Article Clipper- A System for Web Article Extraction

Jian Fan[1], Ping Luo[2], Suk Hwan Lim[1], Sam Liu[1], Parag Joshi[1], Jerry Liu[1]

HP Labs Palo Alto[1]
1501 Page Mill Road
Palo Alto, CA, 94304, USA

HP Labs China[2]
No.1 Zhong Guan Cun East Road
Beijing 100084, China

{jian.fan, ping.luo, suk-hwan.lim, sam.liu, parag.joshi, jerry.liu}@hp.com

ABSTRACT

Many people use the Web as the main source of information in their daily lives. However, most web pages contain non-informative components such as side bars, footers, headers, and advertisements, which are undesirable for certain applications like printing. We demonstrate a system that automatically extracts the informative contents from news- and blog-like web pages. In contrast to many existing methods that are limited to identifying only the text or the bounding rectangular region, our system not only identifies the content but also the structural roles of various content components such as title, paragraphs, images and captions. The structural information enables re-layout of the content in a pleasing way. Besides the article text extraction, our system includes the following components: 1) print-link detection to identify the URL link for printing, and to use it for more reliable analysis and recognition; 2) title detection incorporating both visual cues and HTML tags; 3) image and caption detection utilizing extensive visual cues; 4) multiple-page and next page URL detection. The performance of our system has been thoroughly evaluated using a human labeled ground truth dataset consisting of 2000 web pages from 100 major web sites. We show accurate results using such a dataset.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval– *Information filtering, Selection process*

General Terms

Algorithms, Experimentation

Keywords

Web article extraction, information extraction, page segmentation.

1. INTRODUCTION

The World Wide Web has become the main source of information for many people. There are many types of web pages, including news, blogs, shopping, maps, financial information, photos, videos, email and more. However, most web pages are designed for on-screen viewing only and may not be suitable for other purposes. Repurposing web content generally requires re-layout of the content. This means that not only the content but also the structural roles of various content components should be recognized. For example, the text of a caption and its associated image must be identified and their spatial proximity must be maintained to ensure a proper new layout.

Extracting informative content from web article pages is a very challenging problem and an active area of research [1][2][3]. Most existing methods only attempt to identify the approximate region of the main body within a web article page. Pasternack and Roth recently presented a method based on the maximum subsequence segmentation, a global optimization over token-level classifiers that generate scores for words using features of trigrams and tags [1]. The article text, as defined by the positions of the first and last tokens, is found by a simple searching algorithm. Wang et al. proposed a template-independent wrapper [2]. Their method is to identify two minimum sub-trees containing title and article body, based on a Support Vector Machine (SVM) using simple content and spatial features. However, as was pointed out by [3], the main body region may also contain some non-informative contents such as advertisements and further identifying them is not trivial. Ping et al. utilized visual features for identifying paragraphs as the basic subsequences for local classifiers and the maximum subsequence algorithm [3]. The use of visual features gives the algorithm the capability to filter out most non-informative elements within the main text region. For extracting the title from an HTML document, Hu et al. proposed a machine learning approach with a large set of features based on both HTML tags and visual attributes [4]. They achieved roughly 70% in both precision and recall. Another important and difficult problem that has not been addressed by most existing methods is the identification of image captions. Rowe and Frew reported a method that relied on placement, format and text content as clues for identifying photographs and captions from semi-randomly selected Web pages [5]. They reported 41% recall with 41% precision for caption identification.

In this demonstration, we show a complete system that identifies both the content and the structural roles of content components from news and blog type web pages. We extract the title, main text body as composed of paragraphs, images and captions, and print links. We also detect and traverse "next page" links for multi-page articles.

2. SYSTEM DESCRIPTION

The system block diagram is shown in Figure 1. Provided as its URL, the article web page is parsed and rendered using a headless Webkit to obtain a complete DOM+visual representation. This is followed by a search for a print link (i.e., printable version) in the web page and uses it for analysis if present. The main article text is detected first and the information is utilized to search for other components. Then, the next page detection module checks to see

KDD'11, August 21–24, 2011, San Diego, California, USA.

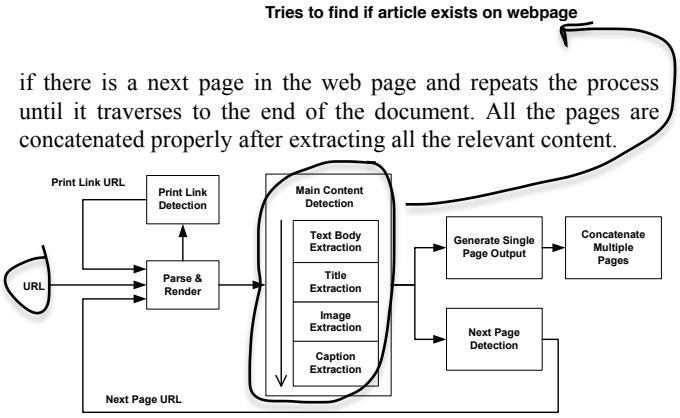

Figure 1. System flowchart.

if there is a next page in the web page and repeats the process until it traverses to the end of the document. All the pages are concatenated properly after extracting all the relevant content.

2.1 Print-link Extraction

Web pages from some web sites provide a hyperlink that leads to a print-friendly web page containing mainly the article itself. Content extraction using these print-friendly pages is generally easier and more reliable. But there are many variations of the print-link representations that made the print link detection a non-trivial problem. First, the link can be text-based, image-based, or both. For text-based links, there is a lexicon of phrases used to represent the print links, such as "print", "print article", "print-friendly version", "print story", etc. For image-based links, in addition to a printer-resembling image icon a print phrase may or may not be present. To complicate the matter further, not all the links contain a valid URL, but instead the pages may be dynamically generated either by the client Javascript or by the server, which are not retrievable using the DOM-based extraction technique since no valid URL is available in the DOM. We observed that almost all print links follow one of the following HTML representations: (1) <a>print phrase, (2) <a><img>, (3) <a><img>print phrase, where the <a> and <img> are the HTML hyperlink and image tags, respectively. Note that in cases 2, even though there is no print phrase (only an image icon is used), a print phrase is typically embedded in the attributes of the <img> or <a> tag, such as the "alt" or "title" attribute.

Our approach to the print-link extraction problem takes on two stages: (1) the detection of the print-link, (2) the retrieval of the print-friendly URL from the link attributes. Since a mistake in retrieving the print-friendly URL would cause "catastrophic" failure in the article extraction we aim at achieving a very high precision. This is accomplished by using a print phrase dictionary to find an exact match with the link text content or with the image attribute values. It is important to populate the print phrase dictionary with appropriate phrases since it impacts both the precision and recall. The precision is controlled by using only print phrases encountered in actual web pages, and the recall is maximized by having a comprehensive dictionary.

We estimate that there are roughly 90% of the news article pages have a print link, of which about 35% of them have valid print-friendly URLs. Our solution achieved over 99% precision and over 97% recall.

2.2 Main Article Text Extraction

The detection of the main article text is the cornerstone of the system. Our approach is based on the observation that the article body usually consists of contiguous paragraph blocks occupying the main area of a web page. It includes two steps. First, the consecutive sequence containing the main text body is identified using the Maximum Scoring Subsequence (MSS). Second, discriminative measures based on visual features are applied to purify the main text.

2.2.1 *Maximum Scoring Subsequence*

uses a segement classifier

Based on the visual attribute of line-break of DOM nodes, we identify the text segments that generally correspond to paragraphs [3]. Among the sequence of text segments $\vec{s} = (s_1,...,s_n)$ (where s_i is a text segment, indexed in the order that they appear in the HTML file), we aim to identify the sequence of $\vec{s}$ which exactly bounds the text body. Specifically, each segment s_i is assigned a real value $v_i = F(s_i) \cdot Length(s_i)$, where F is a segment level classifier that gives a real value based on the prediction whether the segment belongs to the main text, and *Length*(s_i) is the string length of the text in s_i. The classifier F utilizes the visual features of position, font size, color, number of links and more. Please refer to [3] for more details. Note that the classifier output *F(x)* may be negative. The *MSS* of $\vec{s}$ is identified as $(s_a,...,s_b)$ corresponding to $(a,b) = \arg\max_{x,y} \sum_{i=x}^{y} v_i$.

2.2.2 *Refinement using Other Visual Features*

Because *MSS* is limited to selecting a set of consecutive text segments, it may still include some unwanted text content, such as advertisements and link-lists to related stories. It may also include image captions. To remove unwanted text and captions from the article text, we utilize the following visual attributes:

1. Font size. We observed that in most web pages article text has the same font size while captions may have a smaller font size. Therefore the most frequent (in character count) font size is determined and text with smaller font size is pulled from the article text.
2. Horizontal overlap. We observed that text paragraphs belong to an article often overlap significantly in horizontal extent while the unwanted content generally have only a small amount of horizontal overlap with the paragraphs. We leverage this property to further remove those unwanted elements.
3. Horizontal alignment. Some web pages use text alignment (left, center, right) as the sole visual attribute to differentiate captions from others. Similar to the font size, alignment of majority text paragraphs is determined and text with different alignment is removed from the article text.
4. Visual frames. Many web pages use a visible frame to group image and its caption. In order to utilize this property, we detect frames by the border properties as well as background image of HTML elements within the region. Text within a frame will be pulled from article text if two conditions are met: 1) the frame does not contain the whole article text region; 2) text within the frame is not under the node of "blockquote".

2.3 Title Extraction

Title is a unique component of an article. Accordingly, it is often annotated with special html tags (H1-H6) and given visual prominence. Taking these properties into account, our title extraction method includes two steps. In the first step, title candidates are selected according to the following criteria:

1. The horizontal starting position must not exceed the horizontal center position of the main text region, and the top position must not below the top quarter of the main text region.

2. The font size must not be smaller than the font size of the main article text. Text elements with font size equal to that of the main article text are eligible only if they are either tagged with "*H1*" to "*H6*" or they are all bold.

In the second step, we compute a numerical score for each candidate and choose the one with the highest score as the title. The score makes use of the following HTML elements and visual features:

1. Title field. This is a text string delimited by the "<title>" and "</title>" tags in an html file.
2. Header tag "*H1*" to "*H6*". We give higher weight to text elements under a DOM node with the header tags. "*H1*" and "*H6*" have the highest and the lowest weight, respectively.
3. Font size. Title usually has large font size that makes it visually significant.
4. Horizontal alignment with the main text body. Our observation is that most titles are either left or center aligned to the main article text.

For each of the features, a sub score is computed. The matching between a title candidate *s* and the title field *t* is measured based on the Levenshtein distance: $m_{s,t} = 1 - \min(d_{s,t}, L_t)/L_t$, where $d_{s,t}$ is the Levenshtein distance between string s and *t*, and L_t is the string length of the title field *t*. The sub score $m_{s,t}$ has values in the range of 0 to 1 with 1 representing a perfect match.

The spatial alignment between a title candidate and the main article text region is measured by: $\alpha = 1 - \min(dl, dc)/w_{main}$, where *dl* and *dc* are the distance between the lefts and the horizontal centers of the elements and the main article text, respectively, and the w_{main} is the width of the main text region.

The overall score *Q* for a title candidate *s* is computed using the formula $Q_s = (m_{s,t} + \lambda) \cdot \beta^{(7-h)} \cdot (0.1 + \alpha)^{\omega} \cdot f_s$, where $m_{s,t}$ is the matching score between the title candidate *s* and the title field *t*, α is the spatial alignment score, f_s is the font size in points, *h* is the header level (*h*=1 to 6 correspond to *H1* to *H6* tags and *h*=7 for strings with non-header tags), and parameters λ=*0.5*, β=*1.15* and ω=*2* are fixed and empirically determined [6].

2.4 Image and Caption Extraction

The goal is to identify images and associated captions that belong to the article. Ultimately this requires image understanding and recognition and is an unsolved problem. Our approach is based on heuristics. Our search of images is limited to the region that encompasses the main text and the title. We rely on simple heuristics based on image dimensions and the presence of the word "advertisement" to eliminate advertisement images. If the image dimension is smaller than a set threshold (e.g., 50 pixels), or the text "advertisement" is found closely above or below the image, the image is deemed as non-informative and excluded.

For each informative image, a search for the corresponding caption is conducted. The primary conditions for a caption candidate are that it must not be part of the article text, its font size must not be larger than the font size of the article text, and it does not consist of all link text. In addition, we search for three spatial image-caption layouts in the following order:

1. Image and caption text are enclosed in a frame. For a given image, we look for a frame containing the image, and then look for qualified text within the same frame. If such text is found, it is annotated as a caption and associated with the image, and the search is called off for that image.
2. Caption below the image. This is the most commonly encountered layout. Additional conditions for qualified captions include: 1) vertical distance to the bottom of the image must be less than two line spaces; 2) must overlap horizontally with the image; 3) must not have a horizontal line or article text in between the caption and the image.
3. Caption on the right side of the image. This is a less common layout. As with layout 2, a search region to the right and similar conditions are imposed.

2.5 Next-page link detection

Many web articles are spread over multiple web pages and require users to click through "next-page" links. An article extraction system is incomplete if it cannot extract a complete article. However, the problem is more difficult than it appears. First, there are many variations of the next-page link, including links with the text of numbers, "next", "next page", the characters ">>", or some combination of these. Second, we must distinguish the next-page link from other kinds of "next" links, including: a) the "next" link for the next article in the same story column, which has a different title and may be totally unrelated to the current article; b) the link for the next page of user comments, which is divided over several pages; c) the "next" link for the next image, which leads only to changing the image.

Our solution to the next-page detection problem takes on four stages:

1. Group all the links in the page into link clusters and rank them in the order as they appear in the HTML file. If the two successive links have the same vertical coordinates, put them into the same link cluster.
2. Identify the link cluster that contains the next-page link. The features we use include: a) the similarity between the link URL and the current page URL; b) the possible number on each link; c) the possible number at the end of the URL (this number is usually the page number of the corresponding page). Given the current page number, denoted by *k*, if the text on the true next-page link is a number, it must be equal to (*k*+1); or if the URL of the true next-page link ends with a number, it must be also equal to (*k*+1).
3. Remove the links to the next-page comment and next image. This is based on our previous works in comment section detection and image caption detection.
4. Compare the current page content with that in the next-page candidate link. The emphasis of the first three steps is on achieving a very high recall. However, in this final step we aim to achieve high precision. More specifically, after we extract the titles and text bodies from the current page and the next-page candidate we compare these results. Only if the two titles are the same and the two text bodies are different, we declare it to be the next-page link.

3. PERFORMANCE EVALUATION

We have built an extensive system to evaluate the performance of our article extraction system automatically. We have collected 2000 web pages from top 50 news web sites and top 50 blog sites such that each site is represented by 20 web pages. All the web pages were saved locally using a Firefox plug-in such that all the contents embedded in the web page are saved properly. In this way, these pages can be accessed in the future even when some of the web pages are either no longer online, or modified. For all the

saved 2000 web pages, we had human experts extract and label the informative content of [Title], [Main Body], [Image URL], [Image Caption], [Print Link] and [Next Page Link], and saved the results as text files. These saved text files served as ground truth data to be compared against the generated extraction result.

Performance evaluation was carried out using the saved original web pages. No URL from print-link was used. For quantitative measures, the commonly used precision and recall are computed. For text content (title, article text and caption), we adopt the "bag-of-words" method used in [1] for simplicity. Specifically, for a set of extracted words W_P and a set of labeled (ground truth) words W_L, the precision and recall are defined as $P = |W_P \cap W_L| / |W_P|$ and $R = |W_P \cap W_L| / |W_L|$. For URL-type content (image URL, print-link and next-page-link), we treat each URL as a word and applied the above formula.

Figure 2 shows the average precision and recall of the main text and title and image caption for the 100 web sites. Our main text extraction method achieved very high accuracy across web sites with diverse content and styles. The lowest performance came from ehow.com and answers.com. The title extraction method also preformed very well. The lowest scores were from howtodothings.com and blogspot.com. In comparison, the performance of image caption detection is significantly lower. For several web sites (azcentral.com, gamespot.com, nba.com), we failed to detect most captions. The reasons for the failure includes: 1) image and caption outside the main article region (azcentral.com), 2) caption being link (gamespot.com), 3) captions have the same style as the main text and require semantic understanding (nba.com).

4. CONCLUSION

We have described a system for automatically extracting web article content and key components of the layout. We built an extensive ground truth dataset for performance testing. Our evaluation results show excellent performance across diverse web sites.

We would also like to point out that web article extraction remains to be a challenging problem especially for blog-type pages that may not be professionally designed, and for images and captions that inherently requires semantic understanding. We plan to continue our research in these areas.

5. REFERENCES

[1] J. Pasternack and D. Roth, Extracting article text from the web with maximum subsequence segmentation, WWW, April 20–24, 2009, Madrid, Spain., 971-980. DOI= http://dx.doi.org/10.1145/1526709.1526840

[2] J. Wang, *et al*., Can we learn a template-independent wrapper for news article extraction from a single training site?, SIGKDD, June 28-July 1, 2009, Paris, France, 1345-1353. DOI= http://dx.doi.org/10.1145/1557019.1557163

[3] P. Luo *et al*., Web Article Extraction for Web Printing: a DOM+Visual based Approach, DocEng, Sept. 15-18, 2009, Munich, Germany, 66-69.DOI= http://dx.doi.org/10.1145/1600193.1600208

[4] Y. Hu et al., Title Extraction from Bodies of HTML Documents and its Application to Web Page Retrieval, ACM SIGIR, 2005, New York, US, 250-257. DOI= http://dx.doi.org/10.1145/1076034.1076079

[5] Neil C. Rowe, Brian Frew, Automatic caption localization for photographs on World Wide Web pages, Information Processing & Management, Vol. 34, Issue 1, Jan. 1998, 95-107. DOI= http://dx.doi.org/10.1016/S0306-4573(97)00048-4

[6] J. Fan, P. Luo, P. Joshi, Title identification of web article pages using HTML and visual features, Proc. of SPIE-IS&T Electronic Imaging, vol. 7879, 2011. DOI= http://dx.doi.org/10.1117/12.876708

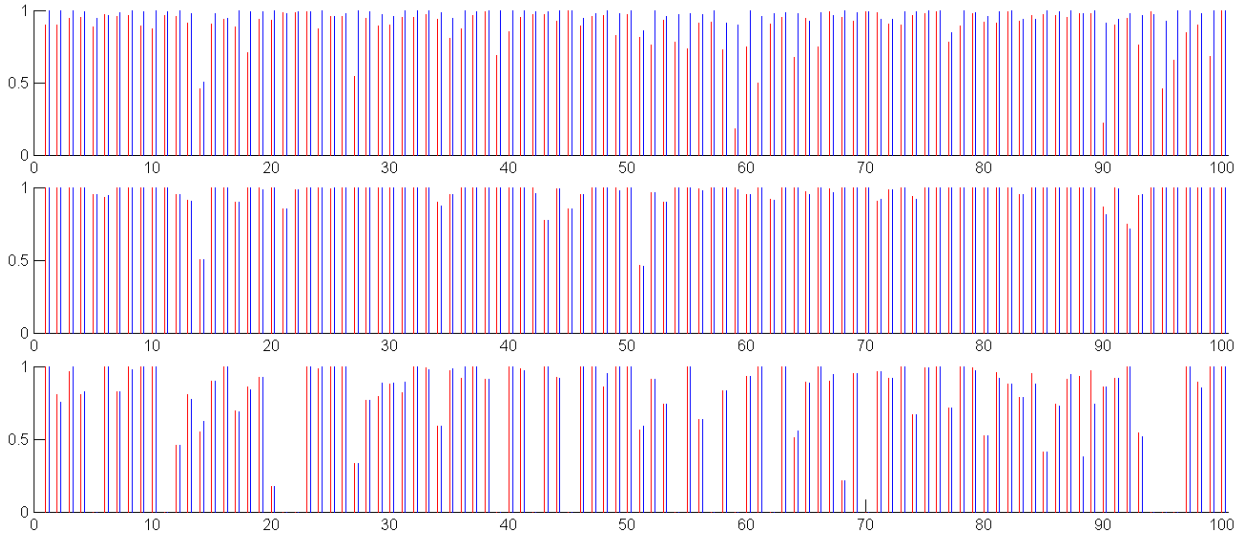

Figure 2. Precision (in red) and recall (in blue) for 100 web sites. (top) Main article text, (middle) title, (bottom) image caption. The missing lines in the bottom plot correspond to sites/pages with no image caption.