EXTRACTING NEWS FROM SERVER SIDE DATABASES BY QUERY INTERFACES

HAO HAN
Kanagawa University
Hiratsuka, Kanagawa 259-1293, Japan

ABSTRACT

Web news has become an important information resource, and we can collect and analyze Web news to acquire desired information. In this paper, an effective and efficient Web-based knowledge acquisition approach is proposed for extracting Web news full content from news site databases using site-side news search engines as query interfaces. We do not crawl the news sites to collect news pages. Instead, we use news search engines affiliated to the news sites to search for the desired news articles directly from the news site databases. We give the search keywords to the search engines and extract the full content of the news articles without the process of machine learning or pattern matching. This approach is applicable to general news sites, and the experimental results show that it can extract a large amount of Web news content from news site databases automatically, quickly, and accurately.

Keywords: Web-based Tools, Knowledge Acquisition, Web News Application, Information Extraction, Site-side Search Engine, Query Interface, Database

INTRODUCTION

The Internet has marked this era as the information age. The Web is rapidly moving towards a platform for mass collaboration in content production and consumption, and the increasing number of people are turning to online source for their daily news. Web news has become an important information resource and traditional news papers have developed significant Web presences. Fresh content on a variety of topics, people, and places is being created and made available on the Web at breathtaking speed. We can collect and analyze these data to acquire the desired information/knowledge and the Web news article content extraction is vital to provide news indexing and searching services. For example, if we want to compare monthly information topics on different countries/regions for previous years from a designated news site such as CNN (http://www.cnn.com/), we need to collect the CNN news articles about each country/region and analyze the content to acquire the desired information (this is for personal use, not anti-copyright republication).

However, the process of collecting news pages is very time consuming. Usually, webpage crawlers are used to collect webpages. They are executed at regular intervals to collect links to the webpages from news sites, and the collection process has to last for a long period of time if we want to collect news pages covering a long time period. Not every collected webpage is usable because there are many non-news webpages, such as blog pages, advertisement pages, and even similar pages with different URLs. Sometimes, we just want to collect news articles on concerned topics such as news articles on “soccer” or “whaling”, and the other collected news articles are undesired. Furthermore, news sites are crawled to find as many news pages as possible, but, actually, it is difficult to acquire old news pages because the latest news is shown prior to the old news.

Also, in a news page, there are usually text parts of advertisements, related comments, as well as news titles and content. In order to recognize and extract the news content parts from the full text of news pages, extraction patterns are generated, based on the layout of news pages. Webpage layout is the style of graphic design in which text or pictures are set out on a webpage. Different news sites use different news page layouts and each news site uses more than one layout, as shown in Fig. 1. It is necessary to generate many news content extraction patterns manually or automatically for each news site. This procedure is costly. Moreover, news sites update the layout of news pages irregularly. If a news site updates the layout of its news pages, the corresponding analysis has to be done again.

FIGURE 1: BBC and CNN use different webpage layouts for news pages of the same category

It is therefore not easy to extract news content on specific topics from news sites quickly, and the current methods of news page collection and news content extraction do not work efficiently.

In this paper, we propose an approach to extracting news full content from news site databases quickly and automatically. Usually, news sites provide site-side news search engines for users. These engines are affiliated to the news sites and can directly access the news databases of news sites. We use these news search engines by query interfaces to search for news by giving the query keywords of desired topics, and extract the page URLs and titles of the matched news from the search result pages automatically. We then use an efficient extraction algorithm to extract the full content of the news, without webpage layout analysis. The target news sites, publication dates/periods (e.g., last week, this month, from 2010 to 2012), and query keywords can be designated. The approach is applicable to general news
sites and can extract a large amount of news, including old news published several years ago.

Our main purpose is to provide a practical and easy-to-use Web-based information acquisition tool for news-oriented research.

The organization of the rest of this paper is as follows. In Section 2 we give the motivation for our research and an overview of related work. In Sections 3, 4 and 5, we explain our topic-based Web information/knowledge extraction approach in detail. We test our approach and give an evaluation in Section 6. Finally we conclude our approach and identify future work in Section 7.

MOTIVATION AND RELATED WORK

In order to analyze and compare Web news articles on many major topics, we need to collect news articles on specific topics from one or many designated news sites over a long period of time. The target news sites, topics, and publication dates are selected, and the full content of news articles is extracted from matched news pages. In this paper, a topic is a discrete piece of content about a subject (e.g., countries, sports, companies) and represented by a series of query words in searching process. Some related work has been done on Web news articles collection or extraction. The first type of work is collection of news pages. News titles and URLs of news pages are extracted from news sites or search engines. The second type is extraction of news article full content. This recognizes and extracts paragraphs of news article content from the HTML documents of news pages.

For news page collection, webpage crawlers are often used. They are executed to collect news pages from news sites, and the collection process is time consuming. Several collection approaches and systems have been proposed.

There are increasing numbers of sites distributing news articles by news RSS feeds. Generally, news sites classify news articles into different topics/categories and publish them by RSS feeds. However, different news sites use different categories, and RSS feeds only contain the latest news articles. For example, CNN provides news RSS feeds by fields such as science, sports, etc., while AllAfrica (http://allafrica.com/) offers news RSS feeds grouped by countries/regions. HTML2RSS [15] can extract news titles and news page URLs automatically from news sites, but the extraction range is limited to news sites that consist of lists with special data structures, and users cannot obtain news on designated topics by giving search keywords.

AllInOneNews (http://www.allinonenews.com/) is a news search system based on automatic extraction of search results from search engines [20]. It passes each user query to existing news site search engines, and collects the search results for presentation to the user. However, users of this system cannot select the target news sites, and just collect results from the first search result page.

Yahoo! News (http://news.yahoo.com/) gives the news published between today and one month ago. MSN News and Bing News (http://www.bing.com/news) provide poor search options. Google News (http://news.google.com/) gives a relatively rich search option. However, there is (search result) page number limitation if target news site is assigned and publication date is customized. These methods/systems cannot satisfy the flexible and quick collection of news pages very well. Moreover, these methods cannot perform analysis or comparison of news articles because they cannot extract the full content of each news article.

They cannot easily answer questions like “which countries had a dispute over whaling in recent years and did other countries become involved in the discussions as the dispute continued?”

For news article content extraction, a number of approaches have been proposed for analyzing the layout of news pages with the purpose of manual or semi-automatic example-based information extraction pattern learning, and ultimately extracting news content from general news pages.

Crunch [8] uses HTML tag filters to retrieve content from the DOM trees of webpages, and Hai et al. use regular expressions to extract news content [9]. However, users have to spend much time configuring a desired filter or a regular expression after analyzing the HTML source documents of the webpages. XSLT [11] uses defined path patterns to find nodes that match given paths repeatedly and outputs data using information on the nodes and values of variables. Similarly, ANDES [14] is an XML-based methodology using manually created XSLT processors to perform data extraction. However, the paths need to be found manually from the HTML documents of webpages. IEPAD [2] uses an automatic pattern discovery system. The users select the target pattern that contains desired information from discovered candidate patterns, and then the extractor receives the target pattern and a webpage as input and applies a pattern-matching algorithm to recognize and extract the information. However, there are many different layout styles used in a news site and the users could not prepare all the extraction patterns for each site.

Reis et al. performed a calculation of the edit distance between two given trees for automatic extraction of Web news article content [5]. Fukumoto et al. focused on subject shifts and presented a method for extracting key paragraphs from documents that discuss the same event [7]. This method uses the results of event tracking, which starts from a few sample documents and finds all subsequent documents. However, if a news site uses too many different layouts in the news pages, the learning procedure costs too much time and the precision becomes low.

Zheng et al. presented a news page as a visual block tree and derived a composite visual feature set by extracting a series of visual features, then generated the wrapper for a news site by machine learning [21]. However, it uses manually labeled data for training and the extraction results may be inaccurate if the training set is not large enough. Similar problems may occur to those in [4] and [22].

Webstemmer [17] is a Web crawler and HTML layout analyzer that automatically extracts the main text of a news site and excludes banners, advertisements, and navigation links. It analyzes the layout of each page in a certain Web site and figures out where the main text is located. All the analysis can be done in a fully automatic manner with little human intervention. However, this approach runs slowly at content parsing and extraction, and sometimes news titles are missing. TSReC [12] provides a hybrid method for news article content extraction. It uses tag sequences and tree matching to detect parts of news article content from a target news site. However, for these methods, if a news site changes the layout of news pages, the analysis of layout or tag sequences has to be done again.

Some approaches analyze the features of news pages to generate wrappers for automatic or semi-automatic extraction. CoreEx [16] scores every node based on the amount of text, the number of links, and additional heuristics to detect news article content. However, it does not seem to deal with news pages with a lot of related information in text format, and may miss title information when the news article header appears far away
from the body. Dong et al. developed a generic Web news article content extraction approach based on a set of predefined tags[6]. This method is based on the assumption that the news pages from a news site use the same layout, but, actually, there are many different layouts used in a news site.

There are some layout-independent extraction approaches. TidyRead (http://www.tidyread.com/) and Readability (http://lab.arc90.com/experiments/readability/) render webpages with better readability, in an easy-to-read format, by extracting the context text and removing clutter. They run as plug-ins or bookmarklets of Web browsers. However, the extraction results consist of part of a webpage containing HTML tags. They also contain some other non-news elements such as advertisements and related links. Wang et al. proposed a wrapper to perform news article extraction using a very small number of training pages, based on machine-learning processes, from news sites [18]. The algorithm is based on the calculation of the rectangle sizes and word numbers of news titles and content. However, these approaches still need to set the values of some parameters manually, and could not be proved to extract news articles successfully or automatically if news sites update the layouts of news pages. Full-Text RSS (http://echodittolabs.org/fulltextrss) only returns news article content when the supplied RSS feed has a summary or description of some kind. ESE [19] gives a framework to digest news webpages in finer granularity: to extract event snippets from contexts. “Events” are atomic text snippets and a news article is constituted by more than one event snippet.

These news content extraction methods are still not widely used, mostly because of the need for a high level of human intervention or maintenance, and the low quality of the extraction results. Most of them have to analyze the news pages from a news site before they extract the news article content from this news site. If different target news sites, topics, and publication dates are selected, the analysis of layout needs to be done again. This is costly and inefficient.

Compared to these developed methods, our approach extracts not only the news titles and URLs of news pages, but also the full content of news articles. We use the news search engines affiliated to news sites instead of the often-used Web crawlers. We can get a large amount of news from the news site databases, not only the latest news but also old news. Furthermore, we do not need to delete non-news pages or other undesired news pages from the search results because all the news articles extracted from the search result pages satisfy our designated query keywords. We propose a partial information extraction algorithm specifically for news article content extraction. It is applicable to general news pages, and we do not need to analyze different types of news pages to generate the corresponding extraction patterns for each news site. The full news content is quickly extracted from the matched news pages for further analysis.

OVERVIEW

Our extraction system is made up of two main parts, as shown in Fig. 2: news page collection and news article full content extraction.

First, we collect the desired news pages. We create a submission emulator to emulate the submission process of the search engine of the target news site. We obtain the query keywords of a specific topic and send them to the emulator one by one, then extract news titles and the URLs of news pages from the continuous search result pages.

Secondly, we extract the news article content from the news pages. We employ a layout-independent Web news full contents extraction method based on relevance analysis, which can extract news article content from a news page using only the news title.

NEWS PAGE COLLECTION

We collect news pages from site-side query interfaces. Although many Web sites, such as Amazon, Yahoo, and YouTube, open their search engine services by Web service APIs, most news sites, such as CNN and the BBC, do not provide Web services for their news search engines. We have to extract partial information, such as news titles, news page URLs, and publication dates, from the news search result pages.

We use the following steps to collect the news pages, as shown in Fig. 3.

1. We generate a submission emulator for a designated target news site. We send the search keywords to the submission emulator and receive the news search result pages.
2. We analyze the search result pages to extract the news titles and news page URLs.

Submission Emulator

Usually, in a news Web site, a site-side query interface is provided to get the requests from users and return the search result pages. The users enter the query keywords into a form-input field using the keyboard and click the submit button by mouse to send the query. For request submission, there are POST methods and GET methods, and some news Web sites use encrypted codes or...
randomly generated codes. In order to get search result pages from all kinds of news sites automatically, we use HtmlUnit (http://htmlunit.sourceforge.net/) to emulate the submission operation instead of a URL templating mechanism.

We need to obtain the start webpage, which comprises the form-input field and the submit button of the query interface. Usually, this start webpage is the top page of the news site or a search result page of the news site. Then we analyze the HTML document of this webpage to find the <form> nodes. If a form fulfills the following criteria, we think it is a possible form which includes the necessary form-input field and submit button.

- It contains a text input field.
- There is a button next to this text input field.
- The server-side form handler of this form is within this news site.

If we find more than one possible form in this webpage, we choose the first one as our final selection because the target form is usually at the top side of the Web, as shown in Fig. 4.

We generate a submission emulator and send the search keyword to it. The submission emulator uses HtmlUnit to emulate the submission process (input the search keyword into the text field and click the button to complete the actual submission). Finally, we get the response page (search result page) from the submission process (input the search keyword into the text field and click the button to complete the actual submission).

All the processes of submission emulator generation in our approach are completed automatically after the start webpage is given.

**News Title and News Page URL Extraction**

After we get the search result page, we need to extract the news titles and news page URLs. As shown in Fig. 5, there are links to advertisement pages, video pages, external non-news pages, and other irrelevant information besides the page links to matched news articles. Fortunately, there are some similar features in the news search result pages of most news sites, which can be used to extract the news titles and links to the news pages.

- Each search result set contains similar information, such as news page link, news title, headline, and publication date, at a similar position.
- The news title is contained inside the news page link as text value.

- Matched news are listed in a column and spread over multi-pages.

All the news search result pages from a news site search engine have the same webpage layout.

We extract all the link nodes from the HTML document for the search result page, and find the news page link nodes. Through our analysis of search result pages of many news sites, we found that the news page link node has some common features in its text value and path. The path is an XPath expression used to select nodes or node sets in an HTML/XML document. We use HTMLParser (http://htmlparser.sourceforge.net/) as our HTML document parser, and calculate the possibility value of each link node using the following steps. Usually, a larger possibility value means that the corresponding link node is more likely to be a news page link node.

1. We split the text value of the link node into a word list \( WL \), using whitespaces as the delimiter, and get the length of \( WL \) as \( L_1 \) \((L_1 \geq 1)\).
2. We calculate the number of occurrences of a search keyword in \( WL \) as \( L_2 \) \((L_2 \geq 0)\).
3. We get the path of the link node, including the ID and class value. We calculate the number of occurrences of “news” and “search”, and “result” in the path. We sum these three values to get the total value \( L_3 \) \((L_3 \geq 0)\).
4. We calculate the possibility value of each link as \( P \), using the formula

\[
P = \frac{L_1 \times (L_2 + \alpha)}{L_2 \times (L_3 + \beta)}
\]

where \( \alpha = 0 \) if \( L_2 > 0 \), and \( \alpha = 1 \) if \( L_2 = 0 \). Similarly, \( \beta = 0 \) if \( L_3 > 0 \), and \( \beta = 1 \) if \( L_3 = 0 \). We also used a least-squares method to analyze and calculate the data collected from many search result pages manually to acquire the most fitting \( \alpha \) and \( \beta \). However, in the actual experiments, they did not give noticeably higher precision. We cannot ensure that the search keyword occurs in the news title because the search range contains not only the news title but also the news content, which is not visible in the search result page. Also, the value of \( L_3 \) may be 0 in many news sites. We use \( \alpha \) and \( \beta \) to avoid the possible occurrence of \( P=0 \). They work well and do not result in negatives in actual extractions in our experiments.

In some news search result pages, the link node with the largest possibility value is not a link node of the news page (e.g.,
We think the final selected child node is the link node of a news page. We use the path of this node to extract the list of nodes with similar paths by the following steps since the nodes of news page links have similar paths.

1. We use the path to find the corresponding node $N$.
2. We get the parent node $P$ of $N$.
3. We get the subtree $S$ whose root node is $P$.
4. We get the node list $L$ in which each node has the same path as $N$, without considering the orders of child nodes of $P$ under $S$.

a link to contextual advertising or a blog). Usually, the news page links are listed in similar structures (XPath) and not mixed with other non-news links. We use the following steps to detect the news page link nodes range and find a news page link node.

1. We sum the possibility values $P_N$ of all the link nodes, and get the root-mean-square $R_P$ as a threshold:

$$R_P = \sqrt{\frac{\sum P_N^2}{|N|}}$$

Where, $|N|$ is the sum of link nodes.

2. For each node $N$, we calculate $P_N$ as follows:

$$P_N = \begin{cases} P_N & (N \text{ is a link node}) \\ \sum_{n \in \text{Child}_N} P_n & (P_n > R_P) \end{cases}$$

Where, $\text{Child}_N$ is a set of child nodes of node $N$. Fig. 6 shows an example of calculation of $P$.

3. We select the child node whose value is the largest among the sibling nodes from the root node to the leaf nodes, as shown in Fig. 7.

The news search results are spread over multi-pages and we need to extract the page number links for our continuous query and extraction. The extraction of these links has to satisfy the following rules.

1. The text values of links are a series of numbers such as 1, 2, 3, . . .
2. The $\text{href}$ attribute values of links have similar lengths.
3. The links are listed in a row.

We use the paths of news page link nodes to extract the news page link nodes directly from the second result page. If we search for the news for many search keywords continually in a news site, we use these paths to extract the news page links and page number links directly.

### Publication Date Extraction

The publication dates of news articles are necessary if we collect the news articles for a specific period. For example, we need to extract the news articles on baseball for the last five years if we want to find out which team was the annual focus of attention in the last five years.

Different news sites choose different formats for date information such as "September 7, 2012", "Sep. 7, 12", and "2012-9-7". Table 1 shows the most-used format patterns for date information.

We use these patterns to find the publication dates in search results. Usually, a news site displays the publication date at the same position in a search result and uses the same pattern for all publication dates in all search results. After we find a publication date in a search result, we can use similar paths and the same format pattern as this found date to extract other
news article publication dates from search results easily as described in [13].

**NEWS FULL CONTENT EXTRACTION**

We extract the full content of news articles from our collected news pages. The news pages from different news sites use different page layouts and news sites update their news page layout irregularly. We propose our news article content extraction algorithm, which is independent of the layout of news pages and applicable to general news pages. We detect the position of a news title in the news page and extract the body of the news article (paragraphs of news article content).

The first process detects the position of a news title in the obtained news page. The news title is an important piece of information for the recognition of news content from the full text of news pages. If we correctly locate the position of the title in a news page, the position of the news content text is easily found because the content text is a list of paragraphs, usually closely preceded by the title. In addition, for a news article, the content describes the same topic as the news title in detail, and the words constituting the title usually occur frequently in the news content.

Usually, the real news title in a news page is the same as the news title in the search result page. However, this is not always the case. In some news sites, we find that the news title in the search result page is totally different from the real news title, but is the same as some other character strings in the news page, such as the related links, as shown in Fig. 9. Moreover, some news titles are so short and simple that we can find two or more strings the same in the news pages. An exact match is therefore not appropriate for this process. Instead, we calculate the similarity score between each string (node) of the news page and the news title extracted from the search result page. If the score is higher than a predetermined threshold \( \text{Sim} \), the string covered by the node is judged to be a news title. If there is no node whose score is higher than \( \text{Sim} \), no string is judged to be the news title. On the other hand, if there is more than one node with a higher score than \( \text{Sim} \), all of the strings covered by the nodes are judged to be news titles.

The second process detects part of the news article body and extracts the whole body. Since the body of a news article is usually preceded by its title, the process first tries to find the news article body in a “content range”, and, if it cannot find the body in the range, it tries to find the body in a “reserve range”. The “content range” and “reserve range” are parts which might include the news article body. They are determined as follows.

1. If only one string is judged to be a news title in the previous process, the following part and the preceding part are a content range and a reserve range, respectively (see Fig. 10(a)).
2. If no string is judged to be a news title, the whole of the news article page is a content range and no reserve range exists (see Fig. 10(b)).
3. If more than one string is judged to be a news title, for each of the strings except the last string, the range between itself and the next string is a content range. The part preceded by the last string is also a content range. The part followed by the first string is a reserve range (see Fig. 10(c)).

First, we specify a part of the news article body. Then we calculate the possibility score of each leaf node with non-linked text in each of the content ranges. If there are some nodes with higher scores than a predetermined threshold \( \text{Pos} \), we consider that the node with the highest score covers part of the news article body. Otherwise, we consider that the node with the highest score in the reserve range covers part of the news article body. Since a news article body is usually a continuous text, it can be extracted by taking leaf nodes around the specified nodes. However, in some

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Element</th>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>YY</td>
<td>Year</td>
<td>2-digit number</td>
<td>12</td>
</tr>
<tr>
<td>YYYY</td>
<td>Year</td>
<td>4-digit number</td>
<td>2012</td>
</tr>
<tr>
<td>MMM</td>
<td>Month</td>
<td>Test</td>
<td>Sep., September</td>
</tr>
<tr>
<td>MM</td>
<td>Month</td>
<td>Number</td>
<td>9, 09</td>
</tr>
<tr>
<td>DD</td>
<td>Day</td>
<td>Number</td>
<td>7, 07</td>
</tr>
<tr>
<td>Mark</td>
<td>Delimiter</td>
<td>Character</td>
<td>, /</td>
</tr>
</tbody>
</table>

**FIGURE 9:** News title in a CNN search result page is different from the real news title, but the same as a related link in the news page.

**FIGURE 10:** Content range and reserve range.
cases, some information which is not related to the article, such as advertisements, is inserted in the article body. In order to avoid taking such information, we also set limits to filter them. Finally, we get a list of nodes which cover the whole news article body. The whole body can be extracted by getting the node value (text) from each node in the list. After we analyzed a large number of news pages from many news sites, we set the Sim and Pos as 0.6 and 100 respectively based on the statistical results [10].

EVALUATION

In this section, we describe experiments to test our algorithms and analyze the experimental results in order to evaluate our approach. We use the news sites listed in Table 2 as our test bed. These news sites are the most popular on-line news publishers, and include global and domestic news sites.

Experiment 1

We selected the countries/regions and their leaders to be used as our test topics. There are 242 countries/regions in the world and most of them have leaders [1]. We used these country/region names and leader names as our search keywords. We collected the news page URLs and titles from 10 (No. 1--No. 10; if the total number of pages was less than 10, we took them all) result pages for each keyword. As shown in Table 2, Page Collection is the average execution time of extracting the news page URLs and titles from one result page (including the submission emulation process), and Content Extraction is the average execution time of extracting the news content from one news page. The submission emulations of two news sites (the Washington Post and Xinhuanet) failed, and the corresponding Content Extraction values are calculated by extracting news content from manually collected/saved result pages. We selected 500 news page URLs randomly and checked them manually one by one, and found that 17 news pages could not be obtained (the server responded with a message such as “page not found”).

Experiment 2

We sent the keywords used in Experiment 1 to the submission emulator, one by one, and extracted 96,095 news titles and page URLs of matched news (published from January 1, 2003 to December 31, 2007, a randomly generated period) from abovementioned news databases. Our computer (CPU: Intel Pentium M 1.30 GHz, Memory: 1.0 GB RAM, Network: 54.0 Mbps Wireless) took about 20 hours to complete this extraction process. We selected 200 news page URLs randomly and checked them manually one by one. The experimental results are listed in Table 3. We found that two news pages could not be obtained, for the reasons described in Experiment 1. Of the remaining 198 news pages, the news article contents of 192 news pages were extracted correctly. In the six extraction failures, some parts of the news article content were not extracted.

Experiment 3

We have crawled and extracted more than 10 million news contents from 38 well-known news sites [10] since May, 2007. We selected 1000 or 500 news articles randomly and checked them manually one by one. We ran this test four times and the experimental results are listed in Table 4.

<table>
<thead>
<tr>
<th>Country/Region</th>
<th>News Site</th>
<th>Page Collection (s)</th>
<th>Extraction (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>CNN</td>
<td>14.3</td>
<td>6.02</td>
</tr>
<tr>
<td></td>
<td>New York Times</td>
<td>4.6</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>Washington Post</td>
<td>x</td>
<td>(6.62)</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>BBC</td>
<td>3.6</td>
<td>3.19</td>
</tr>
<tr>
<td>Africa</td>
<td>All Africa</td>
<td>7.3</td>
<td>2.98</td>
</tr>
<tr>
<td>China</td>
<td>Xinhuanet</td>
<td>x</td>
<td>(2.94)</td>
</tr>
<tr>
<td>France</td>
<td>France24</td>
<td>8.6</td>
<td>3.67</td>
</tr>
<tr>
<td>Japan</td>
<td>Mainichi Daily News</td>
<td>9.3</td>
<td>2.40</td>
</tr>
<tr>
<td>South Korea</td>
<td>Chosun Ilbo</td>
<td>6.6</td>
<td>0.52</td>
</tr>
</tbody>
</table>

TABLE 2: List of news sites and execution times

<table>
<thead>
<tr>
<th>No.</th>
<th>Sum</th>
<th>Success</th>
<th>Failure</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
<td>970</td>
<td>30</td>
<td>97.0%</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>491</td>
<td>9</td>
<td>98.2%</td>
</tr>
<tr>
<td>3</td>
<td>500</td>
<td>485</td>
<td>15</td>
<td>97.0%</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>488</td>
<td>12</td>
<td>97.6%</td>
</tr>
<tr>
<td>Total</td>
<td>2500</td>
<td>2434</td>
<td>66</td>
<td>97.4%</td>
</tr>
</tbody>
</table>

TABLE 3: Extraction results for Experiment 2

<table>
<thead>
<tr>
<th>No.</th>
<th>Sum</th>
<th>Success</th>
<th>Failure</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000</td>
<td>970</td>
<td>30</td>
<td>97.0%</td>
</tr>
<tr>
<td>2</td>
<td>500</td>
<td>491</td>
<td>9</td>
<td>98.2%</td>
</tr>
<tr>
<td>3</td>
<td>500</td>
<td>485</td>
<td>15</td>
<td>97.0%</td>
</tr>
<tr>
<td>4</td>
<td>500</td>
<td>488</td>
<td>12</td>
<td>97.6%</td>
</tr>
<tr>
<td>Total</td>
<td>2500</td>
<td>2434</td>
<td>66</td>
<td>97.4%</td>
</tr>
</tbody>
</table>

TABLE 4: Extraction results for Experiment 3

Analysis and Evaluation

We use the first and second experiments to test our submission emulator and news page collection algorithm. The results prove that our approach can easily and quickly extract the news titles and URLs of news pages covering a long period from news site databases. Our approach is applicable to general news site search engines and does not need methods such as machine learning or extraction pattern matching, which are time consuming when news sites change the layout of search result pages. However, some news sites use external JavaScript files comprising complicated JavaScript functions to realize submission requests, or minor syntax errors occur in webpages where the search keywords are input. Although most current Web browsers such as Firefox and Internet Explorer can run smoothly on these webpages, our submission emulator still cannot emulate these types of submission processes. We think this is the result of a bug in HtmlUnit and hope that the new version will solve this problem in the future. Furthermore, the emulation processes of some news sites run slowly. For example, emulation of submission of search keywords to the CNN news search engine costs about 10 seconds. Moreover, some old news pages are not obtained although their URLs and news titles are shown in news search result pages.

We use the second and third experiments to test our news full content extraction algorithm. We use a large amount of news to test our algorithm and the experimental results prove that our
The extraction algorithm is highly accurate over long time periods. Although news sites change the layout of news pages irregularly, our extraction method works well and the extraction precision is over 97%. However, in some news pages, a paragraph, usually a news outline, is shown in a different style to that used for the other paragraphs. This type of paragraph looks like a non-news part, such as an advertisement in text format, and is omitted in the extraction. Moreover, some news content is too short to be recognized from the news pages. For example, a news flash about a baseball game result which contains just a short paragraph of 10 words may not be extracted correctly.

Compared with other developed extraction systems, our extraction approach has the following strengths.

1. Our extraction system is easily constructed, even for users who know little about information extraction technologies. The extraction processes such as submission emulator generation, news page collection, and news content extraction run automatically. The system needs little maintenance during long extraction periods. We do not need to analyze the layout of the search result pages and news pages of news sites since our extraction algorithm is independent of the layout of webpages. It does not need to reconfigure extraction, even if the news sites change the layout of news pages.

2. Our extraction system supports the designation of news collection/extraction ranges, such as the target news site, news topic, and publication date. By analysis of extracted news content, we can compare the viewpoints regarding a topic among different news sites, see monthly/yearly variations in a topic, observe co-occurrence of one or two country names, and find other useful information/knowledge. For example, we extracted and analyzed old news to find past trends, as shown in Table 5 (numbers enclosed in parentheses indicate the number of corresponding Web news articles). We counted the monthly frequency of country names together with a topic, namely “whaling”, in CNN news published from July 2007 to December 2007. Japan and some other pro-whaling countries are in dispute over whaling with Australia and some other anti-whaling countries. The results reflect the following: “whaling” news in Japan and Australia increased in November and December. Additionally, we find that some other countries, such as the USA, may also be involved in the arguments. Table 6 shows monthly frequencies of country/region names together with “Japan”. This reflects Japan’s international relations (e.g., trade with neighboring countries/regions or “six-party talks”). The co-occurrence of Japan and Myanmar in October 2007 is also higher than in other periods. In this period, Japan canceled a grant to Myanmar to protest against the nation’s crackdown on pro-democracy demonstrators.

3. Our extraction system runs quickly because the algorithms are simple and efficient. For the extraction of a large amount of news, a simple algorithm of low computational complexity saves a considerable amount of time. For example, content extraction from a CNN news page costs 6.02 ms on average (excluding reading news pages from news sites and saving the extraction results into a local hard disk), which is more efficient than other developed methods.

4. Our extraction approach is not limited to the English news sites. Using the morphological analysis tools like ChaSen [3], we also could extract the news article contents from Japanese news sites well (we do not use the translation technology to translate them into English because of the low precision of current translation tools).

**CONCLUSION**

In this paper, we have presented an effective and efficient approach to quick and automatic extraction of Web news full content from news site databases using site-side query interfaces.

### TABLE 5: Trends in topic “whaling” during July 2007 and December 2007 (country names and the number of news)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UK (13)</td>
<td>UK (14)</td>
<td>USA (7)</td>
<td>Australia (20)</td>
<td>Japan (62)</td>
<td>Japan (96)</td>
</tr>
<tr>
<td>2</td>
<td>USA (10)</td>
<td>USA (6)</td>
<td>UK (4)</td>
<td>Canada (7)</td>
<td>Australia (21)</td>
<td>Australia (60)</td>
</tr>
<tr>
<td>3</td>
<td>Australia (8)</td>
<td>China (6)</td>
<td>Canada (4)</td>
<td>South Africa (5)</td>
<td>Brazil (13)</td>
<td>USA (22)</td>
</tr>
<tr>
<td>4</td>
<td>Japan (6)</td>
<td>Japan (3)</td>
<td>Japan (3)</td>
<td>Georgia (4)</td>
<td>UK (9)</td>
<td>Canada (6)</td>
</tr>
<tr>
<td>5</td>
<td>Russia (5)</td>
<td>Colombia (3)</td>
<td>Australia (2)</td>
<td>UK (4)</td>
<td>USA (8)</td>
<td>UK (4)</td>
</tr>
</tbody>
</table>

### TABLE 6: Co-occurrence of country/region names with “Japan” during July 2007 and December 2007 (country names and the number of news items)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>USA (147)</td>
<td>USA (191)</td>
<td>USA (116)</td>
<td>USA (162)</td>
<td>USA (199)</td>
<td>USA (130)</td>
</tr>
<tr>
<td>2</td>
<td>Australia (72)</td>
<td>China (119)</td>
<td>China (81)</td>
<td>China (60)</td>
<td>China (69)</td>
<td>China (112)</td>
</tr>
<tr>
<td>3</td>
<td>China (66)</td>
<td>Australia (54)</td>
<td>Australia (75)</td>
<td>Myanmar (44)</td>
<td>Australia (69)</td>
<td>Australia (70)</td>
</tr>
<tr>
<td>4</td>
<td>South Korea (42)</td>
<td>India (30)</td>
<td>UK (37)</td>
<td>Australia (36)</td>
<td>UK (36)</td>
<td>Russia (25)</td>
</tr>
<tr>
<td>5</td>
<td>North Korea (35)</td>
<td>Germany (24)</td>
<td>Afghanistan (28)</td>
<td>UK (34)</td>
<td>North Korea (34)</td>
<td>South Korea (23)</td>
</tr>
</tbody>
</table>
We proposed an algorithm for extracting news titles and news page URLs from search result pages. We also proposed an algorithm for extracting news full content from news pages. Our extraction methods are applicable to general news sites. All the extraction processes are completed automatically. Our experimental results on several news sites show that our extraction system works well, and the proposed approach is very promising.

In future work, we will modify our algorithm to improve the accuracy rate even further and solve possible information overload, and observe differences in various topics among countries/regions to obtain useful information/knowledge from news sites. Moreover, we will extend our approach to different kinds of information/knowledge sites and construct the corresponding analytical systems.

ACKNOWLEDGEMENT

We gratefully acknowledge the advice and support from Bin Liu (Yahoo! Japan), Takehiro Tokuda (Tokyo Institute of Technology, Japan) and Keizo Oyama (National Institute of Informatics, Japan). This work was partially supported by a Grant-in-Aid for Scientific Research A (No.22240007) from the Japan Society for the Promotion of Science (JSPS).

REFERENCE