Journal of Theoretical and Applied Information Technology
31st January 2013. Vol. 47 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A CALCULATION METHOD OF DEEP WEB ENTITIES RECOGNITION

[1]FENG YONG, [2]DANG XIAO-WAN, [3]XU HONG-YAN
School of Information, Liaoning University, Shenyang Liaoning
E-mail: [1]fyxuhy@163.com , [2]dangxiaowan@163.com , [3]xuhongyan_lndx@163.com

ABSTRACT

A great amount of duplicated entities exist in the heterogeneous data sources of Web. How to identify these entities is the most important prerequisite for pattern matching and data integration. To solve the deficiency of current entity recognition methods, such as low automatic level and poor adaptability, Deep Web entity recognition method based on BP neural network is proposed in this paper. To make full use of the characteristic of autonomic learning of BP neural network, the similarities of semantic blocks are used as the input of BP neural network. It obtains correct entity recognition model through training, and accomplishes the target of automated entity recognition of heterogeneous data sources. Finally, the feasibility of the proposed method is verified via experiments. The proposed method can reduce manual intervention and promote the efficiency and the accuracy of entity recognition simultaneously.

Keywords: *Deep Web; BP Neural Network; Entities Identify; Similarity*

1. INTRODUCTION

As the development of Web technology and the popularization of application, the resource of the internet information is increased sharply. Web data sources can be divided into Surface Web and Deep Web according to the amount of information it contained. Usually, the Web data source which can not be searched by the traditional search engine is defined as Deep Web [1]. During studying the Deep Web, entities recognition is the primary premise about developing pattern matching, and data integration. So how to recognize the same entities accurately and effectively in the Deep Web has become the research focus in the field of Deep Web.

In Deep Web entity recognition field, relevant scholars have carried out in-depth research, obtained some representative research results, such as Tejada, Kniblock and Minton put forward a series of methods based on pattern matching and public attribute weights which give string conversion rules and apply in the public attribute to recognize the entity. The method increases the accuracy of entities recognition, but it needs a pattern matching as the prerequisite, and needs a lot of manual intervention [2]; Cui, Xiao and Ding put forward the adaptive Web record matching method based on the distance, this method increases the efficiency of entities recognition, but it is difficult to application in bad structured Web data [3]; Sarawagi put forward the method that find equivalent units in a given table model, and then compare the similarity of the corresponding attributes of the units to conduct entities recognition. This kind of method shortens the recognition time, improves the recall rate, but the expandability of the method is poor [4].

Aim to solving the problems of the existing research results such as automation level is not high and the adaptability of heterogeneous Web data source is poor, BP neural network technology is used in entities recognition of Deep Web. The structure of this paper is given as follows. Section 2 introduces the theory of BP neural network. Section 3 presents the Deep Web entities recognition method base on BP neural network in detail. Section 4 gives the specific experiment to verify the practicability of the proposed method; Section 5 summarizes the whole paper and points out the advantages of the proposed method.

2. BP NEURAL NETWORK OVERVIEW

Artificial Neural Network (ANN) is a mathematical model which is built on the basis of autonomous learning to imitate the structure and function of biological neural network. It can complete extremely complex pattern extraction or trend analysis by the analysis of a large number of complex data.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The BP neural network learning process is divided into two stages, namely information forward propagation stage and error back propagation stage. In the information forward transmission stage, the input samples enter from the input layer, and then transport to output layer after processing by hiding layer. If there is large error between the actual output of the output layer and the expected output, the network will conduct the error back propagation stage. In the error back propagation stage, the main task is correcting the threshold values of all neurons and the weights of the connection. Repeating information forward propagation and error back propagation until error value meet the requirements, the learning process is over. So the correct neural network model can be gotten.

3. DEEP WEB ENTITIES RECOGNITION METHOD BASED ON BP NEURAL NETWORK

In heterogeneous Web data sources, the same entity has different forms. Deep Web entities recognition can detect same entities from different data source, in order to lay a solid foundation for eliminating data redundancy and comparing the same entity. Combining the BP neural network with the Deep Web entities recognition can make full use of autonomous learning of ANN to reduce manual intervention and improve the accuracy of recognition. Deep Web entity recognition method which is base on BP neural network is divided into three main steps which are to divide the entities into blocks firstly, calculate semantic block similarity secondly, train entities recognition model thirdly. The detailed introduction is as followed.

3.1 Divide the entities into blocks

The entities are divided into some blocks according to an attribute or the combination of some attributes which have the relevant meaning in a semantic block. Through the analysis of the entities, it can be found that the entity not only contains the information of content, but also includes the metadata information which can influence the entities recognition.

It is necessary to preprocess the content of the entities' information before training the entities recognition model. First, the information of content is divided into blocks. Then the semantic block is processed as a text. Thirdly, the entity of semantic block is compared with other entities blocks. Finally, the similarities value of semantic block are calculated as the input of the BP neural network. This paper uses the partition method based on Web page layout tags for semantic structure.

In the Deep Web sites of a given field, the semantic blocks' contents of the entity are roughly similar, only the extremely individual semantic blocks are owned by a unique site. Therefore, all learning samples entities should be divided into semantic blocks and the divided blocks belong to a known attribute. So the attribute text labels can statistic corresponding to these semantic blocks. Then calculating union set and forming a set which includes all attributes text labels corresponding to these semantic blocks will be done. The set is recorded as A = $\{A_1, A_2,...,A_n\}$, there n means the number of text labels which are corresponded to the semantic blocks. After semantic block division of any entity in the data source, the entity can be represented by a semantic block set S = $\{S_1, S_2,... , S_n\}$, there S_i means attribute text labels of semantic block i. If attribute text label can not be found, S_i is empty.

3.2 Calculate semantic block similarity

After dividing the entity into blocks, since each block has the different contribution to entities recognition, the similarity between the semantic blocks and the entities must be calculated. It is used as the input of the neural network training. Specific steps include dividing the entity A to n blocks to form the corresponding semantic blocks set S={ $S_1, S_2,... , S_n$ }, calculating the similarities between each block with the entity B to get a group of similar values and expressing them as a set of vectors, record as T=($Sim(S_1,B)$,$Sim(S_2,B)$,…,$Sim(S_n,B)$), Sim(Sx,B) which means the similarity for any block of entity A with entity B.

Because of the poor structure of HTML documents, the concept of pattern matching will increase the complexity of the calculation. So in this paper entity A will be divided firstly. Then the similarities between the block with entity B will be calculated. During the calculation, different methods are used according to the different feature attribute types.

(1) Non-string type. For the properties of non-string type, such as numeric, currency type, the similarities can be calculated according to the range distance between them. The block of entity A is recorded as non-string t_1 and the one from entity B is recorded as non-string t_i (may extract more non-strings from entity B). The similarities of each non-string form of entity B with t_1 can be calculated by using formula (1), and the biggest similarity is taken as the corresponding neural

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

network input base on the visual characteristics of t_i.

$$sim(t_1,t_i)=1-\sqrt{\frac{(t_1-t)^2+(t_i-t)^2}{2}}\Big/t \quad (1)$$

there $\bar{t}$ is the average value of t_1 and t_i.

(2) String type. For the properties of string type, the similarity of semantic block and entity *B*'s string part can be calculated by using string edit distance method. The method is realized by counting the minimum operation number the source string m_i conversion to the target string m_j using insert, delete and replace. The formula [2] is used to convert to the similarity base on getting the edit distance.

$$sim_{edit}(m_i,m_j)=\max(0,\frac{\min(|m_i|,|m_j|)-D(m_i,m_j)}{\min(|m_i|,|m_j|)})^2 \quad (2)$$

$$D(mi,mj)=\min\begin{cases}D(m_{i-1},m_{j-1})+C(m_i\to m_j)\\ D(m_{i-1},m_j)+C(m_i\to\varepsilon)\\ D(m_i,m_{j-1})+C(\varepsilon\to m_j)\end{cases}$$

There m_i means string semantic block of entity A, m_j means characters in the text part entity *B*, D(m_i,m_j) means edit distance, C(m_i->m_j)means the cost to replace one character of m_j from m_i, C(m_i-> ε) means to delete one character of m_i, C(ε ->m_j) means to insert one character into m_j.

3.3 Train Entities Recognition Model

This step uses BP neural network to identify Deep Web entity. Firstly, the model of entity recognition needs to be established by BP neural network. The advantage of BP neural network is the weight of each attribute needn't to be calculated. Determining whether the Deep Web entities match or not is according to training inner relationship of the attributes by the self-study of the BP neural network. Then the sample entities are divided into two categories, namely the set of matching entities and the set of unmatched entities. The two types of entities are inputted to BP neural network respectively to train the weights and threshold, and the entity identify model is generated. Finally, the entity data that need to be match are inputted into the trained entity recognition model, then the result will be generated by the model. Specific training steps are as follows.

(1) The entity recognition model is established based on BP neural network. There are *n* nodes in the input layer, which are corresponding to the similarities of *n* semantic blocks and another entity respectively, if no corresponding similarity then enters 0. The output layer has two nodes. When output vector is (1, 0), it represents the entity is matched. While the output is (0, 1), it means the entity is unmatched. The number of hidden layer nodes is determined by formula (3).

$$n=\text{sqrt}(i+j)+d \quad (3)$$

there *n* means the number of hide layer, *i* means the number of input layer, *j* means the number of output layer and *d* is the constant between [1, 10].

(2) The sample entites are divided to matching entity set M (*r*) and the unmatched entity set U (*r*). Similarities of the blocks of entity A and entity B are calculated respectively which denote as Sim(A_i,*B*).

(3) The similarity Sim(A_i,*B*) of M(r) entity set is used as the input of the entity recognition model. The properties text labels of the semantic block are corresponded with the input node. The similarities value Sim(A_i,*B*) are input to the nodes which are corresponding to semantic blocks. If there is no corresponding node, the input value is 0 and the desired output is (1, 0). Neural network's error back propagation is used if there is higher error between the actual output and the desired output. The network will conduct an error back propagation phase to adjust neural network's weights and threshold until the neural network convergence and error precision meet the requirement.

(4) The similarity Sim(A_i,*B*) of U(r) entity set is used as an input, and the desired output is (0,1). The process is same with step (3), until the neural network convergence and the error precision meet the requirement;

(5) The similarity Sim(A_i,*B*) which is used to test is input to the completed entity recognition model and the output S is obtained.

(6) If the error range S in the target model is within the scope of M (*r)*, the tested entities will be matched. On the other hand the error range of S in the target model is within the scope of the U (*r*), the tested entities are unmatched. The target mode range is defined as the upper and lower limits of the training set. In the experiments of this paper, for example, the lower limit of target model M (*r*) is (0.852, 0.148), and the entities error range of S ([0.852, 1], [0, 0.148]) means the entities are matched. The upper limit of target model U (*r*) is (0.308, 0.692), and the entities error range of S ([0, 0.308], [0.692, 1]) means the entities are unmatched. If the error is between ([0.308, 0.852], [0.148, 0.692]) that means it belongs to neither matched set nor unmatched set, an expert is need for manual recognition.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. EXPERIMENTS

The experimental data is derived from two large website, namely Amazon and dangdang. Similar entities are obtained from the two websites by submitting queries to the query interface of the sites. These entities are divided into training set and testing set. Firstly, the entities are divided into semantic blocks. This experiment gets 12 attribute text labels corresponding to semantic blocks as a collection which denotes K = {ISBN, title, author, market price, price, discounts, press, paperback, barcode , ASIN, weight, size}. The similarities between each semantic block of any entity and another entity are calculated in the training set which are used as the input of training entity recognition model. The similarity of the training samples is shown in Table 1.

Table 1 The Similarity Of The Training Samples

Training Set	No	ISBN	title	author	market price	price	discounts	Press	paperback	barcode	ASIN	weight	size
Similarities of M(*r*)	1	0.93	0.92	0.90	0.89	0.91	0.88	0.79	0.85	0.75	0.73	0.70	0.00
	2	0.96	0.90	0.83	0.89	0.80	0.90	0.73	0.79	0.80	0.81	0.70	0.64
	3	0.92	0.89	0.87	0.94	0.80	0.77	0.78	0.82	0.70	0.72	0.00	0.77
	4	0.90	0.93	0.92	0.90	0.88	0.85	0.80	0.71	0.73	0.77	0.69	0.00
	5	0.97	0.91	0.94	0.90	0.92	0.88	0.81	0.75	0.80	0.69	0.71	0.40
Similarities of U(*r*)	6	0.21	0.20	0.19	0.34	0.22	0.15	0.08	0.17	0.11	0.06	0.08	0.01
	7	0.09	0.23	0.12	0.20	0.15	0.19	0.08	0.09	0.07	0.07	0.00	0.09
	8	0.17	0.23	0.18	0.22	0.17	0.15	0.09	0.11	0.10	0.07	0.11	0.10
	9	0.21	0.18	0.18	0.13	0.17	0.08	0.09	0.12	0.11	0.09	0.00	0.00
	10	0.29	0.18	0.12	0.10	0.16	0.09	0.07	0.13	0.12	0.04	0.00	0.08

Finally, the entity recognition model is established on the basis of BP neural network. Assume that the maximum number of iterations of the neural network is not limited until the error is to meet the requirements, the objective error is 0.001, and learning rate is 0.2. There are 12 input nodes, 2 output nodes of the neural network. The number of nodes in the hidden layer is 10 according to formula (3). After training the entity recognition model, the lower limit of the M(*r*) target mode is (0.852, 0.148), the upper limit of U(*r*) target mode is (0.308, 0.692). A pair of matched entity *A* and entity *B* and a pair of unmatched entity A and entity C which are got in the test set are used to test. Entity A is divided to blocks and the similarities are computed respectively with entity B and entity C. The results of similarities computation are shown in Table 2. The test result of the entity A and entity B is (0.962, 0.038) which is located in the range of target mode (0.852, 1] [0, 0.148]) range. So the entity A and entity B match. The test result of entity A and entity C is (0.06, 0.994) which is located in range of the target mode ([0, 0.308] [0.692, 1]). So the entity A and C is unmatched.

Table 2 Similarities Of The Test Entities

	ISBN	title	author	market price	price	discounts	press	paperback	barcode	ASIN	weight	size
Similarities of semantic block of A and entity B	0.94	0.93	0.95	0.90	0.88	0.85	0.89	0.81	0.77	0.14	0.79	0.00
Similarities of semantic block of A and entity C	0.19	0.23	0.25	0.11	0.30	0.12	0.09	0.10	0.07	0.15	0.06	0.05

With the increasing of numbers of samples, the accuracy of the model will increase correspondingly. But the amount of training samples is too much which will result time complexity increased. This paper gives a method by the evaluation index of the F value to measure. F value is calculated by formula (4). Performance of this paper to compare with the method given by literature [5, 6] is shown in Figure 1. F value of the method which is given by literature [5] is only 77% and F value of the method which is given by literature [6] is 83%. But F value of the method which is given by this paper is 93%. The experimental result indicates that the proposed method in this paper has better performance.

F = (accuracy rate * recall rate * 2) / (accuracy rate+ recall rate) (4)

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

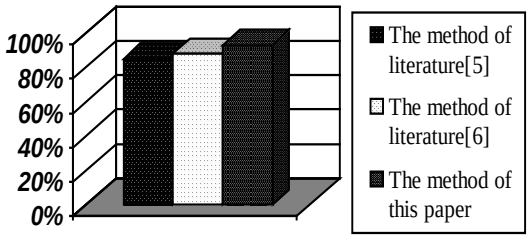

Figure 1 F Value Comparison

5. CONCLUSION

The development on great wealth of Internet resources requests higher on the effectiveness and performance of entity recognition. This paper makes full use of the advantages of the BP neural network. A Deep Web entity matching method is proposed on the basis of BP neural network. The method without the premise of pattern match reduces the complexity of the entity recognition by entities blocks while the similarities of each the semantic block and another entity are used as the input of the neural network training. The performance of the entities recognition is improved by the independent training of BP neural network. The contributions of the proposed method are to improve the accuracy and the efficiency of Deep Web entity recognition, at the same time reduce the manual intervention and solve the problem of low automatic level. The above contributions are verified via the experiments.

ACKNOWLEDGMENTS

This work was supported in part by Social Science Youth Foundation of Ministry of Education of China under Grant Nos. 12YJCZH048, the Natural Science Foundation of Liaoning Province of China under Grant Nos. 20102083 and Hundred, Thousand and Ten Thousand Talent Project of Liaoning Province of China.

REFRENCES:

[1] WANG YAN, SONG BAO YAN, ZHANG JIA YANG et al. Deep Web query interface identification approach based on label coding [J]. Journal of Computer Applications, 2011, 31(5): 1351-1354.

[2] TEJADA SHEILA, KNIBLOCK CRAIG A, MINTON STEVEN. Learning domain-independent string transformation weights for high accuracy object identification [C]//Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM press, 2002: 350-359.

[3] CUN JIAN JUN, XIAO HONG YU, DING LI XIN. Distance-Based Adaptive Record Matching for Web Databases [J]. Journal of Wuhan University, 2012, 58(1): 89-94.

[4] SARAWAGI SUNITA, BHAMIDIPATY ANURADHA. Interactive deduplication using active learning [C]//Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. New York: ACM press, 2002: 269-278.

[5] LI WEN SYAN, CLIFTON CHRIS. SEMINT: A Tool for Identifying Attribute Correspondences in Heterogeneous Database Using Neural Networks [J]. Data and Knowledge Engineering. 2000, 33: 49-84.

[6] QIANG BAO HUA, CHEN LING, YU JIAN QIAO, et al. Research on Attribute Matching Approach Based on Neural Network [J]. Journal of Computer Science, 2006, 33(1): 249-251.