EXTRACTING THE MAIN CONTENT OF WEB DOCUMENTS BASED ON A NAIVE SMOOTHING METHOD

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, Information extraction, Web mining, HTML web pages.

Abstract: Extracting the main content of web documents, with high accuracy, is an important challenge for researchers working on the web. In this paper, we present a novel language-independent method for extracting the main content of web pages. Our method, called DANAg, in comparison with other main content extraction approaches has high performance in terms of effectiveness and efficiency. The extraction process of data DANAg is divided into four phases. In the first phase, we calculate the length of content and code of fixed segments in an HTML file. The second phase applies a naive smoothing method to highlight the segments forming the main content. After that, we use a simple algorithm to recognize the boundary of the main content in an HTML file. Finally, we feed the selected main content area to our parser in order to extract the main content of the targeted web page.

1 INTRODUCTION

Content extraction is the process of identifying the main content (MC) and/or removing the additional items, such as advertisements, navigation bars, design elements or legal disclaimers (Gottron, 2008). (Gibson et al., 2005) in 2005 estimated that around 40 to 50% of the data on the web are additional items. The identification of the MC is beneficial for web search engines. When crawling and indexing the web, knowing the main content part of each web page can be exploited for determining descriptive index terms for a document. In addition, main content extraction (MCE) is also applied in scenarios where a reduction of a document to its main content is of advantage, e.g. on devices that have limited storage or bandwidth capacity, such as mobile phones, screen readers, etc, (Rahman et al., 2001).

In the literature there are several approaches addressing MCE. (Gottron, 2008), (Moreno et al., 2009), (Mohammadzadeh et al., 2011a), and (Mohammadzadeh et al., 2011b) are recent examples. (Mohammadzadeh et al., 2011b) proposed a novel algorithm, called R2L, which is working only on special languages, for example Arabic, Farsi, Urdu, and Pashto. In (Mohammadzadeh et al., 2011a) we extended R2L and compared the new algorithm, called DANA, with eleven algorithms proposed by (Gupta et al., 2003), (Mantratzis et al., 2005), (Finn et al., 2001), (Pinto et al., 2002), (Gottron, 2008), (Moreno et al., 2009). The drawback of DANA and R2L is that they are language-dependent. In this paper, we extend DANA and R2L and introduce DANAg which is an accurate language-independent approach.

The rest of this paper is organized as follows. In Section 2, we review related work. In Section 3, where we introduce our algorithm, DANAg. Section 4 shows the data sets, the evaluation method, and the results and compares DANAg with the latest main content extraction algorithm. In Section 5, we discuss our conclusion and give some suggestions for future work.

2 RELATED WORK

In this section based on the characteristics of MCE , all main content extraction algorithms can be categorized into the following three groups:

Mohammadzadeh H., Gottron T., Schweiggert F. and Nakhaeizadeh G..
EXTRACTING THE MAIN CONTENT OF WEB DOCUMENTS BASED ON A NAIVE SMOOTHING METHOD.
DOI: 10.5220/0003665304620467
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2011), pages 462-467
ISBN: 978-989-8425-79-9

2.1 Dom Tree based Methods

Three algorithms proposed by (Gupta et al., 2003), (Mantratzis et al., 2005), (Cai et al., 2003), and (Debnath et al., 2005) use DOM tree structure to extract the MC area of web pages.

2.2 HTML based Methods

By using HTML tags, methods proposed by (Finn et al., 2001), (Pinto et al., 2002), (Gottron, 2008), (Moreno et al., 2009) , (Lin and Ho, 2002), and (Weninger et al., 2010) are able to extract the MC of web pages.

2.3 Methods based on Character Encoding

(Mohammadzadeh et al., 2011b) proposed a new and simple approach, called R2L, which is independent from the DOM tree and HTML tags and was able to extract 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 a simple rule: the distinction between characters which are used in English and characters used in languages written from right to left. Hence in the first step, R2L separates and stores the R2L characters from English ones, in two different arrays of strings, and then counts the number of these two groups of character for each line of HTML file. In a second step, R2L depicts a density diagram based on the length of English characters and R2L characters of each line to determine the area comprising the main content of an HTML file. It was shown that the area in the density diagram with high density of the R2L characters and low density of the English characters contains the main content of web page with high accuracy. After R2L finds such a main content area, all the R2L characters located in this area are detected as the main content of the HTML file.

The shortcoming of R2L is that it only works on special web pages. Moreover, it loses the Non-R2L characters, for example English characters, of the main content of each line of the HTML file because in the first step 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.

(Mohammadzadeh et al., 2011a) introduced new version of R2L, called DANA, with considerable effectiveness and compared this method to eleven established main content extraction methods. DANA – like the R2L approach – is language-dependent but it could overcome the second drawback of the R2L, losing the Non-R2L characters of the main content. DANA comprises of three phases. The first and the second phase are, to a certain extent similar to the first and second steps of R2L. In the third phase, DANA extracts the main content of the selected main content area, which is produced in a second phase, using an HTML parser used also in (Gottron, 2008) .

3 DANAg

In this section we explain DANAg, which is an extended and language-independent version of DANA. We present our approach by categorizing it into one preprocessing step and four phases as follow. Phases two, three and four are to a certain extent similar to the phases two and three in DANA as introduced by (Mohammadzadeh et al., 2011a). Nevertheless, we describe briefly these phases for the sake of completeness. In the preprocessing step, all JavaScript and CSS codes and comments are removed from the HTML file, as they definitely do not contribute to the main content and might negatively influence the downstream analysis.

3.1 Calculating the Length of Content and Code of Each Line

In the first phase of the algorithm, our aim is to count the number of characters comprising both the content and the code of segments of the HTML file. As segments we use lines of a normalized source code representation. First, we feed the HTML file to our parser to extract all tokens representing the content of the HTML file. Afterward, we search the extracted tokens, which are stored in an array of string, in the HTML file to find the line containing tokens. By this method, we can find the location of all tokens in the HTML file, so we are able to count and store into two one-dimensional arrays T1 and T2, respectively, the number of characters used in content or code of each line in an HTML file.

3.2 Smoothing

Our hypothesis is that in the main content area, the number of characters used in content is greater than the number of characters used in code, so we are investigating the HTML file to find an area with this characteristics. In doing this and to explain, more clearly, our algorithm and for better understanding, we draw a Cartesian diagram, Figure 1, in which for each element of array T1 a vertical line is drawn above

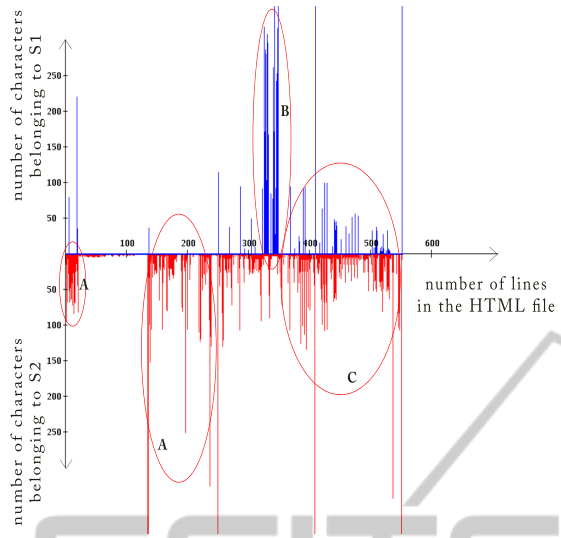

Figure 1: An example plot shows the number of characters uesd in content and code in each line of an HTML file.

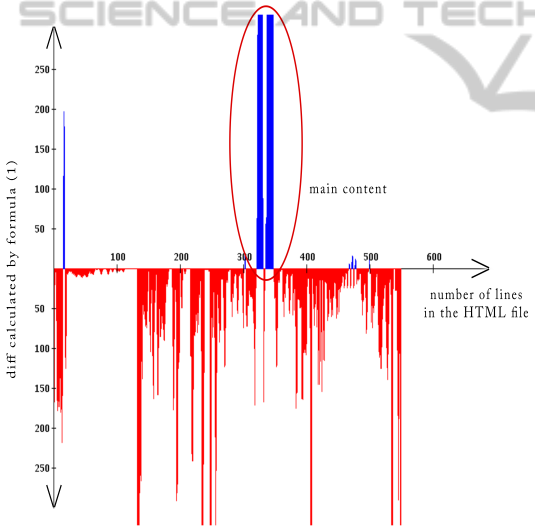

Figure 2: Smoothed diagram of Figure 2.

the x-axis and, similarly, for each element of array T2 a vertical line is depicted below the x-axis.

In Figure 1, the measurement unit for the x-axis is the number of lines in the HTML file. The measurement unit for the y-axis toward up and down is the number of characters in code and content, respectively, of each line of an HTML file.

Three kinds of regions can be observed in Figure 1. Regions like the one with label A consist mainly of HTML tags because, mostly, there is no line above the x-axis. The second type of regions, here labelled with C, belongs to navigation menus and panels because it can be seen both kinds of characters used in content

Table 1: Evaluation corpus of 9101 web pages.

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

and code. Finally, there is only one region, labelled B, in which the number of characters used in content is much more than the number of characters used in code.

To recognize region B easily, we depict Figure 2 using smoothing Formula 1. In Figure 2 for each column i, diff_i is computed and then a vertical line is drawn above or below the x-axis, respectively, if diff_i is greater than or less than zero. As you see, many columns above the x-axis which are parts of menu and additional items are removed.

$$\begin{aligned} \text{diff}_i &= T1_i - T2_i \\ &+ T1_{i+1} - T2_{i+1} \\ &+ T1_{i-1} - T2_{i-1} \end{aligned} \quad (1)$$

Now, we count the number of characters for each region above the x-axis and then among all regions we specify the position of just one region with the maximum number of characters as the first paragraph of the main content area.

3.3 Recognizing the Boundary of the MC Area

After recognizing the first paragraph of the main content area in the previous section, now we are going to extract the other paragraphs of the main content area. In doing this, the algorithm moves up and down in the HTML file to discover all paragraphs making the main content. We know in the HTML file and in the area of the main content, there are some lines between each two paragraphs, which are not concerning to the main content. In other words, there is a gap between each two paragraphs comprising the main content. We configure the gap parameter with a value of 20. So after finding the first paragraph, we are extracting all paragraphs around the first paragraph

Table 2: Evaluation results of Table 1 based on F1-measure.

	BBC	Economist	Zdf	Golem	Heise	Republica	Spiegel	Telepolis	Yahoo	Manual	Slashdot	Wikipedia
Plain	0.5950	0.6130	0.5140	0.5020	0.5750	0.7040	0.5490	0.9060	0.5820	0.3710	0.1060	**0.8230**
LQF	0.8260	0.7200	0.5780	0.8060	0.7870	0.8160	0.7750	0.9100	0.6700	0.3810	0.1270	0.7520
Crunch	0.7560	0.8150	0.7720	0.8370	0.8100	0.8870	0.7060	0.8590	0.7380	0.3820	0.1230	0.7250
DSC	**0.9370**	0.8810	0.8470	0.9580	0.8770	0.9250	0.9020	0.9020	0.7800	0.4030	0.2520	0.5940
TCCB	0.9140	0.9030	0.7450	0.9470	0.8210	0.9180	*0.9100*	*0.9130*	0.7580	0.4040	0.2690	0.6600
CCB	0.9230	**0.9140**	0.9290	0.9350	0.8410	0.9640	0.8580	0.9080	0.7420	0.4200	0.1600	0.4030
ACCB	0.9240	0.8900	**0.9290**	*0.9590*	*0.9160*	*0.9680*	0.8610	0.9080	0.7320	**0.4190**	0.1770	0.6820
Density	0.5754	0.8741	0.7081	0.8734	0.9060	0.3442	0.7609	0.8043	*0.8856*	0.3539	**0.3622**	0.7077
DANAg	0.9240	0.8996	0.9120	**0.9794**	**0.9450**	**0.9704**	**0.9488**	**0.9317**	**0.9518**	0.4012	0.2092	0.6455

with maximum 20 lines between each two consequent paragraphs comprising the main content.

3.4 Extracting the Main Content

In the last phase of our algorithm, we feed the main content area which is discovered in the previous phase to a parser (Gottron, 2008). The output of the parser is, following our hypothesis, the main content tokens of the HTML file.

4 EXPERIMENTS

4.1 Data Set

As evaluation corpus we use 9,101 web pages from different web sites (see Table 1). This dataset has been introduced in (Gottron, 2008).

4.2 Evaluation Methodology

By applying the classical information retrieval performance measures – Recall, Precision, and F1-measure, (Gottron, 2007), and using Hirschberg's algorithm (Hirschberg, 1977), which finds longest common subsequence between two strings, we calculate the accuracy of DANAg.

4.3 Results

The average F1 scores of DANAg and other MCE approaches show in Table 2 in three groups. In column one, diffrent algorithms we compare with DANAg are shown. Link quota filter (LQF) proposed by (Kaiser et al., 2005) and used in Crunch framework which is introduced by (Gupta et al., 2003). (Pinto et al., 2002) construct Document Slope Curve (DSC) which is one of the prominent solution for CE. Content Code Blurring (CCB), Adapted Content Code Blurring (ACCB) and Token based Content Code Blurring (TCCB) all

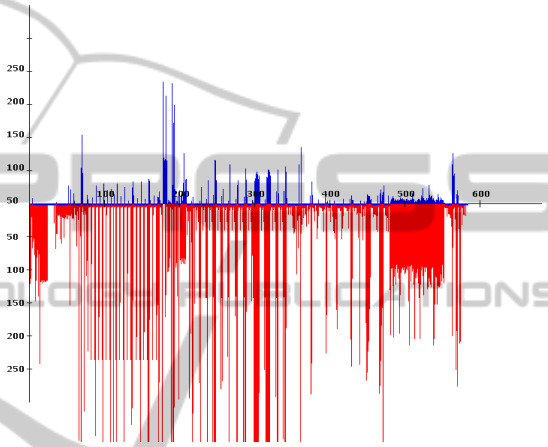

Figure 3: The original diagram of wikipedia.

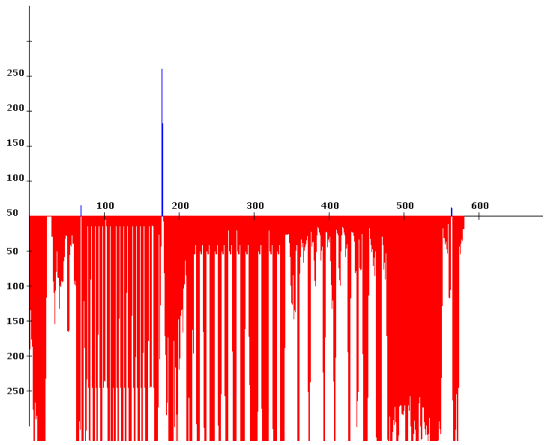

Figure 4: The smoothed diagram of wikipedia.

introduced by (Gottron, 2008). And the last algorithm, called Density, propose by (Moreno et al., 2009). In each column, we highlight the highest F1 score, bold one, and the highest F1 score among all algorithms except DANAg, italic value. The following results are considerable.

- In the middle part of the Table 2, DANAg achieves

```
</div>
      <p id="story_continues_2">&amp;quot;Especially the under-fives and the pregnant
women, they&amp;#039;re suffering from malnutrition and communicable disease like the
measles, diarrhoea and pneumonia,&amp;quot; he said.</p>

       <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>&amp;quot;The next few months are critical,&amp;quot; he said.</p>
       <p>Last week Somalia&amp;#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 &amp;quot;hidden agenda&amp;quot;. </p>
       <p>The drought is said to be the worst affecting by the Horn of
Africa&amp;#039;s in 60 years.</p>

       <p>The International Committee of the Red Cross (ICRC) is reporting a
dramatic rise in malnutrition rates even in the part of Somalia normally
considered to be the breadbasket of the country. </p>
       <p>Somalia, wracked by 20 years of conflict, is worst affected.</p>
       <p>Some 3,000 people flee each day for neighbouring countries such as
Ethiopia and Kenya which are struggling to cope.</p>
</div>
```

Figure 5: One Paragraph of BBC HTML file.

```
<span class="mw-headline" id="Present_significance">Present
significance</span></h2>


<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
requires IE to transform unstructured data into something that can be reasoned
with. A typical application of IE is to scan a set of documents written in a <a
href="/wiki/Natural_language" title="Natural language">natural language</a> and
populate a database with the information extracted.<sup id="cite_ref-4"
class="reference"><a href="#cite_note-4"><span>[</span>5<span>]</span></a></sup></
p>

<h2><span class="editsection">[<a href="/w/index.php?
title=Information_extraction&amp;action=edit&amp;section=3" title="Edit section:
IE tasks and subtasks">edit</a>]</span>
```

Figure 6: One Paragraph of Wikipedia HTML file.

F1 score higher than six web pages, golem, heise, republica, spiegel, telepolis, and yahoo. As can be seen, ACCB is the best algorithm, on three web pages, between all other algorithms after DANAg.

- The left side of Table 2 shows three web pages that DANAg achieves F1 score less than DSC, CCB, and ACCB approaches. But as it can be seen, the differences between F1 score of DANAg and last three mentioned methods are 0.013, 0.0144, and 0.017 and it shows DANAg could be acceptable on these web pages as well.
- In the right side of Table 2, we see three web pages, manual, slashdot, and wikipedia which DANAg and other algorithms could not achieve considerable F1 score. For better explanation about these web pages, we depict original and smoothed diagrams of an example of wikipedia web page, see Figure 3 and Figure 4. The recall, precision and F1 scores of this web page achieved by DANAg are 0.3636, 0.8889, and 0.5161, respectively. In Figure 3, we determinted the main content area that should be extracted, but Figure 4 shows that only small part of the web page was obtained.

The Wikipedia web pages are fully structure because many HTML tags used in each line of the Wikipedia HTML files in comparison with normal HTML files, for example BBC web pages. In Figure 5 and Figure 6, we demonstrate one small part of BBC and Wikipedia web pages. By looking in these two figures we observe that the ratio of HTML tags in the Wikipedia web pages are more than the BBC web pages, so if we calculate the length of content and code of each line in the Wikipedia HTML file then the length of code is greater than the length of content. Hence, formula one guess, by mistake, these lines as the extraneous items, not as the main contents. In contrast, for normal web pages, such as BBC , formula one

could recognize the main content , correctly, because normal web pages are not fully structure.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed DANAg: an extended and especially language-independent new version of DANA, with considerable effectiveness. Results show that DANAg determines the main content with high accuracy on many collected data sets. Achieving an average F1-measure > 0.90 on the test corpora used in this paper, it outperforms the state of the art methods in MCE.

In the future and for next research step, we will try to extend DANAg to use machine learning methods to group several areas in an HTML file contributing to the main content of web pages. This allows for discarding the parameter setting for gaps between main content blocks and to overcome the problem observed on certain documents in the evaluation corpora.

REFERENCES

Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. (2003). Extracting content structure for web pages based on visual representation. In *APWeb, volume 2642 of Lecture Notes in Computer Science*, pages 406–417. Springer.

Debnath, S., Mitra, P., and Giles, C. L. (2005). Identifying content blocks from web documents. In *Lecture Notes in Computer Science*, pages 285–293, NY, USA. Springer.

Finn, A., Kushmerick, N., and Smyth, B. (2001). Fact or fiction: Content classification for digital libraries. In *Proceedings of the Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries*, Dublin, Ireland.

Gibson, D., Punera, K., and Tomkins, A. (2005). The volume and evolution of web page templates. In *WWW '05: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web.*, pages 830–839, New York, NY, USA. ACM Press.

Gottron, T. (2007). Evaluating content extraction on html documents. In *Proceedings of the 2nd International Conference on Internet Technologies and Applications*, pages 123–132, University of Wales, UK.

Gottron, T. (2008). Content code blurring: A new approach to content extraction. In *19th International Workshop on Database and Expert Systems Applications*, pages 29–33, Turin, Italy.

Gupta, S., Kaiser, G., Neistadt, D., and Grimm, P. (2003). Dom-based content extraction of html documents. In *Proceedings of the 12th international conference on World Wide Web*, pages 207–214, New York, USA. ACM.

Hirschberg, D. S. (1977). Algorithms for the longest common subsequence problem. *J. ACM, 24(4)*, pages 664–675.

Kaiser, G., Gupta, S., and Stolfo, S. (2005). Extracting context to improve accuracy for html content extraction. In *Special Interest Tracks and Posters of the 14th International conference on World Wide Web*, pages 1114–1115.

Lin, S.-H. and Ho, J.-M. (2002). Discovering informative content blocks from web documents. In *KDD '02: Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 588–593, New York, NY, USA. ACM.

Mantratzis, C., Orgun, M., and Cassidy, S. (2005). Separating xhtml content from navigation clutter using dom-structure block analysis. In *Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia*, pages 145–147, New York, USA. ACM.

Mohammadzadeh, H., Gottron, T., Schweiggert, F., and Nakhaeizadeh, G. (2011a). A fast and accurate approach for main content extraction based on character encoding. In *TIR'11: Proccedings of the 8th Workshop on Text-based Information Retrieval.*, Toulouse, France.

Mohammadzadeh, H., Schweiggert, F., and Nakhaeizadeh, G. (2011b). Using utf-8 to extract main content of right to left languages web pages. In *ICSOFT 2011: Proceedings of the 6th International Conference on Software and Data Technologies.*, pages 243–249, Seville, Spain.

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

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

Rahman, A. F. R., Alam, H., and Hartono, R. (2001). Content extraction from html documents. In *WDA 2001: Proceedings of the First International Workshop on Web Document Analysis.*, pages 7–10.

Weninger, T., Hsu, W. H., and Han, J. (2010). CETR – content extraction via tag ratios. In *Proceeding of International World Wide Web Conference.*, Raleigh, USA.