Effective and Efficient Web Reviews Extraction Based on Hadoop

Jian Wan, Jiawei Yan, Congfeng Jiang, Li Zhou, Zujie Ren, and Yongjian Ren

Hangzhou Dianzi University, School of Computer Science and Technology,
Hangzhou 310018, China
yanjw1988@gmail.com

Abstract. The rapid development of Web 2.0 brings the flourish of web reviews. Traditional web review data extraction methods suffer from poor performance in dealing with massive data. To solve this problem, we propose an effective and efficient approach to extract web reviews based on Hadoop. It overcomes inefficiency when dealing with large-scale data, and enables the accuracy and efficiency in extracting the massive data sets. Our proposed approach consists of two components: a review record extraction algorithm based on node similarity, and a review content extraction algorithm based on the text depth. We design a Hadoop-based web reviews automatic extraction system. At last, we test the extraction system using the massive web reviews page sets. The experimental results show that this extraction system can achieve accuracy of more than 96%, and also can obtain a higher speedup, compared with the traditional web extraction.

Keywords: web reviews, information extraction, massive data, cloud computing, Hadoop.

1 Introduction

Nowadays, hundreds of millions of users can publish various data freely on the Internet, which makes web information data grows rapidly. How we can get valuable information quickly from such a huge-volume data has become a challenging issue.

In the e-Commerce field, a large number of product reviews are emerged, such as commodity reviews, news comments and so on. Mining reviews generated by web users has become an important solution to improve the search quality of goods, which has attracted much attention in the industrial and academic field.

The premise of mining review data is to extract review data. Compared with the general information extraction[1][2][3], review data extraction is much more complex. Review contents of web pages are much diverse and flexible. Web users generally edit the review content with various data types, such as pictures, tables, videos etc. These facts decrease the accuracy and efficiency of review data extraction. With the data volume grows rapidly, traditional information extraction is usually hard to satisfy the performance requirement.

A. Ghose et al. (Eds.): ICSOC 2012, LNCS 7759, pp. 107–118, 2013.

In this paper, we design an effective method for extracting massive review data through Hadoop[4] framework. Nowadays, existing applications of Hadoop mainly focus on the data statistics tasks, and tasks logic is simple and the task is easy to cut. This uses Hadoop framework to implement parallel processing of the web review data extraction. However, due to the complexity of the procedure, design an effective review extraction algorithm and divide review extraction tasks into multiple nodes of Hadoop efficiently become the key problems.

In order to solve these problems, we propose a Hadoop-based web automatic review extraction approach. This paper's mainly contributions as follows:

- We propose a review record extraction algorithm based on the node similarity. This algorithm has effectively improved the accuracy of extraction, by using visual information of a web page and node similarity technology.
- We propose a review content extraction algorithm based on the text depth. According to the different text length of all review record sub-tree nodes, this algorithm uses a method based on text depth to confirm the location of the review content area and generates the corresponding wrapper of review extraction with the same template web pages.
- We design a Hadoop-based web review automatic extraction system. The system can effectively finish extracting review content accurately from web review pages.

The rest of the paper is organized as follows: Section 2 describes the review record and review content extraction algorithm and implementation; Section 3 introduces the Hadoop-based web review automatic extraction system; Section 4 shows the experimental evaluation of this extraction system; Section 5 presents the related work; Section 6 concludes this paper.

2 Web Review Automatic Extraction Approach and Implementation

The Hadoop-based web review automatic extraction approach consists of four modules: 1) Web page parsing; 2) Review record extraction; 3) Review content extraction; 4) Review content storage. Fig 1 shows the overall workflow of the approach.

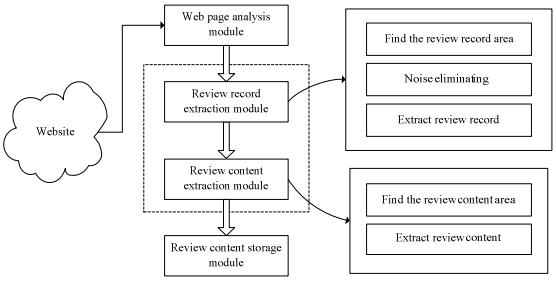


Fig. 1. The procedure of web review automatic extraction approach

2.1 Web Page Parsing

Parsing web page and building the DOM[5] tree is a necessary step in today a number of web information extraction algorithms. In this paper, we mainly study the pure HTML web pages. The HTML file can be transformed into a DOM tree easily, because of HTML tags come in pairs. Each tag pair consists of a start tag (<>) and a closing tag (</>). Each node of the DOM tree is mapped by a pair of tags in the web page file, and the child nodes of this node are mapped by the nested tags.

2.2 Review Record Extraction Algorithm and Implementation

In web review pages using the same template to render, review records are always arranged in the specific location. Based on the observation of a large number of real web review pages, we extract review records according to the following rules:

- *Rule 1*: The review record area contains one or more sub-trees.
- *Rule 2*: Review records with the same template contain the same amount of sub-trees.

The entire review record extraction algorithm contains three parts: 1) Locate the review record area of the web page; 2) Eliminate noisy information in the review record area; 3) Extract the review records, forming as the input of the review content extraction algorithm.

Locate the Review Record Area

As web page tags are lack of semantic information, visual information is used to be merged into the DOM tree. Visual information on the web page has been proved as very effective feature information in the web data extraction algorithm, such as the algorithm in [6]. Based on the feature of the web review page, which there will be published time after web user comment on a product or problem, this paper identifies the review record area.

To facilitate the exposition of these rules, we define some symbols to represent the corresponding component of a DOM tree. Table 1 gives the description of these symbols.

Table 1. Symbols description

Symbols	Content
Root	the root node of web page's DOM tree
Tregion	the minimal sub-tree contains whole review records in DOM tree
Rregion	the root node of *Tregion*
Treview	the tree contains review record in *Tregion*
Tnoise	the tree contains non-review record in *Tregion*
Timenode	the node contains the publish time content in *Tregion*

The algorithm begins to depth-first traverse the DOM tree, and then finds out the nodes with a *Timenode* type child, which often can be *Treview* nodes. And then we find the review records' minimum ancestor node, which is the root node of the review record area *Rregion* according to the *Rule 1*.

Eliminate the Noisy Information

Eliminating the noisy information sub-tree from the review record area is a key step in the review record extraction algorithm. In order to eliminate these noisy information sub-trees, this paper invents a tree similarity algorithm, which calculates the similarity between any two sub-trees of the review record area. For sub-tree *X* and *Y*, with root node *x* and *y* respectively, the similarity of *X* and *Y* is calculated as follows:

$$TS(X,Y) = DS(x,y) + LCS(X,Y) \tag{1}$$

DS (x, y) is the depth similarity of node x or y in the *Tregion*. *LCS (X, Y)* is the similarity of the longest common sequence of sub-tree *X* and *Y*.

The formula to calculate *DS (x, y)* is given as follows:

$$DS(x,y) = tan\left(\frac{\pi(depth(x,y) - avgDepth(Tregion))}{2L}\right) \tag{2}$$

Here, *depth (x, y)* is the depth of the node *x* or *y* in *Tregion*, *avgDepth (Tregion)* is the average depth of all nodes of *Tregion* in DOM tree and *L* is stand for the difference between maximum and minimum depth of the nodes in *Tregion*. So the value of *DS (x, y)* depends on the *depth (x, y)* and *avgDepth (Tregion)*. We use trigonometric functions to effectively enlarge the depth differences between the nodes, and so that improve accuracy of the following noise eliminating algorithm.

The tree similarity algorithm takes several steps: Firstly, match the root node *x* and *y* of the sub-tree *X* and *Y*. If not match, terminate the algorithm, *TS (X, Y) = 0*; If match, calculate the value of *DS (x, y)* and *LCS (X, Y)*, and then *TS(X, Y) = DS(x, y) + LCS(X, Y)*. The value of *DS (x, y)* can be calculated according to the formula (2). *LCS (X, Y)* is computed by summing the longest common subsequence's similarity values of the sub-tree *X* and *Y*.

The pseudo-code of tree similarity algorithm is illustrated as follows:

```
Algorithm TS // Tree Similarity.
Input: X, Y; //Two subtrees
Output: s; //The similarity between X and Y
Begin
x=the root of X;
y=the root of Y;
if NodeMatching(x,y) is false then
return 0;
return DS(x,y)+LCS(Children(x),Children(y));
End
```

The global similarity of the adjacent two sub-trees is used to describe the variation trend between the review record sub-trees and noise sub-trees. If the noisy

information sub-tree exists, the maximum ratio is produced in the first review record sub-tree and the noise sub-tree just before it, and the smallest ratio is produced between the last review record sub-tree and the noise sub-tree after it. Here, global similarity value of a *Tregion* is defined as the maximum value of similarities between each child and its brothers, which are formulized by *gs*.

Base on the above ideas, we propose a noise eliminating algorithm, which can be described as follows:

Firstly, calculate the similarity between any two children sub-trees in the *Tregion* using the tree similarity algorithm. Then, gain the global similarity of each child sub-tree and calculate the ratio of the global similarity between the adjacent child sub-trees. Finally, according to the maximum and minimum value, identify the location of the first review record sub-tree *Start* and the last review record sub-tree *End*.

The pseudo-code of noisy eliminating algorithm is presented as follows:

```
Algorithm EN // Eliminate Noise.
Input: Tregion, n; // Tregion is the tree of review re-
gion, n is the numbers of review trees.
Output: Start, End;  //Start is the first review record
position, End is the last review record position
Begin
T[n]=Children(Tregion);
for i=1 to n do
for j=1 to n do
gs[i]=Max(TS(Ti,Tj));
for i=2 to n do
  Start=Max(gs[i]/gs[i-1]);
End=Min(gs[i]/gs[i-1]);
return Start and End;
End
```

Extract Review Record

Extracting review records from the review record area is the final step of the review record extraction algorithm. Specific ideas described as: use the location information from the noise eliminating algorithm as the output of the Map function directly.

The pseudo-code of extracting the review records is illustrated as follows:

```
Algorithm RE // Record Extract.
Input: URL,Page; //Web review page's URL and content
Output: URL,Treviews;  //Treviews is the review tree
Begin
method Map(URL, Page)
Tregion=Region(Page);
TS(Children(Tregion));
EN(Tregion);
for i=Start to End do
EMIT(URL,Treviews);
End
```

2.3 Review Content Extraction Algorithm and Implementation

In each sub-tree of the review records, the review content corresponds to a complex sub-tree rather than a simple leaf node. Therefore identifying the minimal sub-tree, which contains the whole review content, is the key to the review content extraction algorithm. The review content extraction algorithm consists of two steps: 1) Locate the review content area; 2) Extract the review content from the review content area.

Locate the Review Content Area

In this paper, we define a minimal sub-tree, which contains all the review content in the review record sub-tree, as the review content sub-tree, which is formalized by *Tcontent*. Its root node is formalized by *Rconent*. The semantics are the same among the review content in the review record sub-trees. So we can find out one review record sub-tree's *Rcontent* node, and identify the review content area, which is the location of the review content in the same template pages.

Since of the length of all review text contents in *Tcontent* is most inconsistent, we propose a novel approach of extracting review content based on the text depth of the contents. The detailed approach is described as follows: Firstly, find out the longest length node among the all review record sub-trees. Then, calculate the depth value of this node, and compare to the depth value of this review record sub-tree. If the value is equal, it means that the node is a review content node; If not equal, we recalculated to find the second longest node, and repeat this process. Finally, after finding the review content node, we travel up the review record sub-tree to find the common node of the review content node and the *Timednode*, which are the review record sub-tree's *Rconent*.

Extract Review Content

After identifying the review content area in the sub-tree of the review record, we get the real review content in loop output all text content under the sub-tree of the review record with the same page template. These text contents are the final content of the review extracted, because that the semantics of the order of review records on the same template is consistent with the rules.

The pseudo-code of extracting the review content is given as follows:

```
Algorithm CE // Content Extract.
Input: URL,Treviews;   //Web review page's URL and review
trees
Output: URL,Contents;  // Web review page's URL and re-
view contents
Begin
method Reduce(URL, Treviews)
Ntext=getMaxNode(Treviews);
flag=Compare(Depth(Ntext),Depth(Treview));
if(falg=0)
Rancester=getAncestor(Ntext,getTimeNode(Treviews));
Tcontent=Children(Rancester)
EMIT(URL,getText(Tcontent));
End
```

3 Web Review Automatic Extraction System Design

The input of the Hadoop-based web review automatic extraction system is the web review pages, and outputs are the review text contents, which are extracted from the input pages. The system is able to efficiently extract the review content accurately from the web review pages. The detailed data processing of the system is showed in Fig 2.

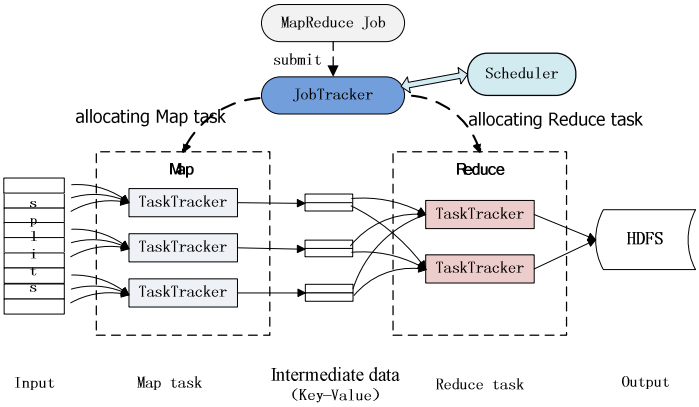


Fig. 2. Data processing in the Hadoop-based Web review automatic extraction system

The workflow of Hadoop-based web review automatic extraction system is described as follows:

1. Each map task is assigned to a part of the input file, which is called a split. In this system, the content of split is the web review page. And the total number of splits determines the number of map tasks.
2. The execution of the map tasks can be divided into two steps: The first step is to read the input splits (web review page), and turn them into the data records (key-value pairs, where as a web page URL - web content). The second step is to apply the map function (implementation of the review record extraction algorithm) to each record.
3. Next is the execution of the reduce task, which is divided into three steps: Firstly, the shuffle phase, gains the intermediate data (review record sub-tree), which are generated by the map task. Each partition of the intermediate data will be transferred to each reduce task. Secondly, in the sort phase, the records with the same key value will be combined together (the same page of review record sub-trees are grouped together). Finally, in the reduce phase, the user-defined reduce function (execute the review content extraction algorithm) will be executed on each record and the corresponding list of values (the review tree in each web page).
4. Finally, the output of the reduce function (review content) will be written to the temporary storage unit of the HDFS. When all the reduce tasks are finished, the output files in the HDFS will be changed the temporary storage unit into the final storage unit automatically.

4 Experiment Evaluation

In this chapter, experiments of web review automatic extraction system based on Hadoop will be conducted to evaluate the overall performance and efficiency of this novel review extraction approach.

4.1 Data Source Collection

The data set of user review content extraction is generated from the Internet. In order to crawl the review data and save it to the local disk, a dedicated crawler is designed. The rules by what the crawler recognizes review pages from web pages are based on regular expressions for distinguishing review page's URL, which is compiled artificially. And the accuracy of the URL recognition accuracy is more than 99.99%. Therefore, the crawler's affection for the experiments is ignored in this paper.

In order to evaluate the extraction results, manual works have been done to count the useful information records. In this paper, the non-review contents are recognized as useless information and are discarded.

4.2 Experiment Platform

In the experiment, 10 PCs are used to form a small Hadoop cluster. Hadoop version is 0.20.2, and the replication of HDFS is set to 1. One PC is as Namenode, and others are as Datanodes. Each PC has Intel Xeon CPU E5150 (2.66 GHz), 4 GB main memory and 75 GB hard disk. The operating system installed on PCs is CentOS 5.6 Final.

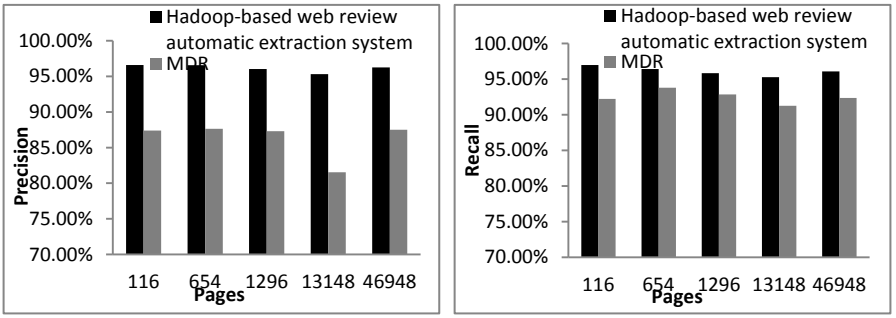
In order to evaluate the scalability of extraction system performance, each PC will be installed 5 virtual machines, so there're 50 nodes in all. Then the 50 nodes are used to build the simulated experiment environment.

4.3 Experiment Results and Analysis

In the field of web information extraction, common evaluation criteria are *Precision*, *Recall* and *F-measure*. *Precision* means the ratio between the number of the correct review pages that has been extracted and the whole number of pages that has been extracted as review pages. *Recall* means the ratio between the number of review pages that has been extracted by this system and the number of all review pages in a data source. The formula of *F-measure* is defined as follows:

$$F - measure = 2\, Recall \times Precision \,/\, (Recall + Precision) \tag{3}$$

These parameters are used to evaluate this novel Hadoop-based web review automatic extraction approach. For each web pages set (the data source), the MDR[7] extraction system is worked in contrast. Fig 3 shows the *Precision* and *Recall* comparison of the two systems.



(a) *Precision* (b) *Recall*

Fig. 3. Experimental extraction results comparison in Hadoop-based web review automatic extraction system and MDR extraction system

According to the formula (3) and Fig 3, we calculate the *F-measure* of Hadoop-based web review automatic extraction system and MDR extraction system. Fig 4 shows the actual *F-measure* of the two systems.

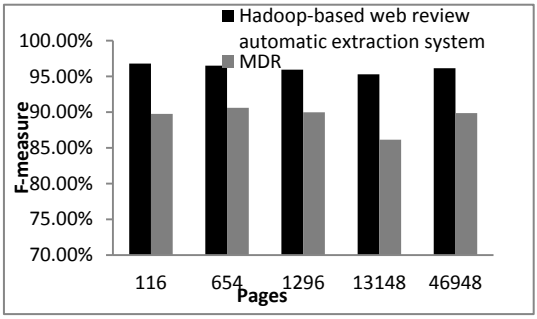
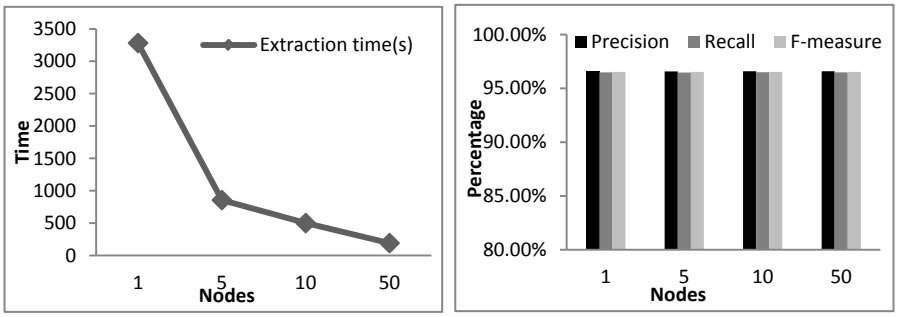


Fig. 4. F-measure of Hadoop-based web review automatic extraction system and MDR extraction system

Through the analysis of experimental results in Fig 3 and Fig 4, we found that the overall accuracy of web review automatic extraction system based on Hadoop is about 7% higher than the web extraction system MDR, and can reach more than 96%. In addition, when increasing the number of web pages, the accuracy of the extraction system will have some minor fluctuations, but overall is relatively stable. The main reasons of the fluctuations can be expressed as: Review record in the DOM tree's structure very differs from the visual information. So it tends to result in extracting the review records by mistake or ignoring as the noise information, affecting the accuracy of the entire experiment. This indicates that the proposed extraction algorithm has basically reached the requirements of practical applications.



(a) Time scalability (b) Accuracy scalability

Fig. 5. The scalability of Web review automatic extraction system based on Hadoop

In addition to the need for reliable accuracy, the scalability of Web data extraction approach is also very important. In this experiment, we have conducted a scalability test on the Web review automatic extraction system, which is deployed on different nodes' Hadoop cluster. In order to strengthen the effect of the experimental results and combine with the actual experimental environment, we have done the simulation experiments on the different Hadoop nodes to verify the scalability by using the same web review page data sets. The experimental results are showed in Fig 5.

Fig 5 (a) shows that, with the number of nodes from 1-50 in the expansion process of the extraction system, the time of review content extraction declines significantly. And Fig 5 (b) shows that the accuracy of the extraction system has been maintained above 96%, of which the variance of the precision and recall rate of less than 0.056, when the number of nodes from 1-50 in the expansion process of the extraction system. So the web review automatic extraction system has a very good scalability.

Speedup is the ratio of the time consumption by the same task in the single-processor systems and parallel or distributed processor system, used to measure the performance and effectiveness of the parallelization of parallel or distributed systems or procedures. In this paper, the speedup is the ratio of the time consumption of the extraction system running on one PC and the Hadoop cluster.

By analyzing the experimental results, we find that the speedup of the extraction system is a little small, when the test data size is not large. This case mainly due to: Since in the communication of Hadoop cluster nodes needs to consume a certain amount of time, it's difficult to play Hadoop well to deal with the large-scale mass data, when the size of the test data set is too small. But with the increasing data size, the extraction system will be parallel executed and the speedup of the extraction system becomes bigger.

5 Related Works

Information Extraction is a branch of natural language processing. It refers to extract structured data from resources by analysing the grammar or semantics of the content.

Researchers of IE fields have built many studies and systems of NLP[8], to extract the interested contents from a large number of text materials. However, most of the systems could only be applied in a narrowed domain of the text materials and are difficult to transplant to the new domains.

Web-IE[9] is a branch of IE in the field of Internet application. Web-IE is an extraction and structure from the massive data that has been distributed all over the Internet, which may be structured, half-structured or unrestrained. Typical automatic Web-IE systems are as follows: RoadRunner[10], EXALG[11] and IEPAD[12]. RoadRunner is designed to make a template by comparing similar web pages. Then we can use the template to extract information from massive web pages. RoadRunner treats the web pages as a flow of text and concludes the pattern by various kinds of heuristic algorithms. EXALG uses the similar method with several improvements and has a different algorithm in concluding patterns. While attempting to discover the frequent appearance of continuous marks, IEPAD is able to locate and extract data by constructing a PAT tree. But it could only be used on pages that do not include nested structures.

Hadoop is mainly used in parallel computing and distributed storage of massive data. Nowadays, it is widely used in mass intensive data process.

Rini et al. [13] proposes the GreenHDFS, a self-adaptive, energy-conserving variant of the HDFS. It can cut down on energy consumption of Hadoop cluster and reduce the running cost. The [14] replaces the HDFS of Hadoop by a new BlobSeer data management service based and concurrent optimized data storage layer. It improves the performance of the MapReduce parallel framework immensely. The [15] proposes an EHAD (Elastic Hadoop Auto-Deployer) system. This system is able to create or destroy virtual machine nodes, and also can deploy or distribution the environment configuration of Hadoop in virtual machine nodes. This system greatly improves the flexibility of the Hadoop's configuration.

6 Conclusion

In this paper, we propose a Hadoop-based web review automatic extraction approach, which has better performance on dealing with massive data than traditional web review data extraction methods. This new approach consists of two core algorithms: a review record extraction algorithm based on the node similarity and a review content extraction algorithm based on the text depth. The experiments have proved that our web review extraction approach has been not only reached the accuracy over 96%, but also achieved a high speedup.

References

1. Riloff, E.: Automatically constructing a dictionary for information extraction tasks. In: Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI), pp. 811–816 (1993)

2. Kim, J., Moldovan, D.: Acquisition of linguistic patterns for knowledge-based information extraction. IEEE Transactions on Knowledge and Data Engineering 7(5), 713–724 (1995)
3. Seymore, K., McCallum, A., Rosenfeld, R.: Learning Hidden Markov Model Structure for Information Extraction. AAAI Technical Report WS, pp. 37–42 (1999)
4. Apache Hadoop, http://hadoop.apache.org
5. Document Object Model, http://www.w3.org/DOM/
6. Liu, W., Meng, X., Meng, W.: Vision-Based Web data records extraction. In: Zhou, D. (ed.) Proc. of the Int'l Workshop on the Web and Databases (WebDB), pp. 20–25 (2006)
7. Liu, B., Grossman, R.-L., Zhai, Y.: Mining Data Records in Web Pages. In: Proc. of the ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining (KDD), pp. 601–606 (2003)
8. Soderland, S.: Learning information extraction rules for semi-structured and free text. Journal of Machine Learning 34(1-3), 233–272 (1999)
9. Chang, C.H., Kayed, M., Girgis, M.R., Shaalan, K.F.: A survey of Web information extraction systems. IEEE Trans. Knowledge and Data Engineering 18(10), 1411–1428 (2006)
10. Crescenzi, V., Mecca, G., Merialdo, P.: RoadRunner: towards automatic data extraction from large Web sites. In: Proceedings of the 26th International Conference on Very Large Database Systems (VLDB), Rome, Italy, pp. 109–118 (2001)
11. Wang, J., Lochovsky, F.H.: Data extraction and label assignment for Web databases. In: Hencsey, G., White, B. (eds.) Proc. of the Int'l Conf. on World Wide Web (WWW), pp. 187–196. ACM Press, Budapest (2003)
12. Chang, C.-H., Lui, S.-C.: IEPAD: Information extraction based on pattern discovery. In: Proceedings of the Tenth International Conference on World Wide Web (WWW), Hong-Kong, pp. 223–231 (2001)
13. Kaushik, R.T., Bhandarkar, M., Nahrstedt, K.: Evaluation and Analysis of GreenHDFS: A Self-Adaptive, Energy-Conserving Variant of the Hadoop Distributed File System. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science, pp. 274–287 (2010)
14. Nicolae, B., Moise, D., Antoniu, G., Bouge, L., Dorier, M.: BlobSeer: Bringing high throughput under heavy concurrency to Hadoop Map-Reduce applications. In: 2010 IEEE International Symposium on Parallel & Distributed Processing, pp. 1–11 (2010)
15. Mao, H., Zhang, Z., Zhao, B., Xiao, L., Li, R.: Towards Deploying Elastic Hadoop in the Cloud. In: 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 476–482 (2011)