Applying Semantic Links for Classifying Web Pages

BEN CHOI & QING GUO

Computer Science, College of Engineering and Science
Louisiana Tech University, Ruston, LA 71272, USA

pro@BenChoi.org

Abstract. Automatic hypertext classification is an essential technique for organizing vast amount of Internet Web pages or HTML documents. One the of problems in classifying Web pages is that Web pages are usually short and contain insufficient text to clearly identify its category. Text classification mechanisms, by analyzing only the contents of the document itself, are relatively ineffective in classifying short Web pages. This paper proposes a new hypertext classification mechanism to address the problem by analyzing not only the Web page itself but also its linked Web pages referred by the URLs contained within the page. The URLs are treated as semantic links. The hypothesis is that the linked Web pages contain related information to help identifying the category of the Web page. Experimental results show that the proposed approach could increase the accuracy by 35% over the approach of analyzing only the Web page itself.

1 Introduction

Automatic hypertext categorization is an important technique for organizing vast amount of information available on the Internet and Intranet. However, Web pages or HTML documents tend to be short and usually contain insufficient text to clearly identify its category.

Many text classification mechanisms, by analyzing only the contents of the document itself, are relatively ineffective in classifying short Web pages. Rocchio [15], for instance, is a classification mechanism where a training set of documents are used to construct a prototype vector for each category, and category ranking given for a document is based on a similarity comparison between the document vector and the category vectors. Vector Space Model [1] [2] is a relatively new approach introduced by Vapnik in 1995 for solving two-class pattern recognition problems. The methods is defined over a vector space where the problem is to find a decision surface that best separates the data points in two classes. Bayes Rule [5] categorization comes from ideas in probability and information theory. A growing number of machine learning methods include LLSF [6]-a regression model, kNN [7]-a nearest neighbor classifier,

This research was supported in part by Center for Entrepreneurship and Information Technology (CEnIT), Louisiana Tech University, Grant iCSe 200123.

Ben Choi and Qing Guo, "Applying Semantic Links for Classifying Web Pages," Developments in Applied Artificial Intelligence, IEA/AIE 2003, Lecture Notes in Artificial Intelligence, Vol. 2718, pp. 148-153, 2003.

RIPPER [8] and Charade-rule learning algorithms [9], Swap-1 [10], Widrow-Hoff [11], EG [12], and Experts [13]-inductive learning algorithms. Other techniques include fuzzy retrieval [3], neural network approaches [4], and so on.

This paper proposes a new hypertext classification mechanism to address the problem by analyzing not only the Web page itself but also its linked Web pages referred by the URLs containing within the page (see Fig. 1). The URLs are treated as semantic links. The semantic relationship between a Web page and its linked pages can be generalization ("is-a"), association ("use"), aggregation ("has"), and composition ("part-of"). The hypothesis is that the linked Web pages (page 1 to n) contain information related to Web page 0. The related information is used to facilitate the identification of the category of Web page 0. If a Web page does not contain any URL, then a standard classifier is used.

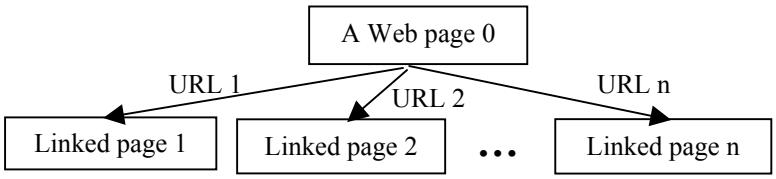

Fig. 1. Relation of a Web page and its linked pages

2 Proposed Page and Linked Pages (PLP) Classification

2.1 Architecture

Our proposed Page and Linked Pages (PLP) classification method consists of extracting surrounding information of a Web page by analyzing its linked Web pages. The overall architecture of the PLP classification is described in Fig. 2.

The Web page categorization process begins by using a standard text classifier to classify the Web page. For each URL contained within the Web page, the process retrieves and classifies the linked Web page. The process, then, computes the similarity between the resulting page category and each resulting linked page category, and assigned the similarity as a weight to each linked page. It groups linked pages that belongs to a same category and computes the weight of the group as the sum of the weights of its members. It selects the group having the highest weight and compares the highest weight with a given threshold. If the highest weight is larger than the threshold, then the process changes the page category to the category of the group that has the highest weight, otherwise it keeps the original category of the page.

2.2 Algorithm

We constructed the text classifier (see Fig. 2) based on term weighing approach [14]. In general the approach takes a document and builds a dictionary of its words (also called terms). It eliminates common words (also called stop words) such as pronouns,

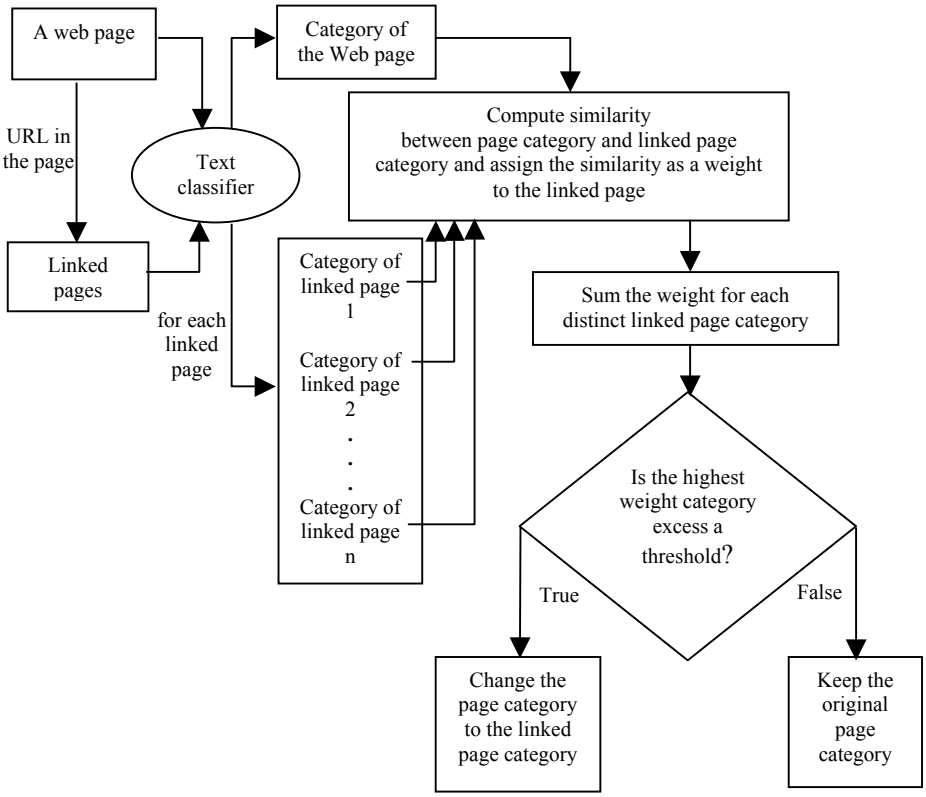

Fig. 2. Structure of PLP Classification

articles, and prepositions. For each word in the dictionary, it counts the number of occurrences in the document. The number is normalized to become a percentage. In short, a document D is represented by a set of ordered pair of a term t and its corresponding weight v (as shown in equation below).

$$D = \{(t_1, v_1), (t_2, v_2), \cdots, (t_m, v_m)\}$$

Similarly, a category C is also represented by a set of ordered pair of term s and its corresponding weight u as shown below:

$$C = \{(s_1, u_1), (s_2, u_2), \cdots, (s_n, u_n)\}$$

To compute the similarity between a document D and a category C, first obtain the intersection E of the set of terms for D and the set of terms for C, that is,

$$E = \{t_1, t_2, \cdots, t_n\} \cap \{s_1, s_2, \cdots, s_m\}$$

For each term in E computes the product of its corresponding weights v and u. The summation of all the products is a measure of similarity. This number is usually normalized to maximum similarity as 1.

We used the same process to compute the similarity between the page category P and a linked page category L (Fig. 2). The resulting similarity is then assigned as a weight to the linked page. Some of the linked pages may belong to the same category. We compute the sum of weight for each group of linked pages that belong to a category, says G. For each category of linked pages, we calculate the weight for it by using equation below.

$$w_l = \sum_{j=1}^{q} Similarity(P, L_j) \cdot \eta$$

Where w_l is total weight of category G, L_j is the category of linked page j in list of linked pages, q is total number of linked pages, and $\eta = \begin{cases} 0 & \text{if } L_j \neq G \\ 1 & \text{if } L_j = G \end{cases}$

$$W_k = \max(w_1, w_2, \cdots, w_q)$$

where W_k is the highest weight in category list, let that category be K. The final category C for the page are decided by equation below:

$$C = \begin{cases} K & \text{if } W_k \geq \tau \\ P & \text{if } W_k < \tau \end{cases}$$

where K is the category having the highest weight in category list, and P is the page category. The τ is a threshold that determines how much influence the linked information will affect the page category. It is obtained empirically based on results of category similarity matrix (Table 1).

3 Experiments and Performance

3.1 Experiments

We experiment using 18 categories (Table 2) and a pre-classified training set of 2000 pages selected from top-level Yahoo. Using these pages we build a term frequency

Table 1. Category Similarity Matrix

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
A	1	.58	.85	.30	.44	.49	.41	.33	.27	.21	.47	.66	.73	.30	.55	.30	.61	.39
B	.58	1	.60	.19	.30	.39	.29	.32	.28	.19	.35	.62	.36	.19	.57	.29	.49	.24
C	.85	.60	1	.61	.49	.74	.62	.47	.24	.23	.58	.70	.80	.41	.82	.62	.76	.46
D	.30	.19	.61	1	.33	.52	.49	.28	.03	0	.25	.35	.48	.31	.65	.65	.53	.24
E	.44	.30	.49	.33	1	.43	.33	.53	.37	.36	.37	.65	.50	.34	.53	.42	.61	.45
F	.49	.39	.74	.52	.43	1	.58	.60	.28	.18	.71	.64	.73	.42	.71	.61	.81	.44
G	.41	.29	.62	.49	.33	.58	1	.47	.16	.10	.38	.52	.54	.38	.69	.54	.72	.29
H	.33	.32	.47	.28	.53	.60	.47	1	.55	.42	.62	.83	.45	.59	.56	.52	.80	.46
I	.27	.28	.24	.03	.37	.28	.16	.55	1	.48	.39	.74	.24	.30	.34	.15	.51	.40
J	.21	.19	.23	0	.36	.18	.10	.42	.48	1	.37	.60	.19	.27	.25	.11	.42	.31
K	.47	.35	.58	.25	.37	.71	.38	.62	.39	.37	1	.71	.50	.40	.46	.35	.70	.48
L	.66	.62	.70	.35	.65	.64	.52	.83	.74	.60	.71	1	.68	.56	.70	.47	.87	.79
M	.73	.36	.80	.48	.50	.73	.54	.45	.24	.19	.50	.68	1	.39	.67	.46	.73	.46
N	.30	.19	.41	.31	.34	.42	.38	.59	.30	.27	.40	.56	.39	1	.40	.34	63	.32
O	.55	.57	.82	.65	.53	.71	.69	.56	.34	.25	.46	.70	.67	.40	1	.77	.74	.48
P	.30	.29	.62	.65	.42	.61	.54	.52	.15	.11	.35	.47	.46	.34	.77	1	.66	.36
Q	.61	.49	.76	.53	.61	.81	.72	.80	.51	.42	.70	.87	.73	.63	.74	.66	1	.56
R	.39	.24	.46	.24	.45	.44	.29	.46	.40	.31	.48	.79	.46	.32	.48	.36	.56	1

Table 2. Top 18 Categories

Letter	Category	Letter	Category	Letter	Category
A	Companies	G	Health	M	Regional
B	Computers	H	Humanities	N	Religion
C	Economy	I	Movies TV	O	Science
D	Education	J	Music	P	Social Science
E	Fine Arts	K	News and Media	Q	Society & Culture
F	Government	L	Recreation	R	Sports

vector to represent each of the categories. To categorize a new Web page and its linked pages we stripped the HTML tags, compute the word frequency vector of the pages, and calculate the similarities of the page and the categories. In order to speed up the classification process, we pre-calculated the similarities of the top 18 categories and save them in a category similarity table (Table 1). The threshold we use in the experiment is 0.65.

3.2 Performance

We randomly selected 32 Web pages from Yahoo top-level categories. These pages contain total 195 linked pages, average 6.1 links per web page. All 227 pages were classified by page classifier and the PLP classifier. In addition, we tested a large number of pages as part of our personal Web search filter project reported elsewhere.

Chart 1 shows the results of correct category numbers against Yahoo document categories by using page only classification and by using PLP classification. Chart 2 shows the correct percentage rate by using page only classification and by using PLP classification.

By using page only classifier to classify a web page, the correct percentage rate against Yahoo is 56%. Using the proposed PLP classifier, the correct percentage rate increase 35%. In this way, the correct percentage rate against Yahoo is 91%. These results show that the proposed PLP classifier could significantly increase the accuracy of automatic Web page classification. We anticipate if the proposed PLP classifier is used in conjunction with better page only classifier, the overall accuracy will improve further.

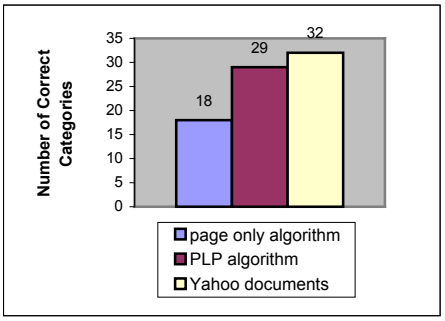

Chart 1

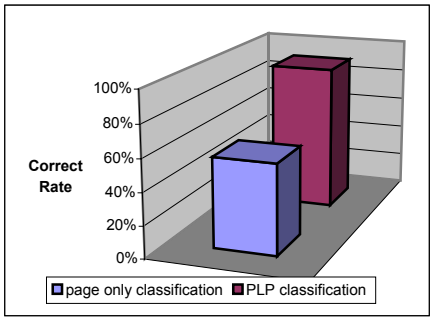

Chart 2

4 Conclusion

We described a new approach to automatically classify hypertext documents. The proposed approach exploits surrounding information extracted from analyzing linked documents. Our experimental results show that the proposed classifier could increase the overall accuracy of the classification by 35%. Much future research could be conducted on exploiting additional semantic information included in HTML documents.

5 References

1. G. Salton, A. Wong, and C.S. Yang, A Vector Space Model for Automatic Indexing, *Communications of the ACM*, 18, pp. 613–620, 1975
2. D. D. Lewis, R. E. Schapire, J. P. Callan, and R. Papka, Training Algorithms for Linear Text Classifiers, In SIGIR '96, Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 298-306, 1996
3. A. Bookstein and W.S. Cooper, A General Mathematical Model for Information Retrieval Systems, Library Quarterly, 46, pp. 153–167, 1976
4. E. Wiener, J.O. Pedersen, and A.S. Weigend, A Neural Network Approach to Topic Spotting, In Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95), 1995.
5. Joachims Thorsten, A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, Proceedings of International Conference on Machine Learning (ICML), 1997.
6. Y Yang and C. G. Chute, An Example-based Mapping Method for Text Categorization and Retrieval, *ACM Transaction on Information Systems* (TOIS), 12(3), pp.252-277, 1996.
7. Bulur v. Dasarathy, Nearest Neighbor (NN) Norms: NN pattern Classification Techniques, McGraw-Hill Computer Science Series. IEEE Computer society Press, Las Alamitos, California, 1991.
8. William W. Cohen and Yoram Singer, Context-sensitive learning methods for text categorization, In SIGIR 96: Proceedings of the 19th annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.307-315, 1996
9. J. G. Ganascia, Deriving the Learning Bias from Rule Prosperities, *Machine Intelligence* 12, pp. 151-167, Clarendon Press, Oxford, 1991.
10. C. Apte, F. Damerau, and S. M. Weiss, Automated Learning of Decision Rules for Text Categorization, *ACM Transactions on Information Systems*, 1994
11. B. Widrow and S. D. Stearns, *Information Retrieval*, Butterworths, London, Second edition, 1996
12. J. Kivinen and M. K. Kivinen, Worst-case Loss Bounds for Single Neurons, In Advances in Neural Information Processing System, In SIGIR'94, pp.192-201
13. William W. Cohen and Yoram Singer, Context-sensitive learning methods for text categorization, In SIGIR 96: Proceedings of the 19th annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996, 307-315
14. G Salton and C. Buckley, Term weighting approach in automatic text retrieval, *Information Processing and Management*, 24(5), pp. 513-523, 1988
15. Joachims Thorsten, A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, Proceedings of International Conference on Machine Learning (ICML), 1997.