Mining Web Pages Using Features of Rendering HTML Elements in the Web Browser

F.J. Fernández, José L. Álvarez, Pedro J. Abad, and Patricia Jiménez

Abstract. The Web is the largest repository of useful information available for human users, but it is usual that Web Pages do not provide an API to get access to its information automatically. In order to solve this problem, Information Extractors are developed. We present a new methodology to induce Information Extractors from the Web. It is based on rendering HTML elements in the Web browser. The methodology uses a KDD process to mining a dataset with features of the elements in the Web page. An experimentation over 10 web sites has been made and the results show the effectiveness of the methodology.

Keywords: Wrapper generation, web data extraction, data mining.

1 Introduction

The Web offers interest information for human users through Web Pages. However, in recent years, its growth has generated a large volume of information of interest, especially in business and, therefore the research community. The aim is developing automatic techniques to facilitate the processing of this large volume of information to extend the functionality of traditional web.

Information Extraction (IE) from Web Pages focus on extracting the relevant information from semi-structured Web Pages, unify them and offers it in a structured format to the end-users. The algorithms that perform the task of IE are referred to as extractors. Extractors can be classified in four classes [5] according to the intervention of an expert user: hand-crafted IE Systems, supervised IE Systems, semi-supervised IE Systems and unsupervised IE Systems.

Unsupervised IE Systems are based on the hypothesis that "Web Pages have a common template". Examples of these systems are MDR [2], IEPAD [13], RoadRunner

F.J. Fernández · José L. Álvarez · Pedro J. Abad · Patricia Jiménez
Departament of Information Technologies, University of Huelva
Ctra. Huelva-La Rábida, 21819 Palos de la Frontera, Huelva, Spain
e-mail: {javier.fernandez,alvarez,pedro.abad,
patricia.jimenez}@dti.uhu.es

[14], or EXALG [1]. Supervised IE Systems need of annotating Web Pages by an expert user. Examples of these systems are WIEN [11], SoftMealy [4], STALKER [8], WHISK [12], SRV [7], DEPTA [18], and ViDE [15]. In Semi-supervised IE Systems, the annotation process can be automated somewhere: OLERA [3].

Therefore, in supervised IE Systems, the extractors must query the Web to collect the Pages, labeling the interest contents on the HTML code, and finally, training to learn patters that allow extracting content of interest from new Web Pages.

This paper presents a new supervised IE System based on rendering the HTML elements in the web browser. Three types of features have been defined: layout features, style features and content features. These features are calculated from HTML elements on the Web Pages. Knowledge Discovery in Databases (KDD) process is applied on this dataset to learn patters to extract interest content in new web Pages.

The rest of the paper is organized as follows: section 2 presents the methodology. Section 3 shows the cases of study and the results. Finally, section 4 concludes this work.

2 Methodology

Almost of the extractors are based on pattern recognition to extract the information from the structure of the web page. The goal of extractors is to extract the relevant information with the minimum error. However, today, web pages are full of non-relevant information trying to make the web page more striking. Moreover, the large number of advertisement in web pages makes difficult the identification of relevant information. In this situation, the extractors must to deal with two problems:

- First, the vast amount of non-relevant information to analyze. This produces that extractors spend a lot of time with the lack of the high computational cost of these algorithms. When the number of pages increases, the effectiveness loss becomes more evident.

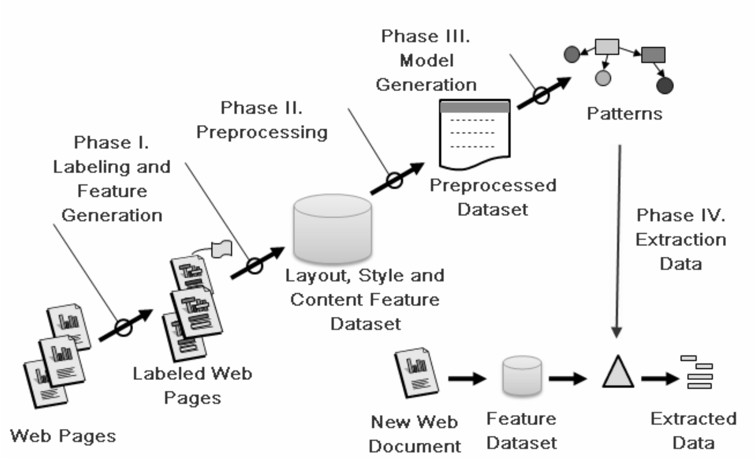

Fig. 1 An overview of the steps that compose the proposed methodology

- The second and harder problem is the loss of effectiveness in the extraction of the goal information. The non-relevant information in the pages makes the web page structure more complex and the extractors need to deal with these complexity.

The code of the non-relevant information in a Web page causes extraction algorithms fail when they try to extract the relevant information.

The proposed methodology is developed in the following steps (Fig. 1):

1. **Web Page labeling and feature generation.** This step involves the labeling of the information to extract and the feature generation of this information. The information selection is performed by the user. After labeling the features of selected information are generated and stored in a database. Features of both, relevant information and non-relevant information are generated and stored.
2. **Preprocessing of the training set.** Techniques of feature selection and classes balancing are applied to the database generated in step 1, which is considered as a training set in a supervised learning process.
3. **Models generation.** A model of the data contained in the training set is generated by a supervised learning algorithm. This model lets classify new information as belonging to one of the classes.
4. **Information Extraction.** The model generated in the previous step is applied to new web pages of the same Web site in order to extract the desired data.

To support this methodology we have developed the W-SOFIE (Web Software Observant For Information Extraction) tool. It automates the steps 1 to 3 and returns the induced model from the training pages.

The next sections describe in detail each one of the steps of the methodology.

2.1 Web Document Labeling and Feature Generation

The Web Pages of interest are those which present information about specific products. These pages display the product details as a product information sheet.

The W-SOFIE tool is able to navigate through these pages in order to label the desired elements. Labeling is performed by the user who also assigns the class to each selected element. Once the user has labeled all interesting elements, the calculation of the interesting area is performed. This process involves selecting the area on the page that contains the labeled elements. Only this area will be treated, the rest of the page is ignored because it does not contain relevant information. The coordinates of the interesting area of each page are stored.

The remaining not-selected elements within the interesting area are automatically labeled as not-interesting. All of these not-interesting elements are assigned to the same class, the not-interesting class.

Once all elements on the page are labeled, W-SOFIE automatically generates the features characterising them, and then, a tuple is stored in the database. This tuple represents an element and consists of the calculated features and the class assigned. Each element on the page is stored in the database, both the interesting elements and not-interesting ones.

Three types of features are generated and stored for the labeled elements:

- **Position features.** X and Y coordinates, high and wide of the element, the distance from the left-top corner of interesting area and the slope of this hypothetical line.
- **Style features.** Font color and style, background color and borders.
- **Content features.** Text density, digit density.

After this step, a database containing a tuple for each element labeled, and the coordinates of the interesting area of all treated pages is generated. The number of treated pages should be sufficient to represent all the possible layouts of the product pages.

2.2 Pre-processing of the Training Set

Previous database is ready for KDD process. This database is considered as a training set. However, before the underlying model can be learned, a preprocessing step is necessary. The database has two main problems:

- The database has high dimensionality. There are a lot of features that represent each element. In addition, much of that information is not relevant. In order to solve this problem, several feature selection algorithms can be used.
- The database is imbalanced. There are a lot of tuples in database belonging to not-interesting class, and few tuples belonging to interesting class. A balancing of database is required to solve this problem.

  After this step, database is suitable for an automatic learning process.

2.3 Models Generation

Because the only elements into interesting area have been considered, a learning of the coordinates of these areas is needed. This way, the first task to perform is determining the coordinates of this area. It is discovered from all stored areas of all labeled pages. An area wrapping all of them is considered as the general interesting area.

After discovering the general interesting area, a variety of classification algorithms can be used to extract a model of the training data. The representation of the model depends on the algorithm used.

It is important to highlight that the learning process is performed off-line, before the information extraction step. Thus, the time taken to generate the model is spending only once, before the information extraction process.

2.4 Information Extraction

The information extraction step starts automatically extracting all elements on a new page within the interesting area. The interesting area considered for a new page is the general interesting area calculated in the previous step. None of all extracted elements have a class assigned. In order to consider these elements as belonging to a class, they are evaluated using the learned model in the previous step. The assigned class by classifier indicates if an element is interesting or not. All desired elements are stored and the not-interesting ones are rejected.

3 Experimentation and Results

In order to test the methodology, it was applied to 10 e-commerce WEB sites related with sales of books. These 10 web sites are the top ten by the Alexa (http://www.alexa.com) ranking.

The 50 Bestsellers pages of each site were tested. The elements to extract were: Title, Authors, Saving, Rates and Availability.

In order to reduce the dimensionality of the training database, four feature selection algorithms were tested: InfoGainAttributeEval (wrapper algorithm), ReliefFAttributeEval (wrapper algorithm), CfsSubsetEval (filter algorithm) and ConsistencySubsetEval (filter algorithm).

The wrapper algorithms return a ranking of the features. Instead, the filter algorithms return a subset of features. In order to select a unique subset of features a hand-crafted selection was performed. The features selected were all of them returned by filter algorithms and the most relevant features (50% of the better rank) in the ranking of the wrapper algorithms. The mean of the size of the resulting database was 18.74% of the original one with a standard deviation of 0.01.

The balancing of the database was performed by oversampling; i.e., replicating examples of the minority class.

Once pre-processing had finished, four classification algorithms were tested: SMO [10], IBk [6], C4.5 [9] and a Stacking meta-model, where three individual models (SMO, IBk, and JRip [17]) were combined. The meta-model was trained from the individual models using the C4.5 classifier.

Table 1 Datasets before pre-processing

	Accuracy percentage				AUC			
Web Sites	C45	SMO	IBK	META	C45	SMO	IBK	META
Amazon	99.04	98.92	99.28	99.28	1.00	1.00	1.00	1.00
Barnesandnoble	98.68	98.52	98.68	97.87	1.00	1.00	1.00	0.94
Bestbuy	97.93	99.58	99.17	99.58	0.98	1.00	1.00	1.00
Buy	99.47	99.47	99.47	99.47	0.96	1.00	0.99	1.00
Ebay	98.18	91.64	98.82	99.14	0.97	0.91	1.00	0.99
Overstock	99.79	99,90	100.00	99.79	0.98	1.00	1.00	0.98
Play	99.71	99.71	99.71	99.41	1.00	1.00	1.00	1.00
Sears	99.57	99.57	100.00	98.22	1.00	1.00	1.00	1.00
Target	99.30	98.42	98.94	99.30	1.00	0.99	1.00	0.98
Walmart	99.66	99.20	99.43	99.31	1.00	1.00	1.00	1.00

Table 2 Datasets after feature selection

	Accuracy percentage				AUC			
Web Sites	C45	SMO	IBK	META	C45	SMO	IBK	META
Amazon	99.03	98.92	99.21	99.10	1.00	1.00	1.00	0.99
Barnesandnoble	98.65	98.57	98.55	98.37	1.00	1.00	1.00	0.99
Bestbuy	99.61	99.34	99.30	98.68	0.99	1.00	1.00	0.99
Buy	99.69	99.41	99.47	99.51	0.96	1.00	0.99	1.00
Ebay	98.97	90.47	99.05	99.11	0.98	0.90	1.00	0.99
Overstock	99.64	99.90	100.00	99.89	0.99	1.00	1.00	1.00
Play	99.70	99.70	99.70	99.53	1.00	1.00	1.00	1.00
Sears	99.56	99.48	100.00	98.99	1.00	1.00	1.00	1.00
Target	99.74	92.99	98.08	99.39	1.00	0.93	0.99	0.99
Walmart	99.73	99.20	99.43	99.47	0.99	1.00	1.00	1.00

Table 3 Datasets after feature selection and balancing

	Accuracy percentage				AUC			
Web Sites	J48	SMO	IBK	META	J48	SMO	IBK	META
Amazon	99.96	98.86	99.94	100.00	1.00	1.00	1.00	1.00
Barnesandnoble	99.86	98.44	99.72	99.89	1.00	1.00	1.00	1.00
Bestbuy	99.81	99.75	99.81	99.69	1.00	1.00	1.00	1.00
Buy	99.52	98.78	99.54	99.54	1.00	0.99	1.00	1.00
Ebay	99.94	97.82	99.88	99.93	1.00	1.00	1.00	1.00
Overstock	100.00	99.98	100.00	99.98	1.00	1.00	1.00	1.00
Play	99.89	99.89	99.89	99.85	1.00	1.00	1.00	1.00
Sears	99.84	99.73	100.00	99.32	1.00	1.00	1.00	1.00
Target	99.85	99.32	99.71	99.87	1.00	1.00	1.00	1.00
Walmart	99.99	99.86	99.91	99.96	1.00	1.00	1.00	1.00

The results obtained by 10-folds cross-validation are shown in tables 1 to 3. Values of Accuracy and Area Under ROC are detailed.

Values on tables show a very good accuracy and AUC of all algorithms and web sites. Most values are near 100% accuracy and 1 AUC.

Table 4 presents the mean of accuracy percent obtained for all classification algorithms in each web site.

A statistical analysis was carried out considering the mean of accuracy percent by means of Wilkoxon test [16]. The test did not detect significance (p=0.05) difference between results of dataset before preprocessing and results of dataset with feature selection (result of the test: p=0.492). This way, the conclusion that feature selection does not improve the classification is probed. Moreover, significance (p=0.05) difference was found between results of Datasets after feature selection and balancing and the others (result of the test: p=0.00586 in both cases). Therefore, we can conclude that feature selection and balancing presents the best accuracy.

Table 4 Mean of accuracy percent for each web site

Web Sites	Datasets before pre-processing.	Datasets after feature selection.	Datasets after feature selection and balancing.
Amazon	99.13	99.07	99.69
Barnesandnoble	98.44	98.54	99.48
Bestbuy	99.07	99.23	99.77
Buy	99.47	99.52	99.35
Ebay	96.95	96.90	99.39
Overstock	99.87	99.86	99.99
Play	99.64	99.66	99.88
Sears	99.34	99.51	99.72
Target	98.99	97.55	99.69
Walmart	99.40	99.46	99.93

4 Conclusions

In this paper we have presented a new supervised IE System to extract data in Web Pages based on rendering HTML elements in the Web browser. Layout, Style and Content features are calculated on the elements generating a dataset. A model is induced applying a KDD process to the dataset. Thus, data from new Web Pages can be extracted using this model.

Experimentation with 10 website of e-commerce has been made. The results show the effectiveness of this system.

The bottleneck of our method is the annotation phase, since the user intervention is required. In future works, a semi-automatic process will be used.

Acknowledgments. This work was partially sponsored by Spanish Ministry of Science and Technology and Junta de Andalucía under grants TIN2007-64119, P07-TIC-02602, and P08-TIC-4100.

References

1. Arasu, A., Garcia-Molina, H.: Extracting structured data from web pages. In: ACM SIGMOD International Conference on Management of Data, pp. 337–348 (2003)
2. Liu, B., Grossman, R., Zhai, Y.: Mining web pages for data records. IEEE Intelligent Systems 19, 49–55 (2004)
3. Chang, C.H., Kuo, S.C.: OLERA: Semisupervised web-data extraction with visual support. IEEE Intelligent Systems 19(6), 56–64 (2004)

4. Hsu, C.N., Dung, M.T.: Generating finite-state transducers for semi-structured data extraction from the web. Information Systems 23(8), 521–538 (1998)
5. Chang, C.-H., Kayed, M., Girgis, M.R., Shaalan, K.F.: A Survey of Web Information Extraction Systems. IEEE Transactions on Knowledge and Data Engineering 18, 1411–1428 (2006)
6. Aha, D.W., Kibler, D., Albert, M.K.: Instance based learning algorithms. Machine Learning 6(1), 37–66 (1991)
7. Freitag, D.: Information extraction from HTML: Application of a general learning approach. In: Fifteenth Conference on Artificial Intelligence, AAAI 1998 (1998)
8. Muslea, I., Minton, S., Knoblock, C.A.: A hierarchical approach to wrap-per induction. In: Agents, pp. 190–197 (1999)
9. Ross Quinlan, J.: C4.5: Programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
10. Platt, J.C.: Fast Training of Support Vector Machines using Sequential Minimal Optimization, pp. 185–208 (1999)
11. Kushmerick, N., Weld, D.S., Doorenbos, R.B.: Wrapper induction for information extraction. In: The 15th International Joint Conference on Artificial Intelligence (IJCAI), pp. 729–737 (1997)
12. Soderland, S.: Learning information extraction rules for semi-structured and free text. Journal of Machine Learning 34(1-3), 233–272 (1999)
13. Csie, T.C., Hui Chang, C.: IEPAD: Information extraction based on pattern discovery. In: The 10th International Conference on World Wide Web, pp. 681–688 (2001)
14. Crescenzi, V., Mecca, G., Merialdo, P.: RoadRunner: Towards automatic data extraction from large web sites. In: The 27th International Conference on Very Large Data Base, pp. 109–118 (2001)
15. Liu, W., Meng, X., Meng, W.: VIDE: A Vision-Based Approach for Deep Web Data Extraction. IEEE Transactions on Knowledge and Data Engineering 22, 447–460 (2009)
16. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics 1, 80–83 (1945)
17. Cohen, W.W.: Fast Effective Rule Induction. In: Twelfth International Conference on Machine Learning, pp. 115–123 (1995)
18. Zhai, Y.: Web data extraction based on partial tree alignment. In: Proceedings of the 14th International Conference on World Wide Web (WWW 2005), pp. 76–85 (2005)