Focused Crawling by Exploiting Anchor Text Using Decision Tree

Jun Li *
Department of General System Studies
The University of Tokyo
jun@graco.c.u-tokyo.ac.jp

Kazutaka Furuse
Institute of Information Sciences & Electronics
University of Tsukuba
furuse@cs.tsukuba.ac.jp

Kazunori Yamaguchi
Information Technology Center
The University of Tokyo
yamaguch@mail.ecc.u-tokyo.ac.jp

ABSTRACT

Focused crawlers are considered as a promising way to tackle the scalability problem of topic-oriented or personalized search engines. To design a focused crawler, the choice of strategy for prioritizing unvisited URLs is crucial. In this paper, we propose a method using a decision tree on anchor texts of hyperlinks. We conducted experiments on the real data sets of four Japanese universities and verified our approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*search process*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*graph and tree search strategies*

General Terms

Algorithms, Experimentation, Performance

Keywords

Focused Crawling, Anchor Text, Decision Tree Learning, Shortest Path

1. INTRODUCTION

Recently, topic-oriented search engines and personalized searching tools are getting popular. Unlike general-purpose search engines, these applications only need to crawl relevant pages from the WWW. Focused crawlers, which fetch relevant pages efficiently, were proposed in recent literature such as [1, 2]. To design a focused crawler, the choice of strategy for prioritizing unvisited URLs is crucial. In this paper, we propose a method to utilize anchor texts for determining the priorities. Our approach is motivated by the following two observations: (1) In many cases, anchor texts on hyperlinks are good summaries on the target pages; (2) Other methods, which are used in the conventional search engines and focused crawlers, tend to underestimate low in-degree pages, therefore miss low in-degree relevant pages.

*Contact User: #703, #15 Bldg, Komaba 3-8-1 Meguro-Ku, Tokyo, Japan. Zip: 153-8902.

WWW 2005, May 10–14, 2005, Chiba, Japan.
ACM 1-59593-051-5/05/0005.

2. METHODOLOGIES AND ALGORITHMS

2.1 Assumptions

We have two assumptions about the web space in which our crawler is designed for crawling. First, we assume that our crawler is crawling in a limited URL domain, e.g., the web site(s) of a university or a company. Second, we assume that there exists an entry page to the URL domain, e.g., the home page of a university. So, our crawler is supposed to crawl in the limited URL domain starting from the entry page. In the following, we let $G = (V, E, r)$ denote the web graph of a limited URL domain, where V is the set of web pages, E is the set of hyperlinks between these web pages, and r is the entry page.

2.2 Modeling Anchor Text Using Decision Tree

The number of terms in an instance of anchor text is small compared to that of the whole content of a web page. To effectively exploit the information contained in anchor texts, we employ a decision tree to predict the relevance of the target pages.

2.3 Training Data and Feature Selection

For a web graph $G = (V, E, r)$, we first crawl all the pages in V and identify the relevant pages in V by using a properly trained SVM (Support Vector Machine) classifier. A user needs to prepare some relevant and irrelevant example pages of topic in mind for the classifier. In the following, we represent the classifier by a function C such that $C(v) = \mathit{true}$ if v is classified as a relevant page, and $C(v) = \mathit{false}$ otherwise.

Second, for each page $t \in \{v \mid C(v) = \mathit{true},\ v \in V\}$, we compute the shortest path from the entry page r to t by the Dijkstra's algorithm. We denote the union of all the pages on each of these shortest paths as a set S.

Third, let $l = (b, e) \in E$ be a hyperlink, where b and e denote the source and target pages of l respectively, and let $f(l)$ be a function returning the anchor text associated with l. We use $P = \{f(l) \mid l = (b, e) \in E \wedge b \in S \wedge e \in S\}$ as positive examples and $N = \{f(l) \mid l = (b, e) \in E \wedge b \in S \wedge e \notin S\}$ as negative examples of the decision tree learning. We simply ignore those hyperlinks whose anchor text is blank. The image of the training data is depicted in Figure 1. The black disks are relevant pages and double circles are irrelevant pages on the shortest paths. Anchor texts on thick edges are used as positive examples and those

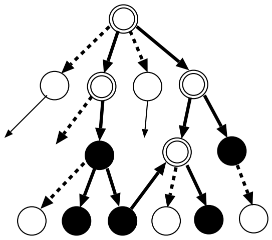

Figure 1: Relevant Pages and Shortest Paths

on dotted edges are used as negative examples. As feature space, we use $F = F_p \cap F_n$ where F_p and F_n are the set of words appeared at least once in P and N, respectively.

2.4 Decision Tree Construction

Given an instance of anchor text a, let $g(a)$ be a function returning the set of features (terms) appeared in a. Note that $g(a) \subseteq F$. The decision tree we are going to construct is a Boolean function $B(g(a))$. This function is constructed by applying the ID3 algorithm [3] to the positive and negative examples in 2.3. If there exists a term set s that cannot separate positive examples and negative examples perfectly and there are no other terms that could be used to further distinguish these examples, we define $B(s)$ based on the probability. Let P_s, N_s be the set of positive and negative examples that contain all the terms in s. If $\frac{|P_s|}{|P|} \rangle \frac{|N_s|}{|N|}$, then the positive case is more probable than the negative case, so we define $B(s) = \mathit{true}$. Otherwise, we define $B(s) = \mathit{false}$.

2.5 Focused Crawling

We use $B(g(a))$ to determine the priority of unvisited URLs, and call the crawler *decision tree crawler* (DTC, for short).

3. EXPERIMENTS

We conducted experiments on the data sets of four Japanese universities on the target topic “lecture”. These four universities were the University of Tokyo (UT), Kyoto University (KU), Keio University (KO), and Waseda University (WU).

We compared the efficiency of our DTC to two crawlers: (1) a standard breadth-first crawler; (2) a traditional focused crawler, which gives high priorities to all the children of relevant fetched pages.

3.1 Performance on the Same Data Set

The performance of DTC on the data set of UT from which the DTC is constructed, along with the performances of the other two crawlers, are depicted in Figure 2. In the figure, the number of crawled relevant pages (Y-axis) is plotted against the number of all crawled pages (X-axis). The performance of the idealistic crawler which crawls only the pages on the shortest paths is also depicted. The figure shows that DTC outperforms the other two crawlers significantly. To crawl 50% (recall) of the relevant pages, DTC only needs to crawl about 3% of the entire pages. Furthermore, most of the lengths of the shortest path from the entry pages to these relevant pages range from 6 to 9 (not depicted), which means that the decision tree guides the crawler to find deep relevant pages effectively.

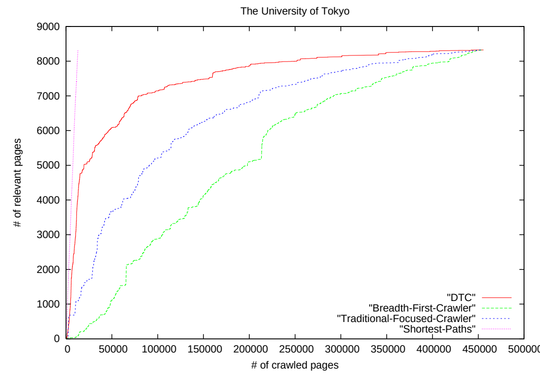

Figure 2: Crawling Performances on UT

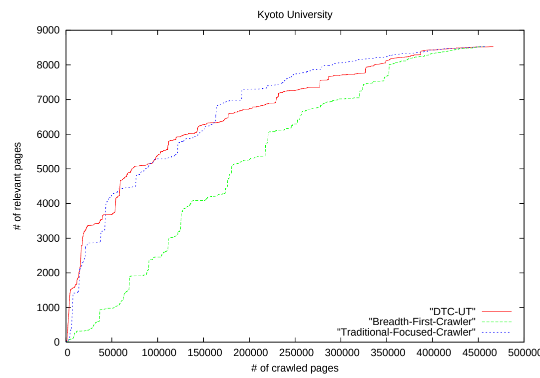

Figure 3: Crawling Performances on KU

3.2 Performance on Different Data Sets

To evaluate generalization performance, we conducted experiments by crawling the other three data sets using the DTC trained from the data set of UT. In all these experiments, DTC outperforms the breadth-first crawler. Compared to the traditional focused crawler, DTC is competitive with it on the data sets of KU and WU, but slightly inferior to it on the data set of KO. The performances on the data set of KU is depicted in Figure 3. It needs further study to improve the generalization performance and make it steady.

4. CONCLUSION

In this paper, we proposed a focused crawler guided by anchor texts using a decision tree. We also showed the effectiveness of the proposed crawler by experiments on the data set of four universities.

5. REFERENCES

[1] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific Web resource discovery. In *The Eighth International World Wide Web Conference*, Toronto, Canada, May 1999.

[2] M. Diligenti, F. M. Coetzee, S. Lawrence, C. L. Giles, and M. Gori. Focused Crawling Using Context Graphs. In *Proc. of the 26th VLDB Conf.*, pp. 527–534, 2000.

[3] J. R. Quinlan. Induction of decision trees. *Machine Learning*, Vol. 1, No. 1, pp. 81–106, 1986.