Automatic Feature Extraction of Search Sites
Sachio Hirokawa, Seiichiro Watanabe, Yasunori Koga, Tsuyoshi Taguchi

Abstract — Many web sites are offering its own search facility. We call such a site as a search site. We show an automated feature extraction method for search sites, which enables a dynamic integration of search sites with high quality. It is based on keyword sampling and wrapper generation.

Keywords — Wrapper, Search Engine, Latent Semantics Indexing

I. INTRODUCTION

The flood of information on the Internet is a serious problem for people and companies. Search engines are keys to get rid of this flood of information. We use search engines, e.g., Yahoo!, Alta-vista, google, which search for the information we need over the WWW. The number of companies are increasing that provide their own information with their own search engines. We call such web sites as “Search Sites” compared with general search engines.

One of the problem of general search engines is the quality of search result. The search results tend to contain many irrelevant pages. On the other hand, a search site of a company focuses on their information and the quality is guaranteed. But 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, Sony, Panasonic, Victor, Hitachi and Pioneer which produce DVD Players. A user can search and compare DVD players with one interface.

Each search site has a different interface for query and has a different pattern of search results. The followings are main problems to achieve the automatic integration of search sites.

- Wrapper generation
- Feature extraction
- Interface for selection
- Integration of search results

We developed a wrapper generation method in [9], [5]. We can integrate search sites if we are given a list of sites for integration. CompletePlanet 1 estimates that there are more than 100,000 searchable databases available on the Web. The problem we face is how to choose appropriate sites for each purpose of integration. In this paper, we propose a framework for feature extraction of search sites. It is based on keyword sampling and Latent Semantics Indexing [1]. Novelty of our approach is that we do not use the raw search result, but we use a wrapper to extract contents from the search result. By an experimental evaluation, we show that our method returns the high quality feature of sites compared with naive method that uses raw search result.

II. WRAPPER GENERATION

To integrate search sites, we need extract the contents of the search results. However, the search result may contain unnecessary headlines and advertisements around the contents. And the format varies site to site.

Fig. 1. Integration of Search Sites

![Integration of Search Sites](image1)

Fig. 2. An example of search result

![An example of search result](image2)

Each search site has a different interface for query and has a different pattern of search results. The followings are main problems to achieve the automatic integration of search sites.

S. Hirokawa is with the Computing and Communications Center, Kyushu University, Hakozaki 6-10-1, Fukuoka 812-8581, JAPAN. E-mail:hirokwa@cc.kyushu-u.ac.jp. S. Watanabe and Y. Koga are with Graduate School of Information Science and Electrical Engineering, Kyushu University. T. Taguchi is with Nomura Research Institute, Ltd.

1 http://www.completeplanet.com/
HTML tags as follows and this pattern occurs 8 times in the HTML file.

```html
```

The pattern can be represented by a sequence of HTML-tags:

```html
<P><DT><A>HREF="http://www.clayzee.com/organizations/art\_and\_craft/israel/index.htm"> </A></DT><DD><BR>
```

We developed a method to discover these tag-patterns from the search results [9], [5]. To determine the feature of a search site, we use the extracted contents instead of raw search result. We show that this process improves the quality for feature extraction.

### III. Feature Extraction by Wrapper Method

We made an experiment for evaluation of our method. We chose the following 23 search sites from CompletePlanet. There are 20 categories at the top level. We chose 4 categories:

(1) Education
(2) Employment & Jobs
(3) News, Media, Entertainment
(4) People, Companies, Orgs

We picked up 7, 4, 5 and 7 sites from each category. (See Table ref(list-of-search-sites).) To obtain the feature of each search site, we send several keywords and analyzed the search result. We chose 133 sampling keywords which consist of the names of the top directory and the names of subdirectories of the four categories.

<table>
<thead>
<tr>
<th>Category</th>
<th>Search Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Education</td>
<td>Clayzee Forums, Education, Research</td>
</tr>
<tr>
<td>Employment &amp; Jobs</td>
<td>Welcome to the Clearinghouse, Massachusetts Department of Education</td>
</tr>
<tr>
<td>News, Media, Entertainment</td>
<td>Yahoo! News and Media, Headhunter.net Job Seeker, Headhunter.net Entertainment</td>
</tr>
<tr>
<td>People, Companies, Orgs</td>
<td>Advanced Job Search, Growing North Dakota, Jobs.com</td>
</tr>
</tbody>
</table>

A search result contains the links to the related documents in the sites together with simple explanation of the documents. It may contain some other information which is not directly related to the keyword or to the document a user looks for. The specialized Yahoo! sites, e.g., 3, 12, and 18 in Table I, has links to Alta Vist, Google, Britannica.com and to other directories in Yahoo! The site 23 of osOpinion displays the today’s news on the right half of the screen. It is not related to the search result. Thus, the contents that surround the search result misleads the feature extraction.

In this paper we compare the qualities of feature extracted from raw search result and wrapper output (Fig. 3).

![Feature of the Site](image)

**Fig. 3.** Naive Method vs Wrapper Method

### IV. Comparison of Naive Method and Wrapper Method with Latent Semantics Indexing

We cannot retrieve all the contents of the database behind the site. So we try a sampling with 133 keywords. We obtain 133 HTML files as search results for each keyword. The file obtained for “activities” surely contains the keyword “activities”. But other search result, e.g., for “education”, may contain the keyword. Given a site, we construct a vector of width 133. Each component of the vector corresponds to a keyword and represents the number of files that contain the keyword. Thus we obtain a 133 × 23 term-site matrix. The matrix $A_{ij}$ for naive method and $B_{ij}$ for our wrapper method is defined as follows. Fig. II shows some part of $B_{ij}$.

$$A_{ij} = \#\{k \mid W_k \in \text{Search}(S_i, W_j)\}$$

$$B_{ij} = \#\{k \mid W_k \in \text{Wrapper}(\text{Search}(S_i, W_j))\}$$

Here $S_i$ is the $i$-th search site and $W_j$ is the $j$-th keyword. Note that most of the third column of the matrix $B_i$ are 0 in Table II. The third column corresponds to Yahoo! restricted to Education. It returns search result only for “Statistics”, “School”, “Science”, “Radio”, “Mathematics”, “K-12”, “Education”, “Center” and “Country”.

This matrix reflects the feature of the search sites. We compare the matrices $A$ and $B$ using Latent Semantic Indexing [1]. We apply the singular value decomposition (SVD)
TABLE II

<table>
<thead>
<tr>
<th>Activities</th>
<th>Education</th>
<th>People</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 30 0</td>
<td>1 2 3</td>
<td>20 21 22 23</td>
</tr>
<tr>
<td>Administration</td>
<td>0 11 0</td>
<td>0 3</td>
</tr>
<tr>
<td>Agriculture</td>
<td>0 11 0</td>
<td>0 0</td>
</tr>
<tr>
<td>Aid</td>
<td>1 32 0</td>
<td>8 0 7</td>
</tr>
<tr>
<td>Alternate</td>
<td>0 1 0</td>
<td>1 0 0 0</td>
</tr>
<tr>
<td>Approaches</td>
<td>0 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>Aptitude</td>
<td>0 1 0</td>
<td>0 0 0 0</td>
</tr>
<tr>
<td>Arts</td>
<td>37 12 0</td>
<td>26 3 1 2</td>
</tr>
</tbody>
</table>

... Unemployment 0 1 0 1 0 0 4
| Universities | 25 6 0   | 8 0 0 2 |
| Vocations    | 0 0 0    | 1 0 0 0 |
| Yellow       | 1 3 0    | 12 2 0 1 |

\[ X = U \Sigma V \]  

for \( X = A \) and \( X = B \) and obtain the triangular matrix \( \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_{23}) \) of singular values, the document vector \( U \) and the term vector \( V \).

V. Evaluation

Fig. 4 shows the scree plots of the singular values for the naive method and for the wrapper method. It is clear that latter is much sharp than the former.

Fig. 4. Scree Plots of Singular Values

Fig. 5 and Fig. 6 are the two-dimensional plot of terms and sites. Two dimensional approximation \( U_2 \) and \( V_2 \) are obtained with principal singular vector and the first non-principal singular vector. The site \( S_i \) and the keyword \( W_j \) are projected onto the document vector \( D_i \) and to the term vector \( K_j \) as follows.

\[ D_i = (0, \ldots, 0, 1, 0, \ldots, 0)^{t} U_2 \Sigma_2^{-1}, \]
\[ K_j = (0, \ldots, 0, 1, 0, \ldots, 0) V_2 \Sigma_2^{-1}. \]  

Fig. 5. Two-dimensional plot of terms and sites for the naive method

Fig. 6. Two-dimensional plot of terms and sites for the wrapper method

In Fig. 6, the sites 1:Education & Research, 4:Native Education, 7:Billilingual, 5: Massachusetts Department of Education form a cluster along the line \( y = x \). These sites correspond to the principal component of “Education”. The secondary component lies along the line \( y = -x \) and contains the sites of 12:Yahoo!News, 13:News.au, 15:CNN and 14:NewSynthesis. On the other hand, in the scree plot Fig. 5 of the naive method, there clustered 7 sites of various kinds on the principal direction and they cannot be distinguished one from another. The secondary component contains only two sites (3:Yahoo!Education, 18:Yahoo!Reference) and four keywords (133:Yellow, 13:Shopping, 14:Category 85:Page).

Fig. 7 and Fig. 8 compare the values of components of the principal singular vector concerning to sites. Fig. 7 plots pairs \( (a_i, b_i) \) of \( i \)-th components of the principal vector for
sites \(i = 1, \cdots , 23\), where \(a_i\) is the value for the naive method and \(b_i\) is the value for wrapper method. The line represents the diagonal line \(y = x\). The sites near the line, i.e., 14 (NewsSynthesis.com), 15 (CNN financial network), 16 (Regional News Search), have the same importance for both method. The sites below the line decrease their importance in wrapper method. Such sites are 9 (North Dakota Economic Department of Economic Development and Finance), 3 (Yahoo! Education) 18 (Yahoo! Reference) and 23 (osOpinion). These sites display many general information surrounding the search result. These information influences the feature in the naive method. The wrapper eliminates these parts and the value of these sites decreases. It explains why the sites in the second principal component (Fig. 5) of naive method disappear in the wrapper method. Fig. 8 compares the values of the components of the principal singular vector concerning to keywords. The keywords below the line are 102 (Resume), 133 (Yellow), 14 (Category), 37 (Events) and 85 (Page). They decreases their importance for the wrapper method. This is because these keywords are very general and not related to the search result.

![Fig. 7. Comparison of Singular Vectors for Sites](image1)

![Fig. 8. Comparison of Singular Vectors for Keywords](image2)

VI. RELATED WORK

Integration of multiple search engines is known as a metasearch engine [8]. Most of their target are general search engines which may overlap each other. On the other hand, the target of integration in this paper are independent search sites which 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 filtering and ranking to the search results. Metasearch engines select appropriate search engines for each query [3], [2]. Therefore, the usefulness of a metasearch engine depends on selection of search engines and ranking of search result. Gravano et. al. [2] applied database selection techniques for each query. But they need vectorized information of each search sites in advance. Another difference of our approach to metasearch engines is that we integrate search sites for a particular purpose and not for a single query. Once you compile your favorite search sites, the integrated search sites becomes your personal search engine.

Each search site has different format of query and search result. To combine the search sites with one interface, we need wrappers for all search sites to be integrated. Kushmerick et. al. [6] introduced a learning algorithm to generate a wrapper from several examples. Our wrapper generation [9] is based on the observation that the search result contains repetition of the same tag sequence. So we do not need examples.

VII. CONCLUSION AND FURTHER WORK

We proposed the wrapper method, which is an automatic feature extraction method for search sites. Novelty of this method is to use the wrapper which is automatically generated for each site. We compare the naive method and the wrapper method for 23 sites and 133 keywords. We evaluated with the latent semantic indexing. We showed that the feature extracted by this method is much sharp compared with the naive method. This is due to the fact that the wrapper extracts the contents and eliminates decorative links and advertisements none of which are related to the keywords.

We can apply the feature extraction for clustering the search sites. We can map the search sites on the conceptual structure to help users select search sites for integration. Similar idea was applied in [7] to categorize documents using Yahoo! as an ontology. Invisible Web², SearchEngine Guide³ and BrightPlanet classify various search engines in hierarchies. But these organization is done manually. Ipeirotis et. al. [4] proposes a strategy of automatic classi-

²http://www.invisibleweb.com
³http://www.searchengineguide.com/
fication. Further research will be necessary to compare our method with others.

ACKNOWLEDGMENTS

The authors would like to acknowledge the supports of Tetsuya Nakatoh, Masato Noguchi, and Miyuki Sakai.

REFERENCES