The Core of a Topic-Specific Search Engine: How to Create It

V.V. KLUEV
Software Engineering Center
University of Aizu
Tsuruga Ikki-machi Aizu-Wakamatsu City
Fukushima 965-8580
JAPAN

Abstract: A technique for gathering scientific, narrow topic-related documents from the Internet is presented. It has been successfully applied to compile a large Japanese collection of algorithms and their applications.

Key-Words: Search Engine, Similarity Metrics, Crawler

1 Introduction

It is well known that the World Wide Web is unique in the sense that everyone can put information of any kind into a computer, connect it to the Internet, install a server on it, and the information will be accessible to anyone anywhere. No one controls publishing on the World Wide Web. From this, there is a lot of potentially useful data, but on the other hand, there is a huge amount of useless data on the net as well. This is the main disadvantage of the Internet. According to some estimations [1], the volume of data on the net grows at an exponential rate. The role of search engines as tools to find appropriate information increases, because without being indexed, data cannot be found and accessible. On the other hand, the more powerful search engines are, the smaller percentage of data from the net can be indexed by them. The rate of improving search tools is much less than the rate of the data growth on the net. We agree with a note made in [10]: Searching is not only one of the most common tasks performed on the Web but one of the most frustrating.

Maybe it is time to stop thinking about a catalogue of documents like the Yahoo catalogue on the net. We need to take a step forward: the time for catalogues of topic-specific servers has come. General purpose search engines like Google and AlltheWeb are playing a significant role in finding starting points on the net to search for detailed information to answer user queries. To make these steps easier for people, topic-specific search engines can be helpful. These search engines have to be tools to sort out the chaotic world of electronic documents on the net.

The core of any search engine is a collection of documents consisting of the links to the Internet locations of the documents themselves. How to create it: by hand or automatically? We advocate a semi-automatic style. The current level of information technologies is good enough to provide researchers with the necessary methods and tools.

In this paper, we discuss one of the possible ways to do that.

The rest of the paper is organized as follows. The next section gives a short overview of related work. Section 3 presents the main heuristics applied to the crawl and to the decision making process. A proposed metrics to evaluate the level of a document relevancy can be found in this section as well. Runs of our experiments and their conditions are in Section 4. Section 5 discusses our approach and obtained results. Final remarks and our plans for future work are in the Conclusion section.

2 Related Work

A crawler is a key component of any search engine as a tool to collect data to index from the net. Focused crawlers usually filter an input stream to select documents relevant to the topic of interest.
Several approaches were proposed to design a focused crawler; some of them can be seen in [5, 6, 9, 11, and 12]. All of the approaches use heuristics based on the intuition what related documents connected by links and links extracted from one document will follow to semantically close ones. Methods differ from each other in the following: What the filter is and how to detect a relevant document. Most of the techniques adopt the idea of

Page Rank algorithm [10]. Some approaches [9] take into account a vision of any document as a bag of words and compounds. As it was noted in [10], context extraction is far less practical, and because of this, none of the search techniques is able to deal effectively and efficiently with the huge volume of information growing on the net.

There is another question. Do we need to crawl topic-specific information in any language which is different from English? The most popular languages after English on the Web are German and Japanese. The percentage of the pages written in these languages is about 60% (English), 7% (German) and 6% (Japanese) [3]. The big problem with the most popular search engines like Google, AlltheWeb, Alta Vista, etc. according to study [3] is as follows: They do not perform any morphological analysis in any language. They were initially designed to search for English documents. Search results in foreign languages are relatively poor. The answer to the aforementioned question is positive. Nowadays, a natural language processing technique can provide researchers only with morphological analyzers. They are helpful in segmenting texts in Asian languages and classifying components of sentences in many languages.

3 Heuristics

We start our discussion with an indexing technique of crawled documents. This question is important for Asian languages. Japanese is no exception. There are no spaces between words to delimit them. To solve this problem, two approaches have been proposed. The first one is to use a morphological analyzer. The basic idea behind the second one is to adopt bi-gram indexing. Because an average Japanese word consists of one or two characters, it makes possible to break down a document into sequence sets which include exactly two characters (kanji). The breaking process of the text should be done by sliding a two character window by step in one character. An exception is katakana subsequences. Each such subsequence should be considered as a separate indexing item. Subsequences consisting of hiragana characters and English words may be dropped. A disadvantage of this approach results in a huge index because a lot of artificial “words” produced. On the other hand, an accuracy of searching according to the both indexing methods is practically at the same level [8].

Which approach works better in the crawling? It is not clear. We have combined them both.

A raw filter includes Japanese words and compounds and their weights as well.

Techniques and heuristics used in our experiments are as follows:

a) **Bi-gram technique** Its aim is to break crawled documents and filter components into the simple units.
b) **String matching** To find an occurrence of any word from the raw filter, every retrieved document was scanned.
c) **Segmenting documents** The Chasen lexical analyzer [2] was applied to detect lexemes which match lexemes from the filter.
d) **A variant of the vector space model** The aim of this metric is to estimate the similarity of discovered documents and their relevance to the topic of interest.
e) **Selecting valuable terms** A proposed schema in [7] to select such terms from a retrieved document set to include them into the variable part was adopted: Words occurring in less that 0.2% of the documents were removed as the low-frequency words. The high-frequency words occurring in more than 20% of the documents were discarded as well from the selecting process.
f) **Detecting topic-related documents connected by links** If documents A and B were recognized as relevant to the topic of interest and they have a link to document C, then score of C has to be increased because it is more likely that it is also relevant.
g) **The assistance of AltaVista** This search engine was employed to get URLs pointed to documents automatically recognized as relevant. Up to 10 of such URLs for each document were added to the queue of links to visit and evaluate.

The main assumptions about the filter are as follows. It consists of the four parts: core, lexeme, bi-gram and variable.

- The core includes Japanese words as they are (this may be a word or a word collocation; this part contains only that words from the initial filter, which can be divided into lexemes by the morphological analyzer);

- The lexeme part covers lexemes obtained as an outcome of the Chasen program applied to the filter;
- The bi-gram part consists of components obtained from the initial filter;
- The variable part includes the most important lexemes and bi-grams from the vector space model point of view. This part of the filter is changed after discovering several relevant documents. It is created as a result of an analysis of new retrieved documents.

3.1 Filtering Technique

Note, thresholds for each part of the filter were tuned independently: The criteria were to recognize 50 - 70% of the document core as relevant to the topic of interest.

Our crawler followed the rules explained below after downloading each discovered document.

1. A database of all URLs extracted from the automatically recognized relevant documents was kept. Each record included a special counter showing how many times the corresponding URL was mentioned at different locations.
2. The weight of each filtered component in a document (Japanese word, lexeme or bi-gram) was doubled, if this component occurred in the title.
3. The variable part in the filter was changed after every 200 undoubtedly relevant recommendations (the definition see below). This set was considered as a small collection and its important lexemes and bi-grams were inserted into the filter.
4. The score of newly discovered document was calculated as follows: Each document was filtered using each part of the filter separately. It was marked as undoubtedly relevant if its score was above the thresholds of two of them, or if its score was above only one of them and at least Z (Z is equal to 3) undoubtedly relevant documents had links to it. The document could be transformed to this category (undoubtedly relevant) later if the counter of links to it exceeded Z. Note that the crawler computed the counters of links for each document evaluated as relevant according to any metrics. Formulas to calculate a document score are as follows:

$$Score(D) = \sum_i Filter_i * D_{feature\_vector} + PageRank(D)$$

$$D_{feature\_vector} = d_1,...,d_k,...,d_m,...,d_n,...,d_p,...,d_r$$

$$PageRank(D) = \alpha * \sum L_i$$

$$d_i = \begin{cases} 1 \mid 0, i \le k \\ tf_i / N_j, i > k, j = \begin{cases} 1, k < i \le m \\ 2, m < i \le n \\ 3, n < i \le p \\ 4, p < i \end{cases} \end{cases}$$

Where components 1 to k are word components; components k+1 to m are lexemes; components m+1 to n are bi-grams; components n+1 to p are variable lexemes; p+1 to r are variable bi-grams. L is a link from the undoubtedly relevant page. tf is a term frequency of the corresponding component; N is a number of lexemes, bi-gram or variable parts of these components. Lexemes and bi-grams for each variable part of the filter were created from Japanese nouns with the assistance of the Chasen analyzer.

4 Experiments

A powerful computer on the base of Intel Pentium 4 Processor was dedicated for our experiments to crawl the collection. It was equipped with 512 Mb of RAM, 390 Gb of RAID disk and a fast Ethernet network connection. Linux 7.3 was running on it.

Only html and plain text documents were crawled and analyzed. After retrieval every M (M is equal to 2500) documents, the crawler have cleaned up the queue of URLs to visit: up to N (N is equal to 10000) elements remained in it, and up to K (K is equal to 100) of them were related to the same server.

Because the lexical analyzer Chasen did not manage to segment some retrieved documents well, it ran into a loop once a day on average. As result, we needed to restart the crawler and put the corresponding URL in a queue of URLs prohibited for visiting.

5 Discussion

A technique to make a decision about the relevance of the crawled documents was adopted from our

Table 1 Changing Semantics of Filter Components

Raw filter components	Lexemes	Bi-gram
並列処理 (parallel processing)	並列　処理 (parallel processing)	並列　列処　処理
情報検索システム(information retrieval system)	情報　検索　システム (information search system)	情報　報検　検索　システム
二分検索 (binary search)	二　分　検索 two minute search	二分　分検　検索
2 分木 (binary tree)	2　分木 two minute wood	分木

successful experience in compiling English scientific documents [9]. Their key components are as follows:

- The threshold was tuned during the training the crawler on the document core;
- Documents were considered as undoubtedly relevant to the topic of interest if their scores calculated according to several metrics were above each threshold;
- After detecting that a document was relevant to the topic of interest, every new document retrieved from the same directory of the same server was also marked as relevant without taking into account its score;
- Every link inherited an average arithmetic sum of the score from its parent documents. The queue was arranged according to a score of its elements in decreasing order.
- Automatically detected relevant documents were selectively checked by human inspection.
- To recognize duplicates, the same technique was used.

Table 1 gives an illustration of changing semantics of the filter's components. Meaning some components in many cases changed completely after processing by the lexical analyzer. To evaluate the effectiveness of crawling, data from Table 2 can be used. The crawler was running 10 to 15 hours per day because of the aforementioned reason. About 32% of all links are "dead". This is quite a high number; it illustrates permanently changing the nature of the net. On average, 37% of visited documents were automatically estimated as relevant. About 56% of them were marked as undoubtedly relevant. This is 21% of our set.

Can we trust these results? We selectively checked 900 documents or 4.4% of undoubtedly relevant ones (one document from half of the visited hosts). It was discovered that about 10% of them were non relevant to our topic. What should we do with another part of the relevant documents which met the requirement only one of the metrics? How many actually relevant documents are among this set (15361 documents)? How many documents on algorithms and their applications were lost during crawling? Are they in the set consisting of 45920 documents (this amount is equal to the number of successfully downloaded pages minus the number of relevant pages)?

Table 2 Statistics for two weeks in March 2004

Number of visited URLs	131905
Number of visited hosts	23281
Number of URLs from .jp domen	96176
Number of successfully downloaded pages	81531
Number of relevant pages	35711
Number of actually relevant pages	20350
Number of hosts with relevant information	1866

We should note the estimation of the crawled data and its quality. What is a good document? Is there a border between relevant and non relevant document? These questions are important in the topic specific search engine area.

Our aim was to create a large collection of algorithms and their applications which could be used in teaching in schools and universities. Our point of view is as follows: 1) documents including a description of any algorithm were marked as relevant; 2) documents which could be used in finding documents with algorithm descriptions were also marked as relevant (for example, home page of any university computer department). Even making this assumption, it was a subjective decision about the relevance of documents in many cases.

We could be able to crawl only linked documents. It means a document could be potentially visited by

crawler, only if there is a link to it from another document. Documents pointed to automatically recognized ones as relevant were visited and estimated as well because of the assistance of Alta Vista. Our crawler did not get the whole directories as the general purpose ones usually do. This is a weakness of our approach.

How long can our collection be fresh? Our links to the source documents will become broken very fast. This is a nature of the net: Nothing remains static on it. Table 3 is a small example of the growing number of broken links in the core of our collection. According to the study [4], up to 15% of the crawled pages from the set of 151 million could not be downloaded in about 3 months.

Table 3 Degradation of the collection core

Core collection	September 2002	January 2003	March 2004
Number of documents	167 (100%)	122 (73%)	99 (60%)

6 Conclusion and Future Work

A large Japanese collection of algorithms and their applications were compiled from the Internet. Usage of several metrics to make a decision about relevance of crawled documents was effective. It made possible the reduction of human intervention in the checking process of compiled data. Proposed metrics can be useful in compiling such data from the net. However, the amount of non relevant documents in the collection is relatively high. How to reduce it without manual analysis? This question is still unanswered. How many important documents were lost during the crawling process? What is an effective way to keep the collection up to date? Additional research should be done to answer these questions.

The main point for the next step in our research is the creation of methods to search inside a narrow topic-specific collection.

We believe that directories of topic specific search engines are steps on the way to put the structure on the existing Web and they are tools to simplify the search. They can be created in a semiautomatic style. Our experience is one illustration how to do that.

Acknowledgment

We would like to thank students of the University of Aizu Mr. Takashi Teramura and Mr. Akihito Takahashi for their help in constructing a document core of our collection and the filter for the crawler.

This research was funded by Fukushima Prefectural Foundation for Advancement of Science and Education grant F-11, 2003.

References:

[1] FAST Data Search, http://www.fastsearch.com/us/products/fast_data_search

[2] ChaSen, http://chasen.aist-nara.ac.jp/hiki/ChaSen/.

[3] Judit Bar-Ilan, Tatyana Gutman, How do search engines handle non-English queries? – A case study, I*n Proc. Of the Twelfth World Wide Web Conference*, Hungary, 2003.

[4] Dennis Fetterly, Mark Manasse, Marc Najork and Janet L. Wiener, A Large –Scale Study of the Evolution of Web Pages, *In Proc. Of the Twelfth World Wide Web Conference*, Hungary, 2003.

[5] Soumen Chakrabarti, Martin van den Berg and Byron Dom, Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery, *In Proceedings of 8th International World Wide Web Conference (WWW8)*, 1999.

[6] Ah Chung Tsoi, Daniele Forsali, Marco Gori, Markus Hagenbuchner, and Franco Scarselli, A Simple Focused Crawler, *In Proc. Of the Twelfth World Wide Web Conference, Hungary*, 2003.

[7] Bhushan Mandhani, Sachindra Joshi, and Krishna Kummamuru, A Matrix Density Based Algorithm to Hierarchically Co-Cluster Documents and Words, *In Proc. Of the Twelfth World Wide Web Conference*, Hungary, 2003.

[8] *Proceedings of the Third NTCIR Workshop*, Tokyo, 2002.

[9] V. Kluev, Compiling Document Collection from the Internet, *ACM SIGIR Forum*, Vol. 34, Number 2, pp. 9 – 14, Fall 2000.

[10] Wen-Chen Hu and Jyh-Haw Yeh, World Wide Web Search Engines, in *Architectural Issues of Web-Enabled Electronic Business*, Idea Group Pub, 2002, pp. 154 – 169.

[11] Michael Chau and Hsinchun Chen, Personalized and Focused Web Spiders, in *Web Intelligence*, Springer Verlag, 2003, pp. 197 – 213.

[12] V. Kluev, Results Merging with the OASIS System: An Experimental Comparison of Two Techniques, *IEICE Transactions on Information and systems*, VOL.ED86-D,No. 9, September 2003, pp.1773 – 1780.