AUTOMATIC MAIN TEXT EXTRACTION FROM WEB PAGES

Phan Thi Ha*, Ha Hai Nam

Posts and Telecomunications Institute of Technology

*Email: hathiphan@yahoo.com

Received: 10/6/2012; Accepted for publication: 19 April 2013

ABSTRACT

This paper presents a novel method for extracting body text from web pages used for building text corpus. The algorithm for extracting body text proposed by Aidan Finn [1] is extended with some enhancements in this research. The experimental results on a set of websites show that the proposed method significantly improves the performance of body text extraction without decrease in accuracy compared to the original algorithm.

Keywords: HTML, BTE, body text etraction, main content text.

1. INTRODUCTION

The web is a huge repository of information. Hence, it has increasingly become a critical source for building large text corpus. Using text extracted from web pages for building text corpus was proposed in the late 1990s [2]. Grefenstette and Nioch [3] introduce an open source tool for collecting text from internet named BootCaT. Keller and Lapata [4] have proved the validation of using text corpus collected from web pages for liguistics research. Keller and Lapata manually and automatically compared language models derived from text corpus extracted from web pages with ones derived from traditional text corpus. One of the advantages of collecting text from web pages is the update of the corpus; and it also allows us to objectively discover various language features. Most of the web pages are encoded using HTML to represent their contents. An HTML document normally contains much of additional information in addtion to the main content such as external links, advertisements, banners, logos…. Web content cleaning is the process of removing boilerplates and extracting only consistent valuable text material from web pages, which is called We Content Extraction. For the web pages whose structures are frequently changed, the extraction of the main content is more challenging [5]. Much research on web content extraction has been carried out [1, 6 - 10] for different applications. For instance, research on web mining and information retrieval can employ WCE for preprocessing HTML documents in order to minimize noise for more accurate results. Another application of WCE is to reduce the size of web pages for portable devices such as mobile phones, PDAs…

The proposed method is used for extracting the main content from web pages of different fields. The extracted texts are then used for building large Vietnamese text corpus. Extraction

time is a critical factor for building large corpus. Hence, the focus of the proposed method, which is based on the method proposed by Aidan Fin, is on performance in terms of extraction time. In order to improve the execution time, HTML tags that are not part of the main content of the web page are removed using a method that minimizes the number of loops for calculating tag and mark densities. In the proposed method, tag and mark densities are used for determining the text regions to be extracted. The improved algorithm has been implemented; and the results have been evaluated against original algorithm proposed by Aidan Fin. The evaluation showed that the extraction time is significantly reduced using improved algorithm while the quality of extracted contents are equivalent.

The remainder of the paper is organised as follows. In Section 2, we describe briefly the HTML features. In Section 3, we introduce the improved Body Text Extraction (BTE) algorithm followed by the experiments and evaluation in Section 4. We present our conclusions and directions for future work in Section 5.

2. STRUCTURE OF A HTMT FILE DOCUMENT

Most of the web pages are in HTML format. At a simple level, an HTML document consists of texts and mark-up tags. The structure of a sample HTML document is shown in figure 1. The <HTML> element indicates that the content of the document is in HTML format. A declarative header section is followed and delimited by the <HEAD> element. The body of the HTML document, which contains the actual content of the HTML document, is implemented by the <BODY> elemement. The <H1>, <P> and <B> elements are used to format the headings, paragraphs and bold text of the HTML document, respectively.

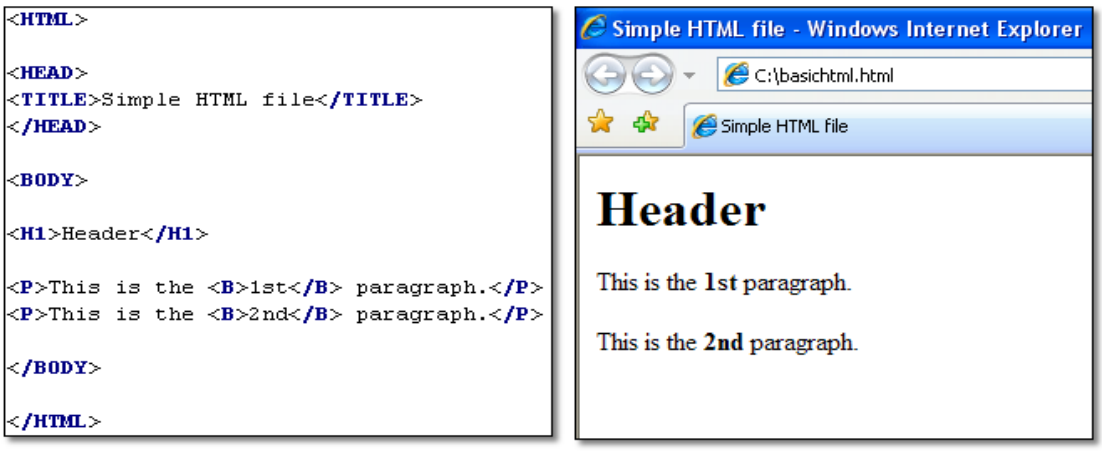

Figure 1. A simple HTML document (left) & its presentation on a browser (right).

In practice, HTML documents contain different markup elements inherited from older versions of HTML. The markups elements are used in various ways without conforming to any standard. For instance, the embedded scripts, erroneous tags appear frequently in HTML documents. The HTML elements concern the syntax of the content. Hence, HTML is not dependent on the main content that are formated by HTML elements. In other words, HTML element are semantics independent. Not all the texts inside the BODY element are about the

main content of the web page. In fact, some trivial texts are also contained in BODY element such as advertisements, external links, annotations, copyright information…

3. IMPROVED BODY TEXT EXTRACTION ALGORITHM

Based on observations of a number of news sites, Aidan Fin has noticed that the main content of news sites normally has most texts and least markup tags in HTML code [1]. Table 1 shows the statistics of some Vietnamese well-known news sites with regard to ratios of texts and tags appear within main content. Text is counted in character.

Table 1. Ratios of texts and tags appear within main content for some Vietnamese news sites.

News Site	Texts main content/Total texts of the page	Tags in main content/Total number of tags of the page
vietnamnet.vn	53%	9%
dantri.com.vn	55%	10%
vnexpress.net	52%	3%

In typical sites, the beginning and end sections often have many of boilerplates and HTML tags. And the middle section of the sites, which contains the main content, often has small number of tags which is likely the content to be extracted. Based on this observation, the BTE algorithm is developed using information about the density of characters and tags for main content identification. The key idea of BTE algorithm proposed by Aidan Fin is described as follows:

Find two points i, j such that the number of HTML tags extracted from sections below i and above j and the number of text-tokens extracted from section between i and j are maximized. The result is the set of text-tokens extracted from the section within [i,j].

Aidan Fin has experimentally used the BTE algorithm to extract the main content for classifying the digital library documents that focused on sports, politics. The original algorithm proposed by Aidan Fin is extended in this research in order to achieve better execution time. The improved algorithm is used to extract the body texts from Vietnamese news sites which are later used for building large Vietnamese text corpus.

Main steps of the improved algorithm are as follows:

Step 0: An HTML document is downloaded. The HTML code of the document is cleaned by removing tags and code snippets that do not contain the relavant content such as the codes that are outside of BODY element and tags of <*input*>, <*script*>, <*img*>, <*marquee*>,<!--...-->, <*iframe*>… Tag library has been collected from URLs[1],[2].

Step 1: For the cleaned remainder of the HTML code, create two arrays named `binary_tokens[]` and `tokens[]`, respectively:

[1] http://mason.gmu.edu/~montecin/htmltags.htm#htmlformat

[2] http://www.w3schools.com/tags/

- `binary_tokens[]` includes the elements that take values of either 1 or -1: `binary_tokens[i]=1` whre *i* is an HTML tag that is:
  + Start tag/Unknown start tag in the form of <...> such as `<html>`, `<p color=red>`…
  + End tag/unknown end tag in ther form of </...> such as `</html>`, `</p>`…
  - `binary_tokens[i]=-1` corresponds to a word.
- `tokens[]` includes elements whose values are word/tags corresponding to elements of the array `binary_tokens[]`. For example, at the position of 23, `binary_tokens[23]=1`, `tokens[23] =<td>`…
- Combine consecutive elements that have the same value of the `binary_tokens[]` into one element of an array named `encode[]`. This combination helps reduce the size of `binary_tokens[]`. And it is based on our observation of tag desity in HTML code.

Step 1 is algorithmically described in Algorithm 1 which has complexity of O(n)

Algorithm 1 BINARY_TOKEN ()

Input: text data in HTML
tagHTML[] – The full set of HTML tags

Output: binary_token[]– elements take value of –1 or 1 mapped from data
endcode[]–stores elements of binary_token[] after combining the consecutive elements that have same value

```
1. BEGIN
2. Declare an integer variable called i, j, k;
3. Initialize i,j,k to zero
4. Declare an array variable called tokenized_data to store
   the tokens derived from input data
5. Declare an array variable called endcode[]
6. Set all elements of endcode[] to null;
7.   for each t in tokenized_data[])
8.      if (t in tagHTML[]) binary_token[i]=1;
9.      else binary_token[i]=-1;
10.       token[i] = tokenized_data[i]
11.       i = i + 1;
12.   endfor
13. /*Combine the consecutive elements that have same value
   of binary_token[] and save to endcode[]*/
14.   for(k=0 to i)
15.       x= binary_token[k];
16.       if( abs(x+ endcode[j])< abs(endcode[j]))
```

```
17.          j = j + 1;
18.       endif
19.       endcode[j]= endcode[j]+x;
20.       k = k +1 ;
21.    endfor
22. END.
```

Step 2: Find *i, j* in `binary_tokens[]` derived from Step 1 such that the number of elements of -1 (corresponding to words) between [i,j] is maximal and the number of elements of 1 (correspoding to tags) is minimal.

Step 3: Extract the data from section between [i,j] and remove HTML tags.

Step 2,3 are algorithmically described in Algorithm 2 which has complexity of $O(n^2)$.

Algorithm 2 EXTRACT_BODYTEXT

Input: token[] – Calculated in Algorithm 1
endcode[] – Calculated in Algorithm 1

Output: Extracted body text

```
23. BEGIN
24. Declare a string variable called  body_txt;
25. Declare integer variables called, i,j,i_max,j_max
26. i_max = 0;
27. j_max = length(endcoded[])-1;
```

*/*Find [i_max, j_max] in endcode[] such that the number of elements that take negative value is maximal and the number of elements that take positive value is minimal */*

```
28. for ( i←0 to length(endcoded[])-1)
29.    if (encoded[i] > 0)
30.         Continue;
31.    for (j←i to lenght(encoded[]))
32.        if ( encoded[j] > 0)
33.          continue;
34.        j = j + 1;
35.    endfor
36.    Declare an integer variable called s
37.    Initialize the variable s to zero
38.    for (k in encoded[i..j])
39.        {s ←s+k;
40.         if (min>s)
```

```
41.              i_max ← i+abs(encode[i-1];
42.              j_max ← j + 1
43.          endif
44.     endfor
45.     i = i + 1;
46. endfor
47. start= $\sum_{k=0}^{i\_max} abs(endcode[k]$ ; end= $\sum_{k=0}^{j\_max} abs(endcode[k]$
48. body_text ←token[start..end];
49. Delete all start tag and end tag in body_text
50. return(body_text);
51. END.
```

The idea for the first improvment of the proposed method is based on following observations: a) the main content of an HTML document is always inside a pair of parent tags within BODY element; b) there are HTML code snippets that do not contain the main content such as javascript code and tags of <img> <input> <select> <option>. Step 0 makes sure that HTML code snippetes that do not contain main content are removed. This step reduces the amount of data for the next processing steps.

The second improvement is based on the fact that the density of the word tokens in the main content of an HTM document is rather high while the tag desity is high ouside the main content. The number of loops for identifying main content delimiters [i, j] is reduced by minimizing the size of the array named `binary_tokens[]`. The size minimization of `binary_tokens[]` is realized by counting consecutive elements that have same value and storing the counted number in an array named `encode[]`. For example, an HTML code snippet that has 2 tags and 5 consecutive words is encoded in `binary_tokens[]` as {1,1,-1,-1,-1,-1,-1}; and this is then compresed and saved in `encode[]` as {2,-5}. This helps reduce the execution time of the Algorithm 2 for extracting main body text. The complexity of the improved method is $O(n^2)$ compared to the original one whose complexity is $O(n^3)$.

4. EXPERIMENT AND EVALUATION

The proposed method has been experimented and evaluated against the orginal one proposed by Aidan Fin. The experiment has been conducted by running the original BTE method proposed by Aidan Fin and the improved method on the same set of HTML documents collected from predefined URLs. The results show that:

a) both methods are independent of languages, character encodings;

b) both methods work best on news sites but they have poor accuracy on sites with special structures such as social networks, shopping sites due to complex distribution of the main content across the page layouts;

c) the improved method remove completely the HTML codes while the original one cannot remove all the HTML codes in most of our runs on experimental data sets. Table 2 summarizes the main results of the experiment.

Table 2. Comparisons of improved and original BTE methods.

Results	Improved BTE	Original BTE
Support multilingual sites	Yes	Yes
Support BTE from news sites	Good	Good
Average Extraction Time (for 100 news sites)	24.5822007691 (s)	238.116690548 (s)
Support BTE from sites with special structures such as social networks, shopping sites	Poor	Poor

5. CONCLUSIONS AND FUTURE WORK

This paper describes a body text extraction method from news sites which is used for building large Vietnamese text corpus. The proposed method, which is based on original one proposed by Aidan Fin, provides new features to improve the extraction accuracy and execution time. Improvement in execuation time is achieved through reducing the loops needed for identifying the delimiters of the main content of HTML documents. The compression of similar elements of the array that indicates the tag or word leads to reduction in amount of processing data.

The experiment results show a significant improvement in extraction time of the improved method compared to the original one while the accuracy remains the same. However, both the improved and original BTE methods have good accuracy for the news sites, but they have poor accuracy for the sites with special structures such as social networks, shopping sites…

The future work would be to improve the accuracy of the improved BTE method for the sites whose distribution of the main content is complex and to adapt the improved BTE method for formats other than HTML.

REFERENCES

1. Aidan F., Nicholas K. and Barry S. - Fact or Fiction: Content Classification for Digital Libraries. Proceedings of the Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries, Dublin City University, Ireland, 2001.
2. Jones R. and Ghani R. - Building a corpus for a minority language from the web, Proceedings of theStudent Workshop of the 38th Annual Meeting of the Association for Computational Linguistics, 2000.
3. Grefenstette G. and Nioche J. - Estimation of english and non-english language use on the www, Proceedings of RIAO (Recherche d'Informations Assistée par Ordinateur), Paris, 2000.
4. Keller F. and Lapata M. - Using the web to obtain frequencies for unseen bigrams, Computational Linguistics **29** (2) (2003) 459–484.

5. David G., Kunal P. and Andrew T. - The Volume and Evolution of Web Page Templates, Proceddings of WWW\'05: Special interest tracks and posters of the 14th international conference on World Wide Web, 2005.
6. Aidan F., Rahman R., Alam H. and Hartono R.- Content Extraction from HTML Documents, Proceedings Workshop on Web Document Analysis, Seattle, USA, 2001.
7. Ziegler C. N. and Skubacz M. - Content extraction from news pages using particle swarm optimization on linguistic and structural features, Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, IEEE Computer Society, Washington, USA, 2007.
8. Ion M., Steve M. and Craig K. - A hierarchical Approach to Wrapper Induction, University of Southern California, 1999.
9. Tim W., William H. H. and Jiawei H. - CETR - Content Extraction via Tag Ratios", Proceedings of the 19th international conference on World wide web, 2010.
10. Sandip D., Prasenjit M., Nirmal P. and Lee G. - Automatic Identification of Informative Sections of Web-pages, 2005.

TÓM TẮT

TRÍCH RÚT TỰ ĐỘNG VĂN BẢN CHÍNH TỪ CÁC TRANG WEB

Phan Thị Hà*, Hà Hải Nam

Học viện Công nghệ Bưu chính viễn thông

*Email: hathiphan@yahoo.com

Bài báo này trình bày một phương pháp trích rút nội dung thân văn bản (BTE-Body Text Extraction) từ các trang web phục vụ cho việc xây dựng kho ngữ liệu nghiên cứu từ vựng. Trong phương pháp đề xuất, thuật toán trích rút văn bản được xây dựng dựa trên thuật toán đề xuất bởi Aidan Finn [1] với một số cải tiến. Kết quả thử nghiệm với một tập các trang web cho thấy thuật toán đề xuất có hiệu năng được cải thiện đáng kể so với thuật toán trong khi vẫn đảm bảo giữ nguyên độ chính xác khi trích rút văn bản so với thuật toán gốc.

Từ khóa: HTML, BTE, trích rút thân văn bản, văn bản nội dung chính.