A Comparison of Three Pre-processing Methods for Improving Main Content Extraction from Hyperlink Rich Web Documents

Moheb Ghorbani[1], Hadi Mohammadzadeh[2] and Abdolreza Nazemi[3]
[1]*Faculty of Engineering, University of Tehran, Tehran, Iran*
[2]*Institute of Applied Information Processing, University of Ulm, Ulm, Germany*
[3]*School of Economics and Business Engineering, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany*

Keywords: Main Content Extraction, Pre-processing Algorithms, Hyperlink Rich Web Documents.

Abstract: Most HTML web documents on the World Wide Web contain a lot of hyperlinks in the body of main content area and additional areas. As extraction of the main content of such hyperlink rich web documents is rather complicated, three simple and language-independent pre-processing main content extraction methods are addressed in this paper to deal with the hyperlinks for identifying the main content accurately. To evaluate and compare the presented methods, each of these three methods is combined with a prominent main content extraction method, called DANAg. The obtained results show that one of the methods delivers a higher performance in term of effectiveness in comparison with the other two suggested methods.

1 INTRODUCTION

A huge volume of web pages being mainly text is placed on the web every day. A significant proportion of this data is published in news websites like CNN and Spiegel as well as information websites such as Wikipedia and Encyclopedia. Generally speaking, every web page of the news/information websites involves a main content (MC) and there is a great interest to extract it at a high accuracy because the MC can be saved, printed, sent to friends and etc. thereafter.

In spite of the numerous studies which have been done during the recent decade on extraction of the MC from the web pages and especially from the news websites, and although many algorithms with an acceptable accuracy have been implemented, they have rarely paid attention to two critical issues, namely pre-processing and post-processing. Thus, these MC algorithms were not fully successful in some cases. Particularly, the MC extraction algorithms have often failed to extract the MC from the web pages which contain a great number of hyperlinks like for example Wikipedia. This paper will introduce and compare three different methods which can be used for pre-processing of the MC extraction algorithms based on HTML source code elements. Each of the three presented methods is combined with a DANAg (Mohammadzadeh et al., 2013) algorithm as a pre-processor in order to be able to compare them with each other. The obtained results show that one of the suggested methods is very accurate.

This paper is organized as follows: Section 2 reviews the related work briefly, while the pre-processing approaches are discussed in Section 3. The data sets and experiments are explained in Section 4, and Section 5 makes some conclusions.

2 RELATED WORK

Algorithms and tools which are implemented for main content extraction usually employ an “HTML DOM tree structure” or “HTML source code elements” or in simple words HTML tags. Algorithms can also be divided into three categories based on the HTML tags including “character and token-based” (Finn et al., 2001), “block-based” (Kohlschütter et al., 2010), and “line-based”. Most of these algorithms need to know whether the characters in an HTML file are components of content characters or non-content characters. For this purpose, a parser is usually used to recognize which type of the component they are. Character and token-based algorithms take an HTML file as a sequence of characters (tokens) which certainly contain the main content in a part of this sequence. Having executed the algorithms of this section, a sequence

Ghorbani M., Mohammadzadeh H. and Nazemi A..
A Comparison of Three Pre-processing Methods for Improving Main Content Extraction from Hyperlink Rich Web Documents.
DOI: 10.5220/0004947503350339
In *Proceedings of the 10th International Conference on Web Information Systems and Technologies* (WEBIST-2014), pages 335-339
ISBN: 978-989-758-024-6

of characters (tokens) is labeled as the main content and is provided to the user. BTE (Finn et al., 2001) and DSC (Pinto et al., 2002) are two of the state-of-the-art algorithms in this category. Block-based main content extraction algorithms, e.g. boilerplate detection using shallow text features (Kohlschütter et al., 2010), divide an HTML file into a number of blocks, and then look for those blocks which contain the main content. Therefore, the output of these algorithms is comprised of some blocks which probably contain the main content. Line-based algorithms such as CETR (Weninger et al., 2010), Density (Moreno et al., 2009), and DANAg (Mohammadzadeh et al., 2013), consider each HTML file as a continuous sequence of lines. Taking into account the applied logic, they introduce those lines of the file which are expected to contain the main content. Then, the main content is extracted and provided to the user from the selected lines. Most of the main content extraction algorithms benefit from some simple pre-processing methods which remove all JavaScript codes, CSS codes, and comments from an HTML file (Weninger et al., 2010) (Moreno et al., 2009) (Mohammadzadeh et al., 2013) (Gottron, 2008). There are two major reasons for such an observation: (a) they do not directly contribute to the main text content and (b) they do not necessarily affect content of the HTML document at the same position where they are located in the source code. In addition some algorithms (Mohammadzadeh et al., 2013) (Weninger et al., 2010) normalize length of the line and, thus render the approach independent from the actual line format of the source code.

3 PRE-PROCESSING METHODS

In this section, all kinds of the pre-processing methods are explained in detail. Hereafter, these methods are referred to as Filter 1, Filter 2, and Filter 3, for further simplicity. In this contribution, only the presented pre-processing methods are combined with one of the line-based algorithms which is called DANAg (Mohammadzadeh et al., 2013).

3.1 Filter 1

Algorithm 1 shows the simple logic used in Filter 1. It can be seen that one just needs to remove all the existing hyperlinks in an HTML file which is done at line 4 of this algorithm. The only disadvantage of this pre-processing method is that by removing the hyperlinks, the anchor texts are also removed. As a result, this will cause the hyperlinks in the extracted main content to be lost. Thus, their anchor texts, which must be seen in the main content, will no longer exist in the final main content. Consequently, the application of Filter 1 will reduce either the accuracy or the amount of recall (Gottron, 2007). In the ACCB algorithm (Gottron, 2008), as an adapted version of CCB, all the anchor tags are removed from the HTML files during the pre-processing stage, i.e. Filter 1.

Algorithm 1: Filter 1.

1: $Hyper = \{h_1, h_2, ..., h_n\}$
2: $i = 1$
3: **while** $i <= n$ **do**
4: $h_i.remove()$
5: $i = i + 1$
6: **end while**

3.2 Filter 2

The idea behind Filter 2 which is shown in Algorithm 2 implies that the all attributes of each anchor tag are removed. With respect to Algorithm 2, which shows the pseudocodes of Filter 2, one can understand that an anchor tag contains only an anchor text.

```
<a>anchor text</a>
```

An advantage of Filter 2 over Filter 1 is that some anchor texts related to the anchor tags, which are located in the main content area, can be extracted by using Filter 2. In other words, the amount of recall (Gottron, 2007) yielded from application of Filter 2 would be greater than the one obtained from Filter 1.

Algorithm 2: Filter 2.

1: $Hyper = \{h_1, h_2, ..., h_n\}$
2: $i = 1$
3: **while** $i <= n$ **do**
4: **for** *each attribute of* h_i **do**
5: $h_i.remove(attribute)$
6: **end for**
7: $i = i + 1$
8: **end while**

3.3 Filter 3

In the third pre-processing method, called Filter 3, all the HTML hyperlinks are normalized. In other words, the purpose of this method is to normalize the ratio of content and code characters representing the hyperlinks. Filter 3 is addressed in the AdDANAg (Mohammadzadeh et al., 2012) algorithm.

For further simplification and better comprehension, the underlying approach of Filter 3 is described using a typical example. In the following HTML code, the only attribute is `href="http://www.spiegel.de/"`.

```
<a href="http://www.spiegel.de/">Spiegel Web
Site</a>
```

Now, length of the anchor text is calculated and saved for each hyperlink (in this example: `Spiegel Web Site`) into a variable called *length*. Then, the attribute part of the opening tag is substituted with a string of space characters (␣) with a length of $(length-7)$ where the value 7 comes from the length of `<a></a>`. Therefore, the new hyperlink for this example should be as below:

```
<a ␣␣␣␣␣␣␣␣␣␣>Spiegel Web Site</a>
```

The above-mentioned explanations of Filter 3 are summarized in Algorithm 3. As can be observed in this algorithm, the `while` loop which is repeated for n times calculates the length of the anchor text related to each hyperlink and stores in the LT variable. Then, a string of LT-7 length is made from the space character and then is inserted into a string variable "Str". Finally, the attribute part of the hyperlink is replaced with the Str string.

Algorithm 3: Filter 3.

1: $Hyper = \{h_1, h_2, ..., h_n\}$
2: $i = 1$
3: **while** $i <= n$ **do**
4: $length = Length(h_i.anchor\ \ text)$
5: $String\ \ Str = new\ \ String(\text{“␣”}, length-7)$
6: $substitute(h_i.attributes, Str)$
7: $i = i+1$
8: **end while**

4 DATA SETS AND RESULTS

To evaluate all the three pre-processing algorithms, two suitable data sets are introduced by (Gottron, 2008) and (Mohammadzadeh et al., 2013). Composition and size of the evaluation data sets are given in Tables 1 and 2.

The first dataset contains 2,166 web documents in Arabic, Farsi, Pashto, and Urdu and has been collected from 10 different web sites for evaluation of the main content extraction in right-to-left language web documents. The second corpus contains 9,101 web pages in English, German, and Italian from 12 different web sites and has been established for evaluation of the main content extraction in western language web documents.

Tables 3 and 4 list the obtained results, i.e. recall, precision and F1 score (Gottron, 2007), from combining each of the filters introduced in this paper with the DANAg algorithm. Tables 3 and 4 are again divided into three parts: the first part contains 4 rows and

Table 1: Evaluation corpus of 2,166 web pages.

web site	size	languages
BBC	598	Farsi
Hamshahri	375	Farsi
Jame Jam	136	Farsi
Al Ahram	188	Arabic
Reuters	116	Arabic
Embassy of Germany, Iran	31	Farsi
BBC	234	Urdu
BBC	203	Pashto
BBC	252	Arabic
Wiki	33	Farsi

Table 2: Evaluation corpus of 9,101 web page.

web site	size	languages
BBC	1,000	English
Economist	250	English
Golem	1,000	German
Heise	1,000	German
Manual	65	German, English
Republica	1,000	Italian
Slashdot	364	English
Spiegel	1,000	German
Telepolis	1,000	German
Wiki /	1,000	English
Yahoo	1,000	English
Zdf	422	German

compares the recalls; whereas the second part compares the precision; and finally, the third section compares the F1 scores. By looking at Tables 3 and 4, one can make the following conclusions:

- As seen in the third part of both Tables 3 and 4, Filter 3 has acquired a better F1 score in comparison with the other two filters in most of the 18 cases. In contrast, Filter 2 has obtained the minimum amount of F1 score as compared to Filters 1 and 3.
- Based on the first part of Tables 3 and 4, it can be observed that Filter 3 has the maximum recall only in 11 web sites out of the total number of 22 web sites, while Filter 3 attains the maximum F1 score in 18 web sites.
- In web sites where the values of recall obtained from Filter 2 or 3 are equal to that of Filter 1, one may judge that the web site does not have any hyperlink in its MC. For example, it can be seen on Economics and ZDF web sites that the recall is equal for all the three filters.
- When Filter 1 has a recall equal to the one in a web site such as Reuters, it can be argued that the web site certainly includes no hyperlink in its MC, thus the other two pre-processors of Filters 2 and

Table 3: Comparison between Recall, Precision and F1 on the corpus in Table 1.

		Al Ahram	BBC Arabic	BBC Pashto	BBC Persian	BBC Urdu	Embassy	Hamshahri	Jame Jam	Reuters	Wikipedia
recall	DANAg	0.942	0.990	0.959	0.997	0.999	0.949	0.993	0.963	1.0	0.613
	Filter 1	0.942	0.987	0.961	0.997	0.999	0.953	0.953	0.963	1.0	0.853
	Filter 2	0.942	0.989	0.961	0.997	0.999	0.953	0.942	0.963	1.0	0.886
	Filter 3	0.942	0.987	0.959	0.997	0.999	0.949	0.993	0.97	1.0	0.81
precision	DANAg	0.970	0.988	0.929	0.994	0.999	0.902	0.989	0.970	0.897	0.912
	Filter 1	0.969	0.952	0.929	0.973	0.999	0.833	0.611	0.97	0.897	0.869
	Filter 2	0.969	0.691	0.918	0.961	0.999	0.831	0.498	0.97	0.897	0.852
	Filter 3	0.969	0.987	0.929	0.994	0.999	0.902	0.989	0.976	0.897	0.915
F1	DANAg	0.949	0.986	0.944	0.995	0.999	0.917	0.991	0.966	0.945	0.699
	Filter 1	0.949	0.969	0.944	0.985	0.999	0.884	0.716	0.966	0.945	0.852
	Filter 2	0.949	0.804	0.939	0.979	0.999	0.883	0.624	0.966	0.945	0.861
	Filter 3	0.949	0.985	0.944	0.996	0.999	0.917	0.991	0.973	0.945	0.852

Table 4: Comparison between Recall, Precision and F1 on the corpus in Table 2.

		BBC	Economist	Zdf	Golem	Heise	Republica	Spiegel	Telepolis	Yahoo	Wikipedia	Manual	Slashdot
recall	DANAg	0.893	0.966	0.963	0.997	0.945	0.997	0.942	0.979	0.955	0.578	0.680	0.318
	Filter 1	0.913	0.967	0.963	0.993	0.976	0.995	0.946	0.979	0.954	0.810	0.687	0.399
	Filter 2	0.922	0.967	0.963	0.745	0.965	0.994	0.946	0.979	0.952	0.760	0.690	0.440
	Filter 3	0.890	0.967	0.963	0.999	0.964	0.996	0.941	0.980	0.953	0.787	0.686	0.372
precision	DANAg	0.991	0.855	0.882	0.963	0.945	0.955	0.969	0.919	0.950	0.782	0.359	0.174
	Filter 1	0.991	0.830	0.880	0.941	0.900	0.872	0.943	0.914	0.948	0.927	0.355	0.208
	Filter 2	0.935	0.732	0.812	0.707	0.830	0.792	0.938	0.914	0.944	0.882	0.356	0.192
	Filter 3	0.991	0.855	0.880	0.989	0.911	0.954	0.974	0.919	0.948	0.927	0.357	0.197
F1	DANAg	0.924	0.990	0.912	0.979	0.955	0.970	0.949	0.932	0.952	0.645	0.401	0.209
	Filter 1	0.939	0.884	0.910	0.965	0.931	0.914	0.938	0.930	0.950	0.856	0.403	0.248
	Filter 2	0.916	0.827	0.871	0.724	0.884	0.865	0.937	0.930	0.948	0.809	0.404	0.239
	Filter 3	0.922	0.900	0.910	0.994	0.931	0.970	0.951	0.932	0.950	0.840	0.404	0.236

3 have calculated the recall value as one.

- When Filter 2 has a higher recall and a lower precision than the other two filters, it can be concluded that a major part of the extraneous items has been selected as the MC. It is well known that the menus are regarded as one of the additional items in the web pages and each item in the menu is usually built by an anchor tag. Therefore, by application of Filter 2, it would be possible to consider menus as the MC in some of the web sites such as BBC Arabic. However, the value of recall is equal to 0.989 in the web site of BBC Arabic, which is excellent. On the other hand, the value of precision is reported to be 0.804 which is rather poor and indicates existence of some words in the final MC which can hardly be taken as a part of MC.

5 CONCLUSIONS AND FUTURE WORKS

In this paper, three simple pre-processing methods are proposed which can be combined with the line-based main content extraction methods. These methods have been compared with each other and the results show that Filter 3 yields better output values. Especially on hyperlink rich web documents such as Wikipedia, Filter 3 clearly outperforms to the other 2 pre-processing methods.

For the future work, it is recommended to combine the already introduced pre-processing methods with some other state-of-the-art main content extraction approaches, such as CETR (Weninger et al., 2010), Density (Moreno et al., 2009), and ACCB (Gottron, 2008).

ACKNOWLEDGEMENTS

We would like to thank Thomas Gottron for providing us the dataset which was used in the evaluation of this work.

REFERENCES

Finn, A., Kushmerick, N., and Smyth, B. (2001). Fact or fiction: Content classification for digital libraries. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*.

Gottron, T. (2007). Evaluating content extraction on HTML documents. In *ITA '07: Proceedings of the 2nd International Conference on Internet Technologies and Applications*, pages 123 – 132, Wrexham, Wales, UK.

Gottron, T. (2008). Content code blurring: A new approach to content extraction. In *DEXA'08: 19th International Workshop on Database and Expert Systems Applications*, pages 29 – 33, Turin, Italy. IEEE Computer Society.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 441–450, New York, NY, USA. ACM.

Mohammadzadeh, H., Gottron, T., Schweiggert, F., and Nakhaeizadeh, G. (2012). The impact of source code normalization on main content extraction. In *WEBIST'12: 8th International Conference on Web Information Systems and Technologies*, pages 677 – 682, Porto, Portugal. SciTePress.

Mohammadzadeh, H., Gottron, T., Schweiggert, F., and Nakhaeizadeh, G. (2013). Extracting the main content of web documents based on character encoding and a naive smoothing method. In *Software and Data Technologies, CCIS Series, Springer*, pages 217 – 236. Springer-Verlag Berlin Heidelberg.

Moreno, J., Deschacht, K., and Moens, M. (2009). Language independent content extraction from web pages. In *Proceeding of the 9th Dutch-Belgian Information Retrieval Workshop*, pages 50 – 55.

Pinto, D., Branstein, M., Coleman, R., Croft, W. B., King, M., Li, W., and Wei, X. (2002). QuASM: a system for question answering using semi-structured data. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 46 – 55, New York, NY, USA. ACM Press.

Weninger, T., Hsu, W. H., and Han, J. (2010). CETR: content extraction via tag ratios. In *Proceedings of the 19th International Conference on World Wide Web*, pages 971 – 980. ACM Press.