Duplicate identification model for deep web

Liu Linan Kou Yue Sun Gaoshang Shen Derong Yu Ge

(College of Information Science and Engineering, Northeastern University, Shenyang 110004, China)

Abstract: A duplicate identification model is presented to deal with semi-structured or unstructured data extracted from multiple data sources in the deep web. First, the extracted data is generated to the entity records in the data preprocessing module, and then, in the heterogeneous records processing module it calculates the similarity degree of the entity records to obtain the duplicate records based on the weights calculated in the homogeneous records processing module. Unlike traditional methods, the proposed approach is implemented without schema matching in advance. And multiple estimators with selective algorithms are adopted to reach a better matching efficiency. The experimental results show that the duplicate identification model is feasible and efficient.

Key words: duplicate records; deep web; data cleaning; semi-structured data

The duplicate identification problem is a core problem arising in data cleaning, where different records refer to the same real-world entity. However, there are some limitations in current duplicate identification methods[1] which are mostly associated with relational models and based on schema matching[2-3]. In this paper, we focus on duplicate identification for the deep web by analyzing the results returned from multiple data sources.

1 Duplicate Identification Model for Deep Web

The duplicate identification model is used to find the duplicates based on XML with unknown attributes and schema information. Fig. 1 shows the framework of the model. The related definitions are described as follows:

Definition 1 Attribute value r_{ij} is the attribute value of an entity.

Definition 2 Entity record O_{mi} is a result record extracted from data sources, and composed of all its attribute values, where m and i are the m-th data source and the i-th entity record, respectively; and j is the j-th attribute value of the entity record.

Definition 3 Homogeneous records are the results extracted from the same data source. They have the same schema (i. e. DOM tree structure) and attribute names.

Definition 4 Heterogeneous records are the results extracted from different data sources. They have different schemas and attribute names.

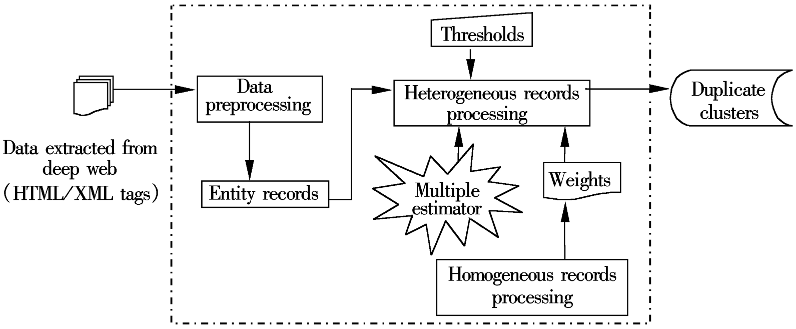
Data extracted from deep web (HTML/XML tags)
Data preprocessing
Entity records
Thresholds
Heterogeneous records processing
Multiple estimator
Weights
Homogeneous records processing
Duplicate clusters

Fig. 1 Framework of duplicate identification model for deep web

2 Data Preprocessing Module

Before comparing the extracted results, parsing XML or HTML data records as the following two parts is necessary: creating DOM trees and generating entity records.

Creating DOM tree: The approach uses DOM to parse XML documents as a DOM tree. So, it can manipulate the DOM trees and the nodes instead of manipulating XML files.

Generating entity records: The DOM tree structures of query results from different sites are different, and the schema information of each node in the DOM trees is also unknown. So, all the records need to be transformed into a unified schema. For the DOM tree of each record, it extracts the text of the leaf nodes and then builds a set of them. The set is represented as $O_{mi} = \{r_{mi1}, r_{mi2}, r_{mi3}, \dots\}$.

3 Heterogeneous Records Processing Module

The purpose of the heterogeneous records processing module is to find the duplicates of the extracted results which describe the same real world entity. The heterogeneous records processing module calculates the similarity of two heterogeneous records by employing a set of special estimators distinguished by attribute types. So each estimator only deals with a certain type of items of the entity records.

3. 1 Multiple estimators

In order to reduce the comparison times, it only compares the attribute values of the same type. The basic idea is that

Received 2008-04-15.

Biographies: Liu Linan (1983—), female, graduate; Shen Derong (corresponding author), female, doctor, professor, shenderong@ ise. neu. edu. cn.

Foundation item: The National Natural Science Foundation of China (No. 60673139).

Citation: Liu Linan, Kou Yue, Sun Gaoshang, et al. Duplicate identification model for deep web[J]. Journal of Southeast University (English Edition), 2008, 24(3): 315 - 317.

the attribute values of each entity record are divided into different blocks according to their types, such as text, numeric and so on. In this way, one estimator can conduct an efficient evaluation on one type of block.

3.1.1 Similarity evaluation

Given two entity records, a score must be assigned to them to represent their approximate semantic distance. In our model, each estimator can only calculate the similarity of one block in which the evaluated attribute values refer to the same type. Specific identification strategies can be tailored by combining the selected matching algorithm on a domain's demand.

In our model, multiple estimators with selective algorithms are adopted to reach better matching efficiency. The implemented estimators cover more data types, such as text, numeric, etc. And in each estimator, there are various algorithms to calculate similarity. In the following, two implemented estimators are described in detail.

- Text estimator: This estimator calculates the similarities of character-based blocks, by means of the following three algorithms: Q-grams[4], Affine gap distance[5], and Jaro distance[6].
- Numeric estimator: This estimator calculates the similarities of numeric blocks, such as ISBN. In our model, two algorithms given in the numeric estimator are as follows: 1) Accurate distance: When two numeric data n_1 and n_2 are the same, they can precisely judge the two entities as referring to the same real-world object, such as the ISBN; thus, if two numbers are exactly the same, the similarity between them is 1, otherwise, it is 0; 2) Range distance: When two numeric data n_1 and n_2 are similar, their margin is less than a threshold δ. And the similarity between two numeric data n_1 and n_2 is $s(n_1, n_2) = 1 - \sqrt{\frac{(n_1 - \bar{n})^2 + (n_2 - \bar{n})^2}{2}} / \bar{n}$. Where n_1 and n_2 are the values of the two numeric data, and $\bar{n}$ is the average of n_1 and n_2.

As described above, two kinds of estimators are provided in our model. Not only can estimators be combined flexibly, but also the algorithm can be selected based on specified needs. In the future, new estimators can be flexibly added to our model, and also new algorithms will be easily extensible.

3.1.2 Similarity combination

Duplicates are identified mainly based on calculating the similarity between entity records, and we combine the similarity results from different estimators as follows:

1) Aggregation of similarity results: In this step, similarity values calculated by multiple estimators are aggregated to form a combined similarity value for each pair of entity records, which is denoted as $s_{\text{est}} = \sum \omega_{\text{esti-name}} \times s_{\text{esti-name}}(r_i, r_j)$, where S_{est} is the aggregation of the similarity values from multiple estimators; $\omega_{\text{esti-name}}$ is the weight of an estimator, and these weights are computed in the homogeneous records processing module described in section 4, and $s_{\text{esti-name}}(r_i, r_j)$ is the similarity value calculated by an estimator.

2) Similarity evaluation: Each similarity value calculated by an estimator is based on a single data type, so each one is insufficient to describe the overall similarity of two records. Consequently, in the second step, all the attribute values of the entity records are considered as a whole block, and it calculates the similarity $S_{\text{tf.idf}}$ between two entity records by using the tf. idf algorithm.

3) Similarity combination: Finally, the similarity value between two entity records is the combination of S_{est} and $S_{\text{tf.idf}}$: $\text{sim}(r_i, r_j) = (s_{\text{est}} + s_{\text{tf.idf}})/2$.

3.2 Duplicates identification

After calculating the similarity of the heterogeneous records, the module should compare the similarity of records and obtain duplicates sets as follows:

1) Compare the initial entity records extracted from the deep web one by one, and then calculate the similarity degrees of the entity records.

2) Find the entity records pair with the highest similarity value as the two nodes of the undirected acyclic graph. Add nodes to the graph as above, until all the nodes are contained in the graphs.

3) The process of the consequent entity records extracted is based on the results of the initial extracted records, the graph created in 2) and the threshold of the specific domain given by the expert. For the consequent query results, first, they are compared with the representative nodes in the graphs(the node which has the most edges joined with it.). After comparing the similarities with all the representative nodes in the graphs created in 2), the node with the similarity value exceeding the threshold will be added to the graph. If all the similarity values are less than the threshold, a new graph will be created and the entity record will be added to it as a node. As a result, the nodes in a graph are duplicates.

4 Homogeneous Records Processing Module

The DOM tree structure of homogeneous records is the same as above. We use two kinds of homogeneous records to calculate the weights of different estimators. Similar homogeneous records are similar to each other but do not correspond to the same object; different homogeneous records have different attribute values and correspond to different entities.

According to the similar homogeneous records and the different homogeneous records, the weights of different estimators are calculated. The process is described as follows:

1) The calculation of similar homogeneous records: First, it calculates the similarity degree s_{ij} of corresponding nodes in the DOM trees by using multiple estimators described in section 3; secondly, it calculates the sum of the similarity degrees of the nodes with the same data types. And then it calculates the average of the sum, and generates the average similarity of each data type. For a set of homogeneous records, it calculates the weight of an estimator: $\omega_{\text{esti-name}_i} = 1 - (\sum S_{ij})/n$. And the weight of a data type in a specific domain also relies on all the homogeneous records, denoted as $\omega'_{\text{esti-name}} = \left(\sum_{i=0}^{n-1} \omega_{\text{esti-name}_i}\right)/n$. Where i is the i-th data source; $\omega_{\text{esti-namei}}$ is the weight of the estimator relied on in the records in the data source i; and n is the number of data sources used.

2) The calculation of different homogeneous records is similar to that of homogeneous records. We generate the

weights $\omega''_{\text{esti-name}}$ based on different homogeneous records.

Finally, the approach generates all the weights of the estimators based on similar homogeneous records and different homogeneous records together, and the formula is as $\omega_{\text{esti-name}} = \omega'_{\text{esti-name}} + \omega''_{\text{esti-name}}$.

5 Experiments

We adopt a comprehensive evaluation for the proposed duplicate identification model on 50 complex web databases over two domains: book and used car.

According to the matching thresholds in a specific domain given by the experts, there are two clusters of entity records: duplicates and non-duplicates. The similarity of the duplicates is mainly on [0. 4, 0. 7] and the similarity of non-duplicates is mainly on [0, 0. 2]. Fig. 2 and Fig. 3 show the results of identification with and without noisy attributes. We can see that the method does improve the overall identification accuracy, especially in the book domain. Therefore, removing the noisy attributes before identifying the duplicates can improve the accuracy and efficiency of identification.

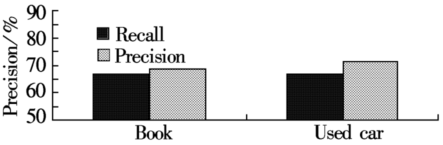

Fig. 2 Result of identification with noisy attributes

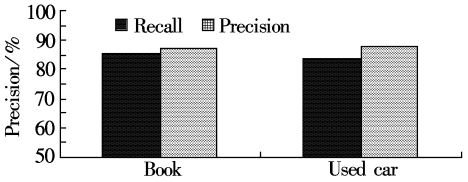

Fig. 3 Result of identification without noisy attributes

6 Conclusion

In this paper, we propose a duplicate identification model. In this model, we address the problem of semi-structured and unstructured data on the deep web. The problem is crucially important while identifying duplicates from different data sources on the deep web. Unlike traditional ones, it is implemented without schema matching. In the future, we will construct more estimators for our model, and improve the self-adaptability.

References

[1] Lee Mong Li, Hsu Wynne, Kothari Vijay. Cleaning the spurious links in data [J]. *IEEE Intelligent Systems*, 2004, **19**(2): 28 – 33.

[2] Ma Weiying. Instance-based schema matching for web databases by domain-specific query probing [C]//*Proceedings of the* 30*th VLDB Conference*. Toronto, Canada, 2004: 408 – 419.

[3] He Bin, Chang Kevin Chen-Chuan. Making holistic schema matching robust: an ensemble approach [C]//*Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data*. Chicago, Illinois, USA, 2005: 429 – 438.

[4] Ukkonen E. Approximate string matching with q-grams and maximal matches [J]. *Theoretical Computer Science*, 1992, **15**(2): 191 – 211.

[5] Waterman M S, Smith T F, Beyer W A. Some biological sequence metrics [J]. *Advances in Math*, 1976, **20**(4): 367 – 387.

[6] Jaro M A. Unimatch: a record linkage system: user's manual [R]. Washington, DC: US Bureau of the Census, 1976: 414 – 420.

一种 deep web 数据源下重复记录识别模型

刘丽楠 寇 月 孙高尚 申德荣 于 戈

(东北大学信息科学与工程学院,沈阳 110004)

摘要:使用 deep web 数据源下重复记录识别模型对从多个 deep web 数据源中抽取出来的半结构化和无结构化的数据进行处理. 首先,在数据预处理模块中将所抽取的数据生成实体记录的形式,然后,在异构记录处理模块中利用在同构记录处理模块所得到的权值,计算各实体记录的相似度,得到重复记录. 与传统的重复记录识别模型不同,所提方法是在模式匹配未知的前提下实现的;并且采用带有可选算法的多个相似度估算器以达到更好的匹配效率. 实验证明,该重复记录识别模型是可行且有效的.

关键词:重复记录;deep web;数据清洗;半结构化数据

中图分类号:TP311