN) proposed in [15] to adjust the class distributions to match the priors. Let P^+ and P^- denote the class prior probability estimated from the labeled set for the positive and negative class respectively. The estimated class label f_i for an unlabeled document d_i is readjusted by incorporating the class prior probabilities and is classified as positive class iff

$$P^+ \frac{f_i}{\sum_{i=l+1}^{l+u} f_i} > P^- \frac{1-f_i}{\sum_{i=l+1}^{l+u} (1-f_i)} \tag{7}$$

3 Experimental Setup

For all experiments, the LL-CP algorithm was evaluated using data provided by the second BioCreAtIvE (Critical Assessment for Information Extraction in Biology) challenge[2]. The BioCreAtIvE challenge evaluation was set up in order to apply approaches in information retrieval and text mining to biomedical literature, and to evaluate them

[2] http://biocreative.sourceforge.net/

against a standard set of data for comparison. One of the tracks in the second BioCreAtIvE challenge in 2006 was the extraction of PPIs from text, which includes the retrieval of documents containing information about protein interactions (Protein Interaction Article Sub-task 1).

The documents in the training set provided for the Protein Interaction Article Sub-Task 1 was used as the corpus. This set consists of biomedical publications from the PubMed database, and the documents are split into two categories: those that contain information about protein interactions and those that do not. In total, there were 3536 true positive examples and 1959 true negative examples available. There were also 18930 positive but noisy examples which were not used in the experiments.

Training sets of labeled examples were obtained from the corpus using different sizes, with each containing 10, 25, 50, 75 or 100 documents. In addition, different proportions of positive/negative examples were also used, with 25%, 50% or 75% of the documents in the training sets being positive. Altogether, there were 15 different kinds of training sets used. There were also 5 test sets of unlabeled documents created, each unique set consisting of 250 relevant and 250 irrelevant documents.

3.1 Preprocessing

The documents to be classified were first read in by an XML parser. Only the `CURATION_RELEVANCE` (indicates whether the specified document is in the relevant set), `TITLE` and `ABSTRACT` child elements were saved into memory, while the other elements were ignored. During the parsing of the XML documents, stemming was performed using Porter's algorithm [16]. Stop-words were also removed by comparing words to a list of common words. Punctuation, numbers and other non-alphabet characters were ignored. After parsing, the tf-idf of the document vectors was computed, which in turn was passed into a matrix for performing singular value decomposition (SVD) along with the integer k, which is the reduced number of dimensions required. In order to perform the SVD needed for latent semantic indexing (LSI), JAMA (a Java matrix package)[3] was used. After decomposition, the resultant right-singular matrix V' was then saved as the set of column vectors of reduced dimensionality, and cosine-normalized.

3.2 Evaluation Metrics

In the area of information retrieval, the set of relevant documents matched (or retrieved) by the classifier is normally not exactly the same as the set of relevant documents in the corpus. Correct matches of relevant and irrelevant documents are known as true positives and true negatives respectively, while incorrectly classified documents are known as false negatives or positives. The most commonly used measures to evaluate the effectiveness of an algorithm are precision and recall [17]. Precision is the proportion of documents retrieved by the classifier that are relevant, while recall is the proportion of relevant documents in the entire corpus that were retrieved. The F-measure (or F-score) is a combination of both precision and recall, F-measure$=$ $(2\times\text{Precision}\times\text{Recall})/(\text{Precision}+\text{Recall})$. The ranges of all 3 values fall in the range $[0, 1]$, with a higher value indicating a better classification result.

[3] http://math.nist.gov/javanumerics/jama/

4 Results

This section presents the experimental results on the Protein Interaction Article Subtask 1 of the second BioCreAtIvE challenge.

4.1 Number of Vector Dimensions

Experiments were conducted to find out if LL-CP performs better with fewer vector dimensions, or whether there is an optimal number of dimensions across all data sets.

The LSI of each pair of training and test sets was first computed, then the LL-CP algorithm was run on the resultant document matrix V_k, with k starting from the original length of the document vectors in V and decreasing in intervals of 10 until k was equal to either 10 or 5. For each pair of sets, the number of dimensions that resulted in the highest F-measure was recorded.

As can be seen from Figure 1, the majority of optimal results occurred when the number of vector dimensions was below 20, with the most frequently occurring number of dimensions being 10. This indicates that the top few dimensions returned from LSI are most important in correctly classifying the documents. In addition, all F-measures of more than 0.7 occurred only when the number of dimensions used was less than 400. As a comparison, the average F-measure of all trials run using vector dimensions of 5 or 10 was 0.693, while none of the trials with all dimensions included had a score above 0.6.

The average performance of the training sets was evaluated with the results displayed in Figure 2. The size of the training set varies between 10 documents to 100 documents and the axes in dash lines show the F-measure values 0.66, 0.7, and 0.74. It was found that the best overall size for a training set was at least 50, with better results for larger training sets. From this experiment, the most consistent and best overall spilt of positive and negative examples was 50%-50%, while the training sets with only 25% positive examples fared the worst. This implies that LL-CP either gives better performance with equal numbers of labeled examples from all classes, or with labeled data that mimics the proportions of class labels in the test set. Also, the size of the training set did not appear to make much difference in the classification quality, except where the proportion of positive labeled examples was 75%.

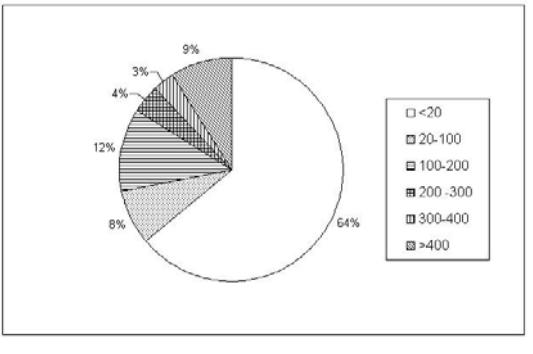

Fig. 1. A breakdown of optimal vector dimensions

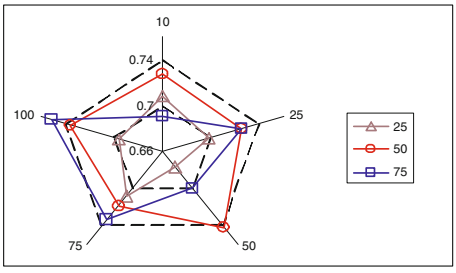

Fig. 2. Relationship between F-measure, size of training set and number of positive examples

The scores of an arbitrarily selected trial against the number of vector dimensions are charted in Figure 3, and are typical of most trials in this experiment. As can be seen from the graph, the scores usually stayed around 0.5 for the most part, indicating that some of the dimensions used were probably too noisy to aid in classification. The precision scores tended towards more gradual changes than the recall scores, which tended to be either 0 or at least 0.5, while rarely being in between the 2 values. Precision was hardly ever higher than recall.

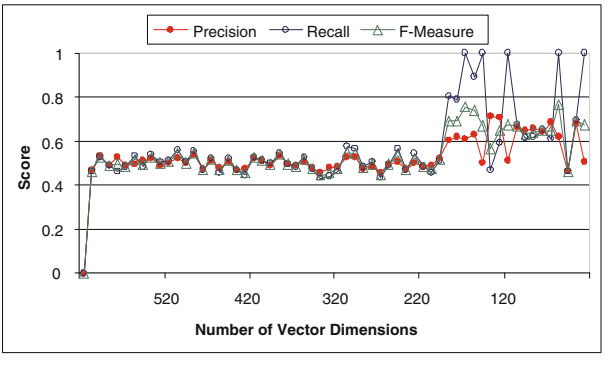

Fig. 3. The effect of vector dimensions on precision, recall and F-measure

4.2 Classification Accuracy of LL-CP

All data sets were tested with 10 vector dimensions. The results were averaged out over each combination of training set size and proportion of positive/negative examples. The results are shown in Figure 4.

As expected, the data sets with 25% positive examples fared the worst, The 50% positive sets gave the best results, probably because they reflected the actual class distribution of the unlabeled documents. The 75% positive sets were more consistent across all the data set sizes than the sets of other proportions. This may be because of the additional positive labeled examples which help to create a strong initial cluster of known positive points, enabling other points to be labeled strongly as positive. Negative examples are less likely to cluster together because irrelevant documents are not likely to

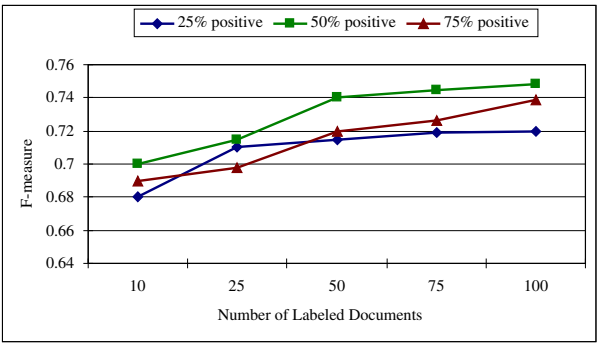

Fig. 4. The F-measures of each group of data with 25%, 50% and 75% positive examples versus different sizes of labeled documents

be similar to other irrelevant documents, whereas most relevant documents will have some features in common. In addition, the focus of this classifier is on labeling *relevant* documents correctly, and the F-measure reflects this.

This implies that if the class distribution of the unlabeled documents are known, then the documents in the labeled set should be chosen to reflect this. However, if it is unknown, then the labeled set should contain more positive than negative examples in order to strongly label the relevant documents.

4.3 Comparison with SVM and Local Learning without Class Priors

This set of experiments compares the performance of three different models, our proposed LL-CP, local learning without incorporating class priors (LL), and the Support Vector Machines (SVMs). Based on previous experiments carried out in this area [5], a radial-basis kernel function was used for SVM with σ set to 0.01 and C set to 2. Since these parameters were obtained from the optimization from PubMed abstracts like those in the BioCreAtIvE corpus, it was assumed that these parameters could be reused for the purposes of our experiments here. The SVM was run only on the 50% positive training sets so that differences in the class distribution between the labeled and unlabeled data would not affect its accuracy.

It can be observed from Figure 5 that the classification performance using SVM was mediocre, with the F-measure never rising above 0.7. Both LL and LL-CP outperforms SVM. The performance difference between LL and LL-CP is negligible when there are only 25 or less labeled documents in the training set, suggesting that incorporating class priors does not help with a limited number of labeled documents. However, LL-CP outperforms LL when the number of labeled documents increases.

The results for the SVM also contrast sharply with the purported accuracy of the SVM used in text classification [18,19,20], which had a recall of 90%, precision of 92% and an overall F-measure of 92%. However, it should be noted that the SVM in that case was trained in several rounds, using articles classified by an expert and by an already trained SVM. Their method of classification also included user feedback and training. The fact that LL-CP could outperform SVM in the absence of such close

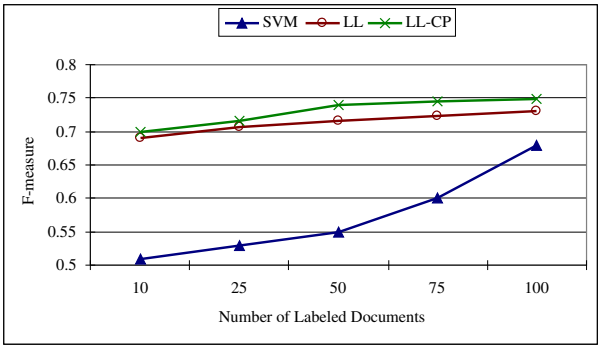

Fig. 5. Average performance of LL-CP and SVM performance over all data set sizes

supervision suggests that it is robust enough to hold its own against more established algorithms such as SVM.

5 Conclusions and Future Work

This paper has investigated a semi-supervised learning algorithm based on local learning with class priors for protein-protein interactions classification from text. Experiments have been carried out on the algorithm to determine the effect of incorporating the class prior knowledge. It was discovered that LL-CP performed better than the local learning method without incorporating class priors.

The algorithm has been applied successfully to the problem of document classification of biomedical literature with promising results, and a brief comparison of LL-CP with SVM has shown that LL-CP performs better in this area. However, biomedical literature is quite different from general text because of its specialized, complex vocabulary, so the findings from this study may not apply to document classification in general. Thus, it is suggested that the use of LL-CP in document classification should be further investigated, and perhaps put to practical use if proven to be superior to other available algorithms.

One disadvantage of using LSI is that it requires a lot of computation time and memory due to the calculation of matrix inverses. In addition, information about individual terms is lost in the calculation of the reduced document matrix, so it is impossible to determine which words were most significant in the classification process. An alternative is to use some other form of feature reduction. For example, substring matching [6] could be explored further in place of performing stemming and LSI, and is ideal for biomedical literature because it would be able to identify and group together biomedical affixes that a normal stemming algorithm would miss.

References

1. Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D., Apweiler, R.: IntAct: an open source molecular interaction database. Nucleic Acids Research 32(1) (2004)

2. Xenarios, I., Rice, D., Salwinski, L., Baron, M., Marcotte, E., Eisenberg, D.: DIP: the database of interacting proteins. Nucleic Acids Research 28(1), 289–291 (2000)
3. Marcotte, E., Xenarios, I., Eisenberg, D.: Mining literature for protein-protein interactions. Bioinformatics 17(4), 359–363 (2001)
4. Chen, D., Muller, H.M., Sternberg, P.W.: Automatic document classification of biological literature. BMC Bioinformatics 7 (2006)
5. Donaldson, I., Martin, J., de Bruijn, B., Wolting, C., Lay, V., Tuekam, B., Zhang, S., Baskin, B., Bader, G., Michalickova, K., et al.: PreBIND and Textomy – mining the biomedical literature for protein protein interactions using a support vector machine. BMC Bioinformatics 11(4) (2003)
6. Han, B., Obradovic, Z., Hu, Z., Wu, C., Vucetic, S.: Substring selection for biomedical document classification. Bioinformatics 22(17), 2136–2142 (2006)
7. Szummer, M., Jaakkola, T.: Partially labeled classification with markov random walks. In: Advances in Neural Information Processing Systems, vol. 14 (2002)
8. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: 18th Annual Conf. on Neural Information Processing Systems (2003)
9. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th International Conference on Machine Learning (2003)
10. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. In: ICML 2006: Proceedings of the 23rd international conference on Machine learning, pp. 985–992 (2006)
11. Wu, M., Scholkopf, B.: Transductive classification via local learning regularization. In: Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS 2007), pp. 628–635 (2007)
12. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: Proceedings of the 9th International Conference on Artificial Intelligence and Statistics, AISTATS 2005 (2005)
13. Wang, F., Zhang, C., Li, T.: Regularized clustering for documents. In: SIGIR 2007: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 95–102. ACM, New York (2007)
14. Mann, G.S., McCallum, A.: Simple, robust, scalable semi-supervised learning via expectation regularization. In: Proceedings of the 24th international conference on Machine learning, pp. 593–600. ACM, New York (2007)
15. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using gaussian fields and harmonic functions. In: The 20th International Conference on Machine Learning, pp. 912–919 (2003)
16. Porter, M.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
17. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2007)
18. Yu, H., Han, J., Chang, K.C.C.: PEBL: Positive Example-Based Learning for Web Page Classification Using SVM. In: ACM SIGKDD International Conference in Knowledge Discovery in Databases (KDD 2002). ACM Press, New York (2002)
19. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building text classifiers using positive and unlabeled examples. In: Third IEEE International Conference on Data Mining, pp. 179–188 (2003)
20. Li, X., Liu, B.: Learning to classify texts using positive and unlabeled data. In: Eighteenth International Joint Conference on Artificial Intelligence, pp. 587–594 (2003)