ECON: An Approach to Extract Content from Web News Page

Yan Guo [1#], Huifeng Tang [3], Linhai Song [12], Yu Wang [12], Guodong Ding [1]
[1]*key laboratory of Network Science and Technology*
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
[2]*Graduate School of Chinese Academy of Sciences, Beijing, China*
[3]*PLA University of Foreign Languages, Luoyang, China*
[#]*guoy@ict.ac.cn*

***Abstract*—This paper provides a simple but effective approach, named ECON, to fully-automatically extract content from Web news page. ECON uses a DOM tree to represent the Web news page and leverages the substantial features of the DOM tree. ECON finds a snippet-node by which a part of the content of news is wrapped firstly, then backtracks from the snippet-node until a summary-node is found, and the entire content of news is wrapped by the summary-node. During the process of backtracking, ECON removes noise. Experimental results showed that ECON can achieve high accuracy and fully satisfy the requirements for scalable extraction. Moreover, ECON can be applied to Web news page written in many popular languages such as Chinese, English, French, German, Italian, Japanese, Portuguese, Russian, Spanish, Arabic. ECON can be implemented much easily.**

***Keywords*-information extraction; Web content extraction; Web mining;**

I. INTRODUCTION

In a Web news page, there is not only actual content of news, but also some noise such as advertisement, links to related news, copyright claimant. Figure 1 shows a sample of Web news page[1]. In the page, the actual content of news is in the anomalistic rectangle, and the noise occupies nearly half of the page. Note that the text in the ellipse is also regarded as noise, because what it expresses is the source of the news.

Efficiently extracting high-quality content from Web news page is crucial for many Web applications such as information retrieval, automatic text categorization, topic tracking, machine translation, abstract summary, helping end users to access the Web easily over constrained devices like PDAs and cellular phones. The extracted results will be the basic data for the further analysis. So content extraction from Web news page has attracted many researchers [1][2][3][4][5][6][7]recently.

Our task is to develop an approach to extract content from a large number of Web news pages, and these pages

This work is partially supported by the State Key Program of National Natural Science of China(Grant No. 60933005), and the National High Technology Research and Development Program of China (Grant No. 2007AA01Z441), and the National Information Security Research Plan of China)

[1]From http://english.peopledaily.com.cn/

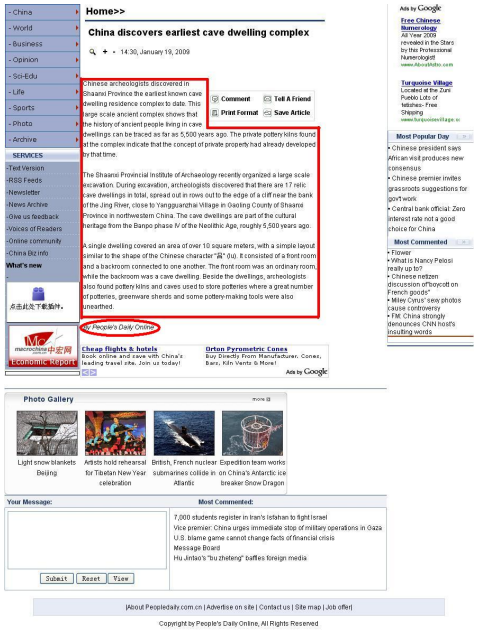

Figure 1. A Sample of Web news page

come from a great many of heterogeneous Web news sites. We mainly deal with news pages written in Chinese. Since our approach will be used in some real applications, the extracting requirements are good adaptivity, both high recall and precise, and also high speed.

To identify the actual content of news in the Web news page is a relatively easy task for a human who can do it just by visual inspection, however it is a hard problem for a computer. There have been many approaches existed to extract content from Web news page[8][1][2][3][4]. Based on the techniques they use, the approaches can be divided into three classes:

1) A wrapper can be generated by wrapper induction system for content extraction from Web news page such as [8]. However one wrapper is usually being generated for only one information source. Since there

978-0-7695-4012-2/10 $26.00 © 2010 IEEE
DOI 10.1109/APWeb.2010.11

are so many heterogeneous news sources, it is not practical to build wrappers for each news source. Therefore, this class of approaches is not fit for our task.

2) Some approaches use some techniques of Web mining, such as classification and clustering, to extract content from Web news page such as [1][2]. These approaches can improve the accuracy of extraction. However most of them need human interventions, and the complexity of the underlying algorithms is not low, so this class of approaches has limited ability for scalable extraction.
3) Some approaches extract content from Web page based on statistics such as [3][4]. These approaches can usually perform the extraction in an unsupervised fashion, which is crucial for our task. However most of them rely on some weights or thresholds that are usually determined by some empirical experiments. It is difficult to find one set of weights or thresholds to satisfy all news pages coming from so many heterogeneous news sources.

Since we can not find one approach to accomplish our task, we provide an approach named ECON(Extracting COntent from web News page) to extract entire content from Web news page fully-automatically. ECON uses a DOM tree to represent the Web news page and leverages substantial features of the tree. It is based on a key observation - actual content contains much more punctuation marks than noise in the same news page. Using this prior with some other observations, ECON finds a snippet-node by which a part of the content of news is wrapped firstly, then backtracks from the snippet-node until a summary-node is found, and the entire content of news is wrapped by the summary-node. During the process of backtracking, ECON removes noise. ECON can achieve high accuracy and fully satisfy our requirements. Since most of the features used in ECON are language-independent, it can deal with Web news page written in many popular languages such as Chinese, English, French, German, Italian, Japanese, Portuguese, Russian, Spanish, Arabic. The basic idea of ECON is so simple that the underlying algorithms of ECON can be implemented much easily.

The rest of this paper is organized as follows: The next section outlines related work; In section III, we will provide the details of ECON; In section IV, our experiments and results will be discussed; Section V concludes with some final remarks and directions for future work.

II. Related Work

The work related to us are mainly wrapper induction, content extraction using Web mining techniques, and content extraction based on statistics.

A. Wrapper induction

Reference [9] presented a good survey on the major Web data extraction approaches and compares them in three dimensions: the task domain, the automation degree, and the techniques used. Here are some classical works: Supervised approaches WIEN [10], STALKER and SoftMealy [11]; Semi-supervised approaches IEPAD [12] and OLERA [13]; Unsupervised approaches Dela [14], RoadRunner [15], and EXALG [16].

B. Using Web mining techniques

Web mining can be used in content extraction. Reference [1] presented an approach to extract real content from Web news pages using a particle swarm optimizer (PSO). In [2], the problem of identifying content from a Web page is treated as a sequence labeling problem. The content of a Web page is identified by using a Conditional Random Field sequence labeling model. In [5], traditional hierarchical clustering techniques are used to extract the desired news from Web news sites. Reference [6] provided an article extraction module using machine learning program Ripper.

C. Based on Statistics

Content extraction can also be performed based on statistics. Reference [7] proposed an approach to partition a Web page into several content blocks according to HTML tables, and to discover informative content blocks based on statistics on the occurrence of the features (terms) in the set of pages. In [4], for content extraction, the content extractor navigates the DOM tree recursively, using a series of different filtering techniques to remove and adjust specific nodes and leave only the content behind. The filters are based on statistics on some features of nodes such as link-to-text ratio. The work in [3] developed a heuristic technique CoreEx for extracting the main article from Web news pages. CoreEx constructs the DOM tree of the page and scores every node based on the amount of text, the number of links it contains and additional heuristics. The target of CoreEx is similar to ECON. However, the underlying algorithms are substantially different. And during the process of extraction, CoreEx uses some weights to score nodes, while ECON does not use any threshold or weight when performing extraction.

III. Our approach

A. The Basic idea

Our approach is based on DOM tree. A Web page can be passed through an HTML parser and described as a DOM tree. Figure 2 shows a DOM tree for a Web news page, (some details are omitted). In the DOM tree of a Web news page, we have some observations as follows:

1) There is such node that the entire content of news is wrapped in it with its subtrees, and any substree of it can not wrap the entire content of news. Such node is called as summary-node. That is, the summary-node

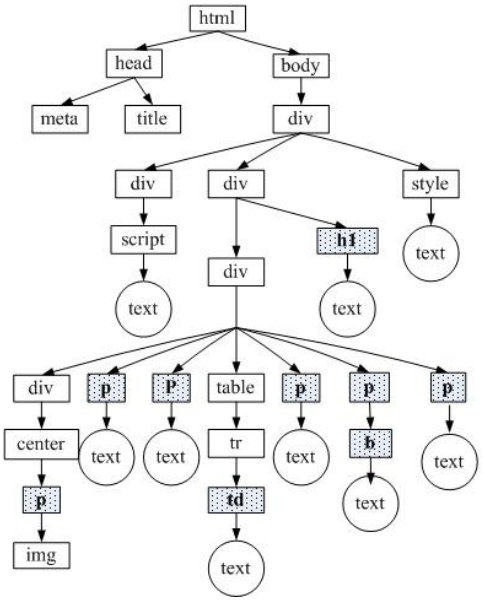

Figure 2. A DOM tree of a Web news page (a)

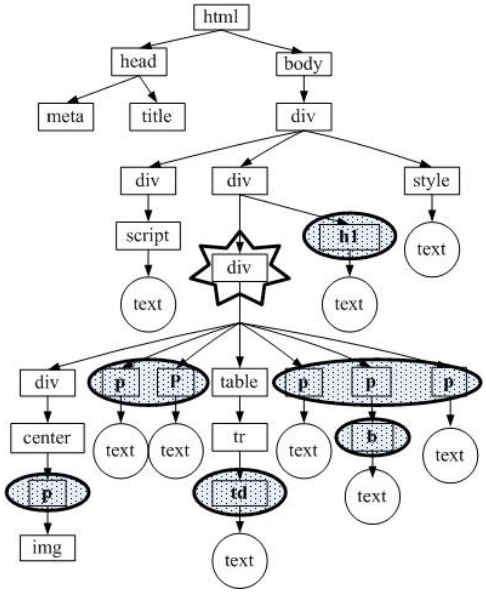

Figure 3. A DOM tree of a Web news page (b)

with its subtrees is the minimal tree which contains the entire content of news. Note that there may be some noise embedded within some subtrees of the summary-node. The summary-node of Figure 2 is in a star in Figure 3. The tag-name of the summary-node is .

2) There is such node that it is the descendant of the summary-node, and it is the father of a text node, and a part of or entire content of news is wrapped in it with its subtrees. Such node is called as snippet-node. There is at least one snippet-node in the DOM tree.

Based on the observations, if one snippet-node can be found, the summary-node can be found by backtracking from the snippet-node, and the actual content of news can be extracted after removing noise from the subtrees of the summary-node. For example, in Figure 3, suppose one snippet-node is found whose tag-name is <p>, and the trace of backtracking is <p>− . ECON is based on this idea.

The key issues of ECON are:

1) How to find one snippet-node to start backtracking?
2) When to stop backtracking to find the summary-node?
3) How to remove noise during the backtracking?

An obvious but very important heuristic is used to help to resolve the above key issues. Actual content contains much more punctuation marks than noise in the same news page. This heuristic can be used to differentiate between the content of news and noise. The heuristic may seem very simple, but the experiments showed that it is really useful.

B. Algorithms of ECON

Before describing the algorithms, some definitions are described firstly.

The node which satisfies either of the conditions is called as big-node:

1) The node is the father of a text-node and the tag-name of the node is not <script> or <style>;
2) The tag-name of the node is <p> or
 or <h1> or <h2> or <h3> or <h4> or <h5> or <h6> or or or
 or <b> or <i> or <tt> or <font>.

The content of news is usually broken into many small pieces by these nodes. In Figure 2, each big-node is shadowed.

One set of nodes that satisfies the conditions is called as text-node-set:

1) The nodes in the set are all big-nodes, and they are at the same level in the DOM tree and they are adjacent;
2) The nodes in the set together wrap a part of or entire content of news or noise, and the text wrapped by the set is called as a text-para.

Each text-node-set of Figure 2 is shadowed in an ellipse in Figure 3.

The number of period and comma in a piece of text is called as punc-num. The punc-num in the text wrapped by a node is called as node-punc-num.

Algorithm of Joint-para:

In a DOM tree of a Web news page, it can be observed that sometimes the entire text of news is broken into many short pieces by some nodes such as <p> and
. If there is a long piece of noise, it is easy to wrongly regard the piece of noise as the start point of backtracking. To guarantee finding a correct start point of backtracking, the algorithm of Joint-para will merge short pieces of text. At the same time, some noise may be embedded within some subtrees, so Joint-para needs to prude some noisy nodes during merging.

The input of Joint-para is a big-node. Joint-para checks its brother nodes to find a text-node-set. And then get the text-para of the text-node-set and compute the punc-num of the text-para. If it is 0, the text-para will be regarded as noise and will not be output. Meanwhile, all the nodes that together wrap the noise piece are pruned. If the punc-num is not 0, the text-para will be output.

An example is illustrated for Joint-para. Figure 4 shows a part of Figure 2. In Figure 4, for the node2 that is a big-node, Joint-para will get its text-para. Joint-para checks node1, since it is not a big-node, Joint-para ignores it and begins to check node3. Since the node3 is a big-node, Joint-para merges the text of node3 and node2. Then Joint-para checks node4. Since it is not a big-node, the algorithm stops checking. If the punc-num of the merged text is 0, Joint-para will prune node2 and node3, and return NULL. If the punc-num is not 0, the merged text will be returned.

Algorithm of Extract-news:

The heuristics to detect when to stop backtracking is from such observation: When backtracking from node1 to node2, if the content of news wrapped by node2 is more than node1, node1 must not be the summary-node, and the node-punc-num of node2 must be more than node1. If node1 is the summary-node, there are two cases as following:

1) The information wrapped by node2 is equal to node1, so the node-punc-num of node2 must be equal to node1;
2) There is more noise wrapped in node2 than node1, and the extra noise does not contain any period and comma, so the node-punc-num of node2 must be equal to node1.

This algorithm of Extract-news is to extract entire content from Web news page. The input of it is a Web news page. Firstly, Extract-news transverse the DOM tree and perform the algorithm of Joint-para for each big-node to get all text-paras. Then select randomly one node from the text-node-set that wraps the longest text-para. The selected node is regarded one snippet-node. Then backtracking starts from the snippet-node. When backtracking from node1 to node2, calculate the node-punc-num of node1 and that of node2 respectively. Then use the node-punc-num of node2 minus the node-punc-num of node1, and get the difference that is called as distance. Thus on the way of backtracking, a sequence of distance can be obtained. The process of backtracking stops at the following condition: The distance appears 0 for the first time. For the distance 0, the child-node is regarded as the summary-node. At last, the content wrapped by the summary-node is extracted as the entire content of news.

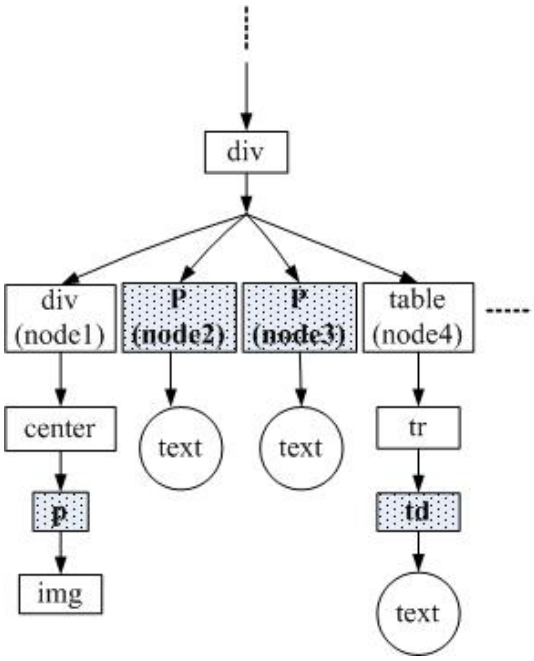

Figure 4. A part of the DOM tree in Figure 2

IV. EXPERIMENTS

All experiments were performed on a machine with 3.4GHz Pentium IV processor and 512MB memory. We made four experiments for evaluating ECON.

The work in [3] developed a heuristic technique CoreEx for extracting the main article from Web news pages. The target of CoreEx is similar to ECON. However, the underlying algorithms are substantially different. According to [3], CoreEx performs well, so we made a comparison between ECON and CoreEx.

For experiments, we collected Web news pages from 30 popular Web news sites written in Chinese. The sites are shown in Table I.

The result is by manual checking. For a Web page, if the extracted result contains only the entire content of the news, the result is regard as **Correctly** extracted. For a Web page, if the extracted result contains not only the entire content of the news, but also some noise, the result is regard as

Table I
WEB NEWS SITES

No.	The Web news site
1	www.sina.com.cn
2	www.sohu.com
3	www.163.com
4	www.qq.com
5	www.qianlong.com
6	www.xinhuanet.com
7	www.people.com.cn
8	chinese.wsj.com
9	www.ifeng.com
10	cn.reuters.com/news/china
11	www.china-cbn.com
12	www.ftchinese.com
13	www.21cbh.com/focus.asp
14	www.dfdaily.com
15	www.ben.com.cn
16	www.nbd.com.cn
17	news.eastmoney.com
18	news.stockstar.com
19	sc.stock.cnfol.com
20	finance.jrj.com.cn
21	news.hexun.com
22	www.wlstock.com
23	news.i918.cn
24	www.cs.com.cn
25	www.secutimes.com
26	www.caixun.com
27	www.zhihuangjin.com
28	www.fundschina.com
29	www.cnfund.cn
30	www.gutx.com

Table II
RESULTS FOR EXPERIMENT1

	Correctly	Wrongly	Missed
ECON	93.7%	5.6%	0.7%
CoreEx	82.6%	9.9%	7.5%

Wrongly extracted. For a Web page, if the extracted result contains only a part of the content of news or contains none of the content of news, the result is regard as **Missed**.

A. Experiment1

In this experiment, the testing set consists of 500 Web news pages from the 30 Web news sites. These pages are selected randomly. Table II shows the comparison between ECON and CoreEx.

ECON extracted correctly an average of 93.7% of the news pages, while CoreEx extracted correctly an average of 82.6%.

An average of 5.6% of the news pages were wrongly extracted by ECON, while 9.9% by CoreEx. Most of the Web pages having been wrongly extracted by ECON have such common features: Some noise contains period or comma, so ECON wrongly regarded them as the content of news. Most of the Web pages having been wrongly extracted by CoreEx have such common features: The noise extracted by CoreEx can not be distinguished obviously from the actual content by HTML code. For example, in Figure 1, the noise in the ellipse is often wrongly regarded as content. ECON can identify such noise successfully because most of the noise have such a common feature: They do not contain any period and comma. It can be seen that using the number of period and comma as a heuristic to identify such noise is very useful.

Table III
RESULTS FOR EXPERIMENT2

	Correctly	Wrongly	Missed
ECON	91%	6%	3%
CoreEx	51%	17%	32%

An average of 0.7% of the news pages were missed by ECON, while 7.5% by CoreEx. Most of the Web pages having been missed by ECON have such common features: The text of the content of news is so short that ECON can not find one snippet-node correctly, so the backtracking is completely wrong from the start. Such page is called as short-page. Figure 5 shows a sample of a part of a short-page. In Figure 5, the text of actual content is in a rectangle, and it is only one sentence. Most of the Web pages having been missed by CoreEx have such common features: they are short-pages, or the content of news is broken into many small pieces by some links or special fonts and so on. The latter page is called as pieces-page. Figure 6 shows a sample of a part of a pieces-page. In Figure 6, some breaking points are in the rectangles. Benefited from the algorithm of Joint-para, ECON dealt with the pieces-page well.

Note that in Figure 5, the information on the photo "(Xinhua/Reuters Photo)" should be excluded from the content. However the part is not separated from the previous sentence by an HTML tag, ECON incorrectly extracted it as content. As referred by the authors of CoreEx, CoreEx can not deal with this situation well either.

The average time cost of performing one news page by ECON is 15 ms, while the time cost of CoreEx is only 11 ms. However, the accuracy of extraction of ECON is higher than CoreEx, and 15 ms is quick enough for the future applications.

To find the summary-node, the average steps of backtracking for ECON are 2.7.

B. Experiment2

For this experiment, the testing set consists of 100 news pages from the 30 Web sites. These testing pages are selected deliberately, and they are all pieces-pages. Table III shows the comparison between ECON and CoreEx.

ECON extracted correctly an average of 91% of the news pages, while 6% were wrongly extracted and 3% were missed. CoreEx extracted correctly an average of 51% of the news pages, while 17% were wrongly extracted and 32% were missed. Therefore, ECON can deal with the pieces-page better than CoreEx. This result is identical to that of Experiment1 (refer to section IV-A).

Figure 5. A Sample of a part of a short-page

Table IV
RESULTS FOR EXPERIMENT3

	Correctly	Wrongly	Missed
ECON	41%	7%	52%
CoreEx	34%	15%	51%

Figure 6. A Sample of a part of a pieces-page

C. Experiment3

For this experiment, the testing set consists of 100 news pages from the 30 Web sites. These testing pages are selected deliberately, and they are all short-pages. Table IV shows the comparison between ECON and CoreEx.

ECON extracted correctly an average of 41% of the news pages, while 7% were wrongly extracted and 52% were missed. CoreEx extracted correctly an average of 34% of the news pages, while 15% were wrongly extracted and 51% were missed. It can be seen that neither ECON nor CoreEx can deal with the short-page well. The reason is that the heuristics used by ECON and CoreEx are based on some features of the text of the content of news. When the text of the content of news is very short, the heuristics may fail to work. It is fortunate that there are only a few short-pages in the real Web news pages.

D. Experiment4

This experiment is devised to check if ECON can be applicable to other languages. For this experiment, the testing set consists of 50 news pages written in 9 popular languages: English, French, German, Italian, Japanese, Portuguese, Russian, Spanish, and Arabic. These testing pages are selected randomly from some Web news sites.

Among all features used in ECON, only the count of punctuation is language-based. We discovered that the usage of punctuation in the 9 popular languages have the following characteristics:

1) The usage of period and comma in English, French, German, Italian, Portuguese, Russian and Spanish is the same as the usage in Chinese. The usage of period in Japanese and Arabic is the same as the usage in Chinese.
2) For the Web news page written in the 9 popular languages, the actual content contains much more punctuation marks than noise in the same news page.

The discovery implied that the heuristic of using the punctuation count to differentiate between the content of news and noise will still work in the news page written in the 9 popular languages.

ECON was used to perform extraction on the testing

Table V
RESULTS FOR EXPERIMENT4

	Correctly	Wrongly	Missed
ECON	90%	6%	4%
CoreEx	92%	2%	6%

set. Table V shows the comparison between ECON and CoreEx. ECON extracted correctly an average of 90% of the news pages, while 6% were wrongly extracted and 4% were missed. CoreEx extracted correctly an average of 92% of the news pages, while 2% were wrongly extracted and 6% were missed. It can be seen that although CoreEx does not use any language-based features while ECON uses the count of punctuation, ECON can deal with the testing pages nearly as well as CoreEx.

V. CONCLUSIONS

In this paper, we have proposed a very simple but powerful approach, named ECON, to fully-automatically extract content from Web news page. Experiments showed that ECON can perform extraction with high accuracy and run fast enough. ECON can fully satisfy our requirements for scalable extraction. ECON can be applied to Web news pages written in many popular languages such as English, French, German, Italian, Japanese, Portuguese, Russian, Spanish, and Arabic. ECON can be implemented much easily.

ECON can not deal with short-page well. So in the future work, we will resolve this problem. Furthermore, since the blog page and forum page have some similar characteristics with news page, we will improve ECON to extract the content from blog page and forum page.

REFERENCES

[1] C.-N. Ziegler and M. Skubacz, "Content extraction from news pages using particle swarm optimization on linguistic and structural features," in *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 242–249.

[2] J. Gibson, B. Wellner, and S. Lubar, "Adaptive web-page content identification," in *Proceedings of the 9th annual ACM international workshop on Web information and data management*. ACM New York, NY, USA, 2007, pp. 105–112.

[3] J. Prasad and A. Paepcke, "Coreex: content extraction from online news articles," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*. New York, NY, USA: ACM, 2008, pp. 1391–1392.

[4] S. Gupta, G. E. Kaiser, P. Grimm, M. F. Chiang, and J. Starren, "Automating content extraction of html documents," *World Wide Web*, vol. 8, no. 2, pp. 179–224, 2005.

[5] D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender, "Automatic web news extraction using tree edit distance," in *WWW '04: Proceedings of the 13th international conference on World Wide Web*. New York, NY, USA: ACM, 2004, pp. 502–511.

[6] K. McKeown, R. Barzilay, J. Chen, D. Elson, D. Evans, J. Klavans, A. Nenkova, B. Schiffman, and S. Sigelman, "Columbia's newsblaster: new features and future directions," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations-Volume 4*. Association for Computational Linguistics Morristown, NJ, USA, 2003, pp. 15–16.

[7] S.-H. Lin and J.-M. Ho, "Discovering informative content blocks from web documents," in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2002, pp. 588–593.

[8] I. Muslea, S. Minton, and C. Knoblock, "A hierarchical approach to wrapper induction," in *AGENTS '99: Proceedings of the third annual conference on Autonomous Agents*. New York, NY, USA: ACM, 1999, pp. 190–197.

[9] C.-H. Chang, M. Kayed, R. Girgis, and K. Shaalan, "A survey of web information extraction systems," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 10, pp. 1411–1428, Oct. 2006.

[10] N. Kushmerick, "Wrapper induction for information extraction," Ph.D. dissertation, 1997, chairperson-Daniel S. Weld.

[11] C.-N. Hsu and M.-T. Dung, "Generating finite-state transducers for semi-structured data extraction from the web," *Inf. Syst.*, vol. 23, no. 9, pp. 521–538, 1998.

[12] C.-H. Chang and S.-C. Lui, "Iepad: information extraction based on pattern discovery," in *WWW '01: Proceedings of the 10th international conference on World Wide Web*. New York, NY, USA: ACM, 2001, pp. 681–688.

[13] C.-H. Chang and S.-C. Kuo, "Olera: semisupervised web-data extraction with visual support," *Intelligent Systems, IEEE*, vol. 19, no. 6, pp. 56–64, Nov.-Dec. 2004.

[14] J. Wang and F. H. Lochovsky, "Data extraction and label assignment for web databases," in *WWW '03: Proceedings of the 12th international conference on World Wide Web*. New York, NY, USA: ACM, 2003, pp. 187–196.

[15] V. Crescenzi, G. Mecca, and P. Merialdo, "Roadrunner: Towards automatic data extraction from large web sites," in *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 109–118.

[16] A. Arasu, H. Garcia-Molina, and S. University, "Extracting structured data from web pages," in *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2003, pp. 337–348.