Print-Friendly Page Extraction for Web Printing Service

Sam Liu
HP Labs
1501 Page Mill Rd
Palo Alto, CA 94304 USA
sam.liu@hp.com

Cong-Lei Yao
HP Labs - China
No. 1 Zhong Guan Cun East Rd
Beijing 100084, China
conglei.yao@hp.com

ABSTRACT

Printing Web pages from browsers usually results in unsatisfactory printouts because the pages are typically ill formatted and contain non-informative content such as navigation menu and ads. Thus, print-worthy Web pages such as articles generally contain hyperlinks (or links) that lead to print-friendly pages containing the salient content. For a more desirable Web printing experience, the main Web content should be extracted to produce well formatted pages. This paper describes a cloud service based on automatic content extraction and repurposing from print-friendly pages for Web printing. Content extraction from print-friendly pages is simpler and more reliable than from the original pages, but there are many variations of the print-link representations in HTML that make robust print-link detection more difficult than it first appears. First, the link can be text-based, image-based, or both. For example, there is a lexicon of phrases used to indicate print-friendly pages, such as *"print"*, *"print article"*, *"print-friendly version"*, etc. In addition, some links use printer-resembling image icons with or without a print phrase present. To complicate matter further, not all of the links contain a valid URL, but instead the pages are dynamically generated either by the client Javascript or by the server, so that no URL is present. Experimental results suggest that our solution is capable of achieving over 99% precision and 97% recall performance measures for print-friendly link extraction.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Service – *commercial services, web-based services.*

General Terms

Algorithms, Design, Experimentation.

Keywords

Web content extraction, Web printing, HTML, DOM.

1. INTRODUCTION

The World Wide Web (WWW) has become the platform of choice for publishers to distribute content to their audiences. A Web page, however, typically includes auxiliary information that has little association with the main content, such as navigation menu and ads. The auxiliary content is generally considered "noise" by information retrieval (IR) systems, and should be removed for Web data mining [2-4]. Recently, there is work in content extraction to provide a better Web printing experience, mainly motivated by trying to improve the poor quality prints rendered by the Web browser [5]. A desired printout would contain only the main content and aesthetically layout in a popular file format, e.g. PDF. A robust solution for accurate Web content extraction, however, is still elusive given the complexity of modern Web pages. Recognizing the Web printing problem, many publishers are now providing print-friendly pages via print-links, especially for pages that are more likely to be printed, such as articles. These print-friendly pages contain essentially the same main content as the original page but in a much cleaner and simpler format, as illustrated in Figure 1. Thus far, all the effort on content extraction is devoted to the original page, which in general is a difficult problem. We propose instead using the print-friendly page as an alternative for content extraction when it is available. This requires the detection of the print-link and the extraction of the print-friendly URL. The key challenge here is to find a solution that can achieve very high extraction accuracy since an incorrect link would lead to a page with the wrong content, which is a "catastrophic" mistake. The target application for this technology is a cloud-based Web printing service as illustrated in Figure 2, but it can also be used for other applications such as search and indexing.

Figure 1a: Example of a news article page

DocEng'11, September 19–22, 2011, Mountain View, California, USA.

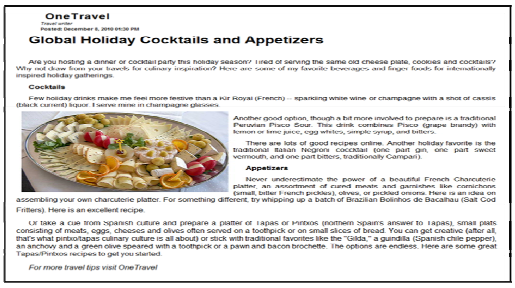

Figure 1b: Example of the print-friendly page

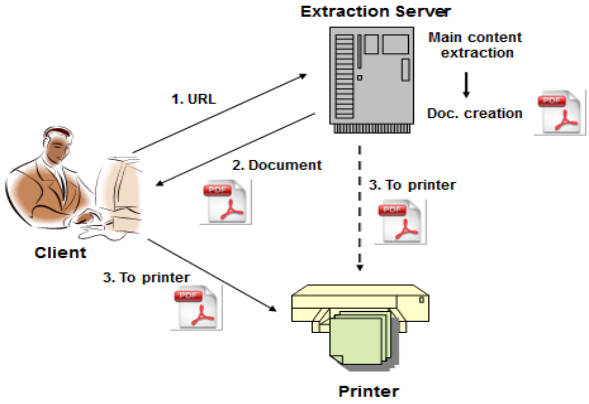

Figure 2: Cloud based Web printing service

As shown in Figure 3, there are many variations of the print-link expressions, making the detection problem more difficult than initially thought. First, the link content between the *"<a>…"* (hyperlink)tags can be text-based, image-based, or both, and the text expression can be a lexicon of phrases of many possibilities, such as *"print"*, *"print article"*, *"print-friendly version"*, etc. Furthermore, some pages use printer-resembling image icons with or without a print phrase present. It turns out that not all the links contain valid URLs because the print-friendly pages can be dynamically generated either by the client Javascript or by the server upon request.

As mentioned earlier, the goal of the system is to provide high extraction accuracy or precision (% of extracted URLs that are correct). In general, systems that require high precision usually have to relax the recall (% of URLs correctly extracted relative to all the valid URLs in the dataset), but experiments show that our solution can maintain a precision of over 99% while achieving a recall of over 97%, a very high performance measure.

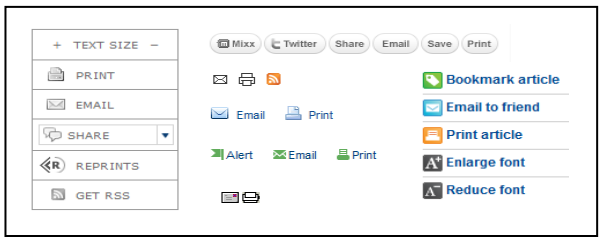

Figure 3: Examples of print-links from various Web pages

2. SYSTEM DESCRIPTION

2.1 Overall Architecture

The processing pipeline of a Web article extraction and reformatting system for Web printing is illustrated in Figure 4. To summarize, the first stage of the system uses the method described in reference [5] to extract the article components such as the title, text body, the associated image and caption. The input to the system is a Web HTML page from which we first attempt to acquire the print-friendly page URL, but if it is not available, content extraction would resort back to the original page. After the article components have been extracted, they are reassembled to create the final aesthetic document in PDF format.

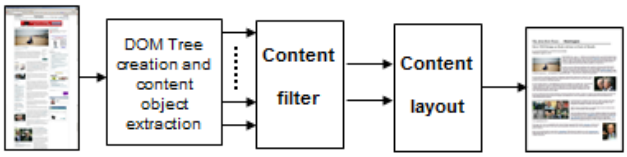

Figure 4: Block diagram of print-link detection

Our solution to the print-link extraction problem places particular importance on the correctness (precision) of the extracted URL, thus we impose a precision target of over 99% while maximizing the recall. The print-link detection strategy uses a print-phrase dictionary to find an exact match with the link text content or the link/image attribute values (described in more details in Section 2.3). It is important to populate the print phrase dictionary with an appropriate set of phrases since it impacts both the precision and recall. We can control the precision performance by using only print-phrases encountered in actual Web pages, and maximize the recall by having a comprehensive dictionary. By using exact print-phrases that only appear on Web pages, there is little chance that the match would result in a false alarm. To further improve the precision, we also compare the extracted (a real URL exists) and original URL domain names. We found that the print-friendly page is also typically archived in the same server as the original page and share the same domain name. It turns out, however, that this is not always the case, but the exceptions are very rare and do not impact the precision (only lower the recall). If a print-friendly URL cannot be extracted, either because there is no print-link on the page or the print-friendly URL is not valid, content extraction would resort back to the original page.

The print-link detection and extraction system basically takes on two stages: (1) detection of the print-link using the DOM (Document Object Model), as illustrated in Figure 5, and (2) print-friendly URL retrieval from the link attributes and the test for its validity, as illustrated in Figure 6. In the following sections, we will provide more details of the various components of the system.

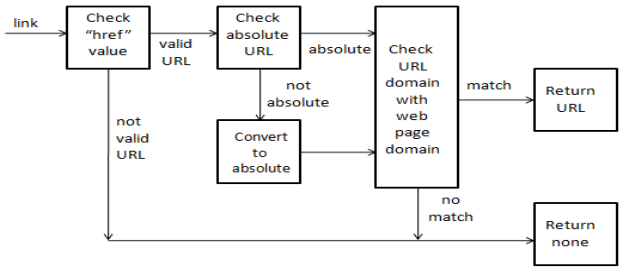

Figure 5: Block diagram of print-link detection

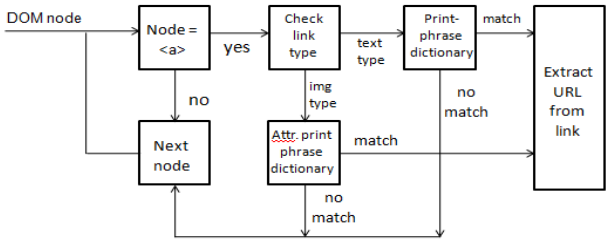

Figure 6: Block diagram of URL extraction and validation

2.2 DOM – Document Object Model

To find the print-link of a Web page, we need to examine the content of the link tag *"<a>"* in the HTML file for a print-phrase. One method of finding the *"<a>"* tags is to simply use text matching, but we instead choose the DOM approach because the DOM is a data structure that captures all the HTML information while avoiding all non-HTML syntaxes such as Javascript [1]. Every HTML file can be mapped to a DOM, which given the embedding nature of HTML, the DOM is a hierarchical data structure representing the organization of the various content objects in the HTML file. Thus the DOM is a data structure that is much cleaner and suited for efficient analysis and manipulation of the Web content than the original HTML file. Each node of the DOM tree is an HTML-Tag element, so we simply just walk the tree to search and examine the *"<a>"* nodes.

2.3 Print-Link Types

We found that all print links in our dataset belong to one of the following types in the HTML representation: (1) *"<a>print-phrase"*, (2) *"<a><img>"*, (3) *"<a><img>print-phrase"*, and (4) *"<a><img>print-phrase"*, where the *"<a>"* and *"<img>"* are the HTML (hyper)link and image tags, respectively (note the attributes are not shown). Examples of the print-phrases are *"print"*, *"print this article"*, *"print story"*, etc. There can be many variations of the print-phrases but they all convey the notion of a print-friendly page. Note that in type 2, there is no print-phrase (only an image icon), but a print-phrase is typically embedded in the attributes of the *"<img>"* or *"<a>"* tag, such as the *"alt"* or *"title"* attribute. Our print-link detection strategy is to simply use a print-phrase dictionary to find an exact match with the link text content or the image attribute values. Again, it is very important to populate this dictionary with the appropriate set of print-phrases since it impacts both the precision and recall. In the following sections, we will provide the details of the four types of print-link expressions found in Web pages.

2.3.1 Type1: <a>print phrase

This is the simplest and the most common type of the four print-link expressions. The link is just represented by a print-phrase, e.g. *"<a> print-phrase "*, as illustrated in following HTML code fragment from a real Web page.

"Print"

The print-phrase in this example is simply *"Print"*, but as mentioned earlier it can take on many expressions. *"Print"* is the most common print-phrase, used by over 80% of the Web pages containing a print-link. A print-phrase is also typically found in the attributes of "<a>" tag, but for this type, we only examine the link text content for the print-phrase.

2.3.2 Type1: <a><img>

The Type 2 print-link expression contains only an image between the *"<a>…"* tags, as illustrated in the following real HTML code fragment.

""

From this example, we see that there is no text content to indicate that this is a print-link (the icon image is providing that information visually). Notice, however, that the *"title"* attribute value of the *"<img>"* tag does contain a print phrase, "*Print This Article*", that can be used to determine this link is indeed a print-link. We also found that the *"alt"* is another attribute that commonly contains a print-phrase. Thus, for image-only print-links, we need to examine the *"title"* and *"alt"* attributes for print-link detection. Our solution, however, is not capable of detecting the print-link if none of these attributes contain a print-phrase, but the miss only impacts recall (precision is not changed).

2.3.3 Type1: <a><img>print phrase

Type 3 print-link expressions simply contain both an image and text print phrase between the *"<a>…"* tags, as illustrated below.

"Print"

Note that this example suggests that we can search for a print-phrase in the link text content or in the attributes of the *"<img>"* tag. For this type, however, we just test for the print-phrase in the link text content, similar to Type 1.

2.3.4 Type1: <a><img> print phrase

The last type is when an image is the link, but the print-phrase is outside of the *"<a>…"* tags, as shown below.

"<img src="/images/print_friendly.gif" width="29" height="15" border="0"> Printer-friendly version"

As in type 2, if the attributes of the *"<img>"* tag contain a print-phrase, then the print-link can be detected. If not, we need to examine the text node adjacent to the *"<a>"* node of the DOM tree. This case, however, is extremely rare. Out of the two 2000 sample Web pages, we only found one such case, so we just apply the Type 2 detection scheme to this type.

2.4 Print Phrase Dictionary

To test for print-link, we simply search for a print-phrase in the text content of the *"<a>"* tag for Type 1 and 3, or the attribute values of the *"<a>"* and *"<img>"* tags for Type 2 and 4. We found that the *"alt"* and *"title"* attributes are the best candidates containing a print-phrase, so no other attribute is examined. Based on empirical studies of over 2000 Web pages, we found that a solution is not robust if it just searches for the word or word fragment *"print"* as the print-phrase. It turns out that there are links with text content such as *"printers"*, *"print run"*, or *"reprint"* that are not true print-links. As mentioned earlier, achieving very high precision is the key objective since any incorrectly extracted URL would lead to a page with the wrong content. Thus to ensure high precision, we use a dictionary containing the exact print-phrases we collected from the dataset

(over 2000 pages from the top 50 popular news sites), and requires an exact match between the link text content (or attribute value) and a print-phrase in the dictionary. To maximize the chances of capturing all of the print-links (recall), the dictionary should contain a comprehensive set of print-phrases from a large enough dataset. Currently, the dictionary size is greater than 20, including the most popular phrases such as *"print"*, *"print article"*, *"printable version"*, and the lesser used phrases such as *"print it"* and *"print story"*. Since the print phrases are simply to convey the notion of print-friendly pages, we do not expect the dictionary to evolve over time. Also note that it is straight forward to internationalize the dictionary to support other languages. With this dictionary, we are able to achieve very high precision and recall performance measures.

2.5 Print-Link URL

The *"href"* attribute value of the *"<a>"* contains the print-link URL (absolute or relative), but not all them are valid. Some of the pages are actually dynamically generated either by the client Javascript or by the server upon request, so no pre-generated print-friendly HTML page is available for content extraction. The following HTML code fragments are examples of print-links with no valid URLs.

"Print" or *"print this story"*

From examining various Web pages with dynamically generated print-friendly pages, the value of the *"href"* attribute usually contains these text symbols or words: *"#", "javascript", or "funct(...)"*, where *"funct"* denotes a Javascript function call. To determine whether a URL is valid, we can simply test for the following text symbols in the in the *"href"* attribute value: *"#", "javascript", "(",* or *")"*.

2.6 URL Domain Validation

Since the goal of print-link detection is to achieve the highest precision possible, we require the extracted URL (non-dynamically generated print-friendly pages) to have the same domain name as the URL of the original Web page. The assumption is that if the extracted URL is indeed the print-friendly page, this page should be pre-generated and archived in the same server, sharing the same domain name as the original page. For example, the following URL is that of an original Web page, "*http://www.foxnews.com/leisure/2010/12/06/gift-guide-gourmands/"*, and the print-friendly page URL is "*http://www.foxnews.com/leisure/2010/12/06/gift-guide-gourmands/print"*, which has the same domain name *"foxnews"* as the original page. By imposing this requirement, it can hurt the recall performance, but based on the evaluation of the dataset, it is very rare (1 out of 2000) that the two pages do not share the same domain name. It is also good to know that even though the recall might be impacted, the precision is not. Also note that the URL in the *"href"* attribute can be relative, which by default it is in the same domain. The overall URL extraction and validation flow diagram is illustrated in Figure 4.

3. EXPERIMENTAL RESULTS

To evaluate our print-link detection and extraction scheme, we collected roughly 2000 Web pages from the top 50 popular news sites. The ground-truths of the print-links and the print-friendly URLs are manually labeled using multiple subjects. As mentioned earlier, for quantitative performance measures, we use the traditional definition of precision (% of extracted URLs that are correct) and recall (% of URLs correctly extracted relative to all the valid URLs). An evaluation system has been build based on Java, using the Java *swing* package for HTML parsing as an efficient method to create the DOM. Our experiment shows that to achieve high precision and recall it is critical that the print-phrase dictionary be comprehensive, containing specific print-phrases for full phrase matching. The dictionary size is currently greater than 20, and we do not anticipate it to grow much, if at all, as we continue to collect more Web pages. With this system, we are able to achieve a precision of over 99% and recall of over 97%. All the false alarms in the precision are caused by miss-classification of invalid URLs as real URLs. With such high precision and recall measures, the system is highly accurate and robust for print-link detection and extraction.

4. SUMMARY

This paper presents a highly robust solution to print-link detection and print-friendly URL extraction from Web pages. The targeted application is a Web printing service, but the technology is applicable to general Web content extraction for search and indexing, data management, targeted advertisement, and more. The print-friendly pages, which are pages created for better print outputs, can be used as a cleaner and simpler alternative for main content extraction. We estimate that over 35% of the Web articles contains usable print-friendly URLs (pre-generated pages). The majority of the articles, however, dynamically create the print-friendly pages upon request, and thus our method cannot retrieve any usable URL for these pages (content extraction is resort back to the original pages). The key goal of the solution is to achieve very high precision while maximizing the recall since any incorrectly extracted print-friendly URL would lead to a page with the wrong content. We have identified there are basically four HTML expressions to represent print-links, and used a dictionary populated with print-phrases collected from over 2000 Web pages from the 50 most popular news sites for print-phrase matching. The print-phrase dictionary is expected to stay fairly stable, and can easily internationalize to support other languages. To further improve the precision, we also impose the requirement that both the extracted print-friendly and original page URLs must share the same domain name. This requirement has only a small negative impact on the recall, but no impact on the precision. With this system, we are able to achieve over 99% precision and 97% recall performance measures.

5. REFERENCES

[1] Le Hégaret, Philippe (2002). "The W3C Document Object Model (DOM)". World Wide Web Consortium. http://www.w3.org/2002/07/26-dom-article.html.

[2] J. Pasternack and D. Roth. "Extracting article text from the web with maximum subsequence segmentation". In Proceedings *of the 18th WWW*, 2009.

[3] Gupta, Suhit et al. "Automating Content Extraction of HTML Documents". World Wide Web: Internet and Web Information System, 8, 2005, 179-224.

[4] Reis, D. et al. "Automatic Web News Extraction using Tree Edit Distance". In Proceedings of the 13th International Conference on World Wide Web, 2004, New York.

[5] Luo, Ping et al. 2009. "Web Article Extraction for Web Printing: a DOM+Visual based Approach". In Proceedings of the 9th ACM Symposium on Document Engineering . DocEng 2009, New York, NY, 66-69.