An Intelligent Topic-Specific Crawler Using Degree of Relevance*

Sanguk Noh[1], Youngsoo Choi[2], Haesung Seo[2], Kyunghee Choi[2], Gihyun Jung[3]

[1]School of Computer Science and Information Engineering,
The Catholic University of Korea, Bucheon, Korea
sunoh@catholic.ac.kr
[2] Graduate School of Information and Communication,
Ajou University, Suwon, Korea
{drabble, retry, khchoi}@ajou.ac.kr
[3] Division of Electronics Engineering,
Ajou University, Suwon, Korea
khchung@ajou.ac.kr

Abstract. It is indispensable that the users surfing on the Internet could have web pages classified into a given topic as correct as possible. Toward this ends, this paper presents a topic-specific crawler computing the degree of relevance and refining the preliminary set of related web pages using term frequency/ document frequency, entropy, and compiled rules. In the experiments, we test our topic-specific crawler in terms of the accuracy of its classification, the crawling efficiency, and the crawling consistency. In case of using 51 representative terms, it turned out that the resulting accuracy of the classification was 97.8%.

1 Introduction

The Internet, the world's end-to-end communications network, is now ubiquitous in everyday's life. It would be necessary for users surfing on the Internet to access a pool of web pages that could be relevant to their interesting topics. It is crucial that the users could have not only the web pages probably related to a specific topic as many as possible, but also the web pages classified into the given topic as correct as possible. This paper, therefore, presents (1) a topic-specific crawler which collects a tapestry of web pages through the calculation of degree of relevance, and (2) a web page classification mechanism to refine the preliminary set of web pages using term frequency/document frequency, the computation of entropy, and the compiled rules.

To crawl any web pages related to a specific topic [1, 3, 5], we use the degree of relevance, which represents how much a web page could be relevant to the topic. For the computation of the degree of relevance, we exploit the relationship between any web page and web pages linked within the web page, and the number of keywords

* This work has been supported by the Korea Research Foundation under grant KRF-2003-041-D20465, and by the KISTEP under National Research Laboratory program.

shown in the web page, compared with the predefined set of key phrases. To identify the contents of web pages [9, 12], we propose a combined mechanism which computes the product of term frequency and document frequency [6, 8] and prioritizes the terms based on the calculation of entropies [10]. Based on a selected set of terms, we assign any web page into a topic (or category). 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. We wish our topic-specific crawler could be used to collect any web pages probably related to a certain topic, and refine them into the category to improve the correctness of the classification.

In the following section of this paper, we will describe the details of our framework for computing the degree of relevance to a specific topic and the weights of terms representing a specific web page. Section 3 describes experimental results to evaluate our topic-specific crawler using benchmark dataset as a case study. In conclusions we summary our work and mention further research issues.

2 An Intelligent Topic-Specific Crawler

2.1 Crawling Strategy

To crawl any web pages related to a specific topic, we need to decide the degree of relevance, which represents how much a web page could be relevant to the topic. For the computation of the degree of relevance, in this paper, we consider the relationship between any web page and web pages linked within the web page, and the number of keywords shown in the web page, given the predefined set of key phrases. The degree of relevance, R_i, of the web page i, can be defined as follows:

$$R_i = (1-\rho)\ \lambda_i / |K| + \rho R_j\ , \qquad (1)$$

where

- R_j denotes the degree of relevance for the web page j, containing the URL of web page i, which is already crawled;
- ρ is a constant reflecting how much R_i could be affected by R_j ($0 < \rho < 1$);
- λ_i is the number of keywords shown in a given web page i;
- K is the pre-defined set of key phrases. $|K|$ is the cardinality of K.

Our topic-specific crawler provides a priority queue with two operations, i.e., enqueue and dequeue, to handle URLs probably related to a certain topic. Let's suppose that there is an URL which might be representative for a given topic. As starting with the URL in the priority queue, the crawler dequeues it from the queue and fetches its contents. Our crawler computes the degree of relevance R_i for the hyperlinks in the web page i. In case that the hostname of the newly found hyperlink is the same of the web page i, the hyperlink cannot be enqueued into the queue and is simply disregarded. According to R_i, then, the hyperlinks whose hostnames are different from that of web page i can be prioritized into the queue. To crawl further web pages, the crawler de-

queues the hyperlinks from the priority queue given their degree of relevance. The above crawling process will be continued until the queue is empty.

2.2 Classifying Web Pages

To refine the set of possibly related web pages, which are collected by our topic-specific crawler, we propose an online web page classification mechanism [6]. Let I be the set of terms in a specific web page, i.e., {1, 2, …, m}, and let J be the set of web pages which are classified into a specific class, i.e., {1, 2, …, n}. From the perspective of information retrieval [8], Term Frequency (TF) and Document Frequency (DF) are combined as follows:

$$W_{i,j} = \frac{TF_{i,j}}{\max_{k \in I} TF_{k,j}} \times \frac{DF_i}{n}, \tag{2}$$

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;
- n is the total number of web pages for a specific class.

The above equation 2 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. By computing the weights of terms, the terms relevant to a topic can be prioritized. The highly representative terms could be used to denote a class in a hierarchy of concepts.

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 [10]. 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 [10]:

$$E_S = -\sum_{i=1}^{n} p_i \ln p_i, \tag{3}$$

where

- S is the set of web pages in our framework, which can be classified into a number of classes n whose probabilities are $p_1, \ldots, p_n$, respectively;
- E_S is the entropy of the set S, representing the expected amount of information to classify the web pages in the set S into a specific class.

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 examples in the set is the class probability, say, p_1, and, similarly, the other ratio is p_2. The entropy of the set S, then, can be computed by $-(p_1 \ln p_1 + p_2 \ln p_2)$.

Given entropy, we compute information gain, $Gain(\alpha)$, which represents the classifying possibility of an attribute α, as follows:

$$Gain(\alpha) = E_S - \sum_{j=1}^{m} \left(\frac{|S_{\alpha_j}|}{|S|} \times E_{S_{\alpha_j}} \right), \tag{4}$$

where

- S_{α_j}, $S_{\alpha_j} \subseteq S$, is the set of web pages containing the attribute α whose value is $j \in \{1, \ldots, m\}$. $|S_{\alpha_j}|$ is the total number of web pages in S_{α_j}.

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 representative terms based upon their information gain.

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:

$$H_\mu = \{\alpha | \alpha \in A, Gain(\alpha) \geq \mu\}. \tag{5}$$

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-DF and information gain. Secondly, we compile the tuples obtained into a set of terms-classification rules using several machine learning algorithms [4]. The compiled rules enable us to classify any web page into a topic.

3 Evaluation

We tested our topic-specific crawler in terms of (1) the accuracy of its classification, (2) the crawling efficiency, and (3) the crawling consistency. In the experiments, we used the benchmark dataset provided by Sinka et al. [11], which consists of Banking and Finance, Programming Language, Science, and Sport. In particular, one subclass of each theme was chosen, i.e., 'Commercial Banks,' 'Java,' 'Astronomy,' and 'Soccer,' respectively.

Regarding the classification accuracy of our crawler, we needed to determine the proper number of terms through the computation of term-frequency/document-frequency and entropy, and measured the classification accuracy using the terms selected and the rules compiled by machine learning algorithms. First, we randomly

chose 100 web pages from each category, thus, 400 web pages total. Given the 100 web pages of each category, the equations 2 and 4 were applied to them to identify the representative and discriminative terms. There were about 241 terms, on the average, per web page and the weights of the terms were computed using TF-DF and entropy.

To determine the terms as inputs of inductive learning algorithms, we applied the equation 5 to the terms, and selected the representative terms whose gain factors were greater than the threshold μ. In the experiment, we changed the thresholds ranging from 0.01 to 0.31, and then, measured the accuracy of terms-classification rules using 3600 web pages which were not used for training purposes. We compiled a set of training tuples into terms-classification rules using C4.5 [7], CN2 [2], and backpropagation [13] learning algorithms. The performances of three machine learning algorithms were measured in terms of classification accuracy, as depicted in figure 1.

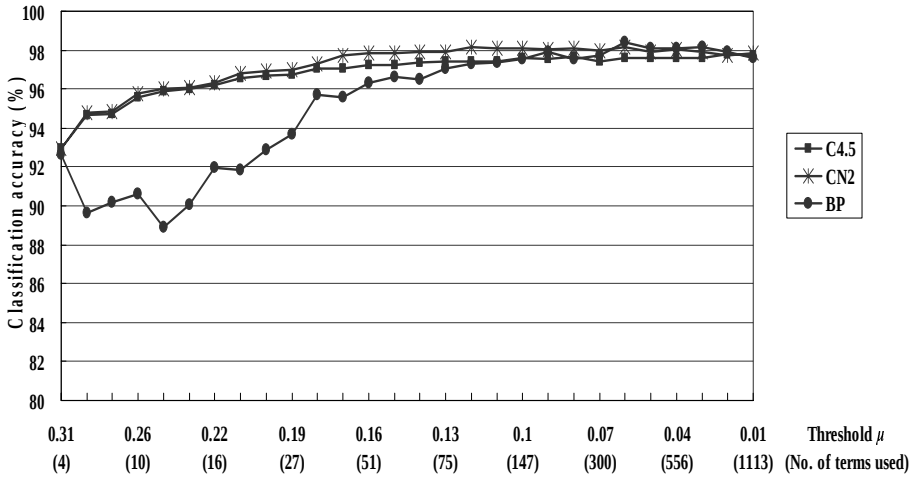

Fig. 1. Classification accuracies of three learning algorithms according to the changes of threshold μ. In horizontal axis, the number of terms as inputs of three learning algorithms was also given below the threshold μ ranging from 0.01 to 0.31.

As the threshold μ decreased, the number of representative terms increased, and the accuracy of classification approximately went up. In figure 1, we could observe the tradeoff between the number of significant terms and the quality of classification. As a result, for the classification accuracy of, say, 97.8%, we could choose 51 terms, as representative ones, out of about two million unique terms, and construct the hierarchy of concepts with these terms. The computation of entropies, thus, enables us to reduce the dimension of resulting terms.

We define the crawling efficiency as a ratio of (the number of URLs related to a specific topic/the total number of URLs crawled). The crawling efficiency enables us to determine the best ρ affecting the degree of relevance, R_i, as described in equation 1. For our topic-specific crawler, we selected 20 terms as the elements of the predefined set of key phrases K, given the resulting classification accuracy of figure 1, and randomly chose ten starting URLs as the seeds of crawling process for each cate-

gory. Since it turned out that CN2 [2] provide the best classification accuracy from figure 1, we used the terms-classification rules compiled by CN2. Using the compiled rules, we measured the crawling efficiency with ρ varying from 0.1 to 0.9 by the interval 0.2, as depicted in figure 2, when the total number of URLs crawled was 10,000.

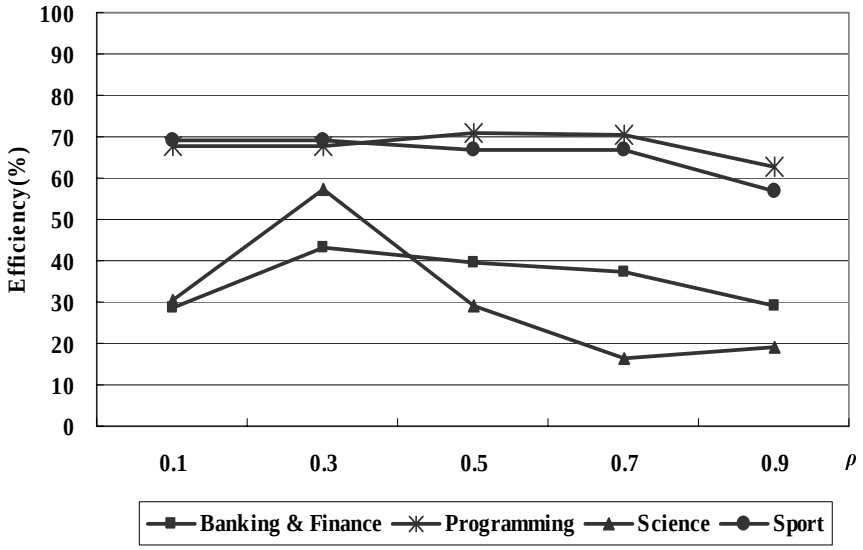

Fig. 2. The efficiency of our topic-specific crawler for four categories, when ρ ranges from 0.1 to 0.9, respectively.

When ρ was 0.1 and 0.9, respectively, the crawling efficiency of our crawler was not good in almost all of four categories. Since the fact that ρ was 0.1 indicated that the degree of relevance for the web page *i* was little affected by that of the web page *j*, the web page *i* belonging to the topic could not be fetched earlier than others. In case that ρ was 0.9, since the degree of relevance for the web page *i* definitely depended on the degree of relevance for the web page *j*, the web page *i* not being related to the topic could be fetched earlier than others. In the experiment, it turned out that the optimal ρ was 0.3 for banking & finance and its efficiency 43% was the best, as shown in figure 2. We could also decide the optimal ρ*'s* for Programming Language, Science, and Sport, namely, 0.5, 0.3, and 0.3, and those crawling efficiencies were 71%, 57%, and 69%, respectively.

In the third experiment, the consistency of our topic-specific crawler was measured in terms of the number of the resulting URLs overlapped [1]. Using our topic-specific crawler with different starting URLs, two sets of URLs crawled were compared to calculate the percentage of URLs overlapped. Our topic-specific crawler, of course, was equipped with classification rules compiled by CN2, as was shown in figure 1,

and the optimal ρ's, as was shown in figure 2. The consistency of our topic-specific crawler, thus, could be summarized into figure 3.

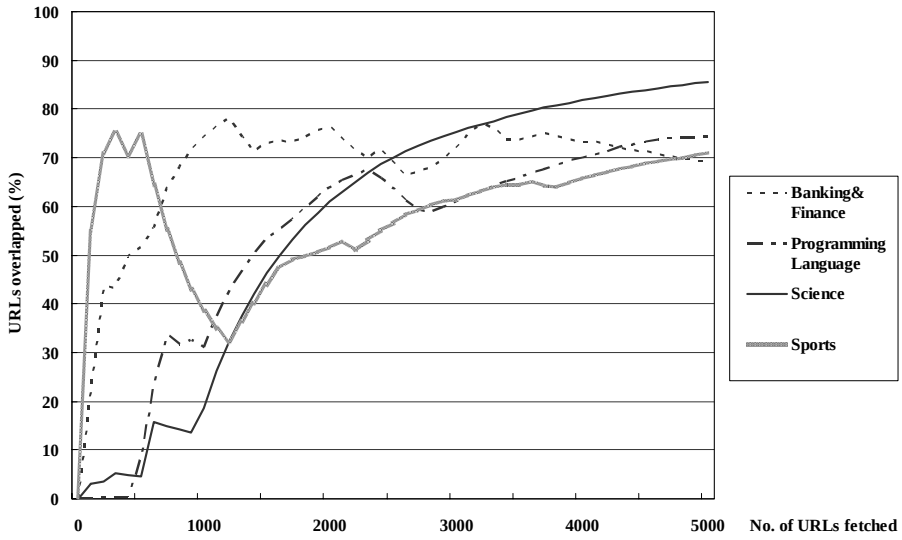

Fig. 3. The crawling consistency of our topic-specific crawler in four categories, when it fetched 5000 URLs total for each category.

For all of four categories, the percentage of the URLs overlapped was above 70%, up to 85%. The experimental results imply that our topic-specific crawler is fairly consistent, regardless of the starting URLs randomly chosen. We wish our topic-specific crawler could be consistent for any other topic.

4 Conclusions

To collect any web pages probably related to a certain topic, we exploited the degree of relevance considering the relationship between any web page and web pages linked within the web page, and the number of keywords found in the web page. To refine the set of possibly related web pages, which were collected by our topic-specific crawler, further, we calculated term frequency/document frequency and entropy for representative terms. Using inductive learning algorithms and a neural network algorithm, we compiled the tuples obtained into a set of terms-classification rules. In the experiments, the best classification performance for our topic-specific crawler was achieved when CN2 was used for compilation, and its classification accuracy using the compiled rules was 97.8% with 51 representative terms. We also measured the crawling efficiency, which enables us to determine the best ρ affecting the degree of relevance. Lastly, to benchmark our topic-specific crawler within our framework, its consistency was tested with different starting URLs. It turned out that the topic-specific

crawler was fairly consistent, given the resulting URLs overlapped. For future research, we will expand our crawler to collect related web pages as broadly as possible, and to provide its best performances given a specific topic. We will continuously apply our crawler to various topics and test it in a real network infrastructure on the Internet.

References

1. Chakrabarti, S. et al.: Focused crawling: a new approach to topic-specific Web resource discovery. In Proceedings of 8th International World Wide Web Conference (1999).
2. Clark, P. and Niblett, T.: The CN2 Induction algorithm. *Machine Learning Journal*, Vol. 3, No.4 (1989) 261-283.
3. Diligenti, M. et al: Focused crawling using context graphs. In Proceedings of VLDB (2000) 527-534.
4. Holder, L.: ML v2.0, available on-line: http://www-cse.uta.edu/~holder/ftp/ml2.0.tar.gz.
5. Menczer, F. et al.: Topic-driven crawlers: Machine learning issues. ACM TOIT (2002).
6. Noh, S. et al.: Classifying Web Pages Using Adaptive Ontology. In Proceedings of IEEE International Conference on Systems, Men & Cybernetics (2003) 2144-2149.
7. Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann. (1993).
8. Salton, G. and Buckley, C.: Term weighting approaches in automatic text retrieval. *Information Processing and Management*, Vol 24, No. 5 (1988) 513-523.
9. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys*, Vol 34, No. 1 (2002) 1-47.
10. Shannon, C. E.: A mathematical theory of communication. *Bell System Technical Journal*, Vol 27, (1984) pp. 379-423 and 623-656.
11. Sinka, M. P. and Corne, D. W.: A large benchmark dataset for web document clustering, *Soft Computing Systems: Design, Management and Applications*, Vol 87, IOS Press (2002) 881-890.
12. Sun, A., Lim, E. and Ng, W.: Web classification using support vector machine, In Proceedings of WIDM (2002).
13. Tveter, D. R.: Backprop Package, http://www.dontveter.com/nnsoft/bp042796.zip (1996).