

Classifying Web Pages Using Adaptive Ontology*

Sanguk Noh

School of Computer Science and
Information Engineering,
The Catholic University of Korea
sunoh@catholic.ac.kr

Haesung Seo, Jaehyuk Choi,

Kyunghee Choi, Gihyun Jung
Graduate School of Information and Communication,
Ajou University
{retry,brahms,khchoi,khchung}@ajou.ac.kr

Abstract – *In this paper, we present an automated Web page classifier based on adaptive ontology. As a first step, to identify the representative terms given a set of classes, we compute the product of term frequency and document frequency. Secondly, the information gain of each term prioritizes it based on the possibility of classification. We compile the selected terms and classification into rules using machine learning algorithms. The compiled rules classify any Web page into categories defined on a domain ontology. In the experiments, 11 terms out of 1,700 terms were identified as representative features given a set of Web pages. The resulting accuracy of the classification was, on the average, 95.2%.*

Keywords: Web classification, ontology, information gain, machine learning

1 Introduction

The Internet, the world's end-to-end communications network, is now ubiquitous in everyday's life. Many enterprises are concerned to detect and restrict inappropriate web pages on the Internet that may be considered irrelevant to their normal tasks for a variety of operational reasons [3]. From the operational perspective, the accesses to the materials considered inappropriate cause the loss of communications bandwidth and the waste of computational resources. We, therefore, identify these materials from the incoming streams in naïve network environments, and classify them into an unwanted category based on an adaptive ontology [11].

This paper presents a Web classification system, which consists of packet capturing module, protocol analyzing module, and web page classifying module. The packet capturing module collects packets from connections between servers and clients on a TCP-based network setting. The protocol analyzer examines header information and identifies the type of application that is supposed to send the packets, for example, Web application or Instant Messenger. The web page classifier

is to analyze the contents of web pages, and determine their classification.

To identify the contents of web pages, we propose a combined mechanism which computes the product of term frequency and document frequency [1, 16], and prioritizes the terms based on the calculation of entropies [18]. Based on a selected set of terms, we assign any Web page into a category within a domain ontology. Our approach to identifying associations between a web page and a predefined category is to use term-classification rules compiled by machine learning algorithms, which provide a high degree of relationship between them, if any.

For the web page classifier, our classifying agents build hierarchical descriptions of objects and features, i.e., ontology. The ontology keeps evolving as the applications expand or proceed. Infrequent updates to the components of the ontology lead to inaccurate classifications. The classifying agents which use data mining and data modeling techniques, therefore, continuously upgrade a list of keywords representing clusters of web pages, and maintain more up-to-date and more complete ontology. Using adaptive ontology, our agents enable the classification system to automatically add new unwanted sites whenever the agents capture the streams on communications network.

We wish our classification system could be used to classify current web pages into proper categories and to generate reports if the web pages belong to unwanted sites. We believe that our system is indispensable to any organization to assure communication network environments to be safe.

In the following section of this paper, we will discuss related approaches to our classifier. Section 3 describes the details of our framework for extracting representative terms, and Section 4 describes a system architecture of the classifier. We show the simulation results in Section 5. In conclusion we summarize our results and mention further research issues.

* This research has been supported by Korea Research Foundation under grant KRF-2002-041-D00465, and by NRL program of KISTEP, Korea.

2 Related Work

There have been several approaches which focus on ontologies to classify Web pages [7, 9, 12, 13]. Prabowo et al. [13] defined ontology as “a single entity which holds conceptual instances of a domain, and differentiates itself from another,” and used ontologies for Web page classification with respect to the Dewey Decimal Classification (DDC) and Library of Congress Classification (LCC). The weakness of their approach is the fact that it is not adaptive when users require more sophisticated classification, even if the approach follows the standard classifications. However, since our approach build an adaptive ontology, we provide a flexible classification reflecting user requests.

To address the question of how to select a reasonable size of features, Ruger and Gauch [15] set a threshold for the minimum and the maximum frequencies of terms, and select only the terms between them. Since their approach depends on the document frequency, they might use some terms, which are frequently found in a specific class of Web pages but cannot separate a class from other classes. However, our approach computes document frequency as well as term frequency to extract the representative and significant terms.

For the classification of Web pages, most of approaches use machine learning algorithms [17, 20, 21]. Sebastiani [17] summarize the role of machine learning algorithms in the field of text categorization (or text classification). Sebastiani pointed out that a strategy should be needed to reduce the dimension of features as inputs of machine learning algorithms. From our observation, it should be required to prove whether those selected features are significant, notable, or worthy to represent each class, since they didn't construct domain ontologies. Our approach shares the efforts to make the size of features manageable but suggests the combined mechanism of document/term frequency and entropy computation given an adaptive ontology for practical and efficient classification.

3 Ontology for classifying the Web Contents

We model a Web page using an ontology and classify Web pages based upon the ontology, as consisting of a hierarchy of concepts. Each class in the ontology contains a tapestry of representing terms. To define the ontology given a domain, therefore, we describe how to select a set of the terms related to the class.

3.1 Feature Extraction

The first step to building domain ontologies is to identify a proper set of terms which represent a specific

class. Intuitively, the representatives of the terms could be explained if a specific term frequently occurs in a Web page and if the term is found in every Web page belonging to a specific class. For example, a domain ontology contains a class “Banking and Finance,” and a term ‘bank’ might be present whenever a Web page is classified into the “Banking and Finance” class. Thus, the ‘bank’ can be a representing term in the “Banking and Finance” class because the term ‘bank’ covers the “Banking and Finance” Web pages.

Let I be the set of terms in a specific Web page, i.e., $\{1, 2, \dots, m\}$, and let J be the set of Web pages which are classified into a specific class, i.e., $\{1, 2, \dots, n\}$. From the perspective of information retrieval [1, 16], Term Frequency (TF) and Document Frequency (DF) are combined as follows:

$$W_{i,j} = \frac{TF_{i,j}}{\max_{k \in I} TF_{k,j}} \times \frac{DF_i}{n} \quad (1)$$

where,

- $W_{i,j}$ is the weight of the term $i \in I$ in the Web page $j \in J$;
- $TF_{i,j}$ is the frequency of the term $i \in I$ in the Web page $j \in J$;
- DF_i is the number of Web pages which the term i occurs.

The above formula (1) indicates that, if a specific term more frequently occurs than other terms within a Web page, and if the term can be found among most Web pages, the weight of the term will be greater than those of the other terms. The bigger weight of the term means that its representativeness outweighs others. By computing the weights of terms, the terms relevant to a category can be prioritized. The highly representing terms should be used to denote a class in an ontological hierarchy.

3.2 Computing Entropies

Having the terms related to classes, we classify any Web page into one of the classes, which are the components of taxonomy. For the classification of a Web page, we compute the entropies of the terms, originally reported in [18]. The entropy of a word (or a term) provides the expected amount of information for correct classification. The lower entropy needs the less information to classify an example Web page. For the efficient classification, therefore, our approach uses the

attributes which have lower entropies. The computation of entropy can be achieved using the following formula [18]:

$$E^S = - \sum_{i=1}^n p_i \log_2(p_i) \quad (2)$$

where,

- E^S is the entropy of the set S of class probabilities $p_1, \dots, p_n$.

For example, when a set consists of two classes of positive and negative, we assume that the ratio of the number of positive examples to the total number of the set is the class probability, say, p_1 , and, similarly, the other ratio is p_2 . The entropy, then, can be computed by $-(p_1 \log_2 p_1 + p_2 \log_2 p_2)$.

Given entropy, we compute information gain which represents the classifying possibility of an attribute, as follows:

$$Gain(\alpha) = E^S - \sum_{j=1}^m \left(\frac{|S_{\alpha_j}|}{|S|} \times E^{S_{\alpha_j}} \right) \quad (3)$$

where,

- $Gain(\alpha)$ is the information gain of an attribute α ;
- S_{α_j} is the set of attribute α whose value is $j \in \{1, \dots, m\}$.

In our framework, attributes present terms within a class. The computed values of gain factors of attributes rank the terms as the possibility of the classification. We choose the terms based upon their information gain.

3.3 Constructing Attributes

To classify a Web page, we need to assemble attributes into a tuple. Let A be a set of attributes. The selection of attributes can be achieved using the following heuristic rule:

$$H_\mu = \{ \alpha | \alpha \in A, Gain(\alpha) \geq \mu \} \quad (4)$$

The threshold μ is to filter out insignificant terms within each class. The resulting H_μ is a set of attributes

which are highly related to the class, based on information gain.

For the automated classification of Web pages, firstly, the attributes are prioritized using the computation of TF-IDF and information gain. Secondly, we compile the tuples obtained into a set of terms-classification rules using several machine learning algorithms [5, 10]. The compiled rules enable us to classify any Web page into a category defined within a domain ontology.

4 Architecture for the Classifier

In this section, we describe system architecture, as depicted in Figure 1, for a Web page classification. Our system consists of three modules: packet capturing module, protocol analyzing module, and contents classifying module. The packet capturing module is to capture the stream of network traffic using libpcap [8] during the connection. The protocol analyzing module is to analyze the packets, separate payloads from the packets, extract readable texts, and ignore multi-media data, binary or compressed data. Using our techniques described in Section 3, the resulting terms-classification rules are obtained. The contents classifying module is then equipped with the compiled rules, and classifies any Web page given a domain ontology.

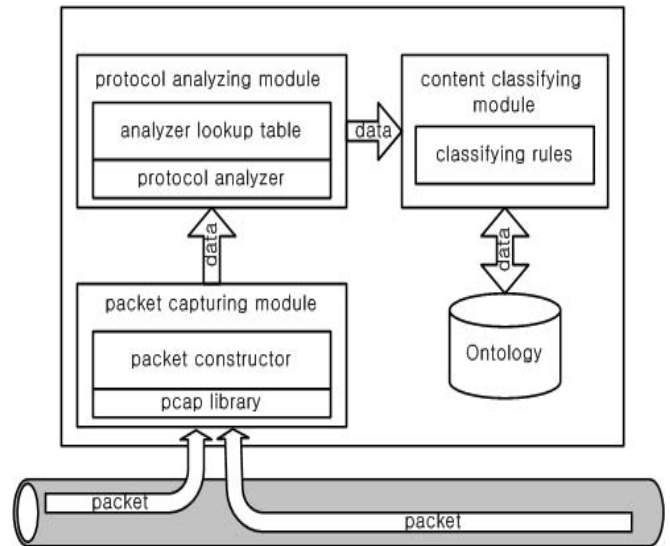


Figure 1. The architecture of Web page classifier

5 Experimental Results

We test the automated Web page classifier using the data set provided by Sinka et al. [19], and measure its performance in terms of classification accuracy. We have implemented the Web page classifier on a 1.6 GHz, Pentium IV PC platform.

The ontology for the data set, as depicted in Figure 2, shows the hierarchy of four themes, which are Banking and Finance, Programming Language, Science, and Sport. For our experiments, one subclass of each theme is chosen. For each theme, the subclasses linked with thick lines in Figure 2 are ‘Commercial Banks,’ ‘Java,’ ‘Astronomy,’ and ‘Soccer,’ respectively.

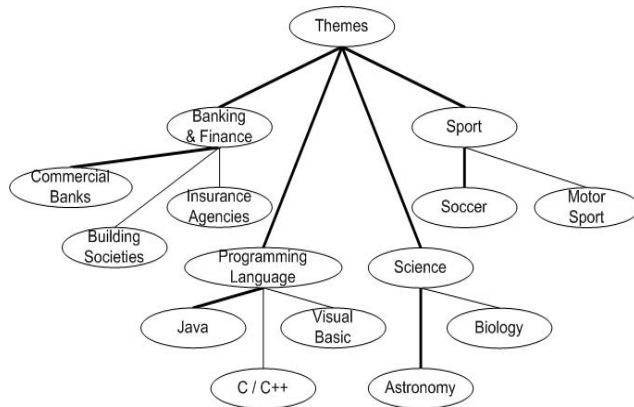


Figure 2. Ontology for the four themes: Banking and Finance, Programming Languages, Science, and Sport

To identify the representing terms for each class, the formula (1) was applied to the data set. The Web pages contained about 1,700 terms, on the average, and the resulting weights of the terms were computed, as described in Table 1 and 2. For instance, ‘bank’ for Banking and Finance theme or ‘java’ for Programming Language theme were representative in each class, respectively.

After having significant terms related to a specific class, we could prioritize the terms based on information gain using the formula (3). We intended to know what attribute could be a reasonably good term to classify a Web page. The resulting classifying powers of the terms are described in Table 3.

Using the formula (4), we selected the terms whose gain factors were greater than the threshold, μ , 0.20. Given the domain ontology, as shown in Figure 2, we could choose 45 terms, as representative ones, out of 6,800 ($= 4 \times 1700$) terms. Finally, a tuple consisted of the 45 terms and a corresponding class, which was one of the four themes. The computation of entropies enables us to reduce the dimension of resulting terms.

Table 1. The weights of the terms for Banking & Finance and Programming Language themes

Themes			
Banking & Finance		Programming Language	
Term	Weight	Term	Weight
bank	0.8681	java	0.9397
servic	0.7710	new	0.8955
inform	0.7570	code	0.8140
person	0.7391	us	0.7846
account	0.7346	time	0.7779
avail	0.7268	com	0.7708
rate	0.7219	class	0.7701
home	0.7087	program	0.7422
interest	0.6641	get	0.7285
year	0.6602	see	0.7285
busi	0.6420	mai	0.7121
credit	0.6405	creat	0.7121
detail	0.6320	page	0.7103
save	0.6219	look	0.6920
contact	0.6219	includ	0.6860

Table 2. The weights of the terms for Science and Sport themes

Themes			
Science		Sport	
Term	Weight	Term	Weight
time	0.7500	footbal	0.7293
observ	0.7472	team	0.7257
univers	0.7404	cup	0.7193
space	0.7203	england	0.6585
new	0.7203	new	0.6134
gener	0.6955	world	0.6081
star	0.6906	game	0.5832
larg	0.6842	club	0.5492
like	0.6726	match	0.5438
mai	0.6711	unit	0.5346
see	0.6704	citi	0.5331
year	0.6663	leagu	0.5191
light	0.6607	home	0.5126
galaxi	0.6603	st	0.5020
data	0.6578	result	0.5006

Table 3. Gain factors of the terms

Themes			
Banking	Programming	Science	Sport
bank:0.40	java:0.57	galaxi:0.38	footbal:0.39
insur:0.28	file:0.27	telescop:0.31	cup:0.38
plc:0.26	server:0.25	astronom:0.31	leagu:0.23
mortgag:0.26	class:0.24	astronomi:0.29	liverpool:0.22
loan:0.25	program:0.23	scienc:0.29	aston:0.22
payment:0.24	user:0.22	univers:0.27	tottenham:0.22
monthli:0.21	code:0.21	observ:0.26	chelsea:0.21
deposit:0.21	http:0.20	measur:0.25	arsen:0.21
account:0.20	window:0.20	light:0.24	villa:0.20
financi:0.20	html:0.20	space:0.23	rover:0.19
tax:0.20	text:0.15	sky:0.23	club:0.18
cash:0.20	creat:0.15	earth:0.22	ranger:0.18
branch:0.20	download:0.14	star:0.21	everton:0.18
invest:0.19	sourc:0.13	mass:0.18	leed:0.17
paid:0.19	england:0.13	scale:0.17	fc:0.17

To implement the automated Web page classifier, we compiled a set of training tuples into terms-classification rules using C4.5 [14], naïve Bayesian classifier [4], CN2 [2], and backpropagation [22] learning algorithms. The compiled rules were used to classify any Web page during online.

We applied the resulting compiled rules to 3,600 Web pages classified into one of four categories, as shown in Figure 2, and measured their performances in terms of classification accuracy. The performance results using four compiled rules were depicted in Figure 3. The performance of CN2, 95.94%, was the best while the performance of naïve Bayesian classifier, 94.08%, was the worst. Thus, we could classify 3,600 Web pages using compiled rules and their average performance was 95.2%.

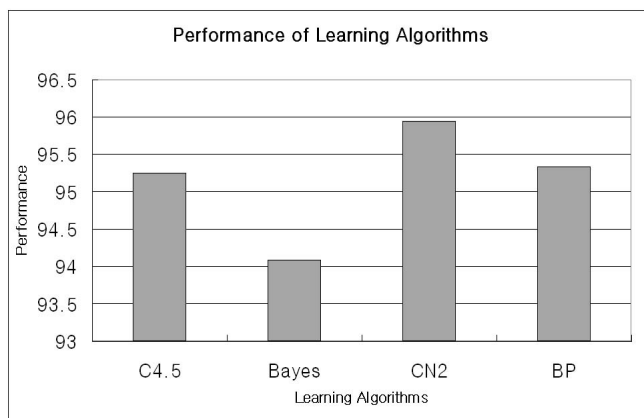


Figure 3. Performance of Learning Algorithms

6 Conclusions

In this paper, we have presented the automated Web page classifier based on adaptive ontology. To extract significant, notable features from a set of terms, we computed the weights of the terms, and further calculated the information gain of the features for ranking their consequences. Given a set of selected terms, classifying rules were compiled using machine learning algorithms.

In the experiments, we tested our Web page classifier and measured its performance in terms of classification accuracy. It turned out that our combined approach to reduce the dimension of terms was very useful. We could observe that 11 terms out of 1,700 terms, on the average, were identified as representative features given a set of classified Web pages. Finally, the resulting accuracy of the classification was, on the average, 95.2%.

For future research, we will investigate the performances when the threshold μ increases. We will figure out the tradeoff between the number of significant terms and the quality of classification. We will apply our current work to various domains [6]. In particular, we already collected more than 30,000 Web pages, which can be classified into harmful Web sites. In further experiments, therefore, our Web page classifier will report whether or not any Web page is harmful.

References

- [1] R. Baeza-Yates, and B. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press/Addison-Wesley, New York, 1999.
- [2] P. Clark, and T. Niblett, "The CN2 Induction algorithm", *Machine Learning Journal*, Vol 3, No.4, pp. 261-283, 1989.
- [3] C. Ding, C. Chi, J. Deng, and C. Dong, "Centralized content-based Web filtering and blocking: how far can it go?", *Proc. of 1999 IEEE International Conference on Systems, Man and Cybernetics*, Vol 2, pp. 115-119, October 1999.
- [4] R. Hanson, J. Stutz, and P. Cheeseman, *Bayesian Classification Theory*, Technical Report FIA-90-12-7-01, NASA Ames research Center, AI Branch, 1991.
- [5] L. Holder, *ML v2.0, Machine Learning Program Evaluator*, available on-line, <http://ranger.uta.edu/~holder/courses/cse6363/ml2.0.tar.gz>.
- [6] K. Huang, and S. Ma, "Chinese web page classification based on statistical word segmentation", *Proc. of 2002 IEEE International Conference on Systems Man and Cybernetics*, Vol 1, pp. 636-640, October 2002.

- [7] C. Jenkins, M. Jackson, P. Burden, and J. Wallis, "Automatic RDF metadata generation for resource discovery", Proc. of 8th International WWW Conference, Toronto, pp. 11-14, May 1999.
- [8] Lawrence Berkeley National Labs Network Research Group, *libpcap*, available on-line, <http://ftp.ee.lbl.gov>.
- [9] Y. Ng, J. Tang, and M. Goodrich, "A binary-categorization approach for classifying multiple-record Web documents using application ontologies and a probabilistic model", Proc. of 7th International Conference on Database Systems for Advanced Applications, pp. 58-65, April 2001.
- [10] S. Noh, C. Lee, K. Choi, and G. Jung, "Detecting Distributed Denial of Service (DDoS) Attacks Through Inductive Learning", To appear in *Lecture Notes in Computer Science*, 2003.
- [11] N. F. Noy, and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology", *Knowledge Systems Laboratory (KSL), Department of Computer Science, Stanford: Technical report*, KSL-01-05, 2001.
- [12] S. Parent, B. Mobasher, and S. Lytinen, "An adaptive agent for web exploration based on concept hierarchies", Proc. of 9th International Conference on Human Computer Interaction, New Orleans, August 2001.
- [13] R. Prabowo, M. Jackson, P. Burden, and H. Knoell, "Ontology-Based Automatic Classification for the WEB Pages: Design, Implementation and Evaluation", Proc. of 3rd International Conference, Singapore, pp. 182-191, 2002.
- [14] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
- [15] S. M. Ruge, and S. E. Gauch, *Feature Reduction for Document Clustering and Classification*, Technical report, Computing Department, Imperial College, London, 2000.
- [16] G. Salton, and C. Buckley, "Term weighting approaches in automatic text retrieval", *Information Processing and Management*, Vol 24, No. 5, pp. 513-523, 1988.
- [17] F. Sebastiani, "Machine learning in automated text categorization", *ACM Computing Surveys*, Vol 34, No. 1, pp. 1-47, 2002.
- [18] C. E. Shannon, "A mathematical theory of communication", *Bell System Technical Journal*, Vol 27, pp. 379-423 and 623-656, July and October 1948.
- [19] M. P. Sinka, and D. W. Corne, "A large benchmark dataset for web document clustering", *Soft Computing Systems: Design, Management and Applications, Frontiers in Artificial Intelligence and Applications*, Vol 87, pp. 881-890, 2002.
- [20] N. Soonthornphisaj, P. Chartbanchachai, T. Pratheeptham, and B. Kijsirikul, "Web page categorization using hierarchical headings structure", Proc. of 24th International Conference on Information Technology Interfaces, Vol 1, pp. 37-42, 2002.
- [21] A. Sun, E. Lim and W. Ng, "Web classification using support vector machine", WIDM'02, Virginia, November 2002.
- [22] D. R. Tsvet, *Backprop Package*, available on-line, <http://www.dontvetter.com/nnsoft/bp042796.zip>, 1996.