Web Page Classification Using Relational Learning Algorithm and Unlabeled Data

Yanjuan Li[1,2]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
[2]School of Information and Computer Engineering, North-East Forestry University, Harbin, China
Email: lyjuan5@gmail.com

Maozu Guo
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
Email: maozuguo@hit.edu.cn

Abstract—**Applying relational tri-training (R-tri-training for short) to web page classification is investigated in this paper. R-tri-training, as a new relational semi-supervised learning algorithm, is well suitable for learning in web page classification. The semi-supervised component of R-tri-training allows it to exploit unlabeled web pages to enhance the learning performance effectively. In addition, the relational component of R-tri-training is able to describe how the neighboring web pages are related to each other by hyperlinks. Experiments on Web-Kb dataset show that: 1) a large amount of unlabeled web pages (the unlabeled data) can be used by R-tri-training to enhance the performance of the learned hypothesis; 2) the performance of R-tri-training is better than the other algorithms compared with it.**

Index Terms—**web page classification, relational tri-training, relational learning, tri-training, co-training**

I. Introduction

In recent years there has been a great deal of interest in applying machine-learning methods to a variety of problems in classifying and extracting information from text [1]. In large part, this trend is sparked by the explosive growth of the World Wide Web. An interesting aspect of the Web is that it can be thought of as a graph in which pages are the nodes of the graph and hyperlinks are the edges. The graph structure of the Web makes it an interesting domain for relational learning [2]. Moreover, Craven, Slattery, and Nigam demonstrated that for several Web-based learning tasks, a relational learning algorithm can learn more accurate classifiers than a common statistical approach [3]. Therefore, many researchers have been done to apply relational learning algorithms to web page classification.

Relaxation labeling [4] is one of the algorithms that work well in web page classification. Relaxation labeling was originally proposed as a procedure in image analysis by Rosenfeld, Hummel, and Zucker. Later, it became widely used in image and vision analysis, artificial intelligence, pattern recognition, and web mining [5]. In the context of hypertext classification, the relaxation labeling algorithm employs a text classifier to assign class probabilities to each node (page) firstly. Then each page is considered in turn and its class probabilities are reevaluated in light of the latest estimates of the class probabilities of its neighbors.

Based on a new framework for modeling link distribution through link statistics, Lu and Getoor [6] proposed a variation of relaxation labeling, in which a combined logistic classifier is used based on content and link information. This approach not only showed improvement over a textual classifier, but also outperformed a single flat classifier based on both content and link features. Angelova and Weikum [7] proposed another variation of relaxation labeling, not all neighbors are used, that is, only neighbors that are similar enough in content are considered.

Besides relaxation labeling, other relational learning algorithms can also be applied to web classification. Loopy belief propagation and iterative classification are compared and analyzed with relaxation labeling by Sen and Getoor [8]. Their performance on a web collection is better than textual classifiers. A toolkit for classifying networked data is implemented by Macskassy and Provost [9], it utilized a relational classifier and a collective inference procedure [10], and its powerful performance is demonstrated on several datasets including web collections.

Although these relational learning algorithms have been successfully applied to web page classification, they need large number of labeled web pages acting as training set in order to guarantee generalization ability. However, labeled web pages are fairly expensive to obtain because they require artificial marking. In contrast, unlabeled web pages are readily available. Therefore, in relational learning domain, how to exploit unlabeled web pages to enhance the performance of the learned hypothesis is an urgent challenge.

In this paper, we apply R-tri-training algorithm, which is proposed in our previous work [11], to web page classification. R-tri-training, as a relational semi-supervised learning algorithm, combines semi-supervised learning and relational learning. Three different relational learning systems are initialized according to the labeled data and the background knowledge, and then the three classifiers are refined by iterating using the unlabeled

© 2011 ACADEMY PUBLISHER
doi:10.4304/jcp.6.3.474-479

data under certain condition. Experiments on Web-Kb using R-tri-training show its outstanding performance.

The remainder of the paper is organized as follows. Section 2 gives the brief description of relational tri-training. In section 2, the selection of base classifier and the sufficient condition for re-training is introduced. Sections 3 discuss the experiment. In section3, the dataset used in this paper is given firstly, and then the effect of unlabeled web pages is discussed, finally the comparison with other systems is proposed. Conclusion is given in section 4.

II. RELATIONAL TRI-TRAINING

Relational learning is a research area which investigates the inductive construction of first-order clausal theories from examples and background knowledge. By using first-order logic as the representational mechanism for hypotheses and examples, relational learning can overcome the two main limitations of classical machine learning techniques, 1) the use of a limited knowledge representation formalism(essentially a propositional logic), and 2) difficulties in using substantial background knowledge in the learning process [12]. However, current relational learning algorithm such as Nfoil [13], Kfoil [14] and Aleph [15] are all supervised learning algorithms. Therefore, they need a large number of labeled examples acting as training set in order to guarantee generalization ability.

Existing semi-supervised learning algorithms attempt to exploit the additional information provided by the large amount of unlabeled data to guide the learning process, and enhance the final performance. A prominent method known as tri-training was proposed by Zhou and LI [16]. However, current semi-supervised learning algorithm belongs to classical machine learning, use propositional logic to represent training examples, therefore, it cannot overcome the two main limitations of classical machine learning techniques mentioned above.

Since relational learning and semi-supervised learning have their own pros and cons, it motivates the integration of the two approaches. Relational tri-training [11] is a framework for combining the power of enhancing the learning performance by exploiting the unlabeled examples of tri-training and the power of knowledge representation of relational learning. This algorithm combined the tri-training based on propositional logic representation and relational learning algorithm based on first-order logic representation. Three different relational learning systems (base classifiers) are initialized according to the labeled data and the background knowledge, and then the three classifiers are refined by iterating using the unlabeled data. That is, under special condition, for one unlabeled data, it is going to be labeled to one classifier as the new training data when the same labeled result are given by the other two classifiers. The final hypothesis (the learned model) is produced via majority voting of the three base classifiers.

A. The selection of base classifier

Relational tri-training algorithm employs Nfoil, Kfoil and Aleph as three different base classifiers. Nfoil was the first system in literature to tightly integrate feature construction and Naive Bayes. Such a dynamic propositionalization was shown to be superior compared to static propositionalization approaches that employ Naive Bayes only for post-processing the rule sets. Kfoil system integrates FOIL with kernel methods. The feature space is constructed by leveraging FOIL search for a set of relevant clauses. The search is driven by the performance obtained by a support vector machine based on the resulting kernel. KFOIL can naturally handle classification and regression tasks. ALEPH is a publicly available relational learning system implemented in a Prolog compiler called Yap (version 4.3.22) that is a generalization of PROGOL [17].

The reason for selecting Nfoil, Kfoil and Aleph as the base classifiers is that the three relational learning algorithms have different search bias. Nfoil and Kfoil belong to top-down relational learning algorithm that learns program clauses by searching a space of possible clauses from general to specific. The learned final classifier is the most general hypothesis and is prone to classify the unlabeled data into positive. Aleph belongs to bottom-up relational learning algorithm that searches for program clauses by starting with very specific clauses and attempting to generalize them. The learned final classifier is the most specific hypothesis and is prone to classify the unlabeled data into negative.

B. The sufficient condition for re-training

Let L denote the initial labeled example set, and U denote the unlabeled example set. Background knowledge is denoted by B. Nfoil, Kfoil, Aleph system is respectively trained from L and B, three different classifiers h_1, h_2, h_3 is obtained. If h_2 and h_3 agree on the labeling of an example x in U, then x can be labeled for h_1. It is obvious that in such a scheme if the prediction of h_2 and h_3 on x is correct, then h_1 will receive a valid new example for further training; otherwise, h_1 will get an example with noisy label. Therefore, in previous work [11], we analyses the PAC theory of concept class $C_{B,d,M}$, then gives the sufficient condition to ensure that the classification accuracy of individual classifier could be improved by re-training on the new training set, and use the sufficient condition as criteria to decide whether the newly labeled example set should be used for re-training. To guarantee the integrity of the article, this paper briefly describes the sufficient condition.

According to the conclusion of Horuath, Sloan and Turan [18], The concept class $C_{B,d,M}$ is polynomially PAC learnable with random classification noise with sample size

$$s = O\left(\frac{1}{\varepsilon^2(1-2\eta_b)^2}\left(\left(2m|M|\right)^{2a^{d+1}}\ln\left(\frac{\left(2m|M|\right)^{a^{d+1}}}{\delta}\right)+\ln\left(\frac{\left(2m|M|\right)^{a^{d+1}}}{\left(1-2\eta_b\right)^2\varepsilon\delta}\right)\right)\right)$$

, where ε is the hypothesis worst-case classification error rate, η (<*0.5*) is an upper bound on the classification noise rate, δ is the confidence , m is the arity of target

© 2011 ACADEMY PUBLISHER

predicate, $|M|$ is the number of modes in M, a is the maximum arity of any predicate in background knowledge, d is depth bound, $C_{B,d,M}$ is concept class with background knowledge B that are represented by Horn clauses of depth at most d satisfying M.

Let

$$s' = \frac{1}{\varepsilon^2 (1-2\eta_b)^2}\left(\left(2m|M|\right)^{2a^{d+1}} \ln\left(\frac{\left(2m|M|\right)^{a^{d+1}}}{\delta} \right) + \ln\left(\frac{\left(2m|M|\right)^{a^{d+1}}}{(1-2\eta_b)^2 \varepsilon\delta} \right) \right),$$

$\mu = s/s'$, then

$$s = \mu s' = \frac{\mu}{\varepsilon^2 (1-2\eta_b)^2}\left(\left(2m|M|\right)^{2a^{d+1}} \ln\left(\frac{\left(2m|M|\right)^{a^{d+1}}}{\delta} \right) + \ln\left(\frac{\left(2m|M|\right)^{a^{d+1}}}{(1-2\eta_b)^2 \varepsilon\delta} \right) \right)$$

(1)

For a specific concept class $C_{B,d,M}$, m, $|M|$, a, d are constants, δ is confidence and is also a constant.

Let $p = \left(2m|M|\right)^{2a^{d+1}} \ln\left(\frac{2m|M|^{a^{d+1}}}{\delta} \right)$, $q = \frac{\left(2m|M|\right)^{a^{d+1}}}{\delta}$,

then p, q are constant, Eq. (1) can be rewritten as:

$$s = \frac{\mu}{\varepsilon^2 \left(1-2\eta_b\right)^2}\left(p + \ln\left(\frac{q}{\left(1-2\eta_b\right)^2 \varepsilon} \right) \right) \quad (2)$$

In each round of R-tri-training, the classifiers h_2 and h_3 choose some examples in U to label for h_1. Since the classifiers are refined in the R-tri-training process, the amount as well as the concrete unlabeled examples chosen to label may be different in different rounds. Let L_t and L_{t-1} denote the set of examples that are labeled for h_1 in the tth round and the (t-1)th round, respectively. Then, the training set for h_1 in the tth round and the (t-1)th round are $L \cup L_t$ and $L \cup L_{t-1}$, whose sample size are $|L \cup L_t|$ and $|L \cup L_{t-1}|$, respectively. Let $e_{1,t}$ (<0.5) denote the upper bound of the classification error rate of h_2&h_3 in the tth round, i.e., the error rate of the hypothesis derived from the combination of h_2 and h_3. Thus, the number of examples in L_t that are mislabeled is $e_{1,t}|L_t|$. Therefore, the classification noise rate in the tth round is:

$$\eta_t = \frac{e_{1,t} \, | L_t |}{| L_t | + | L |} \quad (3)$$

Let ε_{t-1}, ε_t denote the classification error of h_1 in (t-1)-th iteration and t-th iteration, respectively. Eq. (2) could respectively be reformed as Eq. (4) and Eq. (5) in (t-1)-th iteration and t-th iteration.

$$| L \cup L_{t-1} | = \frac{\mu}{\varepsilon_{t-1}^2 (1-2\eta_{t-1})^2}\left(p + \ln\left(\frac{q}{(1-2\eta_{t-1})^2 \varepsilon_{t-1}} \right) \right) \quad (4)$$

$$| L \cup L_t | = \frac{\mu}{\varepsilon_t^2 (1-2\eta_t)^2}\left(p + \ln\left(\frac{q}{(1-2\eta_t)^2 \varepsilon_t} \right) \right) \quad (5)$$

With Eq. (4) and Eq. (5), the sufficient condition of $\varepsilon_t < \varepsilon_{t-1}$ can be given

$$|L_t| \geq |L_{t-1}| \text{ and } \eta_t < \eta_{t-1}$$

Considering Eq. (3), we have

$$|L_t| \geq |L_{t-1}| \text{ and } \frac{e_{1,t} \, | L_t |}{| L_t | + | L |} < \frac{e_{1,t-1} \, | L_{t-1} |}{| L_{t-1} | + | L |}$$

It is the sufficient condition for re-training used in relational tri-training.

III. Experiment

A. Dataset

The Web page dataset, we use for our experiments is the Web-Kb [19] dataset. This data set consists of 4,167 Web pages collected from Web sites of Computer Science departments of four universities: Cornell University, University of Texas, University of Washington, and University of Wisconsin. These pages have been labeled into a number of categories, among which the category student home page is regarded as the target. That is, student home pages (13 percent) are the positive examples and all other pages are negative examples.

B. Effect of unlabeled web page

Experiment settings: we randomly selected 40% of all web pages as test examples to evaluate the performance of the learned hypothesis, while the rest are used as the pool of training examples, i.e., $L \cup U$. In each pool, L and U are partitioned under different unlabeled rates including 80 percent, 60 percent, 40 percent, and 20 percent. For instance, assuming a dataset contains 1,000 examples, 400 examples are selected as test examples, the remaining 600 examples are the pool of training examples. When the unlabeled rate is 80 percent, 120 examples are put into L with their labels while the remaining 480 examples are put into U without their labels. Here, the pos/neg ratio of L, U, and test set are similar to that of the original dataset.

Under each unlabeled rate, three different random partitions of L and U are generated and one independent run is performed for each random partition, then the average classification error rate of three runs is computed and as the final performance evaluation.

Experiment results: The average predictive error rates of the final hypothesis and of each base classifier (Nfoil, Kfoil, Aleph) under different unlabeled rate are respectively shown in table I, II, III, IV, respectively. Where *initial* denotes the performance at round 0, i.e., the performance obtained using only the labeled training examples, *final* denotes the performance when algorithm finished, *improve* denotes the corresponding improvements, and *Hypothesis* denotes the performance of the hypotheses, i.e., the learned models. Table I, II, III, and IV show that on the Web page classification task, relational tri-training can effectively utilize unlabeled data to enhance the learning performance.

TABLE I. Classification Error Rates of the Initial and Final classifiers under 80 Percent Unlabeled Rate

classifier	Classification error rate on test dataset		
	initial	*final*	*improve*
Nfoil	0.0650	0.0442	32.00%
Kfoil	0.0804	0.0684	14.93%
Aleph	0.0788	0.0729	7.49%
Hypothesis	0.0470	0.0394	16.17%

© 2011 ACADEMY PUBLISHER

TABLE II. CLASSIFICATION ERROR RATES OF THE INITIAL AND FINAL CLASSIFIERS UNDER 60 PERCENT UNLABELED RATE

classifier	Classification error rate on test dataset		
	initial	*final*	*improve*
Nfoil	0.0450	0.0402	10.68%
Kfoil	0.0840	0.0716	14.76%
Aleph	0.0792	0.0658	16.92%
Hypothesis	0.0402	0.0342	14.93%

TABLE III. CLASSIFICATION ERROR RATES OF THE INITIAL AND FINAL CLASSIFIERS UNDER 40 PERCENT UNLABELED RATE

classifier	Classification error rate on test dataset		
	initial	*final*	*improve*
Nfoil	0.0428	0.0392	8.41%
Kfoil	0.0632	0.0316	50.00%
Aleph	0.0868	0.0859	0.94%
Hypothesis	0.0356	0.0254	28.65%

TABLE IV. CLASSIFICATION ERROR RATES OF THE INITIAL AND FINAL CLASSIFIERS UNDER 20 PERCENT UNLABELED RATE

classifier	Classification error rate on test dataset		
	initial	*final*	*improve*
Nfoil	0.0579	0.0504	13.66%
Kfoil	0.0759	0.0612	19.37%
Aleph	0.0963	0.0747	22.43%
Hypothesis	0.0449	0.0384	14.48%

Figure 1, 2, 3, and 4 depicts the error rates change of the final hypothesis and three base classifiers (Nfoil, Kfoil, Aleph) along with the re-training iterations. Note that since the relational tri-training algorithm may terminate in different iterations, the error rates at termination are used as the error rates of the iterations after termination.

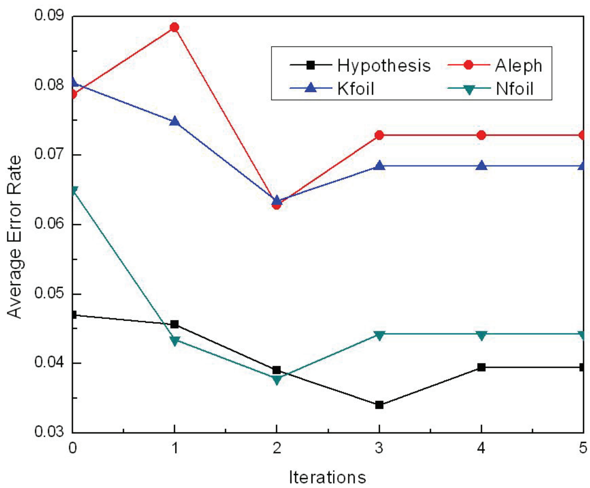

Figure 1. Iterative change of the error rates under 80 percent unlabeled data

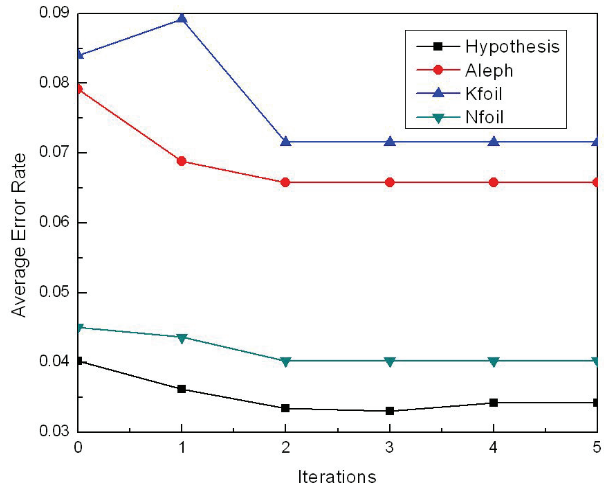

Figure 2. Iterative change of the error rates under 60 percent unlabeled data

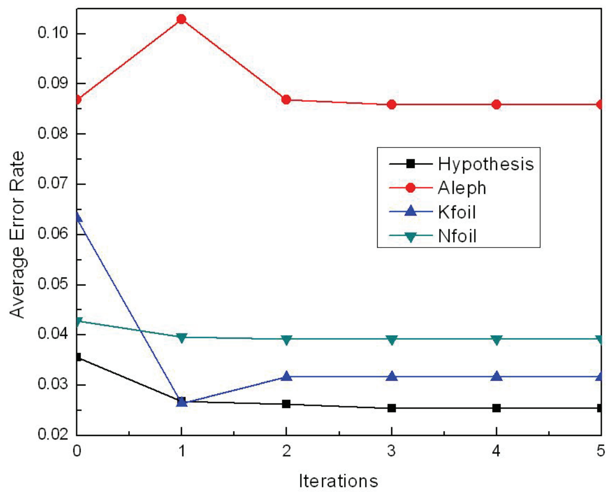

Figure 3. Iterative change of the error rates under 40 percent unlabeled data

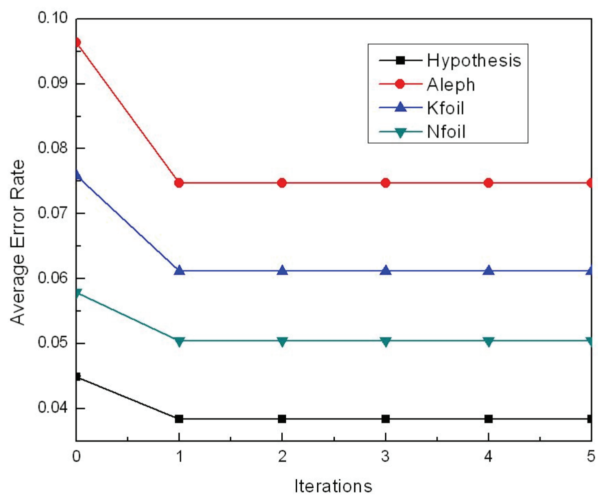

Figure 4. Iterative change of the error rates under 20 percent unlabeled data

Figure 1, 2, 3 and 4 reveal that on all percent unlabeled data, the final hypotheses generated by relational tri-training are better than the initial hypotheses, which

© 2011 ACADEMY PUBLISHER

confirms that relational tri-training can effectively exploit unlabeled examples to enhance the learning performance.

The time complexity of relational tri-training can be analyzed from Figure 1, 2, 3 and 4. It shows that the maximum number of iterations under different unlabeled rate does not exceed five times, and after four iterations, the error rate of hypothesis is no longer obviously changed. Therefore, the time complexity of relational tri-training is very low based on the iteration number when the training is over.

C. Comparison with other systems

1) Algorithms used for comparison

Naive Bayes classifier

Naive Bayes algorithm is often applied to text learning problems. Nuanwan and Boonserm [20] test the performance of naive Bayes using two feature sets which are heading and content feature on Web-Kb dataset.

Co-training algorithm

Co-training algorithm was proposed by Blum and Mitchell [21]. Two individual classifiers are trained separately on two different views, i.e. two independent sets of attributes, and then each individual classifier labels some unlabeled ones for the other to augment the training set for re-training. The standard co-training assumes that the attributes are naturally partitioned into two sets, each of which is sufficient for learning and conditionally independent to the other given the class label. When the assumption is satisfied, the cotrained classifiers could make fewer generalization errors by maxmizing their agreement over the unlabeled data [22].

ICT-ILP

ICT (Iterative-Cross Training) [20] is a semi-supervised learning algorithm. It consists of two learners and each learner gets a small amount of labeled data. The strong learner (classifier1) starts the learning process from the labeled data and classifies unlabeled data (TrainingData2). The weak learner (classifier2) uses these newly labeled data to learn and classifies TrainingData1. ICT-ILP is the ICT which combines the inductive logic programming system into one of the classifiers, that is, the Classifier1 of ICT-ILP is the Progol system.

2) Experiment setting and result

To exploit the experiment result in [20], the experimental configuration is the same as that used in it. We randomly selected 30% of all examples to be initial labeled data. The unlabeled training data consists of 30% of all examples and 40% of all examples were used as a test set.

The performance evaluation is done using the standard precision (P), recall (R) and F_1-measure (F_1). These measurements are defined as follows.

$$recall = \frac{\#correct \quad positive \quad predictions}{\#positive \quad examples}$$

$$precision = \frac{\#correct \quad positive \quad predictions}{\#positive \quad predictions}$$

$$F_1 = \frac{2PR}{P+R}$$

Table V shows the results of experiments constructed on the Web-Kb dataset, where the results of ICT-ILP, Co-training and Naïve Bayes derive from [20]. In table V, for relational tri-training, classifier1, classifier2 and classifier3 denote Nfoil algorithm, Kfoil algorithm and Aleph algorithm, respectively. For ICT-ILP, the classifier1 is Progol system and the classifier2 is Naïve Bayes. For co-training and Naïve Bayes, classifier1 denotes the heading-based classifier and classifier2 denotes the content-based classifier. Note that, all algorithms except relational tri-training algorithm have not classifier3.

TABLE V. PERFORMANCE OF CLASSIFIERS ON THE WEB-KB DATASET

Algorithm	Classifier 1			Classifier 2			Classifier 3		
	P	R	F_1	P	R	F1	P	R	F1
R-tri-training	91.38	95.06	93.18	94.61	95.48	95.05	74.81	91.93	82.49
ICT-ILP	80.00	81.82	80.90	82.61	86.36	84.44	--	--	--
Co-training	73.95	84.69	75.64	79.69	60.72	66.14	--	--	--
Naïve Bayes	74.99	87.24	79.91	76.95	84.18	79.60	--	--	--

Comparing the recall results first, we see that R-tri-training outperform other algorithms on all classifiers. Comparing the precision results and F_1 scores, we see that two classifiers (Nfoil, Kfoil) of R-tri-training outperform all classifiers of other algorithms, and one classifier (aleph) of R-tri-training is comparative with the classifiers of other algorithms. The reason that R-tri-training got the highest performance came from the contribution of relational learning algorithm and unlabeled data.

IV. CONCLUSION

Classification of web page content is essential to many tasks in web information retrieval such as maintaining web directories and focused crawling. The uncontrolled nature of web content presents additional challenges to web page classification compared with traditional text classification, but the interconnected nature of hypertext also provides features that can assist the process. An interesting aspect of the Web is that it can be thought of as a graph in which pages are the nodes of the graph and hyperlinks are the edges.

We apply relational semi-supervised learning algorithm named R-tri-training to web page classification. The relational component is able to describe the graph structure of hyperlinked pages and the internal structure of HTML pages. Furthermore, the semi-supervised learning component is able to exploit unlabeled data to enhance the learning performance. Experiment on Web-Kb is given and analyzed. We compare the result with the supervised Naive Bayes, which is the well-known algorithm for the text classification problem. The performance of R-tri-training is also compared with other semi-supervised learning algorithms which are Co-Training and ICT-ILP. The experimental results show that R-tri-training algorithm outperforms those algorithms.

© 2011 ACADEMY PUBLISHER

In the analysis of the error rates change of classifiers (see figure 1, 2, 3, 4), it is noted that the performance of R-tri-training algorithm are usually not stable because the unlabeled examples may often be wrongly labeled during the learning process. A promising solution to this problem may be using data editing mechanisms to help identify the wrongly labeled examples. Incorporating data editing mechanisms into R-tri-training is an interesting issue to be investigated in future work.

ACKNOWLEDGMENT

This work is supported by the National Nature Science Foundation of China under Grant Nos. 60932008 and 60702033.

REFERENCES

[1] X. Qi, and B.D.Davison, "Web page classification: features and algorithms," *ACM computing surveys*, 2009,41(2):12–42.
[2] M. Craven, and S.Slattery, "Relational learning with statistical predicate invention: Better models for hypertext," *Machine Learning*, 2001,43: 97–119.
[3] M. Craven, S. Slattery, and K. Nigam, "First-order learning for Web mining," *Machine Learning: ECML-98*, 1998: 250–255.
[4] A. Rosenfeld, R.A.Hummel, and S.W.Zucker, "Scene labeling by relaxation operations," *IEEE Transactions on Systems, Man and Cybernetics*, 1976, 6: 420–433.
[5] S.Chakrabarti, *Mining the Web: Discovering Knowledge from Hypertext Data*. San Francisco, CA: Morgan Kaufmann, 2003.
[6] Q.Lu and L.Getoor, "Link-based classification," *Proceedings of the 20th International Conference on Machine Learning (ICML)*, Menlo Park, 2003.
[7] R. Angelova, and G.Weikum, "Graph-based text classification: Learn from your neighbors," *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, 2006: 485-492.
[8] P.Sen and L. Getoor, *Link-based classification*. Technical Report CS-TR-4858, University of Maryland, 2006.
[9] S.A.Macskassy and F.Provost, "Classification in networked data: A toolkit and a univariate case study," *Journal of Machine Learning Research*, 2007,8(5): 935-983.
[10] D. Jensen, J. Neville and B. Gallagher, "Why collective inference improves relational classification," *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, 2004: 593–598.
[11] Y.J.Li and M.Z.Guo, "Relational tri-training: learning first-order rules exploiting unlabeled data," *Proceedings of the 12th china conference on machine learning*, 2010:28-38.
[12] L. De Raedt, *Logical and Relational Learning*. Springer, 2008.
[13] N.Landwehr, K.Kersting, and L.D.Raedt, "Integrating Naïve Bayes and FOIL," *Journal of Machine Learning Research*, 2007, 8: 481–507.
[14] N.Landwehr, A.Passerini, L. De Raedt, and P.Frasconi, "kFOIL: Learning simple relational kernels," *Proceedings of the twenty-first national conference on artificial intelligence and the eighteenth innovative applications of artificial intelligence conference*, Boston, 2006, 7: 16–20.
[15] http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/aleph
[16] Z.H.Zhou and M.Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on knowledge and Data Engineering*, 2005,17(11):1529–1541.
[17] S.Muggleton, "Inverse entailment and Progol," *New Generation Computing Journal*, 1995, 13: 245–286.
[18] T.Horváth, R.Sloan, and G.Turán, "Learning Logic Programs with Random Classification Noise," *Proceeding of the 6th workshop on inductive logic programming(ILP-96)*, 1996: 97–118.
[19] S.R.Singh, H.A. Murthy and T.A.Gonsalves, "Feature selection for text classification based on Gini coefficient of inequality," *Proceedings of the Fourth International Workshop on Feature Selection in Data Mining*, 2010: 76–85.
[20] N. Soonthornphisaj and B. Kijsirikul, "Combining ILP with Semi-supervised Learning for Web Page Categorization," *International journal of Computational Intelligence*, 2005: 271–274.
[21] A. Blum, and T.Mitchell, "Combining labeled and unlabeled data with co-training," *Proceedings of the 11th annual conference on computational learning theory*, Wisconsin, 1998: 92–100.
[22] S. Golman and Y. Zhou, "Enhancing supervised learning with labeled data," *Proc. Of the 17th Int'1 Conf. on Machine Learning(ICML2000)*, San Francisco, 2000: 327–334.

Yanjuan Li is a Ph.D. candidate of Harbin Institute of Technology of China. She received B.S. degree in Computer Science and Technology College from Harbin Normal University in 2002, and received M.S. degree in Department of Computer Software and Theories from Harbin University of Science and Technology in 2005. His research interests include machine learning, data mining, pattern recognition and knowledge-based system. His current work is focused on learning with unlabeled examples, relational learning and its application to web page classification.

Maozu Guo is a professor at the Department of Computer Science & Engineering of Harbin Institute of Technology. He received the B.S. and M.S. degrees in computer science from Harbin Engineering University, China, in 1988 and 1991 respectively, and PhD degree in computer science from Harbin Institute of Technology, China, in 1997. As a visiting scholar, he once worked at AI group of the Computer Science Department at Lund University in Sweden for the academic year 2002-2003. His research interests are in machine learning, data mining, computational biology and bioinformatics. In these areas he has published over 90 technical papers in international journals or conference proceedings. He is on the editorial boards of Journal of Test and Measurement Technology and Journal of Optics and Precision Engineering. He served as program committee member for various international conferences and chaired many native conferences. He is a committee member of Chinese Association for Artificial Intelligence (CAAI), and the councilor of CAAI Machine Learning Society.

© 2011 ACADEMY PUBLISHER