

# A Comparative Study on Main Content Extraction Algorithms for Right to Left Languages

Houriye Esfahanian<sup>1</sup><sup>a</sup>, Abdolreza Nazemi<sup>2</sup><sup>b</sup> and Andreas Geyer-Schulz<sup>2</sup><sup>c</sup>

<sup>1</sup>Non-Governmental Non-Profit College, Refah, Tehran, Iran

<sup>2</sup>Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

**Keywords:** Main Content Extraction, Evaluation Methods, Boilerplate Detection, Right to Left Languages.

**Abstract:** With the daily increase of published information on the Web, extracting the web page's main content has become an important issue. Since 2010, in addition to the English Language, the contents with the right to left languages such as Arabic or Persian are also increasing. In this paper, we compared the three famous main content extraction algorithms published in the last decade, Boilerpipe, DANAg, and Web-AM, to find the best algorithm considering evaluation measures and performance. The ArticleExtractor algorithm of the Boilerpipe approach was scored as the most accurate algorithm, with the highest average score of F1 measure of 0.951. On the contrary, the DANAg algorithm was selected with the best performance, being able to process more than 21 megabytes per second. Considering the accuracy and the effectiveness of the main content extraction projects, one of the two Boilerpipe or DANAg algorithms can be used.

## 1 INTRODUCTION

Over the last few years, the growth of the Web has led to increased information sharing and the need for efficient organization and extraction of valuable data. Information retrieval (IR) methods are used to store and find relevant information on the Web, but the increase in information requires more server capacity and leads to higher costs. The main content extraction (MCE) methods help control the amount of information on the Web by extracting the essential content, including the title (Mohammadzadeh et al., 2012) and the publication date while discarding unnecessary information. This enables efficient storage of valuable web page content without any additional clutter. The components of a website, such as the header, the footer, the sidebar, and the sections that are repeated on most of the pages of a website called Boilerplate.

There are plenty of usages in the area of the MCE, including search engine optimization and text-to-speech conversion (TTS).


Due to the growing volume of content published in right-to-left (R2L) languages on the Web,


languages like Persian and Arabic now rank among the top 20 most widely published languages. In this article, we specifically concentrated on extracting the main content (MC) from web pages containing R2L content. In this paper, we compare three algorithms in extracting R2L main content from the web pages so we can understand the strengths and weaknesses of each algorithm, and we will be able to create better algorithms with higher efficiency and effectiveness in the future.


This paper will discuss the related works in (2) and then explain the methods in (3). In (4), we will evaluate and compare presented algorithms, and in the last section (5) we will discuss the conclusions.

## 2 RELATED WORKS

In the last two decades, scientists in the field of MCE from the web pages have provided various approaches, being able to extract MC with high accuracy and maximum performance. Some MCE algorithms aimed to find non-main-content such as HTML tags, javascript, and CSS codes so that they

<sup>a</sup> <https://orcid.org/0000-0002-0890-2872>

<sup>b</sup> <https://orcid.org/0000-0002-1157-1066>

<sup>c</sup> <https://orcid.org/0009-0000-5237-3579>

can be removed, and consequently, the rest of the file would be MC. The second group of researchers has been concentrating on finding MC directly on the web pages without considering the non-main-content tags.

## 2.1 MCE Based on Deleting Non-Main-Content

The main objective of (Finn et al., 2007)'s study was to establish a digital library by using the BTE (Body Text Extraction) approach to extract and categorize MC of web pages based on HTML tags. They utilized a diagram to display the distribution of tags in different sections of the web page, ultimately identifying MC through a region with consistent distribution.

(Debnath et al., 2005) introduced two algorithms named FeatureExtractor and K-featureExtractor. The probability of being MC for each block is calculated considering the specified features and tag set. Based on the probability, the non-informative blocks are removed, and secondly, the informative content is extracted. K-featureExtractor applied the K-means algorithm to choose the best set of blocks, instead of one block, to extract a web page's MC.

(Weninger & Hsu, 2008) extracted MC from the web pages with an inline algorithm, which is called TTR (text tag ratio). In TTR, HTML and non-HTML tags are counted in each line, and the ratio is stored in a one-dimensional array. Finally, the content and non-content will be clustered based on the ratio.

(Mantratzis et al., 2005) with creating a DOM tree from the web pages, removing non-important tags like <a> tag, and specifying important tags, has extracted MC by considering the ratio of hyperlinked text to the overall text.

(Gottron, 2008) introduces three algorithms (CCB, ACCB, TCCB) that help diagnose the MC region visually. CCB and ACCB are based on characters, and TCCB is based on tokens. CCB focuses on creating a one-dimensional image by employing content code blurring (CCV) and calculating the code content ratio (CCR), while ACCB disregards anchor tags to enhance the accuracy of extracting wiki-style web documents.

## 2.2 MCE Based on Detecting Main Content

(Chakrabarti et al., 2007) proposed a method to detect page-level templates on web pages. The method involved building a DOM tree of the web pages, assigning "templateness" score to each node based on

specified features, and concluding that a node is a template if all its children are templates.

To remove the non-content in a web page, (Gupta et al., 2003) fed an HTML file into the parser to create a DOM tree. MC is extracted by removing and modifying nodes based on specific filters. (Fernandes et al., 2007) proposed a method to enhance search results by identifying important blocks. They first create a DOM tree of the web pages and then divide them into blocks using the VIPS algorithm. By evaluating the diversity of content within each block, they determine the MC of a web page.

(Vieira et al., 2006) developed a technique to identify website templates, using a process involving providing web pages as input, extracting a common subtree from the DOM tree of pages using the "RTDM-TD" algorithm, signing each extracted subtree, and ultimately detecting and removing the template subtree found in all web pages.

## 3 METHODS

This section will evaluate three algorithms created over the past decade, namely DANAg, Boilerpipe, and Web-AM (Table 1). In addition to L2R content (e.g., the English language), the DANAg and Web-AM algorithms are focused on MCE with R2L content. In this article, we will evaluate all these algorithms.

### 3.1 Boilerpipe

Boilerplate algorithm was presented by (Kohlschütter et al., 2010) and is used for identifying non-content (boilerplate) features such as shallow text features, mean word length, mean sentence length, and absolute number of words, which are based on quantitative linguistics.

Using a machine-learning classification model, it separates the HTML text into content and non-content. First, it converts web pages into blocks of text with a sequence of characters and HTML tags, and then it calculates hyperlink density and the word count in blocks. Blocks with a higher density of hyperlinks are tagged as boilerplate, and blocks with a higher density of normal words are tagged as MC. In addition, text blocks with lower-than-usual lengths are considered as boilerplate and are eliminated in the next stage.

Table 1: Pseudocodes of Algorithms.

Algorithm 1: DefaultExtractor.	Algorithm 2: DANAg.	Algorithm 3: Web-AM Algorithm.
<pre> curr_linkDensity &lt;= 0.333333 prev_linkDensity &lt;= 0.555556 curr_numWords &lt;= 16 next_numWords &lt;= 15 prev_numWords &lt;= 4:     BOILERPLATE prev_numWords &gt; 4:     CONTENT next_numWords &gt; 15:     CONTENT curr_numWords &gt; 16:     CONTENT prev_linkDensity &gt; 0.555556 curr_numWords &lt;= 40 next_numWords &lt;= 17:     BOILERPLATE next_numWords &gt; 17:     CONTENT curr_numWords &gt; 40:     CONTENT curr_linkDensity &gt; 0.333333:     BOILERPLATE                     </pre>	<pre> T = {rm}, R = {r1, r2, ..., m} i = m while i &gt; 1 do     if d(ri, ri-1) ≤ gap then         T = T ∪ {ri-1}     else         break     end if     i=i-1; end while i = m while i &lt; n do     if d(ri, ri+1) ≤ gap then         T = T ∪ {ri+1}     else         break     end if     i=i+1; end while return T                     </pre>	<pre> Input: An HTML Tree (T) Output: Article Text procedure MAIN(T)     SeedNode ← getseednode(T)     ArticleText ← extracttext(SeedNode, T)     return ArticleText procedure GETSEEDNODE(T)     hashMap(String, Integer) ← new     hashMap()     for each node n in T do         hashMap.put(n.path, n.text.length)     max ← 0     seed-node ← “ ”     for each path,length in hashMap do         if length &gt; max then             max ← length             seed-node ← path     return seed-node procedure EXTRACTTEXT(SN, T)     Content,Header ← “ ”     ContentSize ← 500     HeadingSize ← 50     for each node n in T do         Text ← n.text         if n.path == SN.path then             if Text.length ≥ ContentSize then                 Content ← Content + Text             else if Text.length ≥ Header then                 Header ← Text             Content ← Header + Content     return Content                     </pre>

Finally, the MC of the web pages is extracted by removing boilerplate blocks. This approach has various methods to extract MC, including:

### 3.1.1 DefaultExtractor (DE)

DefaultExtractor is a generic full-text extractor based on the number of words/link density classifier.

### 3.1.2 CanolaExtractor (CE)

A full-text extractor was trained on krdwr and Canola databases. These corpora are provided in krdwr<sup>4</sup> project. This is a version of the DefaultExtractor, which was trained on the Canola dataset.

### 3.1.3 ArticleExtractor (AE)

ArticleExtractor is an extension of the DefaultExtractor, which is tuned towards news articles.

### 3.1.4 ArticleExtractor (AE-Py)

We used the code available in Github<sup>5</sup> written in Python2. We converted it to Python3 using 2to3 library<sup>6</sup>.

## 3.2 DANAg

DANAg was introduced by (Mohammadzadeh et al., 2011a). In DANAg, after pre-processing the HTML elements for removing the CSS and JavaScript codes and comments, two one-dimensional arrays are

<sup>4</sup> <https://krdwr.org/>

<sup>5</sup> <https://github.com/Zhiz0id/boilerpipepy>

<sup>6</sup> <https://pypi.org/project/2to3/>

created based on the HTML code length and the length of text in each line, respectively. Then, MC can be identified and stored in a smoothing array by calculating the difference between the lengths of the text and the HTML code in each line. If a line has a negative smoothing value, it means the code density is bigger than the text density, and if a line has a positive smoothing value, it indicates the text will be longer than the HTML code. After identifying the lines with the positive values, the MC of an HTML file can be identified and extracted.

### 3.3 Web-AM

(Aslam et al., 2019) introduced the Web-AM which uses the "ArticleExtractor" algorithm to extract MC from web pages. First, it creates a DOM tree using parsing the HTML output. Afterward, all nodes in the tree that contain a text longer than a *threshold* parameter, usually 500 characters, will be selected and named seed-nodes. The nodes at the same level of the tree with the same tag name as seed-node are marked as cluster-nodes. Finally, the contents of the cluster-nodes are extracted as MC.

## 4 EVALUATION

### 4.1 Datasets

The first dataset was created by (Mohammadzadeh et al., 2011b). This dataset includes ten news websites that are written in R2L languages, such as Arabic, Persian, Urdu, and Pashto (Table 2). The second dataset was created by CURWEB (Aslam et al., 2019)

Table 2: Information of corpus.

Web site	URL	Size	Languages
BBC	bbc.co.uk/persian/	598	Farsi
Hamshahri	hamshahronline.ir/	375	Farsi
Jame Jam	jamejamonline.ir/	136	Farsi
Al Ahram	ahram.org/	188	Arabic
Reuters	ara.reuters.com/	116	Arabic
Embassy of Germany	teheran.diplo.de/Vertretung/teheran/fa/Startseite.html	31	Farsi
BBC	bbc.co.uk/urdu/	234	Urdu
BBC	bbc.co.uk/pashto/	203	Pashto
BBC	bbc.co.uk/arabic/	252	Arabic
Wiki	fa.wikipedia.org/	33	Farsi
Total		2,166	

Table 3: CURWEB dataset.

Category	Quantity	Found
Forum	6	5
GeneralSites	26	19
Litature	4	3
NewsSites	66	60
Religious	21	21
Uncategorized	78	72
Total	201	180

and includes Urdu language websites (Table 3).

For evaluating the results, we require three separate data types. First, the original HTML files. The second one is Gold Standard files, which contain MC and are extracted manually. The last type is cleaned files, where their contents are extracted using algorithms.

Two important points concerning the CURWEB dataset should be mentioned. First, the HTML files were unavailable, so the URLs only are considered for evaluation. Due to a 404 error on some URLs, we had no access to all files (Table 3). The second point is that we have changed the content of the Gold Standard based on the new definition presented in the introduction section.

### 4.2 Evaluation Methodology

Two sets of golden and cleaned data and information retrieval criteria, such as precision, recall, and F1, are used to evaluate and compare algorithms in terms of accuracy and performance (Gotttron, 2007). In these equations, golden and cleaned data, respectively, include the content extracted manually and the content extracted using algorithms, and LCS (Longest Common Subsequence) value represents the shared content between golden and cleaned files. All the criteria used, precision, recall, and F1, are between 0 and 1. A value of 0 is considered the worst, whereas a value of 1 is considered the best.

### 4.3 Results and Discussion

We calculated recall, precision, and F1 for all three approaches on two datasets. The results were demonstrated in Tables 4 to 10. On Arabic domains (Ahram, BBC Arabic, Reuters), DANAg outperformed other algorithms with F1 = 0.961, while the F1 measure for ArticleExtractor is 0.957 (Table 9). In addition, DANAg has a precision of 0.95 and is ranked after the Web-AM (0.986 Table 8), and concerning the recall, it is ranked as the second algorithm (0.976) after ArticleExtractor (0.994 Table 7).

According to the Urdu dataset (including BBC Urdu, Forum, General Sites, Literature, News Sites, Religious, and Uncategorized), Web-AM, ArticleExtractor, and again ArticleExtractor demonstrates the highest precision, recall, and F1 scores, with values of 0.928, 0.948, and 0.951 respectively. A similar ranking is resulted in the Persian database (including BBC Persian, Embassy, Hamshahri, and Jamejam), corresponding values of 0.995, 0.979, and 0.979. In the news datasets (BBC Arabic, BBC Urdu, BBC Persian, BBC Pashtoo), DANAg has the highest recall value of 0.985 compared to the second approach, ArticleExtractor (0.98). The ArticleExtractor has the highest precision, with a value of 0.995, compared to the Web-AM algorithm, with a value of 0.99. In total,

ArticleExtractor is the best algorithm for extracting MC from news websites (F1 = 0.987).

In the Wiki dataset, the DefaultExtractor demonstrated the highest F1-measure of 0.817. Additionally, the CanolaExtractor and the Web-AM achieved remarkable recall and precision values, with a recall of 0.83 and a precision of 0.994, respectively.

In general (Table 10), based on the entire two datasets and the average performance of all algorithms, we can summarize the following results: The Web-AM usually has the highest precision value. It can be inferred that the algorithm can extract the MC cleanly and with minimal boilerplate (being able to detect boilerplate precisely). High-precision results in the context of IR prove that the algorithm has the great ability to detect and remove boilerplates.

Table 4: Evaluation results based on Recall on CURWEB dataset.

Datasets	Boilerpipe				DANAg	Web-AM
	AE	CE	DE	AE-Py		
Forum	0.944	0.854	0.938	<b>0.955</b>	0.948	0.887
GeneralSites	<b>0.938</b>	0.914	0.874	0.907	0.843	0.876
Litrature	<b>0.939</b>	0.833	0.83	0.846	0.666	0.678
NewsSites	0.94	<b>0.951</b>	0.898	0.943	0.838	0.863
Religious	<b>0.958</b>	0.92	0.91	0.956	0.69	0.89
Uncategorized	<b>0.963</b>	0.944	0.909	0.949	0.763	0.845
Average	<b>0.947</b>	0.902	0.893	0.926	0.791	0.839

Table 5: Evaluation results based on Precision on CURWEB dataset.

Datasets	Boilerpipe				DANAg	Web-AM
	AE	CE	DE	AE-Py		
Forum	<b>0.993</b>	0.846	0.86	0.841	0.991	0.989
GeneralSites	0.933	0.761	0.871	0.879	0.855	<b>0.955</b>
Litrature	<b>0.749</b>	0.667	0.647	0.657	0.647	0.715
NewsSites	0.939	0.719	0.882	0.891	0.848	<b>0.971</b>
Religious	0.944	0.837	0.896	0.886	0.791	<b>0.993</b>
Uncategorized	0.929	0.634	0.858	0.856	0.798	<b>0.954</b>
Average	0.914	0.744	0.835	0.835	0.821	<b>0.929</b>

Table 6: Evaluation results based on F1 on CURWEB dataset.

Datasets	Boilerpipe				DANAg	Web-AM
	AE	CE	DE	AE-Py		
Forum	0.944	0.848	0.891	0.891	0.968	0.934
GeneralSites	0.938	0.801	0.859	0.89	0.848	0.91
Litrature	0.939	0.734	0.715	0.729	0.656	0.695
NewsSites	0.94	0.797	0.872	0.909	0.839	0.912
Religious	0.958	0.859	0.89	0.91	0.704	0.935
Uncategorized	0.963	0.719	0.864	0.882	0.774	0.882
Average	0.947	0.793	0.848	0.868	0.798	0.878

Table 7: Evaluation results based on Recall on dataset in Table 2.

Datasets	Boilerpipe				DANAg	Web-AM
	AE	CE	DE	AE-Py		
Ahram	<b>0.998</b>	0.975	0.755	0.979	0.942	0.874
BBC Arabic	<b>0.999</b>	0.959	0.981	0.976	0.987	0.912
BBC Pashtoo	<b>0.969</b>	0.909	0.932	0.93	0.959	0.899
BBC Persian	0.993	0.975	0.989	0.984	<b>0.997</b>	0.921
BBC Urdu	0.959	0.94	0.845	0.94	<b>0.999</b>	0.807
Embassy	<b>0.976</b>	0.942	0.948	0.97	0.949	0.916
Hamshahri	0.981	0.97	0.966	0.964	<b>0.993</b>	0.846
Jamejam	<b>0.968</b>	0.93	0.912	0.947	0.963	0.785
Reuters	0.986	0.937	0.941	0.945	<b>1</b>	0.736
Wiki	0.684	0.83	0.746	0.788	0.613	0.584
Average	<b>0.951</b>	0.936	0.901	0.942	0.94	0.828

Table 8: Evaluation results based on Precision on dataset in Table 2.

Datasets	Boilerpipe				DANAg	Web-AM
	AE	CE	DE	AE-Py		
Ahram	0.87	0.902	0.767	0.876	0.969	<b>0.972</b>
BBC Arabic	<b>0.997</b>	0.754	0.888	0.773	0.986	0.992
BBC Pashtoo	<b>0.993</b>	0.992	0.992	<b>0.993</b>	0.929	0.991
BBC Persian	<b>0.996</b>	0.825	0.916	0.84	0.994	0.995
BBC Urdu	0.994	0.994	0.994	0.994	<b>0.999</b>	0.982
Embassy	0.952	0.887	0.872	0.886	0.902	<b>0.996</b>
Hamshahri	0.982	0.582	0.859	0.822	<b>0.998</b>	0.994
Jamejam	0.994	0.793	0.893	0.892	0.97	<b>0.997</b>
Reuters	0.91	<b>0.997</b>	<b>0.997</b>	0.906	0.897	0.994
Wiki	0.973	0.825	0.947	0.912	0.912	<b>0.994</b>
Average	0.966	<b>0.855</b>	0.912	0.889	0.955	<b>0.99</b>

Table 9: Evaluation results based on F1 on dataset in Table 2.

Datasets	Boilerpipe				DANAg	WEB-AM
	AE	CE	DE	AE-Py		
Ahram	0.929	0.937	0.748	0.924	<b>0.949</b>	0.919
BBC Arabic	<b>0.998</b>	0.842	0.932	0.86	0.986	0.948
BBC Pashtoo	<b>0.98</b>	0.947	0.961	0.96	0.944	0.942
BBC Persian	0.994	0.893	0.951	0.906	<b>0.995</b>	0.955
BBC Urdu	0.976	0.966	0.912	0.966	<b>0.999</b>	0.881
Embassy	<b>0.962</b>	0.913	0.904	0.925	0.917	0.948
Hamshahri	0.981	0.716	0.908	0.886	<b>0.991</b>	0.912
Jamejam	0.981	0.854	0.896	0.917	<b>0.966</b>	0.874
Reuters	0.946	0.964	<b>0.966</b>	0.925	0.949	0.836
Wiki	0.782	0.808	<b>0.817</b>	0.383	0.699	0.713
Average	<b>0.952</b>	0.884	0.899	0.865	0.939	0.892

ArticleExtractor has the highest recall in all datasets. This shows it can detect and extract MC correctly from all web pages. MC, which is extracted by the above-mentioned algorithm, is much more

identical to the gold standard file. The F1, which is the harmonic mean of recall and precision, measures the algorithm's overall accuracy in terms of distinguishing MC and boilerplate more precisely.

ArticleExtractor achieves a F1-measure of 0.951, while Web-AM and DANAg earn values of 0.887 and 0.886 across the entire dataset. Despite the high accuracy of ArticleExtractor, it has a low performance of 2.644 (MB/s) and causes to be placed in 5th among other approaches.

Table 10 shows the efficiency of all algorithms in megabits per second. With a value of 21.120 (MB/s), DANAg is very efficient in comparison with the second one, CanolaExtractor, with a performance of about 5.201 (MB/s). Seeing that DANAg can quickly extract MC, it can be used in projects where the speed of MC extraction has the highest priority.

## 5 CONCLUSION

Given the growing number of websites with R2L

languages such as Urdu, Farsi, and Pashto, we have examined such websites since they were scarcely have been analyzed.

This paper has compared three of the best MC extraction algorithms in the latest decade in R2L languages. This comparison was performed using data extraction criteria, including precision, recall, and F1, which can be used to determine accuracy.

With values of 0.946, 0.949, and 0.95 for precision, recall, and F1, ArticleExtractor is more precise than the others. We also analyzed the algorithms' performance in identifying the fastest algorithms in processing the input data. At 21.120 MB/s, DANAg is notably superior to the rest.

Finally, features like efficiency and performance can be prioritized to select the intended algorithm according to application and objective.

In the future, we can develop a new algorithm or

Table 10: Average performance of algorithms (MB/s).

Datasets	Performance					
	Boilerpipe				DANAg	Web-AM
	AE	CE	DE	AE-Py		
Ahram	4.868	12.707	6.249	4.701	<b>93.649</b>	1.39
BBC Arabic	2.473	7.977	2.758	2.623	<b>25.267</b>	1.838
BBC Pashtoo	1.052	3.495	1.005	1.575	<b>12.086</b>	0.822
BBC Persian	4.369	10.876	5.894	2.487	<b>28.403</b>	2.665
BBC Urdu	0.715	3.153	0.859	1.505	<b>12.265</b>	0.72
Embassy	0.731	1.622	0.867	2.069	<b>3.006</b>	0.366
Hamshahri	4.6	7.442	6.284	2.033	<b>25.897</b>	2.23
Jamejam	5.177	8.305	7.837	3.539	<b>24.906</b>	1.966
Reuters	0.896	1.367	1.09	1.158	<b>11.357</b>	0.636
Wiki	2.181	3.878	2.956	3.122	<b>13.849</b>	1.124
Forum	0.535	0.494	0.799	3.269	<b>4.119</b>	0.238
GeneralSites	2.137	2.413	2.84	2.251	<b>10.397</b>	0.559
Litratue	0.566	0.86	0.289	2.859	<b>4.187</b>	0.225
NewsSites	3.989	7.867	6.952	4.072	<b>38.54</b>	1.13
Religious	2.839	2.634	3.365	3.522	<b>11.287</b>	0.714
Uncategorized	5.177	8.128	6.874	3.921	<b>18.702</b>	1.346
Average	2.644	5.201	3.557	2.794	<b>21.12</b>	1.123

Table 11: The average of all metrics from every algorithm, based on the entire dataset.

Algorithms		Metrics			
		Precision	Recall	F1	Performance (MB/s)
Boilerpipe	AE	0.946	<b>0.949</b>	<b>0.95</b>	2.644
	CE	0.813	0.923	0.849	5.201
	DE	0.883	0.898	0.88	3.557
	AE-Py	0.869	0.936	0.866	2.794
DANAg		0.905	0.884	0.886	<b>21.12</b>
Web-AM		<b>0.967</b>	0.832	0.887	1.123

framework that covers accuracy and performance to achieve the best outcome.

## REFERENCES

- Aslam, N., Tahir, B., Shafiq, H. M., & Mehmood, M. A. (2019, December). Web-AM: An efficient boilerplate removal algorithm for Web articles. *In 2019 International Conference on Frontiers of Information Technology (FIT)* (pp. 287-2875). IEEE.
- Chakrabarti, D., Kumar, R., & Punera, K. (2007, May). Page-level template detection via isotonic smoothing. *In Proceedings of the 16th international conference on World Wide Web* (pp. 61-70).
- Debnath, S., Mitra, P., & Giles, C. L. (2005). Identifying content blocks from web documents. *In Foundations of Intelligent Systems: 15th International Symposium, ISMIS 2005, Saratoga Springs, NY, USA, May 25-28, 2005. Proceedings 15* (pp. 285-293). Springer Berlin Heidelberg.
- Fernandes, D., de Moura, E. S., Ribeiro-Neto, B., da Silva, A. S., & Gonçalves, M. A. (2007, November). Computing block importance for searching on web sites. *In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (pp. 165-174).
- Finn, A., Kushmerick, N., & Smyth, B. (2001, June). Fact or Fiction: Content Classification for Digital Libraries. *In DELOS*.
- Gottron, T. (2007, September). Evaluating content extraction on HTML documents. *In Proceedings of the 2nd International Conference on Internet Technologies and Applications* (pp. 123-132).
- Gottron, T. (2008, September). Content code blurring: A new approach to content extraction. *In 2008 19th international workshop on database and expert systems applications* (pp. 29-33). IEEE.
- Gupta, S., Kaiser, G., Neistadt, D., & Grimm, P. (2003, May). DOM-based content extraction of HTML documents. *In Proceedings of the 12th international conference on World Wide Web* (pp. 207-214).
- Kohlschütter, C., Fankhauser, P., & Nejdl, W. (2010, February). Boilerplate detection using shallow text features. *In Proceedings of the third ACM international conference on Web search and data mining* (pp. 441-450).
- Mantratzis, C., Orgun, M., & Cassidy, S. (2005, September). Separating XHTML content from navigation clutter using DOM-structure block analysis. *In Proceedings of the sixteenth ACM conference on Hypertext and hypermedia* (pp. 145-147).
- Mohammadzadeh, H., Gottron, T., Schweiggert, F., & Nakhaeizadeh, G. (2011a, October). Extracting the main content of web documents based on a naive smoothing method. *In International Conference on Knowledge Discovery and Information Retrieval* (Vol. 2, pp. 462-467). SCITEPRESS.
- Mohammadzadeh, H., Gottron, T., Schweiggert, F., & Heyer, G. (2012, November). TitleFinder: extracting the headline of news web pages based on cosine similarity and overlap scoring similarity. *In Proceedings of the twelfth international workshop on Web information and data management* (pp. 65-72).
- Mohammadzadeh, H., Schweiggert, F., & Nakhaeizadeh, G. (2011b, July). Using utf-8 to extract main content of right to left language web pages. *In International Conference on Software and Data Technologies* (Vol. 2, pp. 243-249). SCITEPRESS.
- Vieira, K., Da Silva, A. S., Pinto, N., De Moura, E. S., Cavalcanti, J. M., & Freire, J. (2006, November). A fast and robust method for web page template detection and removal. *In Proceedings of the 15th ACM international conference on Information and knowledge management* (pp. 258-267).
- Weninger, T., & Hsu, W. H. (2008, September). Text extraction from the web via text-to-tag ratio. *In 2008 19th International Workshop on Database and Expert Systems Applications* (pp. 23-28). IEEE.