qRead: A Fast and Accurate Article Extraction Method from Web Pages using Partition Features Optimizations

Jingwen Wang and Jie Wang
Department of Computer Science, University of Massachusetts, Lowell, MA 01854, U.S.A.

Keywords: Article Extraction, Text Automation, Density, Similarity.

Abstract: We present a new method called qRead to achieve real-time content extractions from web pages with high accuracy. Early approaches to content extractions include empirical filtering rules, Document Object Model (DOM) trees, and machine learning models. These methods, while having met with certain success, may not meet the requirements of real-time extraction with high accuracy. For example, constructing a DOM-tree on a complex web page is time-consuming, and using machine learning models could make things unnecessarily more complicated. Different from previous approaches, qRead uses segment densities and similarities to identify main contents. In particular, qRead first filters obvious junk contents, eliminates HTML tags, and partitions the remaining text into natural segments. It then uses the highest ratio of words over the number of lines in a segment combined with similarity between the segment and the title to identify main contents. We show that, through extensive experiments, qRead achieves a 96.8% accuracy on Chinese web pages with an average extraction time of 13.20 milliseconds, and a 93.6% accuracy on English web pages with an average extraction time of 11.37 milliseconds, providing substantial improvements on accuracy over previous approaches and meeting the real-time extraction requirement.

1 INTRODUCTION

Web pages, starting with simple layouts in the mid 1990s, have evolved into the dynamic and complex layouts we are experiencing today. A typical contemporary layout of a web page imbeds the main contents such as news articles, stories, reviews, and reports in irrelevant contents such as commercials, navigation menus, guidelines, scripts, and user comments. Fig. 1 shows a real contemporary news page, where the main contents are bounded inside the thick box (added by us for a better visual), including the news title, publication time, source of the news, a video, and the main text; and the rest is junk, including navigation bars and commercials.

While human readers can easily distinguish main contents from irrelevant contents by looking at the page layout displayed on the web browser, it is difficult for a machine to do it well. Early approaches to content extraction include inaccurate empirical rules, more accurate but inefficient DOM-trees and machine learning models. Early extraction methods, while having met with certain success, may no longer meet the real-time requirements of extraction with high accuracy. In particular, we note that constructing a

Figure 1: A concrete example of a modern layout of a news page.

DOM tree on a complex web page is time-consuming, and using machine learning models could make things unnecessarily more complicated.

Wang, J. and Wang, J..
qRead: A Fast and Accurate Article Extraction Method from Web Pages using Partition Features Optimizations.
In *Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015) - Volume 1: KDIR*, pages 364-371
ISBN: 978-989-758-158-8

We present a new algorithm called qRead to achieve a higher efficiency and accuracy in extracting main contents without relying on HTML structures or DOM features. In particular, we first remove the HTML tags and the text wrapped inside certain tags that are obviously irrelevant to the main content. We then partition the remaining text into segments, use the highest ratio of word counts over the number of lines in each segment, and compare the similarity between each segment with title to determine the main content area. Note that we do not completely ignore HTML tags; we will use them when they help.

We carry out extensive experiments on web pages and show that qRead achieves a 96.8% accuracy on web pages written in Chinese and a 93.6% accuracy on web pages written in English. Moreover, qRead only incurs an average extraction time of 11.37 millisecond on Chinese web pages and 13.20 millisecond on English web pages, on a laptop computer with a 2.5 GHz Intel Core i5 processor and 8 GB 1600 MHz DDR3 memory, and thus meeting the real-time requirement. Thus, we may use qRead to extract main contents from a very large volume of web pages in real time.

2 RELATED WORK

Using HTML tags is a simple approach to content extraction (Bar-Yossef and Rajagopalan, 2002; Uzun et al., 2013; Crescenzi et al., 2001; Liu et al., 2000; Adelberg, 1998). Certain tags (such as the comment tags) and everything enclosed can be removed. A block of text without punctuation marks may sometimes be removed, except for lyrics or poems that may not be punctuated. Tags such as `<div>` may indicate that the enclosed text belongs to the main content (Joshi and Liu, 2009). However, we note that comments made by users may be enclosed in `<div>` and so using tags alone often results in poor accuracy.

Constructing DOM trees (Prasad and Paepcke, 2008; Gibson et al., 2005; Kao et al., 2002; Kao et al., 2004; Ziegler and Skubacz, 2007; Chakrabarti et al., 2007; Debnath et al., 2005) and using machine learning techniques (Baluja, 2006) are more accurate but less efficient. DOM-tree methods include Site Style Trees by sampling web pages (Yi et al., 2003), and subtree findings with large bodies of text using text size and Natural Language Processing techniques (Joshi and Liu, 2009). In particular, CoreEX (Prasad and Paepcke, 2008) is a DOM-tree system for content extraction on web pages in English, and ECON is a DOM-tree based system for extracting main contents on news web pages in Chinese (Guo et al., 2010). However, for web pages with multiple layers of embedded tag structures, constructing DOM-trees is time-consuming.

Machine learning methods include the Conditional Random Field sequence labeling model (Gibson et al., 2007) that divides an HTML file into units and determines which unit is content and which is junk; maximum subsequence segmentation, a global optimization method over token-level local classifiers (Pasternack and Roth, 2009); and a stochastic model based on a classification model combining word length and link density (Kohlschütter et al., 2010). The machine-learning approach, however, is also time-consuming, and could make extraction unnecessarily more complicated.

It was noted that using the text-to-tag ratio without DOM trees can help detect main contents (Weninger and Hsu, 2008).

We note that none of the early publications provided time analysis of content extraction.

3 qRead: OUR APPROACH

To achieve real-time extractions with high accuracy, we first remove everything wrapped inside the comment tags, style tags, script tags, and head tags. Since navigation lists are often wrapped in the `<div>` and `</div>` tags with attributes `class=''menu''` or `class=''nav''`, we can filter out most of the navigation lists using these features.

We then eliminate all the HTML tags while maintaining its natural segmentations (e.g., separations between titles, tables, and paragraphs). If a tag takes up a whole line, then removing it will leave an empty line in the text. We note that the web page title or the article title are typically contained in the heading tags such as `<h1>` and `<h2>`. The `<p>` and `<span>` tags usually contain plain text. A paragraph is typically wrapped in the `<p>` and `</p>` tags, or the `<br>` tags. We use these features of tags to maintain the original paragraph structure. We will keep track of such information and mark the `</p>` tags and the `<br>` tags to indicate paragraph break points before removing them. This information will be used when displaying the main content.

3.1 Text Segmentations

The remaining text often contains navigational text. There are two types of navigational text. One type is menu list with keywords in each line. The other type of navigational text is a list of article titles with a phrase or sentence in each line.

We observe that a text line in the area of junk contents is typically much shorter than a text line in the main contents we are interested in, where a text line ends with a carriage return in the source code, an end-of-sentence mark we placed in the filtering phase, or some other symbols. Define line length to be the number of words contained in the line. Fig. 2(a), Fig. 2(b), and Fig. 2(c) depict the line length distributions of three different news web pages obtained from, respectively, the ABC News website, the BBC News website, and the FOX News website, where the main content (the news) in the ABC web page appears in lines 492–494, in the BBC web page appears in lines 178-188, and in the FOX News web page appears in lines 296–317.

Thus, the area of the main content would typically include more words in each line than those in the junk-content area. We therefore consider, in a given text area, the ratio of word counts over the number of lines in the area.

Let L be a line. Denote by $w(L)$ the number of words contained in L. We partition the remaining content into a sequence of text segments, separated by at least λ empty lines, where λ is a threshold value. The value of λ is obtained empirically. We note that paragraphs in the main content are typically separated by HTML tags such as <p> and
 (where no empty lines are necessary), or by a single empty line or double empty lines. In our experiments, we set $\lambda = 3$. In other words, inside each segment S, the first line and the last line are non-tmpty text lines, and there are at most $\lambda - 1$ consecutive empty lines in between. Let

$$S_1, S_2, \cdots, S_k$$

denote the sequence of segments. Let L denote a text line. Then the size of S, denoted by $w(S)$, is the number of words contained in S. That is,

$$w(S) = \sum_{L \in S} w(L) \tag{1}$$

Let $l(S)$ denote the number of non-empty text lines contained in S. Let $d(S)$ denote the word desnity of S, which is the ratio of the number of words contained in S over the number of non-empty text lines contained in S. Namely,

$$d(S) = \frac{w(S)}{l(S)} \tag{2}$$

We observe that the main content is typically the text contained in the segments with the highest density, and we hypothesize that any reasonably well-organized web page with a reasonably long text in the main-content area has this property. Our extensive experiments have confirmed this hypothesis. Thus, our task is to identify the segment with the highest density to identify the area of the main content.

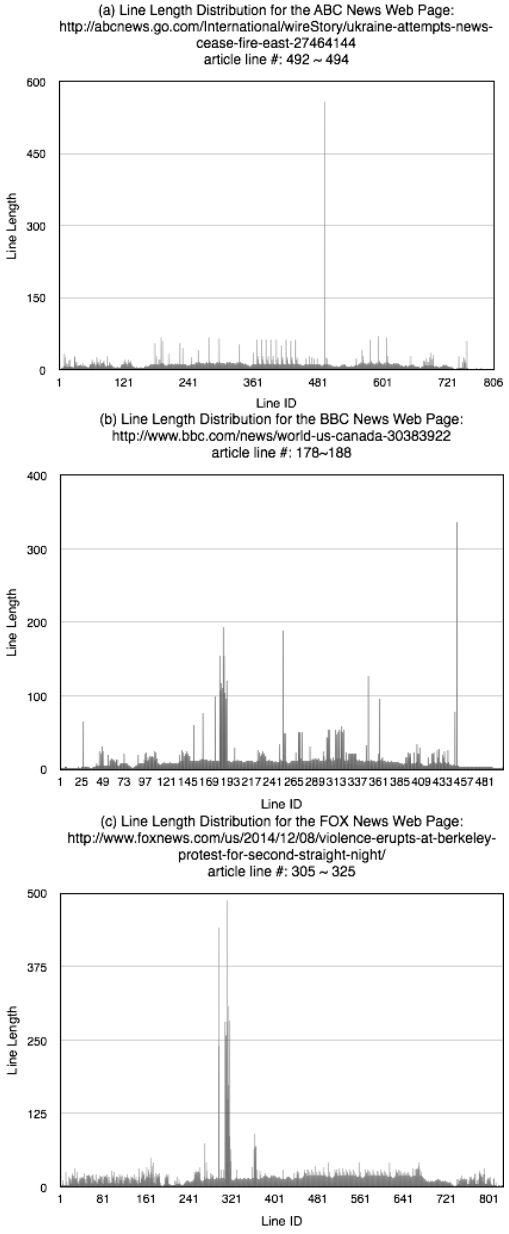
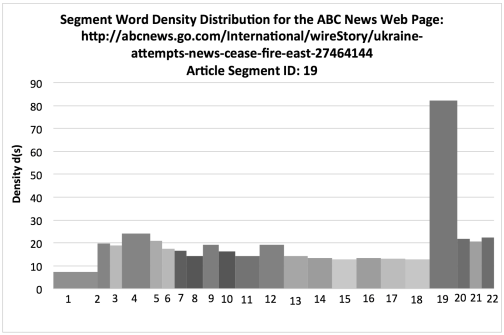
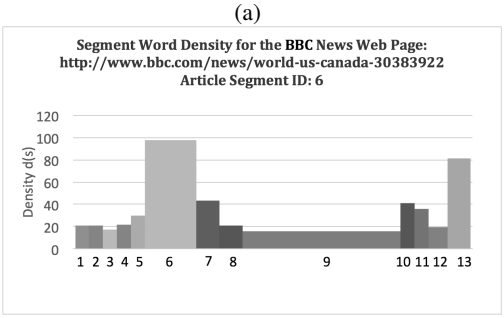


Figure 2: Line length distribution in news web pages.

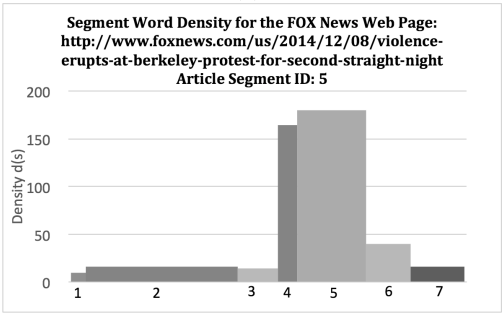
Fig. 3(a), Fig. 3(b), and Fig. 3(c) show the segment density distribution for each of the web pages used in Fig. 2(a), Fig. 2(b), and 2(c). In each figure, the segment with the highest word density is exactly the area of the main content (i.e., the news article contained in the web page).



(a)



(b)



(c)

Figure 3: Segment Word Density distribution in News web pages.

3.2 Similarity Process

It may be difficult to extract the main contents from the following two types of web pages using segmentation densities alone: One type is that the main content is a short article with one or two sentences. The other type is that the main content consists of multiple text segments with the same density. Text area of the highest density is not the article area in either of these two web page types. To improve extracting accuracy, we apply the measure of cosine similarity between text segment and article title to determine if the text segments should be appended to the extraction result.

We first calculate the average similarity between the entire text and the title. We then compare it with the similarity between each segment and the article title. Finally, we append all the segments with both high similarity and density.

4 EXPERIMENTS

We carry out extensive experiments on a commodity laptop computer with a 2.5 GHz Intel Core i5 processor and 8 GB 1600 MHz DDR3 memory. The implementation is written in Java using java platform–Eclipse Kepler, executed with a single thread.

4.1 Datasets

We evaluate qRead on news web pages written in English and news web pages written in Chinese.

1. *English Dataset*: We followed Weninger and Hsu's approach (Weninger and Hsu, 2008) and harvested 13,327 web pages written in English from more than 119 web sites by searching on the keyword "the" from Yahoo's search engine.
2. *Chinese Dataset*: We followed Guo et al.'s approach (Guo et al., 2010) and collected data from the following 35 (Table.1) popular news web sites written in Chinese. We harvested a total of 10,382 web pages.

4.2 Sampling and Extraction Types

We randomly choose 310 web pages from the English dataset and the Chinese data set, respectively, to evaluate the accuracy of qRead and other measures. We repeat this experiment for 20 times and compute the average.

For each sampling of 310 web pages from the English dataset or the Chinese dataset, we remove the ones with the maximum and minimum extraction time, and then compare the accuracy and efficiency of extractions of the main-content for the remaining web pages. Recall that this experiment is repeated 20 times.

We have the following three different types of extraction: accurate extraction, extra extraction, and missed extraction.

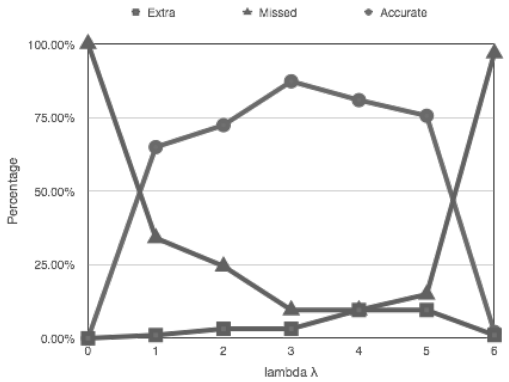
1. An extraction is *accurate* if it contains exactly the complete article without any junk content.
2. An extraction is *extra* if it contains the whole article but with some junk content.
3. An extraction is *missed* if it contains incomplete article or nothing related to the article.

Table 1: Thirty Five Chinese News Web Sites.

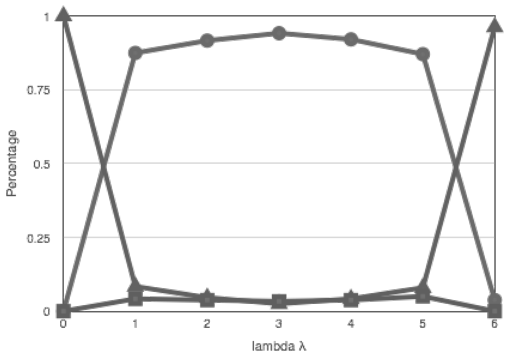
No.	Web site URL
1	www.sina.com.cn
2	www.sohu.com
3	www.163.com
4	www.qq.com
5	www.huanqiu.com
6	www.cankaoxiaoxi.com
7	www.china.com
8	www.zaobao.com
9	www.infzm.com
10	www.qianlong.com
11	www.xinhuanet.com
12	www.people.com.cn
13	chinese.wsj.com
14	www.ifeng.com
15	cn.reuters.com/news/china
16	www.china-cbn.com
17	www.ftchinese.com
18	www.21cbh.com
19	www.dfdaily.com
20	www.ben.com.cn
21	www.nbd.com.cn
22	news.eastmoney.com
23	news.stockstar.com
24	sc.stock.cnfol.com
25	finance.jrj.com.cn
26	news.hexun.com
27	www.cs.com.cn
28	www.secutimes.com
29	www.caixun.com
30	www.zhihuangjin.com
31	www.cnfund.cn
32	news.tom.com
33	news.cntv.cn
34	news.takungpao.com
35	www.chinanews.com

4.3 Extraction Threshold

We note that paragraphs in the main content are typically separated by HTML tags such as <p> and
 (with or without empty lines), or by a single empty line or double empty lines. The threshold value λ is used to partition the text into a sequence of text segments. In our experiments (see Fig. 4), we can see that when $\lambda = 3$, the extracted articles are the most accurate. In particular, from Fig. 4(a) for the English dataset we can see that when $\lambda = 3$, we achieve the highest accurate extraction and the lowest missed and extra extraction percentages. From Fig. 4(b) for the Chinese dataset we can see that, while the extraction accuracy are all very good for $\lambda = 2,3,4,5$, choosing $\lambda = 3$ offers the highest accurate extraction and the lowest missed and extra extraction percentages.



(a) English Dataset



(b) Chinese Dataset

Figure 4: The values of λ and extraction accuracy percentage.

4.4 Extraction Accuracy

In each of the web pages in our datasets, the article contained in it is substantially shorter than the web page itself. This means that every web page in our dataset contains lots of junk contents. We measure the size of a web page using the number of characters, and we call it the length of the web page. We measure the size of an article in the same way.

We manually checked all extractions for accuracy produced by qRead, ECON, and CoreEX.

The comparison results are shown in Table 2. We can see that qRead achieves an extraction accuracy of 96.8% accuracy on web pages written in Chinese with an average extraction time of 13.20 milliseconds on a laptop computer, and a 93.6% accuracy on web pages written in English with an average extraction time of 11.37 milliseconds.

The percentage of extra extractions using qRead is 5.8% on English web pages and 3.2% on Chinese web pages. In most of these cases, junk content is extracted because it is located very closely to the main

Table 2: Accuracy comparisons.

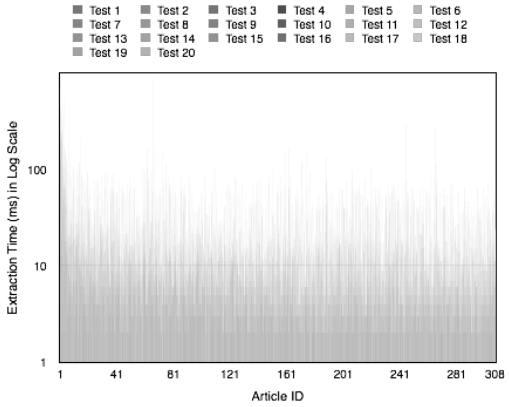
	Accurate	Extra	Missed
qRead (English)	93.6%	5.8%	1.6%
CoreEx (English)	82.6%	9.9%	7.5%
qRead (Chinese)	96.8%	3.2%	1.0%
ECON (Chinese)	93.7%	5.6%	1.0%

content area, making it difficult for qRead to separate the junk content from the article area.

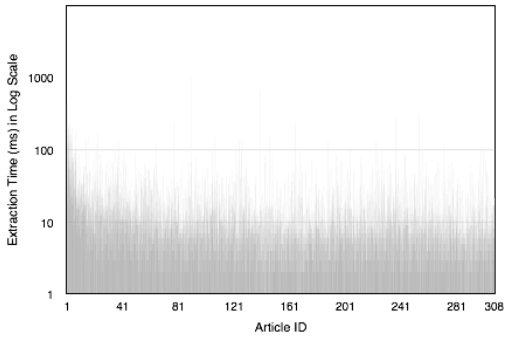
The percentage of missed extractions using qRead is 0.8% on web pages written in English and 1.0% on news web pages written in Chinese. The reason of missed extractions is that the source codes of these web pages interleave the main content, keyword links, and advertising blocks, partitioning the main content area into two or more segments of different word densities. In particular, we note that qRead would not work well on very short articles.

4.5 Time Efficiency

Early publications did not provide time efficiency results on extractions. We depict the running time of qRead in the log scale (see Fig. 5(a) and Fig. 5(b)), one on the English dataset and one on the Chinese

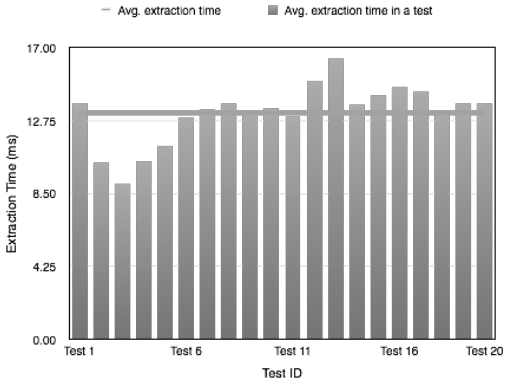


(a) English Dataset

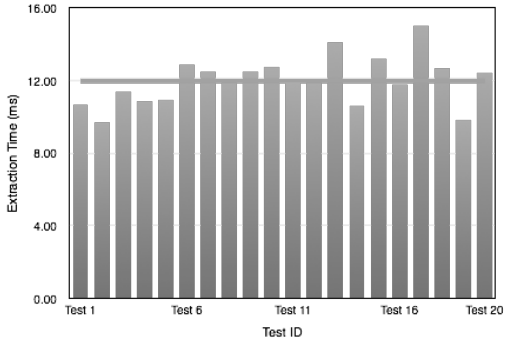


(b) Chinese Dataset

Figure 5: Article extraction time (ms) in Log Scale.



(a) English Dataset



(b) Chinese Dataset

Figure 6: Article extraction average time (ms).

dataset. It appears that the extraction time in Fig. 5(a) are longer than that in Fig. 5(b). This is due to the fact that the structures of English web pages tend to be more complicated than those of the Chinese web pages, and the article lengths in the English dataset are generally larger than the article lengths in the Chinese dataset.

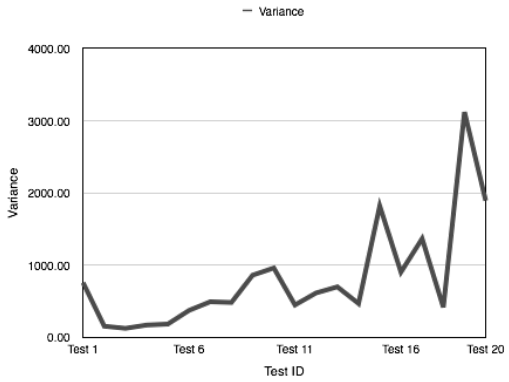
Fig. 6 illustrates the average running time for article extractions using qRead both on the English dataset and on the Chinese dataset. The average running time of each tests are almost the same, all meeting the real-time requirement.

1. It took 87 seconds to process all 20 tests on English web pages with an average extraction time of 13.20 milliseconds (see Fig. 6(a)). The shortest extraction time is 1 millisecond (on Test 17 web page ID 308) and the longest extraction time is less than 75 milliseconds (on Test 19 web page ID 66). Most web pages incurred less than 22 milliseconds of extraction time. The shortest average extraction time is 9.08 milliseconds (on Test 3) and the longest extraction time is 16.39 milliseconds (on Test 12).
2. It took 79 seconds to process all 20 tests of Chinese web pages with an average extraction time of

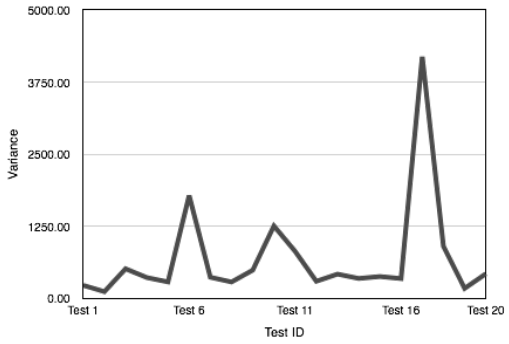
11.98 milliseconds (see Fig. 6(b)). The shortest extraction time is 1 millisecond (on Test 13 web page ID 288) and the longest extraction time is less than 684 milliseconds (on Test 6 web page ID 140). Most web pages incurred less than 20 milliseconds. The shortest average extraction time is 9.67 milliseconds (on Test 2) and the longest extraction time is 15.05 milliseconds (on Test 17).

4.6 Standard Deviation and Variance of Running Time

The standard deviation and variance of extraction running time on each of the 20 tests for each dataset are shown in Fig. 7 and Fig. 8, respectively. We can see that the extraction time is quite stable for different tests. We can see that there is just one outlier in the 20 tests for each dataset: Test 19 for the English dataset and Test 17 for the Chinese dataset. This situation is normal, for the running time of content extractions depends on the size of the web page source code and the HTML complexity, and once in a while (5% to be exact in our experiments) a long and complicated web page would show up.

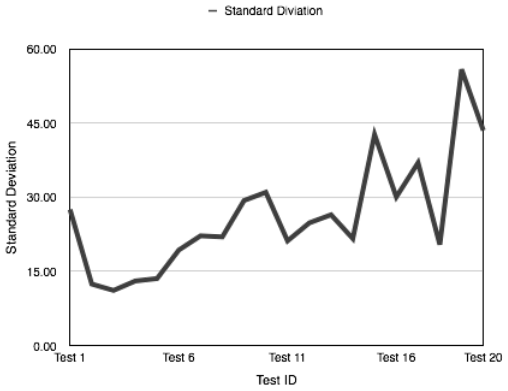


(a) English Dataset

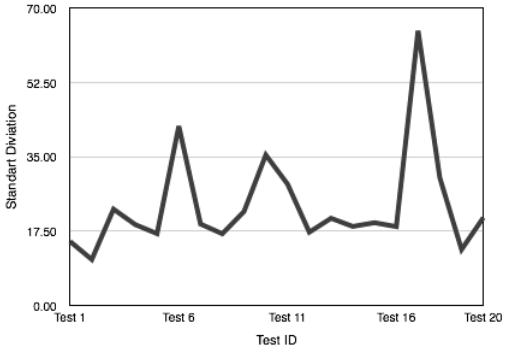


(b) Chinese Dataset

Figure 7: Article extraction tests extraction time variance.



(a) English Dataset



(b) Chinese Dataset

Figure 8: Article extraction tests extraction time standard deviation.

5 CONCLUSIONS

We presented a simple and effective algorithm called qRead for extracting the main contents from web pages with high accuracy, which can be used for real-time extractions. Our main technical contributions include novel usage of segment densities and similarities to identify the main content area, where qRead first filters obvious junk contents, eliminates HTML tags, and partitions the remaining text into natural segments. It then uses the highest ratio of words over the number of lines in a segment combined with similarity between segment and title to identify the main content area. The qRead demonstration page is http://www.wssummary.net/extraction. Our datasets can be found at http://www.wssummary.net/ extraction/ dataset. In future research we will work out new ways to detect very short articles, such as online social web feeds (Jia and Wang, 2014), to improve extraction accuracy to near 100%.

ACKNOWLEDGEMENTS

This work was supported in part by the NSF under grant CNS-1331632.

REFERENCES

Adelberg, B. (1998). Nodose—a tool for semi-automatically extracting structured and semistructured data from text documents. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, SIGMOD '98, pages 283–294, New York, NY, USA. ACM.

Baluja, S. (2006). Browsing on small screens: Recasting web-page segmentation into an efficient machine learning framework. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 33–42, New York, NY, USA. ACM.

Bar-Yossef, Z. and Rajagopalan, S. (2002). Template detection via data mining and its applications. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, pages 580–591, New York, NY, USA. ACM.

Chakrabarti, D., Kumar, R., and Punera, K. (2007). Page-level template detection via isotonic smoothing. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 61–70, New York, NY, USA. ACM.

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

Debnath, S., Mitra, P., Pal, N., and Giles, C. L. (2005). Automatic identification of informative sections of web pages. *IEEE Trans. on Knowl. and Data Eng.*, 17(9):1233–1246.

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

Gibson, J., Wellner, B., and Lubar, S. (2007). Adaptive web-page content identification. In *Proceedings of the 9th Annual ACM International Workshop on Web Information and Data Management*, WIDM '07, pages 105–112, New York, NY, USA. ACM.

Guo, Y., Tang, H., Song, L., Wang, Y., and Ding, G. (2010). Econ: An approach to extract content from web news page. In *Web Conference (APWEB), 2010 12th International Asia-Pacific*, pages 314–320.

Jia, M. and Wang, J. (2014). Handling big data of online social networks on a small machine. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8591 LNCS:676–685. cited By 0.

Joshi, P. M. and Liu, S. (2009). Web document text and images extraction using dom analysis and natural language processing. In *Proceedings of the 9th ACM Symposium on Document Engineering*, DocEng '09, pages 218–221, New York, NY, USA. ACM.

Kao, H.-Y., Chen, M.-S., Lin, S.-H., and Ho, J.-M. (2002). Entropy-based link analysis for mining web informative structures. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, pages 574–581, New York, NY, USA. ACM.

Kao, H.-Y., Lin, S.-H., Ho, J.-M., and Chen, M.-S. (2004). Mining web informative structures and contents based on entropy analysis. *IEEE Trans. on Knowl. and Data Eng.*, 16(1):41–55.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 441–450, New York, NY, USA. ACM.

Liu, L., Pu, C., and Han, W. (2000). Xwrap: an xml-enabled wrapper construction system for web information sources. In *Data Engineering, 2000. Proceedings. 16th International Conference on*, pages 611–621.

Pasternack, J. and Roth, D. (2009). Extracting article text from the web with maximum subsequence segmentation. In *WWW*.

Prasad, J. and Paepcke, A. (2008). Coreex: Content extraction from online news articles. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1391–1392, New York, NY, USA. ACM.

Uzun, E., Agun, H. V., and Yerlikaya, T. (2013). A hybrid approach for extracting informative content from web pages. *Inf. Process. Manage.*, 49(4):928–944.

Weninger, T. and Hsu, W. H. (2008). Text extraction from the web via text-to-tag ratio. In *Proceedings of the 2008 19th International Conference on Database and Expert Systems Application*, DEXA '08, pages 23–28, Washington, DC, USA. IEEE Computer Society.

Yi, L., Liu, B., and Li, X. (2003). Eliminating noisy information in web pages for data mining. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 296–305, New York, NY, USA. ACM.

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