ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 8(30), DOI: 10.17485/ijst/2015/v8i30/88013, November 2015 ISSN (Online) : 0974-5645

Automatic Extraction of Objects and their Attributes from Semi-Structured Web Tables for E-commerce Tasks

Yerzhan Baiburin and Aliya Nugumanova*

Department of Information Technologies, D. Serikbayev East Kazakhstan State Technical University, Ust Kamenogorsk, Kazakhstan; ebaiburin@gmail.com, yalisha@yandex.kz

Abstract

Most business Web documents provide key information for decision-making in the form of text tables. High-speed retrieval and analysis of such tables is of great interest to the business and at the same time is a major challenge for researchers. The problem lies in the fact that text tables in Web documents are rarely self-described, i.e. contain no data schemas. In this paper we consider the special case of the problem, limited by the scope of e-commerce. The main objects of e-commerce are goods, works, services and other objects represented in text tables as sets of characteristics (attributes). In fact, these text tables compactly map domain objects into sets of "attribute-value" pairs. Thus, the extraction of meaningful information from the tables can be interpreted as extraction of sets of "attribute-value" pairs. We need to interpret these attributes correctly using a domain knowledge base, since the search engine does not have any information about neither the nature of these attributes, nor their internal relations. In this paper we use as knowledge base semi-extensible e-commerce ontology, based on economic classifiers of the Kazakhstan unified system of classification and coding of technical, economic and social information.

Keywords: E-Commerce, Information Retrieval, Ontology-Based Text Mining, Table Understanding

1. Introduction

Web text tables are important research objects because they can be treated as databases, although they are not databases by their nature. These tables contain markup tags and can be classified as semi-structured forms of data because their markup is not relational. Text tables cannot be fitted to the relational format, so they require fundamentally new data processing technologies.

The goal of this research is to develop the technology of automatic information extraction from text tables in the Web, related to E-commerce area. The main E-commerce objects are goods, works, services and other objects presented in text tables as sets of characteristics (attributes). In fact, these text tables compactly map domain objects into sets of "attribute-value" pairs. The difficulty is that such a mapping is not unique. In particular, the text table objects of the same class may have different sets of attributes; one table can display the attributes of objects of different classes; attributes of one object can be distributed over multiple tables. The search machine knows only that the tables store attributes of objects, but it knows nothing about them and, moreover, it does not know how they relate to each other. Consequently, the information extraction from text tables is an inverse problem that consists of extraction of "attribute-value" pairs and representing them as a structured domain objects. We divide this inverse problem into 3 tasks:

- Extraction of schemaless data from semi-structured text tables (i.e. sets of "attribute-value" pairs).
- Loading the extracted data into "key-value" storage.
- Semantic interpretation and transformation of the loaded data into a structured object representation.

According to the set goal the structure of the paper can be presented as follows. The next (second) section introduces related works on table analysis and table understanding. The third section presents a novel attribute-centric approach. The fourth section includes

* Author for correspondence

a description of our general methodology and methods used in it to extract and interpret valuable data from text tables. The fifth section gives a description of the experimental (test) part of the work. Conclusion and future work plans can be found in the section 6. The last section contains a list of references.

2. Related Works

Research[1] is one of the early works on the problem of information extraction from web tables. It describes a 5-modular software system. The first module processes hypertext and searches tables. The second module filters detected tables, eliminating meaningless ones, i.e. layout tables. The third module detects table structure. The fourth module differentiates tables' cells according to their purpose and interprets their contents. The fifth module is responsible for the results presentation. At the output, the system provides a sequence of "attribute-value" pairs captured form tables' cells. The capturing algorithm is based on the analysis of cells similarity (homogeneity): cells, similar in type and semantics, are recognized as the values of the same attribute.

Ideas proposed in work[1] have been further developed in research[2]. The authors of this research have extracted about 14 billion raw tables from documents indexed by Google. Then they selected and recognized 15 million tables containing the high-quality semantic relations. The results of this research have been improved a few years later in research[3]. The main weakness of the early researches was that their authors only used the local document information for recognition of objects represented in the tables, although the recognition accuracy can be improved by using of external sources. This weakness was eliminated in later researches that took into account a variety of knowledge sources, including Wikitology[4], DBPedia[5], Freebase[6], Yago[7], a training corpus[8], an automatic ontology[9], Open Linked Data[10]. In[11,12] the tool Google Fusion Tables is used.

The basis of all these later works is entity-centric approach. As the name implies, the approach aims at identifying of domain entities and relations in the tables. For example, research[7] proposes the following typical procedure implemented entity-centric approach. There is a hierarchical directory, which represents domain entities, their types (classes) and relations. It is required to annotate tables using the specified directory in accordance with specified rules. The first rule: each column of annotated table is assigned to one or more directory classes as possible. The second rule: each pair of columns is assigned to directory relation as possible. The third rule: each cell is assigned to the identifier of directory entity as possible. The described approach greatly enhances the tables' recognition (annotation); however, it is poorly applied in the following cases:

- There is no information about analyzed entities in external sources of knowledge (for example for specific domains).
- Knowledge bases are not available in the public space or do not exist at all (for example, a knowledge base in the Kazakh language).
- It is difficult to determine precisely from the context which entities are in question.

The latter problem is very characteristic for extracting information from text tables. In work[13] it is defined as a protagonist detecting problem. To solve it, the authors search the protagonists' candidates as N-grams. According to them there are 3 possible places in the document that may contain information about protagonist: 1) within a table, often in such frequently used columns as "name" or "model"; 2) in the text or in the title; 3) within the document in anchor tags. A significant limitation of the technique is that it can be applied only to the tables presented a single protagonist.

3. Attribute-Centric Approach to Extracting Information from Text Tables

As can be seen from the related works analysis, the entity-centric approach to information retrieval, dominant in modern researches, has serious limitations. The novelty of our research is, firstly, in the rejection of entity-centric approach in favor of an attribute-centric one, and secondly, in the rejection of universal knowledge bases in favor of domain ontology. Attribute-centric approach is based on simple idea that all that a search machine can extract from a text table is a set of "attribute-value" pairs and it is the entire set rather than single pairs that provides valuable information for the objects identification. We can easily illustrate this idea by the following example. Domain expert analyze two tables. One presents the characteristics of hard disks and the second table presents characteristics of processors. Obviously, the domain expert can determine at a first glance which devices are

presented in the tables. The machine can also identify objects in a similar way, if it has the same knowledge as the expert, i.e. relies on the developed domain ontology.

Attribute-centric approach is naturally realized by ELT (Extract Load Transform) model for data collection and NoSQL model for data storage[14]. At the first stage (Extract), all "attribute-value" pairs regardless of their objects are extracted from the tables. Before extraction, it is necessary to recognize the physical structure of the table and to determine how "attribute-value" pairs are arranged: vertically, horizontally, or in other way.

In the second stage (Load) extracted "attribute-value" pairs are loaded into a simple NoSQL "key-value" storage. This allows not caring about the stored data types and the relations, as they are not known yet. Additional advantages of NoSQL are horizontal scalability and high-speed processing of large amounts of data, which is important for e-commerce tasks.

It is only the third stage (Transform) when semantic interpretation of extracted attributes and their mapping into domain objects are being performed. To perform semantic interpretation special expandable domain ontology is used, in contrast to[4–12] where researchers used open sources of knowledge. In this research, we use a semi-automated ontology of e-commerce.

4. Step-by-Step Description of the Proposed Attribute-Centric Approach

4.1 Extracti`ng Data from Semi-Structured Tables

Extracting data from text tables is a multi-stage process, starting with the physical detection of the table and ending with the recognition of its logical structure[15]. It consists of 4 steps:

- Table detection.
- Analysis of table's physical structure (i.e. table segmentation including cells, rows and columns recognition in the table structure).
- Analysis of table's functional structure (i.e. defining of a role of each cell, which indicates attribute or value is stored in this cell).
- Analysis of table's logical structure (i.e. detecting of logical sets of data, i.e. "attribute-value" sets).

As noted in[16] the "physical structure" term is used to refer to obvious features of table's coding such as markers of begin of the table, of rows, of columns, of cells, etc. In web documents such a mark-up is performed usually with HTML-tags. Therefore, the first two steps (table detection and its physical structure analysis) are purely technical problems, which can be simply implemented by HTML-parser[17]. The third and the fourth steps of data extraction are more interesting and challengeable from a point of view of a researcher. The only difficulty of the physical analysis is that often in web documents table tags are used for text layout. In this case, further investigation, whether a table contains important information or it should be excluded from consideration, is required[18,19]. Such an investigation should use the discriminant analysis methods, which allow distinguishing meaningful and meaningless tables based on domain keywords occurrence analysis[20].

The aim of functional analysis is to classify table cells according to the functions (roles) they perform. Totally two functions can be distinguished: an attribute container and a value container. In the simplest case, cells, which are column headers, perform the role of attribute containers and remaining cells in the columns perform the role of value containers; the rows represent data tuples. A more complex sample is in case when attributes and their values are placed in the rows, not in the columns; tuple corresponds to group of rows (super rows), not to one row. This step is the most important in data recognition as everything depends on whether data attributes are correctly determined. Domain ontology plays an important role in this step.

Besides domain ontology, we use a classifiers committee, which allows defining the class (function) of each cell by voting. Each algorithm in the classifiers committee uses its own feature space and its own decision rule, that allows taking into account as many different classification features as possible. Classification features include positions of the cells in the table, theirs format and content, as well as less obvious features such as lexical and/or semantic proximity of cells. Cells containing numeric or alphanumeric values of the same attribute have a lexical proximity. Lexical proximity is estimated using metrics based on regular expressions. Cells which values participate in different semantic relations, such as the genus-species, whole-part, class-instance relations, have a semantic proximity. Semantic proximity is estimated by calculating the distance between attributes in the semantic graph.

The next step is the logical structure analysis. Its aim is to join each value cell to the corresponding attribute cell. Thus, cells are being grouped in "attribute-value" pairs, which should be extracted together[15]. Work[1] describes the following simple algorithm for pairs determining. Each value cell attaches to the attribute cell above and/or beside it. If an attribute does not exist in the row or column, then the value cell attaches to the first cell in its column. Attributes for such orphaned values are being sought in the table title or in the document context. This algorithm does not use lexical-semantic connections between the cells, identified in the previous step. Meanwhile, this information is very useful for the logical analysis of a table. For example, if you know that a group of similar cells semantically related to the parent cell, it is obvious that the parent cell is an attribute and a group of similar cells are values of this attribute. The result of logical analysis is sets of "attribute-value" pairs grouped into data tuples. Each tuple, if possible, should contain a complete set of attributes and their values.

4.2 Loading Data into Cloud "Key-Value" Storage

The distributed computing in a cloud is one of the most promising approaches to the processing and storage of big volumes of data[21,22]. The cloud is based on a distributed infrastructure consisting of pool of configurable computing resources and data storages. As data storage, the cloud uses "key-value" structure instead of relational database. Such a choice in favor of non-relational structure is explained by the fact that in distributed environment relational database performance is significantly reduced.

Non-relational "key-value" storage has a very important advantage: it allows developers to organize storage of schemaless data. This is useful for the development of systems that require open architecture allowing the further scaling without changing the code or data structures. The main difficulty in such a decision is to determine the hashing mechanism, i.e. algorithm determining the nodes to store the information. Work[23] proposes a fundamentally new method of storing records, called hyperspace hashing.

The method represents each record as an independent multi-dimensional space where the axes are the attributes of the record. Thus, the record is attached to the node according to their coordinates obtained by hashing each of record's attribute on the corresponding axes. According to the authors, this method will allow to solve most common tasks from 2 to 13 times faster than traditional hashing methods used in such popular NoSQL systems like MongoDB, Redis, Cassandra etc. The main problem of the proposed method is the so-called "curse of dimensionality" phenomenon associated with the fact that the resources for the data storage and processing increases exponentially when adding another attribute. The authors solve this problem dividing the space into subspaces. In this work, we propose the interpretation of the multidimensional space formed by the records' attributes as a tensor, which can be replaced by a set of the associated conventional matrices. The tensor decomposition is well known in computational mathematics, but "tensor train" a new tensor format has been presented just recently[24]. The "tensor train" is the decomposition, which reduces the original tensor of large dimension N to N 3-dimensional tensors. It allows to significant compressing large dimension data and speed up the computation.

4.3 Transformation of Data into a Structured Object Representation

In this research we propose MapReduce technology[20,25] for transformation of schemaless "key-value" data into structured relationships more suitable for analysis. Such transformations can be performed by conventional user-defined functions; however, MapReduce paradigm supposes more flexible and efficient transformations. In general, the MapReduce algorithm consists of three steps: "Map", "Group" and "Reduce". "Group" step is internal; "Map" and "Reduce" steps are implemented by the user. "Map" converts an input pair {key: value} to a set of intermediate pairs. "Group" is performed within the MapReduce model without user's care. This step combines all the values with the same key and returns a pair {key: value list}. "Reduce" function takes pairs {key: value list} and wraps a list of values into a single value. Thus, the result of the MapReduce algorithm is minimized pairs {key: value}. The main advantage of the MapReduce algorithm is parallelizability, thus it is able to handle huge amounts of data on a set of cores/processors/machines.

The task of data transforming fits well with the MapReduce paradigm. We use the MapReduce algorithm to perform a convolution of attributes on the values (to search synonyms of the same attribute), the convolution of objects on attributes (for domain objects allocation) and a convolution of documents on domain objects. The transformation of data into a structured representation

is based on the e-commerce ontology. E-commerce is a sphere that needs the heavyweight ontology with broad semantic space. For creating and enriching this ontology, we use a set of ready classifiers belonging to a unified system of technical, economic and social information classification and coding.

5. Experiments

For carrying out experiments, http://goszakup.gov.kz/ government site has been chosen. Learning collection comprising 30000 lots represented by tables published on this web site has been automatically generated for its description. List of 42571 attributes has been retrieved as a result of transformation this collection. E-commerce classifiers have been applied to the list which let distribute these attributes to the objects. After learning stage our search engine extracted 10000 lots from the given site. Extracted data were loaded into NoSQL data storage and then transformed with use of learning data and e-commerce classifiers.

6. Conclusion and Future Work

Results of the given approach were offered for analysis to independent business experts. Experts noted following advantages of automatic data extraction: rapidity and topicality. However they noticed that results highly depend on source collection of lots and more representative collection of lots is required for reaching full reflection of attribute-object terminology. However in this case dimension of data used in the given approach will require higher capacity computing power and improving of used e-commerce ontology; further work will be related to this subject.

7. References

1. Chen H, Tsai S, Tsai J. Mining tables from large scale HTML texts. Proceedings of the 18th Conference on Computational Linguistics. 2000; 1. p. 166–72.
2. Cafarella M, Wu E, Halevy A, Zhang Y, Wang D. Web tables: Exploring the power of tables on the web. Proceedings of the VLDB. 2008; 1:538–49.
3. Yakout M, Ganjam K, Chakrabarti K, Chaudhuri S. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. Proceedings of the 2012 ACM SIGMOD. 2012; 97–108.
4. Syed Z, Finin T, Mulwad V, Joshi A. Exploiting a Web of semantic data for interpreting tables. Proceedings of the 2nd Web Science Conference. 2010.
5. Auer S, Bizer C, Kobilarov G, Lehmann J, Cyganiak R, Ives Z. DBPEDIA: A nucleus for a web of open data. Proceedings of the ISWC/ASWC. 2007; 4825:722–35.
6. Bollacker, K, Evans C, Paritosh P, Sturge T, Taylor J. Freebase: a collaboratively created graph database for structuring human knowledge. Proceedings of the SIGMOD. 2008; 1247–50.
7. Limaye G, Sarawagi S, Chakrabarti S. Annotating and searching web tables using entities, types and relationships. Proceedings of the VLDB. 2010; 3(1-2):1338–47.
8. Venetis P, Halevy AY, Madhavan J, Pasca M, Shen W, Wu F, Miao G, Wu C. Recovering semantics of tables on the Web. Proceedings of the VLDB. 2011 Jun; 4(9):528–38
9. Wang J. Wang H, Wang Zh, Zhu K. Understanding tables on the web. Proceedings of the 31st International Conference on Conceptual Modeling; Springer-Verlag; 2012. p. 141–55.
10. Ni Y, Zhang L, Qiu Z, Wang C. Enhancing the open-domain classification of named entity using Linked Open Data. Proceedings of the 9th International Semantic Web Conference on The semantic Web; 2010. Springer-Verlag. 2010; 6496. p. 566–81.
11. Guo X, Chen Y, Chen J, Du X. ITEM: Extract and integrate entities from tabular data to RDF knowledge base. Proceedings of the 13th Asia-Pacific Web Conference on Web Technologies and Applications; 2011. Berlin, Heidelberg. Springer-Verlag. 2011; 6612. p. 400–11.
12. Gonzalez H, Halevy A, Jensen CS, Langen A, Madhavan J, Shapley R, Shen W, Goldberg-Kidon J. Google Fusion Tables: web-centered data management and collaboration. Proceedings of the 2010 International Conference on Management of Data; New York, USA. 2010. p. 1061–6.
13. Crestan E, Pantel P. Web-scale knowledge extraction from semi-structured tables. Proceedings of the 19th International Conference on WWW; 2010. p. 1081–2.
14. Jacobs A. The pathologies of big data. Communications of the ACM. 2009; 52(8):36–44.
15. Silva AC, Jorge A, Torgo L. Design of an end-to-end method to extract information from tables. International Journal of Document Analysis and Recognition. 2006 Jun; 8(2):144–71.
16. Richard Z, Blostein D, Cordy J. Decision-based specification and comparison of table recognition algorithms. Machine Learning in Document Analysis and Recognition. 2008; 90:71–103.
17. Huy HP, Kawamura T, Hasegawa T. How to make web sites talk together: web service solution. Special Interest Tracks and Posters of the 14th International Conference on WWW; 2005. p. 850–5.
18. Wang Y, Hu J. Detecting tables in HTML documents. Lecture Notes in Computer Science. 2002; 2423:249–60.
19. Hurst M. Layout and language: Challenges for table understanding on the Web. Proceedings of the 1st Int'l Workshop on Web Document Analysis; Seattle, WA. 2001. p. 27–30.

20. Nugumanova A, Novosselov A, Baiburin Y, Karimov A. Automatic keywords extraction from the domain texts: Implementation of the algorithm based on the MapReduce model. Proceedings of the International Conference on Current Trends in Information Technology. 2013 Dec 11-12. p. 186–9.
21. Antonopoulus N, Gillam L. Cloud Computing: Principles, Systems and Applications. Springer; 2010.
22. Ahuja SP, Mani S. The state of high performance computing in the cloud. Journal of Emerging Trends in Computing and Information Sciences. 2012; 3(2):262–6.
23. Escriva R, Wong B, Sirer EG. HyperDex: A distributed, searchable key-value store. ACM SIGCOMM Computer Communication Review. 2012; 42(4):25–36.
24. Oseledets I. Tensor-train decomposition. SIAM Journal on Scientific Computing. 2012; 33(5):229–317.
25. Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. Communications of the ACM. 2008 Jan; 51(1):107–13.