Combining Classifiers to Extract Web Data

Qiang Chu, Yongquan Dong*, Ping Ling

School of Computer Science and Technology, Jiangsu Normal University,

Xuzhou, 221116, China

Keywords: Web Data Extraction; Ensemble Learning; Data Integration

Abstract. A lot of data on the web are usually embedded in the semi-structured pages. In order to automatically process the content embedded in Web pages, extracting data from them and making it available to computer applications remains a complex and urgent task. Most of current approaches use a single classifier to extract web data, but relying on a single classifier is not sufficient and different classifier has different performance for a problem. In this paper, we combine multiple classifiers to extract web data. Firstly, we identify the main data regions of web pages, and construct feature sets of text nodes in the regions. Secondly, we choose three kinds of base classifiers and then use the voting method to integrate results of each classifier. Finally, we combine integration results with heuristic rules to get the final extraction results. The experiment results show that our approach outperforms the baseline approaches.

Introduction

The amount of information that is currently available on the Web in HTML format grows at a very fast pace. However, websites are often intended to be browsed by humans, and not computed over by applications. So in order to automatically process the content embedded in Web pages, extracting data from them and making it available to computer applications remains a complex and urgent task. There has been a lot of recent research in the database and AI communities on web data extraction [1], which can identify the information from unstructured or semi-structured Web pages, and turn it into a structured format. In general, according to the used techniques, web data extraction can be divided into the following categories: WDE by natural language processing [2], WDE by wrapper induction [3], WDE by html DOM structure [4], WDE by ontology [5], WDE by machine learning [6] and so on. With the dramatic increase of web data, the first four methods have difficult to process these large scale data. The methods of machine learning for large-scale web data extraction will be the future research trends. However, current main methods usually use a single classifier to extract the data, and the extraction accuracy still needs to be further improved. As different single classifier has different performance on web data extraction, in this paper we propose a method which combines multiple classifiers to extract web data. Firstly, we identify main data regions according to the page structure, then generate feature set of text nodes (We call leaf nodes of the DOM [7] tree of the Web page as text nodes). Secondly, we choose the base classifiers to build classification models with feature set, and then use the voting method to integrate classification results of each base classification model. Finally, we combine integration results with heuristic rules to get the final extraction results. Experimental results on 20 different structural sites prove the validity of our approach.

Our Approach

Framework

The framework is divided into two phrases. The first one is training phrase. In this phrase, we first identify the main data regions of training web sites to determine the area of data extraction. Secondly, we choose multiple base classifiers to build models. The second one is extraction phrase. In this phrase, we first use every trained model on testing set to obtain every classification result.

Secondly, we use the voting method to integrate these classification results, and then combine the integration results with heuristic rules to get the final extraction results.

Main Region Identification

A Web page contains a lot of data regions, but not all data regions are required by the user. Therefore, we must determine the data extraction region at first. In this paper, we define region node as non-leaf node of the DOM tree, and then set three parameters: linkCharacterNum, characterNum, linkDensity. The linkCharacterNum is used to record the number of hyperlink text characters of each region node. The characterNum is used to record the number of all text characters of each region node. The linkDensity is used to record the proportion of hyperlink text characters of each region node, which is shown in equation (1).

$$\text{linkDensity}=\text{linkCharacterNum}/\text{characterNum} \tag{1}$$

We have observed 100 web pages from different websites and find that in the data region required by the user, its linkDensity is always less than 0.5. So in this paper we call the data region whose linkDensity is smaller than a threshold as the main region. According to our observation, we set the threshold to be 0.5. The main region is the concern of web data extraction system.

Model Construction

Base Classifier. In this paper, we select three base classifiers to create the classification model: SVM、KNN and Random Forest.

SVM[8]

SVM is the abbreviation of Support Vector Machine which can be used for classification. It belongs to a family of generalized linear classification. SVM can simultaneously minimize the empirical classification error and maximize the geometric margin. SVM maps input vector to a higher dimensional space where a maximal separating hyperplane is constructed. Two parallel hyperplanes are constructed on each side of the hyperplane that separate the data. The separating hyperplane maximizes the distance between the two parallel hyperplanes.

KNN[9]

KNN (k-Nearest Neighbor, KNN) is a classification algorithm based on distance measure. A sample is classified by a majority vote of its neighbors, with the sample being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). In this experiment, we set k=1, that is to say, the sample is assigned to the class of that single nearest neighbor.

Random Forest[10]

Random Forest builds a forest containing a lot of decision trees randomly, and each decision tree has no correlation with each other. When a new sample is inputted, each decision tree in the forest will judge the sample, give their classification results and choose the classification having the most votes as the final classification result of the new sample.

Classifier Integration. In the previous section, we introduce three base classifiers. However, a single base classifier has its shortcomings in performance. In order to improve the accuracy of data extraction, we use the voting method [11] to combine these three classifiers to obtain the final result. The common voting methods include majority voting, one-vote veto, minimum voting, etc. In this paper, we use majority voting.

Result Combination. The output categories of each attribute contain three types: the first one is the attribute name (denoted as [attribute]-an), the second one is the attribute value (denoted as [attribute]-av) and the third one is the attribute value and name (denoted as [attribute]-nv), where [attribute] represents any attribute; the other one is that the attribute name and attribute value are in one text node. For the first situation, we can directly combine the values with the same catetory "[attribute]-av" as the final extraction result of this attribute. For the second situation, we first separate its value into attribute name and attribute value according to some heuristic characteristics, then combine this value with the adjacent values having the same category "[attribute]-av" and regard the combination value as this attribute's final extraction result.

Feature Set. In this paper, we use three kinds of features for a text node:

(1)Own features. They include whether it contains numbers, whether it contains special characters (for example, ¥, :) and the length of text node.
(2)Path features. They include the path information of text node.
(3)Contextual features. They include the information of the adjacent previous text node and the adjacent next text node. And this information is composed of the own features of the text node.

Experiments

Dataset

In this paper, the training set is collected from 20 websites with different structures. On each website we collect 40 pages. The validation set and testing set are also collected from these 20 websites. On each website we collect 10 pages. We extract 21 kinds of attributes. Each attribute contains three types of category: "[attribute]-an", "[attribute]-av" and "[attribute]-nv".

Evaluation Criteria

We measure the extraction performance via three metrics: precision, recall and F1. A brief definition is given as follows.
Defining m as the existing total number of attributes, t as the number of attributes which are correctly extracted, n as the number of attributes which are wrongly labeled. Then the evaluation criteria is defined as follows: P=t/(t+n), R=t/m, F1=2PR/(P+R),where P is precision representing the confidence of the results, R is recall representing the coverage of correct results, F1 is the result of considering P with R together. As every attribute has its F1, for simplicity, we only use average F1 of all attributes to evaluate the experiment in this paper.

Discussion on Experimental Results

We have designed two experiments to evaluate the effects of our approach. In these experiments, the base classifier Random Forest is abbreviated as RF.
(1)Performance Comparison between Single Base Classifier and Our Approach
In this experiment, we compare the extraction performance of single base classifier with our approach. The result is shown in Fig. 1.

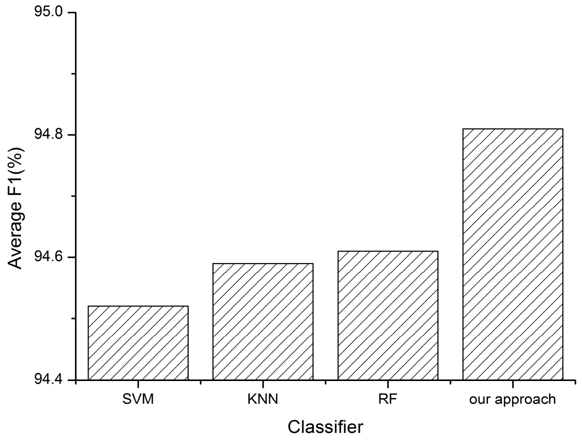

Fig. 1: Average F1 using Single Base Classifiers and Our Approach

In Fig. 1, the average F1 of single base classifiers SVM、KNN and Random Forest are 94.52%、94.59% and 94.61% respectively. The average F1 of ensemble classifier is 94.81%. The result suggests the extraction performance of our approach outperforms that of any single base classifier. This is because although each single base classifier has its own advantages and disadvantages, our approach can utilize complementary information among multiple base classifiers and achieve the best performance.

(2) Effect of Changing the Size of Training Set

In this experiment, we test the influence of training set size on the extraction results. There are 20 websites with different structures in training set. From each website, we randomly select the pages whose size is from 5 to 40 with step size 5 to derive different training sets. The experimental result is shown in Fig. 2.

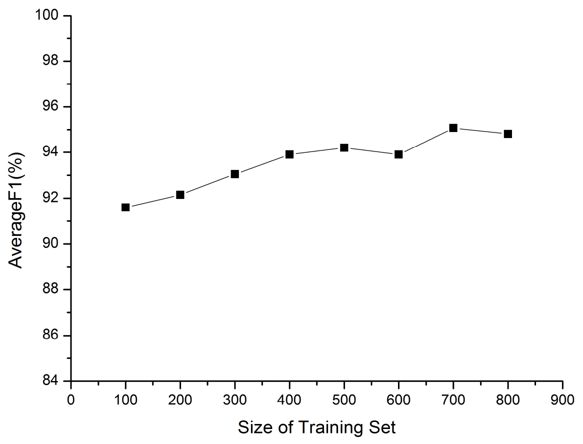

Fig. 2: Average F1 using different size of training sets

In Fig. 2, we can see that when increasing the size of the training set, a gradual improvement in average F1 is obtained. It can also be seen from Fig. 2, when the size of training set is 500, the average F1 is close to its maximum. Thus, our method only needs labelling a few pages to achieve better extraction performance and it can be applied to large-scale Web data extraction.

Conclusion

In this paper, we combine multiple classifiers to extract Web data. Firstly, we clean the Web page, and build the feature set of text nodes in the main regions. Secondly, we choose three base classifiers and use feature set to construct classification models respectively. Then, we use voting method to integrate classification results of each base classifier model. Finally, we use heuristic method to get the final extraction results. The experiments prove that the extraction performance of our approach is better than that of each base classifier.

The text nodes in this paper all contain one attribute. However, the structure of Webpage is complex. Many text nodes contain multiple attributes. Therefore, we will further study the problems that the single text node contains multiple attributes in the future work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 61100167 and 61105129, and the Natural Science Foundation of Jiangsu Province, China under Grant No. BK2011204.

References

[1] Chang, C.H. A Survey of Web Information Extraction Systems. *IEEE Transactions on Knowledge and Data Engineering*, 2006, 18(10):1411-1428.

[2] Soderland, S., Learning information extraction rules for semi-structured and free text. *Journal of Machine Learning*, 34(1-3): 233-272, 1999.

[3] Hsu, C.-N. and Dung, M., Generating finite-state transducers for semi-structured data extraction from the web. *Journal of Information Systems*, 23(8): 521-538, 1998.

[4] Crescenzi, V., Mecca, G. and Merialdo, P., RoadRunner: towards automatic data extraction from large Web sites. *Proceedings of the 26th International Conference on Very Large Database Systems(VLDB)*, pp. 109-118, 2001.

[5] Embley, D.W., Campbell, D.M., Jiang, Y.S., Liddle, S.W., Kai Ng,Y. Quass, D. and Smith, R.D. Conceptual-model-based data extraction from multiple-record Web pages. *Data and Knowledge Engineering*, 31(3):227-251, 1999.

[6] Aidan Finn, Nicholas Kushmerick. Multi-level boundary classification for information extraction . *Proceedings of the 15th European Conference on Machine Learning*(ECML), 2004, pp. 111-122.

[7] World Wide Web Consortium.W3C.The Document Object Model.http://www.w3.org/ DOM.

[8] Vapnik, V. *The Nature of Statistical Learning Theory*. NY: Springer-Verlag. 1995.

[9] Cover, T., Hart, P. Nearest neighbor pattern classification. *IEEE Transaction on Information Theory*, 13(1): 21-27, 1967.

[10] Leo, B. Random Forest. *Machine Learning*, 2001, 45(1): 5-32.

[11] Schapire,R. E., Freund, Y., Bartlett, P., and Lee, W.S. Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. *Annals of Statistics*, 1998, 26(5): 1651-1686, 1998.