# THE IMPACT OF SOURCE CODE NORMALIZATION ON MAIN CONTENT EXTRACTION

Hadi Mohammadzadeh[1], Thomas Gottron[2], Franz Schweiggert[1] and Gholamreza Nakhaeizadeh[3]
[1]*Institute of Applied Information Processing, University of Ulm, D-89069 Ulm, Germany*
[2]*Institute for Web Science and Technologies, Universität Koblenz-Landau, D-56070 Koblenz, Germany*
[3]*Institute of Statistics, Econometrics and Mathematical Finance, University of Karlsruhe, D-76128 Karlsruhe, Germany*

Keywords: Main Content Extraction, Web Mining, HTML Web Pages, Normalization.

Abstract: In this paper, we introduce AdDANAg, a language-independent approach to extract the main content of web documents. The approach combines best-of-breed techniques from recent content extraction approaches to yield better extraction results. This combination of techniques brings together two pre-processing steps, e.g. to normalize the document presentation and reduce the impact of certain syntactical structures, and four phases for the actual content extraction. We show that AdDANAg demonstrates a high performance in terms of effectiveness and efficiency and outperforms previous approaches.

## 1 INTRODUCTION

Content Extraction (CE) is defined as the task of determining the main textual content of an HTML document and/or removing the additional items, such as navigation menus, copyright notices or advertisements (Gottron, 2008).

In recent years, many algorithms and methods have been introduced to provide solutions to this task (Gottron, 2008; Mohammadzadeh et al., 2011a; Moreno et al., 2009; Weninger and Hsu, 2008). However, none of these methods demonstrated perfect results on all types of web documents. A typical flaw is a poor extraction performance on highly structured web documents or documents containing many hyperlinks. The structure of such documents contradicts the typical hypothesis underlying CE methods: the main content forms a rather long and uniformly formatted region of text. Wiki documents are a typical example contradicting this hypothesis. To analyse the behaviour of CE methods in these difficult scenarios, CE evaluation datasets typically also contain documents from Wikipedia. Following this line of thought we will use Wikipedia as a representative example to illustrate our approach throughout the paper. Figure 1 shows an example of a Wikipedia web page with a highlighted main content.

The challenge posed by highly structured documents has been previously addressed by the ACCB algorithm (Gottron, 2008), which coped with the problem by introducing a dedicated pre-processing step. ACCB achieved more accurate results compared to its baseline CCB. Motivated by this success, we introduce in this paper the AdDANAg approach, which incorporates a pre-processing step inspired by ACCB with the current state-of-the-art CE algorithm DANAg (Mohammadzadeh et al., 2011a). We apply AdDANAg to an established collection of web doc-

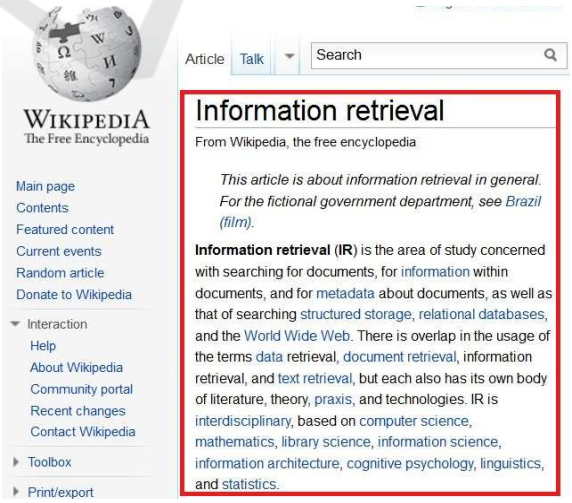

Figure 1: An example of a web document with a highlighted main content.

Mohammadzadeh H., Gottron T., Schweiggert F. and Nakhaeizadeh G..
THE IMPACT OF SOURCE CODE NORMALIZATION ON MAIN CONTENT EXTRACTION.
DOI: 10.5220/0003931906770682
In *Proceedings of the 8th International Conference on Web Information Systems and Technologies* (WEBIST-2012), pages 677-682
ISBN: 978-989-8565-08-2

uments and demonstrate an overall improvement in main content extraction performance on highly structured documents while in general maintaining its very good performance on all other documents.

We proceed as follows: in Section 2, we briefly review related work in the field of CE. Our novel AdDANAg approach is introduced in Section 3. The data sets and experiments are explained in Section 4. In Section 5, we conclude the paper and outline suggestions for future work.

## 2 RELATED WORK

Content extraction algorithms can be subdivided in three families, based on the structures they analyse: the HTML DOM tree, HTML source code elements or the character encoding.

### 2.1 Methods based on DOM Tree Analysis

In this section, we list three of most outstanding methods operating on a DOM tree. These methods were introduced by Gupta et al. (Gupta et al., 2003), Mantratzis et al.(Mantratzis et al., 2005), and Debnath et al. (Debnath et al., 2005) and are called, respectively, Crunch framework, Link Quota Filter(LQF), and finally the FeatureExtractor algorithm and its extension the K-FeatureExtractor. All of these methods first construct a DOM tree from an HTML web page using an HTML parser.

The first method navigates the DOM tree respectively and utilizes a number of heuristic filtering techniques to extract the main content of an HTML web page. The second method, LQF, determines the areas with a high hyperlink density within a web document, so these areas can be separated from the main content. In the last two methods, the "primary content blocks" are identified based on various features. In the first step, they segment the web pages into web page blocks and, second, they separate the primary content blocks from the non-informative content blocks based on their compliance with desired features, such as dominance of text or images.

### 2.2 Methods based on the Analysis of HTML Source Code Elements

Methods operating on source code elements, such as tags and words represent are the most common and effective approach for CE. In this section we highlight five of the most prominent approaches.

Finn et al. (Finn et al., 2001) proposed "Body Text Extraction" (BTE). BTE identifies only a single continuous fragment of the HTML document containing the main content. This method is based on tokenizing a web document into a binary vector of word and tag elements. Afterwards, BTE chooses a fragment containing a high percentage of text against low percentage of tags.

Pinto et al. (Pinto et al., 2002) extended this approach to the "Document Slope Curves" algorithm (DSC). The aim of this method is to overcome the restriction of BTE to be capable to extract more than one continuous block of text tokens. In doing this, in an intermediary step, DSC generates a graph by plotting the accumulated tag token count for each entry in the vector. Then, the approach extracts long and low sloping regions of this graph represent the main content (text without HTML tags). By employing a windowing technique, the approach can identify also a main content which is fragmented into several parts of an HTML document.

Gottron (Gottron, 2008) presented two new algorithms: Content Code Blurring (CCB) and Adapted Content Code Blurring (ACCB) are capable of working either on characters or tokens. CCB finds the regions in an HTML document which contains mainly content and little code. In order to do this, the algorithm, determines a ratio of content to code for each single element in the content code vector (CCV) by using a Gaussian blurring filter, building a new vector, referred to as Content Code Ratio (CCR). Now a region with high CCR values indicates the main content. In ACCB, all anchor-tags are ignored during the creation of the CCV. Two parameters influence the behaviour of these two algorithms, so tuning these two parameters is important in order to produce good results (Gottron, 2009).

Weninger et al. (Weninger and Hsu, 2008) introduced CETR: CE via tag ratios. Their method computes tag ratios on a line-by-line basis and, afterwards, produces a histogram based on results. Finally, by using the k-means clustering method, they group the resulting histogram into the content and the non-content area.

Finally, Moreno et al. (Moreno et al., 2009) proposed Density. This approach has two phases. In the first step, they separate texts from the HTML tags by using an HTML parser; afterwards, the extracted texts are saved in an array of strings L. In the second step, a region in the array L that has the highest density will be determined as a main content.

### 2.3 Methods based on Analysing Character Encoding

The last category of CE methods operates on the character encoding of a document.

In the first, simple algorithm, R2L, was proposed by Mohammadzadeh et al. (Mohammadzadeh et al., 2011c). R2L is independent from the DOM tree and HTML tags and extracts the main content of special language web pages, such as Arabic, Farsi, Urdu, and Pashto, written from right to left. The proposed method relies on the simple rule: the distinction between characters which are used in English characters and right to left language characters. The shortcoming of R2L is that it is language dependent. Moreover, it loses the Non-R2L characters, for example English characters, of the main content because when R2L separates characters, it categorizes all Non-R2L characters incorrectly as the English characters while some of these Non-R2L characters are members of the main content.

The latter flaw was overcome by DANA (Mohammadzadeh et al., 2011b), an extended release of R2L with considerable effectiveness. DANA like R2L approach is still language-dependent but it can cope with Non-R2L characters in the main content.

Finally, Mohammadzadeh et al. (Mohammadzadeh et al., 2011a) presented DANAg, a language-independent version by incorporating an HTML parser. DANAg shows high efficiency and a very accurate extraction of the main content.

## 3 AdDANAg

AdDANAg is inspired by an adaptation of the pre-processing step of ACCB (Gottron, 2008) and the current state-of-the-art content extraction approach DANAg (Mohammadzadeh et al., 2011a). The process behind AdDANAg can be divided into a pre-processing phase and a core extraction phase. While the core extraction phase is taken from DANAg, the pre-processing from ACCB has been adapted and enhanced. Our preprocessing normalizes imbalances in the source code structure that hinder typical CE approaches. The imbalances can be motivated due to technical constraints or domain specific deviations from the typical source code patterns. Important is to note, that they do not imply a semantic change in the main content specifications. Below we explain the pre-processing step of AdDANAg in detail and recall the core extraction phase for the sake of completeness of the paper.

### 3.1 Pre-processing

A common problem of CE methods based on source code analysis is, that on hyperlink rich web documents they cannot detect the main content accurately. This can be explained with the code for hyperlinks prevailing over the actual content items, which contradicts typical assumptions made by the content extraction methods. Illustrative examples are given in Figure 2 showing some paragraphs of a normal HTML file while Figure 3 represents some portions of source code with many hyperlinks. In Figure 2, there is no hyperlink, so approaches like DANAg or CCB have no problems in extracting the main content accurately. In comparison, in Figure 3, there are plenty of hyperlinks which introduces a strong bias towards code structures.

Therefore, in the pre-processing step of AdDANAg, we normalise all HTML hyperlinks using a fast approach based on substitution rules. For better understanding we explain the approach using a illustrative example. Suppose the following hyperlink to be contained in an HTML file:

```
<a href="http://www.BBC.com/">BBC Web Site</a>
```

Here, there is only one attribute, which is `href="http://www.BBC.com/"`. Now, the length of the anchor text (in this example: `BBC Web Site`) is determined and we refer to this value by LT. Then, we substitute the attribute part of the opening tag with a placeholder text of length LT - 5, where the subtracted 5 comes from the length of `<a></a>`. So, using the underscore sign _ as placeholder the new hyperlink for our example should be as below:

```
<a ________>BBC Web Site</a>
```

The purpose of this rule is to normalise the ratio of content and code characters representing hyperlinks. This counterbalances inequalities originating from the URLs in hyperlinks.

```
        <p>Earlier this week Mark Bowden, the UN humanitarian affairs co-ordinator
 for Somalia, told the BBC that the country was close to famine.  </p>
        <p>Last week Somalia&#039;s al-Shabab Islamist militia - which has been
 fighting the Mogadishu government - said it was lifting its ban on foreign aid
 agencies provided they did not show a &quot;hidden agenda&quot;. </p>
        <p>Some 3,000 people flee each day for neighbouring countries such as
 Ethiopia and Kenya which are struggling to cope.</p>
```

Figure 2: Some paragraphs of regular HTML file.

### 3.2 The Core Extraction Phase

AdDANAg follows the common hypothesis, that the main content is composed of a relatively homogeneous region of text. For the web document in Figure 4, this hypothesis holds true for the shown text

```
<p>The present significance of IE pertains to the growing amount of information
available in unstructured form. <a href="/wiki/Tim_Berners-Lee" title="Tim
Berners-Lee">Tim Berners-Lee</a>, inventor of the <a href="/wiki/World_wide_web"
title="World wide web" class="mw-redirect">world wide web</a>, refers to the
existing <a href="/wiki/Internet" title="Internet">Internet</a> as the web of
<i>documents</i> <sup id="cite_ref-2" class="reference"><a
href="#cite_note-2"><span>[</span>3<span>]</span></a></sup> and advocates that
more of the content be made available as a <a href="/wiki/Semantic_web"
title="Semantic web" class="mw-redirect">web of <i>data</i></a>.<sup
id="cite_ref-3" class="reference"><a
href="#cite_note-3"><span>[</span>4<span>]</span></a></sup> Until this transpires,
the web largely consists of unstructured documents lacking semantic <a
href="/wiki/Metadata" title="Metadata">metadata</a>. Knowledge contained within
these documents can be made more accessible for machine processing by means of
transformation into <a href="/wiki/Relational_database" title="Relational
database">relational form</a>, or by marking-up with <a href="/wiki/XML"
title="XML">XML</a> tags. An intelligent agent monitoring a news data feed
```

Figure 3: Some portions of hyperlink rich HTML file.

fragment. The challenge is now to detect this relatively homogeneous region of text in the source code.

The core extraction phase of AdDANAg in general follows the process designed for DANAg (Mohammadzadeh et al., 2011a). This incorporates two additional pre-processing steps: first, to remove all JavaScript codes, CSS style codes, and comments from the HTML file and retain only the HTML code as the output, and second, to normalize the distribution of newline characters in the source code as it operates on the level of lines.


Figure 4: An example of Wikipedia web pages.

The actual extraction phase of AdDANAg is divided into four steps. In the first step, AdDANAg calculates the amount of content and code characters in each line of an HTML file and stores these numbers in two one-dimensional arrays $T$ and $S$, respectively. For the web document shown in Figure 4 the plot in Figure 5 visualizes these arrays as a plot of two types of columns above and below the x-axis with the length equal to the values of $T$ and $S$, respectively. The columns above the x-axis represent the amount of content of each line in the source code, while the columns below the x-axis represent the amount of code characters. The normalisation of the hyperlinks in the pre-processing phase of AdDANAg here takes its effect, as for those lines with many hyperlinks it equally modifies the values in $T$ and $S$; this is the key point of AdDANAg.

AdDANAg algorithm's hypothesis of the main content being composed of a relatively homogeneous region of text corresponds to a dense region of columns positioned above the x-axis. The normalisation avoids a strong fragmentation of these areas due to extensive hyperlink markup. In Figure 5, the actual main content area is located in the lines 15 to 540 and the remaining part of diagram belongs to the extraneous items such as menus and advertisements. Consequently on hyperlink rich web pages, combining the algorithm DANAg with our new pre-processing phase result in a better retention of the columns located above x-axis representing the main content. The aim of AdDANAg is to recognize the main content area based on this representation. This is implemented in the following three steps.

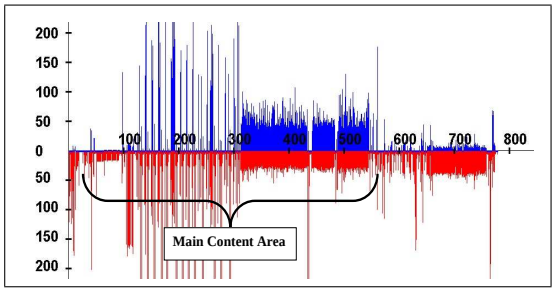


Figure 5: A plot of the amount of content and code characters per line in the source code of the web document shown in Figure 4.

In the next step, AdDANAg computes $diff_i$ through Formula 1 for each line $i$ of the HTML file, and keeps these new numbers in a one-dimensional array $D$, to produce a smoothed plot which can be seen in Figure 6. The plot in Figure 6 draws a column above the x-axis if $diff_i > 0$. Otherwise, a line with length $|diff_i|$ is depicted below the x-axis. Looking at Figure 6, the region of the main content is now the only part of the plot with a dense part of the plot above the x-axis. Thus, it becomes easier to recognize the main content. Considering the positive val-

ues of $D$, AdDANAg identifies all paragraphs of the main content located above the x-axis and, for simplicity, it defines a new set $R = \{r_1, r_2, ..., r_n\}$ of all such paragraphs. Each element $r_j \in R$ denotes only one individual paragraph and $n$ is the total number of recognized paragraphs above the x-axis.

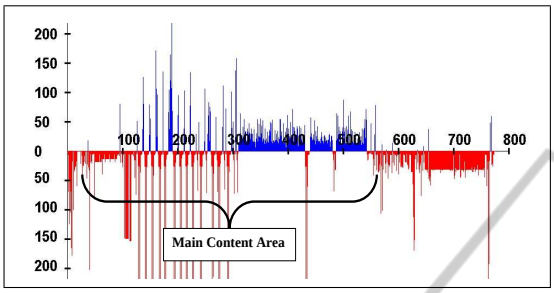


Figure 6: A smoothed plot of Figure 5.

$$diff_i = (T_i - S_i) + (T_{i+1} - S_{i+1}) + (T_{i-1} - S_{i-1}) \quad (1)$$

In the third step, AdDANAg discovers all paragraphs shaping the main content. We refer to the original paper for the technical details (Mohammadzadeh et al., 2011a).

Finally, AdDANAg feeds all these extracted paragraphs to a parser (Gottron, 2008) to obtain the main content tokens of the HTML file.

## 4 DATA SET AND RESULTS

For the purpose of evaluation we use the dataset introduced in (Gottron, 2008) and (Mohammadzadeh et al., 2011c), composed of 9,101 and 2,166, respectively, web pages from different web sites for which the main content is provided as a gold standard. The composition and size of the evaluation data sets are presented in Table 1 and Table 2, respectively.

The F1 scores regarding the accurate extraction of the main content are presented in Table 3 and Table 4. A first observation is that in comparison to DANAg, AdDANAg does not show drawbacks in general for standard web documents. Further, AdDANAg and DANAg for most documents deliver the best results. When focussing on the set of Wikipedia web pages, which have been observed to be extremely difficult for main content, we can observe that AdDANAg clearly outperforms DANAg and all other approaches. The overall average F1 score for both DANAg and AdDANAg in Table 3 are 0.8099 and 0.8284, respectively.

Table 1: Evaluation corpus of 9101 web pages.

| Web site | Source | Size | Languages |
|---|---|---|---|
| BBC | www.bbc.co.uk | 1000 | English |
| Economist | www.economist.com | 250 | English |
| Golem | golem.de | 1000 | German |
| Heise | www.heise.de | 1000 | German |
| Manual | several | 65 | German, English |
| Repubblica | www.repubblica.it | 1000 | Italian |
| Slashdot | slashdot.org | 364 | English |
| Spiegel | www.spiegel.de | 1000 | German |
| Telepolis | www.telepolis.de | 1000 | German |
| Wiki | fa.wikipedia.org | 1000 | English |
| Yahoo | news.yahoo.com | 1000 | English |
| Zdf | www.heute.de | 422 | German |

Table 2: Evaluation corpus of 2166 web pages.

| Web site | Source | Size | Lang. |
|---|---|---|---|
| BBC | www.bbc.co.uk/persian | 598 | Farsi |
| Hamshahri | hamshahrionline.ir | 375 | Farsi |
| Jame Jam | www.jamejamonline.ir | 136 | Farsi |
| Ahram | www.jamejamonline.ir | 188 | Arabic |
| Reuters | ara.reuters.com | 116 | Arabic |
| Embassy of Germany | www.teheran.diplo.de Vertretung/teheran/fa | 31 | Farsi |
| BBC | www.bbc.co.uk/urdu | 234 | Urdu |
| BBC | www.bbc.co.uk/pashto | 203 | Pashto |
| BBC | www.bbc.co.uk/arabic | 252 | Arabic |
| Wiki | fa.wikipedia.org | 33 | Farsi |

## 5 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed AdDANAg as a combination and variation of DANAg and the pre-processing of ACCB, with considerable effectiveness. Results show AdDANAg determines the main content with high accuracy on many web documents. Especially, also on the difficult to handle hyperlink rich web documents AdDANAg shows previously unseen good performance. In future, we are going to extend AdDANAg to propose parameter-free approach.

## ACKNOWLEDGEMENTS

We would like to thank Mina Aramideh for editing some parts of this paper.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257859, ROBUST.

Table 3: The Average F1 Scores of AdDANAg based on the corpus in Table 1.

| | BBC | Economist | Zdf | Golem | Heise | Repubblica | Spiegel | Telepolis | Yahoo | Wikipedia | Manual | Slashdot |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Plain | 0.595 | 0.613 | 0.514 | 0.502 | 0.575 | 0.704 | 0.549 | 0.906 | 0.582 | *0.823* | 0.371 | 0.106 |
| LQF | 0.826 | 0.720 | 0.578 | 0.806 | 0.787 | 0.816 | 0.775 | 0.910 | 0.670 | 0.752 | 0.381 | 0.127 |
| Crunch | 0.756 | 0.815 | 0.772 | 0.837 | 0.810 | 0.887 | 0.706 | 0.859 | 0.738 | 0.725 | 0.382 | 0.123 |
| DSC | **0.937** | 0.881 | 0.847 | 0.958 | 0.877 | 0.925 | 0.902 | 0.902 | 0.780 | 0.594 | 0.403 | 0.252 |
| TCCB | 0.914 | 0.903 | 0.745 | 0.947 | 0.821 | 0.918 | *0.910* | *0.913* | 0.758 | 0.660 | 0.404 | 0.269 |
| CCB | 0.923 | **0.914** | 0.929 | 0.935 | 0.841 | 0.964 | 0.858 | 0.908 | 0.742 | 0.403 | 0.420 | 0.160 |
| ACCB | 0.924 | 0.890 | **0.929** | *0.959* | *0.916* | *0.968* | 0.861 | 0.908 | 0.732 | 0.682 | **0.419** | 0.177 |
| Density | 0.575 | 0.874 | 0.708 | 0.873 | 0.906 | 0.344 | 0.761 | 0.804 | *0.886* | 0.708 | 0.354 | **0.362** |
| DANAg | 0.924 | 0.900 | 0.912 | 0.979 | **0.945** | **0.970** | 0.949 | **0.932** | **0.952** | 0.646 | 0.401 | 0.209 |
| AdDANAg | 0.922 | 0.900 | 0.911 | **0.994** | 0.931 | **0.970** | **0.951** | **0.932** | 0.950 | **0.840** | 0.404 | 0.236 |

Table 4: The Average F1 Scores of AdDANAg based on the corpus in Table 2.

| | Al Ahram | BBC Arabic | BBC Pashto | BBC Persian | BBC Urdu | Embassy | Hamshahri | Jame Jam | Reuters | Wikipedia |
|---|---|---|---|---|---|---|---|---|---|---|
| ACCB-40 | 0.871 | 0.826 | 0.859 | 0.892 | 0.948 | 0.784 | 0.842 | 0.840 | 0.900 | 0.736 |
| BTE | 0.853 | 0.496 | 0.854 | 0.589 | 0.961 | 0.810 | 0.480 | 0.791 | 0.889 | 0.817 |
| DSC | 0.871 | *0.885* | 0.840 | *0.950* | 0.896 | 0.824 | *0.948* | 0.914 | 0.851 | 0.747 |
| FE | 0.809 | 0.060 | 0.165 | 0.063 | 0.002 | 0.017 | 0.225 | 0.027 | 0.241 | 0.225 |
| KFE | 0.690 | 0.717 | 0.835 | 0.748 | 0.750 | 0.762 | 0.678 | 0.783 | 0.825 | 0.624 |
| LQF-25 | 0.788 | 0.780 | 0.844 | 0.841 | 0.957 | 0.860 | 0.765 | 0.737 | 0.870 | 0.773 |
| LQF-50 | 0.785 | 0.777 | 0.837 | 0.828 | 0.954 | 0.856 | 0.767 | 0.724 | 0.870 | 0.772 |
| LQF-75 | 0.773 | 0.773 | 0.837 | 0.819 | 0.954 | 0.852 | 0.756 | 0.724 | 0.870 | 0.750 |
| TCCB-18 | *0.886* | 0.826 | *0.912* | 0.925 | 0.990 | *0.887* | 0.871 | *0.929* | **0.959** | 0.814 |
| TCCB-25 | 0.874 | 0.861 | 0.909 | 0.927 | *0.992* | 0.883 | 0.888 | 0.924 | 0.958 | 0.814 |
| Density | 0.879 | 0.202 | 0.908 | 0.741 | 0.958 | 0.882 | 0.920 | 0.907 | 0.934 | 0.665 |
| DANAg | **0.949** | **0.986** | **0.944** | **0.995** | **0.999** | **0.917** | **0.991** | 0.966 | 0.945 | 0.699 |
| AdDANAg | **0.949** | 0.985 | **0.944** | **0.996** | **0.999** | **0.917** | **0.991** | **0.973** | 0.945 | **0.852** |



# REFERENCES

Debnath, S., Mitra, P., and Lee Giles, C. (2005). Identifying content blocks from web documents. In *Foundations of Intelligent Systems*, Lecture Notes in Computer Science, pages 285–293.

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. (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. IEEE Computer Society.

Gottron, T. (2009). An evolutionary approach to automatically optimise web content extraction. In *IIS'09: Proceedings of the 17th International Conference Intelligent Information Systems*, pages 331–343.

Gupta, S., Kaiser, G., Neistadt, D., and Grimm, P. (2003). DOM-based content extraction of HTML documents. In *WWW '03: Proceedings of the 12th International Conference on World Wide Web*, pages 207–214, New York, NY, USA. ACM Press.

Mantratzis, C., Orgun, M., and Cassidy, S. (2005). Separating XHTML content from navigation clutter using DOM-structure block analysis. In *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, pages 145–147, New York, NY, USA. ACM Press.

Mohammadzadeh, H., Gottron, T., Schweiggert, F., and Nakhaeizadeh, G. (2011a). Extracting the main content of web documents based on a naive smoothing method. In *KDIR'11: International Conference on Knowledge Discovery and Information Retrieval*, pages 470 – 475. SciTePress.

Mohammadzadeh, H., Gottron, T., Schweiggert, F., and Nakhaeizadeh, G. (2011b). A fast and accurate approach for main content extraction based on character encoding. In *DEXA '11: 22th International Workshop on Database and Expert Systems Applications*, pages 167 – 171. IEEE Computer Society.

Mohammadzadeh, H., Schweiggert, F., and Nakhaeizadeh, G. (2011c). Using utf-8 to extract main content of right to left language web pages. In *ICSOFT 2011 - Proceedings of the 6th International Conference on Software and Data Technologies, Volume 1, Seville, Spain, 18-21 July, 2011*, pages 243–249. SciTePress.

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. and Hsu, W. H. (2008). Text extraction from the web via text-tag-ratio. In *TIR '08: Proceedings of the 5th International Workshop on Text Information Retrieval*, pages 23 – 28. IEEE Computer Society.