Automatic Estimation of Query Syntax for Search Sites

Nakatoh, Tetsuya
Computing and Communications Center, Kyushu University

Koga, Yasunori
Graduate School of Information Science and Electrical Engineering, Kyushu University

Uhl, Axel
Interactive Objects Software GmbH

Hirokawa, Sachio
Computing and Communications Center, Kyushu University

出版情報：Proceedings of Pan-Yellow-Sea International Workshop on Information Technologies for Network Era : PYIWIT'02, 2002. Information Processing Society of Japan
バージョン：
権利関係：

Automatic Estimation of Query Syntax for Search Sites

Tetsuya Nakatoh, Yasunori Koga, Axel Uhl, Sachio Hirokawa

Abstract— **General search engines like Yahoo! and Google are indispensable to cope with the flood of information on the Internet. But the quality of search results is not always good enough due to the vast size of the search space that these engines cover. On the other hand, many databases are becoming searchable on the Web, and many companies and organizations are providing their own search facility on the Web. We call such a site a search site. We are developing a system that integrates such search sites.**

To integrate these search sites we need to conceal the difference of the query syntax from the user. But the query syntax of these search sites vary. In this paper we propose a method that automatically determines the query syntax.

Keywords— **Wrapper, Search Engine, Query Syntax**

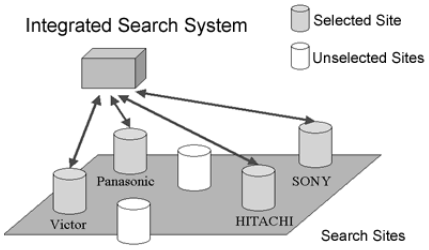

Fig. 1. Integration of Search Sites

I. Introduction

The flood of information on the Internet is a serious problem for people and companies. Search engines are a central tool in getting rid of this flood of information. We use search engines, e.g., Yahoo!, AltaVista, or Google, which search for the information we need over the WWW. One of the problems of general search engines is the quality of search results. The search results tend to contain many irrelevant pages. On the other hand, many companies are providing their own information with their own search engines[9]. We call such web sites "Search Sites" as compared to general search engines. A search site of a company focuses on their information and the quality is guaranteed.

The number of such search sites is increasing rapidly. One problem is that virtually all of them use web-based protocols like HTTP/HTML to expose their search functionality. These are ill-suited for solid and robust integration efforts as they are tailored for human-machine interfaces, rather than being amenable to programmatic access. This problem and possible solutions have been described in [10].

The next problem we face is that we have to visit many search sites one by one to collect all information we need. A solution is the integration of search sites for each purpose. Fig. 1 shows a typical example of integration of search sites of electronic companies like Sony, Panasonic, Victor, Hitachi, and Pioneer that produce DVD Players. A user can search and compare DVD players with one interface.

CompletePlanet† estimates that there are more than 100,000 searchable databases available on the Web. To integrate these search sites we need to conceal the difference of the sites. But the query syntax of these search sites vary. The metasearch engines integrate a small number of targets using manually written wrappers. But the fast expansion and change of the WWW is beyond any centralized manual effort. Automation of all processes is necessary and the following are the main problems in achieving the automatic integration of search sites.

1. Pattern extraction of search result
2. Automatic estimation of query syntax
3. Feature extraction of search sites for site selection
4. Interface for integration of search results

1 and 2 are necessary for automatic wrapper generation. 3 and 4 are necessary for integrating many sites. We developed a wrapper generation method in [5], [7] and [2], where the query is assumed to be a single word. In this paper, we propose a method to estimate the query syntax and extend our method for logical expressions. As for 3 and 4, we proposed a framework for feature extraction of search sites in [8] and [4]. With these methods, we can integrate search sites if we are given a list of sites for integration.

II. Probing Queries

Some sites admit only one keyword. Many sites admits complex queries with more than two keywords. We need to guess whether the site admits complex queries or not. For a simple search, we only have to compose a query syntax with a single keyword. But for a complicated search, we use more than two keywords with logical operators - "*AND*" and "*OR*". The syntax used to construct logical expressions varies regarding the symbols to be used for the logical operators. We need to guess automatically which expression is used in each site.

The HTML document representing the search results contains zero or more answers, which are usually displayed using the same pattern. But it may contain unnecessary headlines and advertisements. We extract the essential

T. Nakatoh and S. Hirokawa are with the Computing and Communications Center, Kyushu University, Hakozaki 6-10-1, Higashi-ku, Fukuoka, 812-8581, JAPAN. E-mail:nakatoh@cc.kyushu-u.ac.jp. Y. Koga is with Graduate School of Information Science and Electrical Engineering, Kyushu University. Axel Uhl is with Interactive Objects Software GmbH, Germany

†`http://www.completeplanet.com/`

parts from the HTML document using the pattern (See [7], [2]). Thus, we can tell the number of answers that the HTML document contains.

We use two kinds of keywords for probing. The keywords we denote by "$\mathcal{A}$" in the sequel of the paper return some search results. The keyword "$\mathcal{Z}$" returns no search result. By analyzing the HTML document representing the search results, we can guess whether the keyword is "$\mathcal{A}$" or "$\mathcal{Z}$". Thus we estimate the query syntax by

1. sending a query in a possible syntax with specific keywords,
2. analyzing the HTML document representing the search results, and
3. evaluating which expression syntax is correct.

III. Complex Queries

A. Blank

A query syntax may accept more than one keyword without logical operators. When logical operators are omitted, we must examine how some keywords handled.

- query = "$\mathcal{A} \sqcup \mathcal{Z}$"‡
  - some results : *OR*§, or second keyword is ignored.
  - no result : *AND*††.
- query = "$\mathcal{Z} \sqcup \mathcal{A}$"
  - some results : *OR*.
  - no result : *AND*, or second keyword is ignored.
- query = "$\mathcal{A} \sqcup \mathcal{A}$"
  - some results : worked.
  - no result : Do not work with two keywords. Only a single keyword is permitted, or a logical operator is necessary.

We can then estimate as shown in TABLE I.

TABLE I
Estimating the meaning of the blank separator

		$\mathcal{A} \sqcup \mathcal{Z}$	
		some results	no result
$\mathcal{Z} \sqcup \mathcal{A}$	some results	*OR*	-
	no result	single	TABLE II

TABLE II
Estimating the meaning of the blank separator

$\mathcal{A} \sqcup \mathcal{A}$	some results	*AND*
	no result	An operator is necessary

‡A blank ("space") character is shown as "$\sqcup$" in this paper.
§A disjunction operator is shown as "*OR*".
††A conjunction operator is shown as "*AND*".

B. Conjunction and Disjunction

A query syntax may allow for or even require explicit use of logical operators. There are several different kinds of symbols used to identify a particular logical operator. Thus, we must evaluate which symbols are used to represent the logical operators for a particular search site. Let K_1 and K_2 be two keywords. Then generally the following logical expressions are considered.

- $K_1 \sqcup \text{AND} \sqcup K_2$
- $K_1 \sqcup \text{and} \sqcup K_2$
- $K_1 \sqcup \& \sqcup K_2$
- $K_1 \sqcup * \sqcup K_2$
- $*K_1 \sqcup * K_2$
- $K_1 \sqcup \text{OR} \sqcup K_2$
- $K_1 \sqcup \text{or} \sqcup K_2$
- $K_1 \sqcup , \sqcup K_2$
- $K_1 \sqcup | \sqcup K_2$
- $K_1 \sqcup + \sqcup K_2$
- $+K_1 \sqcup + K_2$

Now, We write these query syntax as $Q(K_1, K_2)$.

Fundamentally, estimation can be done the same way as for the blank symbol before. But in addition we must check whether an operator symbol is handled as one of the keywords, ignored, or recognized as a meta-character. The following subsections provide the details.

B.1 When a blank separator means *AND*

We obtain a result when the operator works properly as *OR*. In this case, we guess that the operator has the function of *OR*. On the other hand, when we do not obtain a result, the operator is working as *AND*, or ignored. Although this confirmation is not easy, it is unnecessary because the blank separator means *AND*.

- $Q(\mathcal{A}, \mathcal{Z})$
  - some results : *OR*
  - no result : *AND*, operator is ignored or error.
- $Q(\mathcal{Z}, \mathcal{A})$
  - some results : *OR*
  - no result : *AND*, operator is ignored or error.

B.2 When a blank separator means *OR*

When we obtain a result, the operator is working as *OR* or ignored. Although this confirmation is not easy, it is unnecessary because the blank separator means *OR*. On the other hand, we do not obtain search results when the operator works as *AND*. Another reason for no results could be that the operator caused an error. Therefore, we check $\mathcal{A} \sqcup \mathcal{A}$, and confirm that no error arises.

- $Q(\mathcal{A}, \mathcal{Z})$
  - some results : *OR* or operator is ignored
  - no result : *AND* or error
- $Q(\mathcal{Z}, \mathcal{A})$
  - some results : *OR* or operator is ignored
  - no result : *AND* or error
- $Q(\mathcal{A}, \mathcal{A})$

– some results : *AND*
– no result : error

B.3 When a blank separator means neither *AND* nor *OR*

If a blank separator means neither *AND* nor *OR*, those functions can not be used except that *AND* and *OR* operator are written specifically. Therefore, we do not need to estimate the function of the blank separator.

IV. Experiment

We made several experiment of the automatic estimation of query syntax with using actual search sites. In this paper, we report on the experiment for *OR*, *AND*, and BLANK. The TABLE III is the list of search sites of this experiment and the TABLE IV is the list of keywords we used. For simplicity, the keywords were chosen after some manual tests against some more search sites. To improve the keyword list one could perform an automatic analysis for each individual search site, based, e.g., on a dictionary containing a large vocabulary.

TABLE IV
Keywords

$\mathcal{A}$	regional, arts, society, reference, science
$\mathcal{Z}$	plomkijmnhuytgbvfrdcxeszawq

A result of logical operator estimation is shown in TABLE V, using the blank character as operator representation.

TABLE V
blank separator

site ID	$\mathcal{A} \sqcup \mathcal{Z}$	$\mathcal{Z} \sqcup \mathcal{A}$	$\mathcal{A} \sqcup \mathcal{A}$	function
1	0	0	0	single
2	0	0	+	*AND*
3	0	0	+	*AND*
4	+	+	+	*OR*
5	+	+	+	*OR*
6	+	+	+	*OR*
7	+	+	+	*OR*
8	+	+	+	*OR*
9	0	0	+	*AND*
10	+	+	+	*OR*
11	0	0	+	*AND*
12	0	0	+	*AND*
13	0	0	0	single

The meaning of the sign are:

+	Some search results returned.
0	No search result returned.
single	Only a single keyword is permitted, or a logical operator is necessary.

We compared this result with the help texts provided at each of the search sites, and we confirmed that the estimation worked correctly for all sites except one. The result of estimation of the site with ID 1 was different from the syntax as specified in their help text. However, a manual check revealed that the site did not work as explained in the help text.

A result of estimating the logical operator that can be used is shown in the TABLE VI and the TABLE VII. We confirmed that we estimated properly.

TABLE VI
Estimate *OR* operator

site ID	$Q(\mathcal{A}, \mathcal{Z})$	$Q(\mathcal{Z}, \mathcal{A})$	query syntax
1	0	0	–
2	+	+	"$K_1 \mid K_2$" "K_1 *or* K_2"
3	+	+	"$K_1 \mid K_2$" "K_1 *or* K_2"
9	0	0	–
11	0	0	–
12	+	+	"K_1 *OR* K_2" "K_1 *or* K_2"
13	+	+	"K_1 *OR* K_2" "K_1 *or* K_2"

TABLE VII
Estimate *AND* operator

site ID	$Q(\mathcal{A}, \mathcal{Z})$	$Q(\mathcal{Z}, \mathcal{A})$	$Q(\mathcal{A}, \mathcal{A})$	query syntax
1	0	0	0	–
4	0	0	+	"$+K_1$ $+K_2$"
5	0	0	+	"$+K_1$ $+K_2$"
6	0	0	+	"$+K_1$ $+K_2$"
7	0	0	+	"$+K_1$ $+K_2$"
8	0	0	+	"K_1 *AND* K_2"
10	0	0	+	"$+K_1$ $+K_2$"
13	0	0	0	–

V. Related Work

Integration of multiple search engines is known as a metasearch engine [6]. Most of their targets are general search engines that may overlap each other. On the other hand, the targets of our project are independent search sites that do not overlap. They may be homogeneous, like competing electronic companies, or may be heterogeneous, like airlines, hotels and restaurants. The contents are qualified by each site, so that we do not need to apply filtering and ranking to the search results. The wrappers used in metasearch engines are usually written manually. We are proposing the automatic generation of wrappers for search sites.

In [1], Ipeirots et al. used a similar query probing method for feature extraction of text databases. But they used a single keyword and used the number of search results. We proposed a method for complex queries and used the pattern extraction, which Ipeirots et al. admit to be desirable.

Kushmerick et al. [3] introduced a learning algorithm to generate a wrapper from several examples. Our wrapper generation [7] is based on the observation that the search result contains repetition of the same tag sequence. So we do not need examples.

TABLE III
SITE LIST

site ID	URL
1	http://www.jimin.jp/jimin/title.html
2	http://www.jsps.go.jp/
3	http://www.inpaku.go.jp/
4	http://st.jr.chiba-u.ac.jp/mos/
5	http://www.jdin.net/
6	http://www.boj.or.jp/search/search.htm
7	http://www.ctc-g.co.jp/
8	http://www.mei.co.jp/
9	http://www.compaq.co.jp/search/
10	http://www.cisco.com/japanese/warp/public/3/jp/search/public.html
11	http://www.mri.co.jp/top.html
12	http://www.jama.or.jp/indexJ.html
13	http://www.jtnet.ad.jp/WWW/JT/JTI/Welcome.html

VI. Conclusion and Further Work

We proposed the method that automatically determines the query syntax. We can integrate search sites which accept several keywords as their input.

As a future work, we need to estimate a NOT operator. There is a proximity operator(NEAR). And, we must make an experiment by a much broader sample base. That preparation is completed.

Acknowledgments

The authors would like to acknowledge Miyuki Sakai for her earnest support.

References

[1] P. Ipeirotis, L. Gravano and M. Sahami, *Automatic Classification of Text Databases through Query Probing*, Proc. of the ACM SIGMOD Workshop on the Web and Databases (WebDB'00), 2000.

[2] Y. Koga, T. Taguchi and S. Hirokawa, *Wrapper Generation for Search Sites Integration(in Japanese)*, Proc. DEWS'01, 2001.

[3] N. Kushmerick, D. Weld and B. Doorenbos, *Wrapper induction for information Extraction*, IJCAI'97, pp .729-737, 1997.

[4] T. Nakatoh, Y. Koga and S. Hirokawa, *Automatic Classification of Search Sites (in Japanese)*, Proc. DBWeb2001, pp.225-228,2001.

[5] T. Nakatoh, M. Sakai, Y. Koga and S. Hirokawa, *Generation of Query URL for Search Sites*, Proc. SSGRR2002w(CD-ROM).

[6] E. Selberg and O. Etzioni, *The MetaCrawler architecture for resource aggregation on the Web*, IEEE Expert, Vol.12, No.1, pp. 11-14, 1997.

[7] T. Taguchi, Y. Koga and S. Hirokawa, Integration of Search Sites of the World Wide Web, Proc. of International Forum cum Conference on Information Technology and Communication, Vol. 2, pp. 25-32, 2000.

[8] S. Hirokawa, Y. Koga and S. Hirokawa, Integration of Search Sites of the World Wide Web, Proc. of International Forum cum Conference on Information Technology and Communication, Vol. 2, pp. 25-32, 2000.

[9] C. Sherman and G. Price, The Invisible Web, Information Today, Inc, Medfore, New Jersey, 2001.

[10] Axel Uhl and Horst Lichter, "New Wave Searchables: Changing the paradigm of Internet-scale search," in *International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, L'Aquila, Italy, Aug. 2001, SSGRR.