A Layout-Independent Web News Article Contents Extraction Method Based on Relevance Analysis

Hao Han and Takehiro Tokuda

Department of Computer Science, Tokyo Institute of Technology
Meguro, Tokyo 152-8552, Japan
{han,tokuda}@tt.cs.titech.ac.jp

Abstract. The traditional Web news article contents extraction methods are time-costly and need much maintenance because they analyze the layout of news pages to generate the wrappers manually or automatically. In this paper, we propose a relevance-based analysis method to extract the news article contents from the news pages without the analysis of news page layouts before extraction. This method is applicable to the general news pages and we give the implementations of news extraction from different kinds of news sources.

Keywords: News Extraction, Search Engine, RSS Feeds.

1 Introduction

Nowadays, the traditional newspapers have developed significant Web presences. We can extract and analyze the Web news articles to acquire the desired information. Wrappers are generated based on the analysis of layout of news pages by many traditional extraction methods. However, different news sites use the different news page layouts, and each news site uses more than one layout. It is costly and inefficient to analyze the news page layout of each news site for news contents extraction.

In this paper, we propose a novel Web news article contents extraction method, which is independent of news page layout and does not need to analyze the news page layouts before extraction. We calculate the relevance between the news title and each sentence to detect the news paragraphs from the full text of the news page. We give the implementations of news extraction from the general news pages, news site databases, and news aggregation sites. By the experiments, we prove that our method runs conveniently and accurately. The organization of the rest of this paper is as follows. In the next section we give the motivation of our research and an overview of the related work. We explain our Web news article contents extraction method in detail in Section 3. In Section 4, we explain the implementation of our method and give the evaluation. Finally, we conclude our method and give the future work in Section 5.

M. Gaedke, M. Grossniklaus, and O. Díaz (Eds.): ICWE 2009, LNCS 5648, pp. 453–460, 2009.

2 Motivation and Related Work

A lot of approaches have been proposed for extracting the Web news article contents. Reis et al. calculates the edit distance between two given trees for the automatic Web news article contents extraction [1]. Fukumoto et al. focuses on subject shift and presents a method for extracting key paragraphs from documents that discuss the same event [2]. 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. represents a news page as a visual block tree and derives a composite visual feature set by extracting a series of visual features, then generate the wrapper for a news site by machine learning [3]. However, it uses manually labeled data for training and the extraction result may be inaccurate if the training set is not large enough. Webstemmer [4] is a Web crawler and HTML layout analyzer that automatically extracts main text of a news site without having banners, advertisements and navigation links mixed up. All the analysis can be done in a fully automatic manner with little human intervention. However, this method runs slowly at contents parsing and extraction, and sometimes news titles are missing.

These methods are still not widely used, mostly because of the need for high human intervention and maintenance, or the low quality of the extraction results. They have to analyze the news pages from target news sites before extraction. Moreover, if the target news sites update the layout of news pages frequently and irregularly, or the target news pages come from a large number of different news sites, it is difficult to realize the extraction by these methods. To address these problems, we propose a layout-independent method to realize the Web news article contents extraction. Compared with the developed work, our method is applicable to the general news pages, and can extract the news articles contents from all kinds of news pages conveniently.

3 News Article Contents Extraction

We can collect the news articles from news sites, RSS feeds, search engines, aggregation sites and others. The collected news is shown as a link to news page, which includes the news title and URL of news page usually. As shown in Fig. 1, we use the collected URL to get the news page and use the collected news title to find out the news article contents from news page. Firstly, we split the news title to get the keyword list and use them to detect the position of news title in the news page. Then, we recognize one paragraph of news article by using the news title position and keyword list. Finally, we find all the paragraphs of news article contents and extract them out of the full text of the news page. We explain our algorithm step by step in this section.

3.1 Title Keywords Acquisition

The news title is a piece of important information for the recognition of the news article contents from the full text of news page. If we locate the position

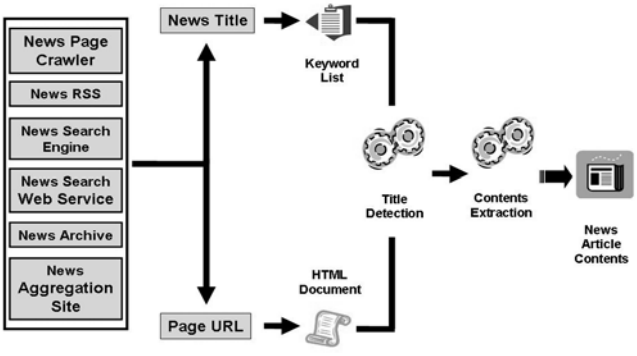

Fig. 1. The outline of Web news article contents extraction

of the news title in the news page correctly, the position of news article contents would be found easily because the contents are a list of paragraphs closely below the title usually. In addition, for a news article, the contents describe the same topic of news title in detail, and the words constituting the title would occur in the news article contents frequently usually. We split the collected title into single words to make a list of keywords as follows. Firstly, we split the news title into a word list using whitespace as the delimiter. Then, we remove the articles, prepositions and conjunctions. Finally, we remove the characters "*'s*" or "*'*" from the words ending with "*'s*" or "*'*". For example, we replace "*Tom's*" with "*Tom*", and replace "*parents'*" with "*parents*".

3.2 Full Text Analysis

An HTML document may be represented as a tree structure. A sentence in a Web page is a visible character string, which is the value of a leaf node. It is possible for each sentence to be the title or a paragraph. We use the following steps to analyze the full text of a news page in order to find the most possible title and paragraphs.

1. We split each sentence into a list of words using the keywords acquisition method described in Section 3.1.
2. We set the words list size as an attribute $WordNumber$, and set the occurrence number of the keywords ignoring case considerations within the words list as an attribute $KeyNumber$ of the corresponding leaf node.
3. We count up the $WordNumber$ of the sibling nodes and set the sum as an attribute $WordNumber$ of their parent node.
4. We count up the $KeyNumber$ of the sibling nodes and set the sum as an attribute $KeyNumber$ of their parent node.
5. We repeat the Step 3 and Step 4 until we set the attribute $WordNumber$ and $KeyNumber$ for <body> as shown in Fig. 2.

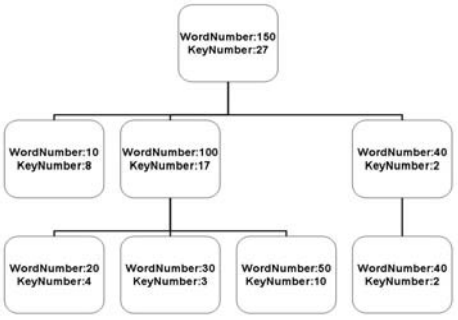

Fig. 2. A full text analysis example

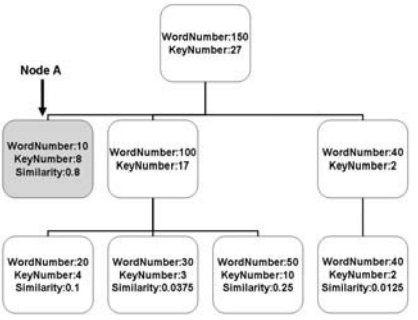

Fig. 3. A news title detection example

3.3 News Title Detection

After the full text analysis, we need to find out the real news title in news page. Usually, the real news title in a news page is same or similar to the collected news title. We use the following formula to calculate the similarity between each sentence of the news page and the collected news title.

$$Similarity = \frac{KeyNumber^2}{WordNumber \times TitleKeywordNumber}$$

Where, $KeyNumber$ and $WordNumber$ are the attribute value of the corresponding node of each sentence respectively, and $TitleKeywordNumber$ is the size of keyword list of the collected news title.

We think a sentence is a possible real news title in the news page if the value of $Similarity$ is more than a predetermined threshold, and a node whose value is a possible news title would be a possible title node. Fig. 3 shows a news title detection example where the size of title keyword list is 8. Assuming that the predetermined threshold is set to 0.6, the node A is judged as the title node.

However, the collected news title is not always same or similar to the real news title in the corresponding news page. In some news sites, we even can find that the collected news title is different from the real news title totally, but same to the other sentences in the news page. Moreover, some news titles are so short and simple that we can find two or more same or similar sentences in news pages. Therefore, there are five different situations about the possible news title and the real title in a news page: 1. There is no possible news title. 2. There is just one possible news title, and it is the real news title in the news page. 3. There is just one possible news title, but it is not the real news title in the news page. 4. There are two or more possible news titles, and one of them is the real news title in the news page. 5. There are two or more possible news titles, but none of them is the real news title in the news page.

3.4 News Paragraph Recognition and News Contents Extraction

Usually, the news article contents part is a list of paragraphs immediately below the title. It becomes easier to find the paragraphs after the real news title is found. However, we can not make certain whether the found possible news title is the real news title in the news page as we describe in Section 3.3. The news paragraph recognition can be classified as the following situations.

1. There is no possible news title and the news paragraphs exist between <body> and </body> (Fig. 4(a)).
2. There is one possible news title and the news paragraphs exist between the end tag of possible title node and </body>. If we can not find out the news paragraphs in this range, we would find them in the reserve range which is between <body> and the start tag of possible title node (Fig. 4(b)).
3. There are two or more possible news titles and the news paragraphs exist between the end tag of each possible title node and the start tag of the next possible title node or </body>. If we can not find out the news paragraphs in these ranges, we find them in the reserve range which is between <body> and the start tag of the first possible title node (Fig. 4(c)).

Although each sentence of each selective range, including the link text, has the possibility to be one of the paragraphs, most of the paragraphs are non-link texts. We give a possibility for each non-link sentence and select one with the highest possibility as the final most possible paragraph if the highest possibility is more than a predetermined threshold. If the highest possibility is less than this predetermined threshold, we would find a sentence with the highest possibility in the reserve range and then compare these two possibilities to select one with the higher possibility as the final most possible paragraph. We use the following formula to calculate the possibility of each non-link sentence.

$$Possibility = WordSum \times (KeySum + 1)$$

Where, $WordSum$ is the sum of the attributes $WordNumber$ of each sentence's corresponding node and its related nodes in the same selective range. $KeySum$ is the sum of the attributes $KeyNumber$ of each sentence's corresponding node

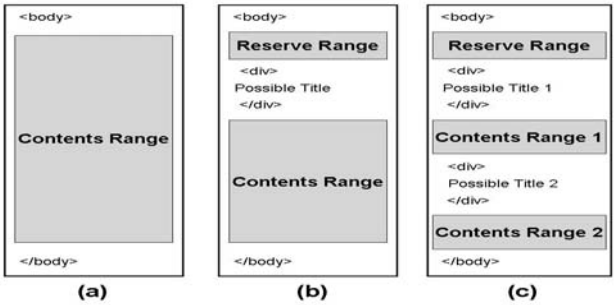

Fig. 4. Contents range and reserve range

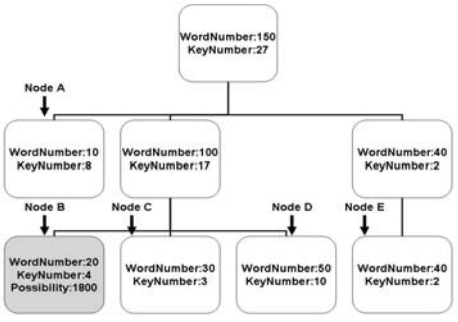

Fig. 5. A paragraph recognition example

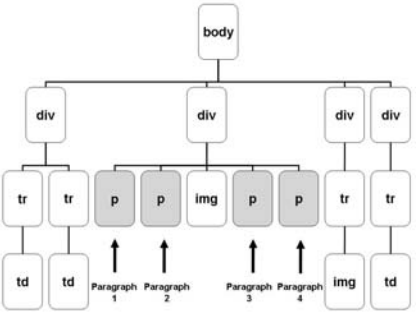

Fig. 6. A full contents extraction example

and its related nodes in the same selective range. For Node A and Node B, Node B is a related node of Node A if Node B satisfies the following conditions.

1. Node B and Node A are sibling nodes, or their parent nodes are sibling nodes.
2. Node B or its parent node is of one of the following nodes: $\sharp text$, , <a>, <p>, <ul>, <ol>, <dd>, <dt>, , <h1>, <h2>, <h3>, <h4>, <b>.

Fig. 5 shows a paragraph recognition example. Node B, C, D and E belong to the contents range, and finally the node B is judged as the paragraph node where the predetermined threshold is set to 100. After the paragraph recognition, we get a paragraph of the news article contents. Usually, the full contents of a news article are a list of continuous paragraphs. However, there is advertisement information such as the image advertising among the paragraphs of a news article in some news sites. We get a list of related nodes of paragraph node, and each one represents a paragraph of the news article contents as shown in Fig. 6. We get the node value from each node as a paragraph. The full contents of news article are the extracted continuous node values.

4 Implementation and Evaluation

In this section, we give the implementation of our proposed news article contents extraction method. After we analyzed a large number of news pages from many news sites, we set the threshold as 0.6 in Section 3.3 and 100 in Section 3.4 respectively based on the statistical results. In the following experimental results, we prove that the thresholds are suitable for the general news pages.

Experiment 1 We extract the news from RSS feeds of 38 popular news Web sites periodically. Since May 2007, we have collected more than 1.8 million pieces of news articles. Our experiments were run periodically using the randomly collected news articles. Our experimental results are listed in Table 1. Here, *Success* means that our extraction method extracts the news article contents

Table 1. Experimental results of extraction accuracy rate (long period extraction)

Period	Sum	Success	Failure	Precision
May 2007 - Aug 2007	1000	970	30	97.0%
Sep 2007 - Jan 2008	500	491	9	98.2%
Feb 2008 - May 2008	500	485	15	97.0%
Jun 2008 - Sep 2008	500	488	12	97.6%
Total	2500	2434	66	97.4%

correctly, and *Failure* means that our extraction method extracts nothing or partial paragraphs or other non-news parts such as advertisements and related stories. Although the news sites update the layout of news pages irregularly, our news article contents extraction method works well during each period and the precision of extraction is over 97%. The experimental results prove that our extraction algorithm is highly accurate during a long period.

Experiment 2 We extract and analyze the topic-based Web news articles from news site databases to observe the difference in the various topics. We select the countries and leaders as our test topics. There are 242 countries in the world and most of them have the leaders. We use these country names and leader names as our search keywords. We send these keywords to site-side news search engines one by one, and collect 121,336 news titles and page URLs of matched news published in the past 6 years (from January 1, 2003 to December 31, 2008) from news database of CNN. Finally, we extract the news article contents from these news pages. We select 250 news pages randomly and check them one by one manually. The experimental result is listed in Table 2. We find that 2 news pages can not be obtained (the server responds the message like "page not found"). Among the rest 248 news pages, the news article contents of 240 news pages are extracted correctly. In the 8 extraction failures, some parts of news article contents are not extracted. We also do the similar experiments on the other news sites. Although the news sites updated the layout of news pages many times in the past 6 years irregularly, our extraction method works well from our experimental result. The experimental results prove that our extraction method is suitable for the extraction of topic-based news articles from news site databases.

Experiment 3 We collect the news from a large-scale news sources by collecting the news titles and URLs of news pages from news aggregation sites. We collect one week's news about "Asia" from Google News as our experimental data. The total results include 1,535 news articles extracted from many different news sites. We select 500 news pages randomly and check them one by one manually. The experimental result is listed in Table 3. Among the 500 news pages, the news

Table 2. Experimental result of extraction accuracy rate (news site databases)

Sum	Extracted	Success	Failure	Precision
250	248	240	8	96.8%

Table 3. Experimental result of extraction accuracy rate (news aggregation sites)

Sum	Success	Failure	Precision
500	483	17	96.6%

article contents of 483 news pages are extracted correctly. In the 17 extraction failures, some parts of news article contents are not extracted or other non-news parts are extracted. Although the news pages comes from the different news sites, our news article contents extraction method works well and the experimental result proves that our extraction method can extract the news article contents from news aggregation sites accurately.

We give the implementation of our proposed news article contents extraction method and the experimental results prove that our extraction algorithm is highly accurate. However, in some news pages, a paragraph, usually the outline of news article, shows in different style compared to other paragraphs. This kind of paragraph looks like a non-news part such as an advertisement in text format, and is omitted in the extraction. Moreover, some news article contents are too short to recognize from the news pages. For example, a news flash about baseball game result, which contains just a short paragraph of ten words, maybe can not be extracted correctly. Compared with other developed methods, our extraction method has different implementations including the extraction from crawled news pages, news search engines and news aggregation sites. The extraction system is constructed easily based on our method and does not need any maintenance over the long period extraction. We do not need to analyze the layout of news pages since our extraction algorithm is independent of the layout of Web pages. It does not need to reconfigure extraction even though the news sites update the layout of news pages, and keeps a high extraction precision.

5 Conclusion

In this paper, we have presented a relevance-based analysis method to realize the news article contents extraction without the analyzing the layout of news pages. Our algorithm is applicable to the general news pages, and can extract the news article paragraphs accurately. Our experimental results show that our method works well with a high accuracy rate in different kinds of implementations. As future work, we will modify our algorithm to improve the accuracy rate even further, and extend its implementations to more news related applications.

References

1. de Castro Reis, D., Golgher, P.B., da Silva, A.S., Laender, A.H.F.: Automatic Web news extraction using tree edit distance. In: The Proceedings of the 13th International Conference on World Wide Web (2004)
2. Fukumoto, F., Suzuki, Y.: Detecting shifts in news stories for paragraph extraction. In: The 19th International Conference on Computational Linguistics (2002)
3. Zheng, S., Song, R., Wen, J.R.: Template-independent news extraction based on visual consistency. In: The Proceedings of the 22th AAAI Conference (2007)
4. Shinyama, Y.: Webstemmer (2007), `http://www.unixuser.org/~euske/python/webstemmer/`