Software Conception for Semantic Interpretation of Spreadsheet Data

Nikita Dorodnykh[0000-0001-7794-4462] and Aleksandr Yurin[0000-0001-9089-5730]

Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of
Russian Academy of Sciences, Lermontov St. 134, Irkutsk, Russia
iskander@icc.ru

Abstract. Spreadsheet data are a valuable source of knowledge in data science and business intelligence applications. However, most commonly, spreadsheets are not accompanied by explicit semantics which are necessary for a machine interpretation of their contents. Information accumulated in spreadsheets is often poorly structured and not standardized. Analysis of this tabular data requires its preliminary extraction and transformation to a structured representation with the subsequent recovering of the implicit semantics. In this paper, we consider a conception of software for semantic interpretation of spreadsheet data in XLSX format and the linked data generation in the form of RDF triplets. We suggest to use DBpedia as a global taxonomy of concepts for understanding and conceptualizing the content of tables. A list of the main functions of this software is also provided. Issues of the further software development are discussed.

Keywords: Semantic Interpretation, Spreadsheet Data Analysis, Canonical Table, Linked Data, DBpedia

1 Introduction

Today, a large volume of arbitrary tables presented in the spreadsheet-like formats (HTML, EXCEL and CSV) has been accumulated worldwide [8]. Arbitrary tables are a valuable data source in business intelligence and data-driven research. However, the information accumulated in them is often poorly structured and not standardized. Typically, they are not accompanied by explicit semantics necessary for machines to interpret their content in the same way as was intended by their creator. In this paper we consider the problem of recovering implicit semantics of tabular data content that had already been extracted and reduced to the canonical (relational) form. This prob-lem is an important task in the field of Semantic Web.

We propose a conception of software for semantic interpretation of spreadsheets presented in the XLSX format. The interpretation associates the table content with external knowledge (concepts of global or subject taxonomies and vocabularies). Linked Open Data (LOD) cloud [2] including global taxonomies (e.g. DBpedia, Wikidata, YAGO, etc.) can be utilized as external vocabularies for these purposes. Usually, due to tabular layout properties, the tabular data content can be easily di-vided into sets of values, so that all values in each set belong to one unknown category. Each value from the set (hyponym) individually can be associated with one or more categories (hyperonyms) from a certain global taxonomy. Then, its environment (hyponym from its set) can be used to avoid ambiguity. As a result, it helps to restore the category that most adequately describes all values of the set.

We suggest to use DBpedia [4] as a global taxonomy of concepts for understanding and conceptualizing the content of spreadsheets. DBpedia is one of the most well-known and major projects aimed at extracting structured information from data cre-ated as part of the Wikipedia project and publishing it in the form of linked data sets.

Thus, the main result of the semantic interpretation is tabular data enriched with references to DBpedia. This data is presented in the RDF (Resource Description Framework) [12] format and describes concepts and relationships of a subject do-main.

2 Related Works

In the last two decades, methodological foundations of semantic interpretation of the data extracted from spreadsheets have been actively developed. These methods are intended to link table content with external domain concepts (ontologies or global taxonomies) using the following methods: knowledge (classes and relationships) automatically collected from the Web [21], automatic annotation of natural science spreadsheets using a combination of structural properties of the tables and external vocabularies [5], natural language processing, including the named entity linking [3, 23, 24] and the word embedding [7], as well as various general-purpose ontologies of Linked Open Data [6, 9-11, 13, 14] or proprietary global taxonomy, Probase [22]. Basically, they rely on the analysis of the natural language content of tables and their context, neglecting their layout and style features. In practice, this is not enough. In particular, some information can be conveyed by various fonts or colors.

Analysis of the up to date research in this area shows the absence of universal li-braries and algorithms for transforming tabular data and interpreting their content. Each approach has its own pros and cons; and usually solves some particular prob-lems and processes only certain data sets. It is also important to highlight that existing approaches mainly target programmers and focus on linking the cell contents of the data block in a table. However, as a rule, table headers are ignored or used only as an additional context in the semantic interpretation process. This paper is aimed to elimi-nate these shortcomings.

3 Software Conception

We suggest a conception of software for the semantic interpretation of tables; this interpretation is based on combining natural language processing techniques and ex-ternal vocabularies. First of all, the extracted data items should be separated into two types: numeric and non-numeric via the named entity recognition. Then data items are linked with concepts (classes, instances, and properties) of an external vocabulary. To examine this approach we used DBpedia but it can be extended by other vocabularies in the future.

Our implementation of this conception supports the following functionality:

- *arbitrary table canonicalization* – analyzing spreadsheets and its transforming from an arbitrary form to a canonical (relational) one by using TabbyXL tool [19];
- *canonical table preprocessing* – tabular data cleansing and formatting in accordance with DBpedia naming conventions, and creating queries to DBpedia in SPARQL language [17];
- *canonical table interpretation* – linking extracted data items of a table with DBpedia classes, instances, and properties;
- *linked data generation* – RDF code generation for linked canonical table cells.

In this paper, we consider table processing only with quantitative indicators that describe different literal data values such as dates or numerical characteristics of an object or its properties. For example, such tables very often reflect data of any measurements.

The four functions mentioned are implemented as separate interconnected modules of web-based software. Next, we consider these functions in detail.

3.1 Arbitrary Table Canonicalization

The proposed semantic interpretation of tabular data is focused on processing only canonical spreadsheets. Canonical tables are a formalized (unified) representation of some subset of arbitrary spreadsheets [20]. The canonization (normalization) process includes: a role analysis (extracting data units from tabular content and comparing them with functional roles) and a structural analysis (restoring relationships between tabular data units).

Thus, let us formally define the canonical table as a set of three columns:

$$CT = \{DATA, RowHeading, ColumnHeading\},$$

where *DATA* is a data block that describes literal data values (named "entries") belonging to the same datatype (e.g., numerical, textual, etc.); *RowHeading* is a set of row labels of the category (left headers from a source arbitrary table); *ColumnHeading* is a set of column labels of the category (top headers from a source arbitrary table). *RowHeading* and *ColumnHeading* columns represent a block of categories that describe data in *DATA* column. The values in cells for heading blocks can be separated by the "|" symbol to divide categories into subcategories. Thus, the canonicalized table denotes hierarchical relationships between categories (headings) in a source arbitrary table.

The detailed description of obtaining canonical tables on the basis of analysis of arbitrary spreadsheet tables in CSV and XLSX formats is presented in [16] and implemented in the TabbyXL tool [19].

3.2 Canonical Table Preprocessing

Preprocessing *DATA*, *RowHeading* and *ColumnHeading* column values in canonical tables is carried out by:

- deleting various symbolic values except letters and numbers;
- discarding all characters after a dot, comma or semicolon for header columns;
- deciphering acronyms and removing other undefined abbreviations;
- identifying and removing measurement units, etc.

The normalization of table cell values in accordance with the requirements of DBpedia global taxonomy is also carried out. Further, it is necessary to build correct SPARQL queries. For example, if a cell value consists of several words, then there no any characters between these words except a space or an underscore.

3.3 Canonical Table Interpretation

The semantic interpretation (linking) of canonical tables containing literal values is carried out in two stages:

- annotating cells in *DATA* column (a data block);
- annotating cells in *RowHeading* and *ColumnHeading* columns (a heading block).

All *DATA* column cell values contain only numerical values and are linked using *http://dbpedia.org/resource/Number* entity (instance). In this case, no additional annotation is required.

Cells in *RowHeading* and *ColumnHeading* column contain string values that express some named entities, so the linking strategy for these cell values described bellow.

The main task of this linking strategy is to assign the corresponding reference classes or entities (instances) from DBpedia global taxonomy to each *RowHeading* and *ColumnHeading* column cell value. Herewith, we use DBpedia segments describing ontology classes: <*http://dbpedia.org/ontology/*> and specific resource (entities): <*http://dbpedia.org/resource/*>. Each value in these cells is considered as one named entity (mention). The algorithm for annotating a heading block in a canonical table consists of four main steps (see Fig. 1):

Step 1. Searching and forming a set of candidate concepts (classes and instances) for each cell value of *RowHeading* and *ColumnHeading* columns.

This step yields a formed set of candidate concepts: $CL = \{CL_1,...,CL_k\}$. Since each header cell value can refer to multiple concepts from the set of candidates CL, it is not enough to simply select the appropriate concept for mapping. To eliminate this ambiguity, we propose two methods (see Step 2 and 3) to determine the similarity between concepts from the set of candidates and header cell values.

Step 2. We suggest a string similarity method to rank the set of candidate concepts CL. The most appropriate concept $cl_i \in CL$ for the header cell value $h_j \in H$ (where H is a set of *RowHeading* or *ColumnHeading* header values) is determined based on the maximum similarity of the character set. The Levenshtein distance can be used for this calculation: $f_{LevDis}(cl_i, h_j)$.

Step 3. Use a similarity method for linking candidate concepts CL (linking similarity). The most appropriate concept for the header cell value is determined by maximum amount of links with other concepts from sets of candidates found for all other cell values in *RowHeading* and *ColumnHeading* columns: $f_{LinkSim}(cl_i^1, cl_j^2)$.

Step 4. Aggregate estimates of methods to determine the most appropriate concept from the set of candidates CL: $f_{RankAgg} = \left(1 - \frac{f_{LevDis}}{100}\right) + f_{LinkSim}$.

Thus, the concept with the highest estimate (rank) is selected as a reference concept from the set of candidates CL for linking with the current value of the header cell.

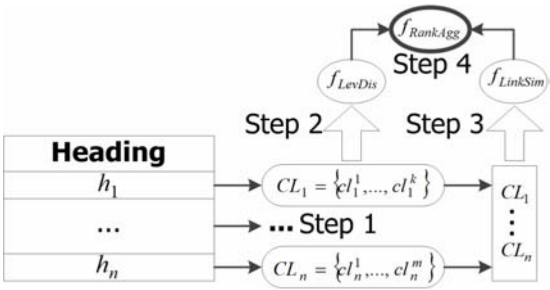

Fig. 1. The algorithm for annotating a headers block of a canonical table.

3.4 Linked Data Generation

We developed a prototype of software for generating linked data in the RDF format from the extracted and linked tabular data. It is designed to be a part of an end-to-end process of the table understanding implemented by our software platform within the TabbyDOC project [15, 18]. In this environment, it can be applied to the semantic interpretation of tables with an arbitrary layout.

Thus, the final result of linking is RDF graph, where each node is a resource or a literal, and each edge is a RDF triplet predicate. Moreover, each row of the annotated canonical table represents a description of a single object (one RDF resource per cell), where the concepts for cell values of *RowHeading* and *ColumnHeading* columns denote this object or its properties, and cell value of the *DATA* column is a quantitative measure (a numerical characteristic of an object). An RDF triplet is also created by default for all values of a *DATA* column. This triplet indicates the number entity: <*http://dbpedia.org/resource/Number*>.

4 Conclusions and Future Work

In this paper, we presented a conception of software for semantic interpretation of tables. The main purpose of this software is to link elements of data and heading blocks of canonical spreadsheets in the XLSX format with the concepts (classes or entities) of the external global DBpedia taxonomy. Annotation results are presented as linked data documents in the RDF format. The main advantage of our approach for the semantic interpretation of tables is that it can be applied to tables with an arbitrary layout. Unlike other similar solutions, our approach is focused on recovering implicit semantics not only for a data block but also for table headers.

The findings suggest that the proposed approach and software can be successfully used in future for annotation of industrial safety inspection reports [1]. Effectiveness evaluation of the semantic interpretation of spreadsheets for our software will be obtained for the Troy200 and Tango datasets. At the same time, TabbyXL experiment results on precision and recall for some data sets are presented in [16].

We expect that the explained principles can be used for designing software for the end-to-end semantic table interpretation in scientific and industrial data-intensive applications.

5 Acknowledgments

This work was financially supported by the Russian Science Foundation (Grant No. 18-71-10001).

References

1. Berman, A.F., Nikolaichuk, O.A., Yurin, A.Yu., Kuznetsov, K.A.: Support of Decision-Making Based on a Production Approach in the Performance of an Industrial Safety Review. Chemical and Petroleum Engineering 50(1-2), 730–738 (2015).
2. Berners-Lee, T. Linked Data (2006), https://www.w3.org/DesignIssues/LinkedData.html, last accessed 2019/08/21
3. Bhagavatula, C.S., Noraset, T., Downey, D.: Tabel: Entity linking in web tables. In: Arenas, M., Corcho, O., Simperl, E., Strohmaier, M., d'Aquin, M., Srinivas, K., Groth, P., Dumontier, M., Heflin, J., Thirunarayan, K., Thirunarayan, K., Staab, S. (eds.) The Semantic Web - ISWC 2015. pp. 425–441 (2015)
4. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S. DBpedia – A Crystallization Point for the Web of Data. Journal of Web Semantics, 7(3), 154–165 (2009). https://doi.org/10.1016/j.websem.2009.07.002
5. de Vos, M., Wielemaker, J., Rijgersberg, H., Schreiber, G., Wielinga, B., Top, J.: Combining information on structure and content to automatically annotate natural science spreadsheets. Int. J. Human-Computer Studies, 103, 63–76 (2017). https://doi.org/10.1016/j.ijhcs.2017.02.006
6. Deng, D., Jiang, Y., Li, G., Li, J., Yu, C.: Scalable column concept determination for web tables using large knowledge bases. Proc. VLDB Endow. 6(13), 1606–1617 (2013). https://doi.org/10.14778/2536258.2536271
7. Efthymiou, V., Hassanzadeh, O., Rodriguez-Muro, M., Christophides, V.: Matching web tables with knowledge base entities: From entity lookups to entity embeddings. In: d'Amato, C., Fernandez, M., Tamma, V., Lecue, F., Cudr´e-Mauroux, P., Sequeda, J., Lange, C., Heflin, J. (eds.) The Semantic Web – ISWC 2017. pp. 260–277 (2017)

8. Lehmberg, O., Ritze, D., Meusel, R., Bizer, C. A large public corpus of web tables containing time and context metadata. Proc. 25th Int. Conf. Companion on World Wide Web. pp. 75–76 (2016). https://doi.org/10.1145/2872518.2889386
9. Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and searching web tables using entities, types and relationships. Proc. VLDB Endow. 3(1-2), 1338–1347 (2010). https://doi.org/10.14778/1920841.1921005
10. Mulwad, V., Finin, T., Joshi, A.: A Domain Independent Framework for Extracting Linked Semantic Data from Tables, pp. 16–33 (2012). https://doi.org/10.1007/978-3-642-34213-4_2
11. Munoz, E., Hogan, A., Mileo, A.: Using linked data to mine RDF from wikipedia's tables. In: Proc. 7th ACM Int. Conf. Web Search and Data Mining. pp. 533–542 (2014). https://doi.org/10.1145/2556195.2556266
12. RDF 1.1 Concepts and Abstract Syntax, https://www.w3.org/TR/rdf11-concepts/, last accessed 2019/08/21
13. Ritze, D., Bizer, C.: Matching web tables to dbpedia - A feature utility study. In: Proc. 20th Int. Conf. on Extending Database Technology. pp. 210–221 (2017). https://doi.org/10.5441/002/edbt.2017.20
14. Shen, W., Wang, J., Luo, P., Wang, M.: LIEGE: Link entities in web lists with knowledge base. In: 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining. pp. 1424–1432 (2012). https://doi.org/10.1145/2339530.2339753
15. Shigarov, A.O., Khristyuk, V.V., Paramonov, V.V., Yurin, A.Y., Dorodnykh, N.O.: Toward framework for development of spreadsheet data extraction systems. CEUR Workshop Proceedings. Information Technologies: Algorithms, Models, Systems (ITAMS 2018), 2221, 90-96 (2018).
16. Shigarov, A.O., Mikhailov, A.A.: Rule-based spreadsheet data transformation from arbitrary to relational tables. Information Systems 71, 123–136 (2017). https://doi.org/10.1016/j.is.2017.08.004
17. SPARQL 1.1 Query Language, https://www.w3.org/TR/sparql11-query/, last accessed 2019/08/21
18. TabbyDOC, Tabular Document Analysis Research Group at ISDCT SB RAS, http://td.icc.ru/, last accessed 2019/08/21
19. TabbyXL, https://github.com/tabbydoc/tabbyxl, last accessed 2019/08/21
20. Tijerino, Y., Embley, D., Lonsdale, D., Ding, Y., Nagy, G.: Towards ontology generation from tables. World Wide Web: Internet and Web Information Systems 8(3), 261–285 (2005). https://doi.org/10.1007/s11280-005-0360-8
21. Venetis, P., Halevy, A., Madhavan, J., Pa¸sca, M., Shen, W., Wu, F., Miao, G., Wu, C.: Recovering semantics of tables on the web. Proc. VLDB Endow. 4(9), 528–538 (2011). https://doi.org/10.14778/2002938.2002939
22. Wang, J., Wang, H., Wang, Z., Zhu, K.Q.: Understanding tables on the web. In: Proc. 31st Int. Conf. Conceptual Modeling. pp. 141–155 (2012). https://doi.org/10.1007/978-3-642-34002-4_11
23. Wu, T., Yan, S., Piao, Z., Xu, L., Wang, R., Qi, G.: Entity linking in web tables with multiple linked knowledge bases. In: Li, Y.F., Hu, W., Dong, J.S., Antoniou, G., Wang, Z., Sun, J., Liu, Y. (eds.) Semantic Technology. pp. 239–253 (2016)
24. Zhang, Z.: Effective and efficient semantic table interpretation using TableMiner+. Semantic Web 8(6), 921–957 (2017). https://doi.org/10.3233/SW-160242