# From web-tables to a knowledge graph: prospects of an end-to-end solution

Alexey Shigarov[1], Nikita Dorodnykh[1], Alexander Yurin[1], Andrey Mikhailov[1] and Viacheslav Paramonov[1]

[1] *Matrosov Institute for System Dynamics and Control Theory, Siberian Branch of the Russian Academy of Sciences, 134 Lermontov St, Irkutsk, 664033, Russian Federation*

## Abstract

The Web stores a large volume of web-tables with semi-structured data. The Semantic Web community considers them as a valuable source for the knowledge graph population. Interrelated named entities can be extracted from web-tables and mapped to a knowledge graph. It generally requires reconstructing the semantics missing in web-tables to interpret them according to their meaning. This paper discusses prospects of an end-to-end solution for the knowledge graph population by entities extracted from web-tables of predefined types. The discussion covers theoretical foundations both for transforming data from web-tables to entity sets (table analysis) and for mapping entities, attributes, and relations to a knowledge graph (semantic table annotation). Unlike general-purpose text mining and web-scraping tools, we aim at developing a solution that takes into account the relational nature of the information represented in web-tables. In contrast to the table-specific proposals, our approach implies both the table analysis and the semantic table annotation.

## Keywords

table understanding, semantic table interpretation, web-tables, data extraction, knowledge graph population, semantic web

## 1. Introduction

The Web stores a large volume of tables. The exploration of the Web crawl discovered hundreds of millions of web-tables containing relational data [1, 2]. There are at least billiards of valuable facts that can be extracted from web-tables. All of these make web-tables an attractive data source in various applications, such as knowledge base construction [3, 4], question-answering systems, and table augmentation [5, 6, 7]. However, in general, web-tables are not interpretable by computer programs. Their original representation does not provide all explicit semantics required to interpret them according to their meaning. This hinders the wide usage of such tabular data in practice.

Reconstruction of the semantics missing in tables is commonly referred to as the "*table understanding*". This problem was first formulated by M. Hurst in 2000 [8, 9]. Over the two last

*ITAMS 2021: Information Technologies: Algorithms, Models, Systems, September 14, 2021, Irkutsk, Russia*

shigarov@icc.ru (A. Shigarov); tualatin32@mail.ru (N. Dorodnykh); iskander@icc.ru (A. Yurin); mikhailov@icc.ru (A. Mikhailov); slv@icc.ru (V. Paramonov)
http://td.icc.ru (A. Shigarov)
0000-0001-5572-5349 (A. Shigarov); 0000-0001-7794-4462 (N. Dorodnykh); 0000-0001-9089-5730 (A. Yurin); 0000-0003-4057-4511 (A. Mikhailov); 0000-0002-4662-3612 (V. Paramonov)

CEUR Workshop Proceedings (CEUR-WS.org)

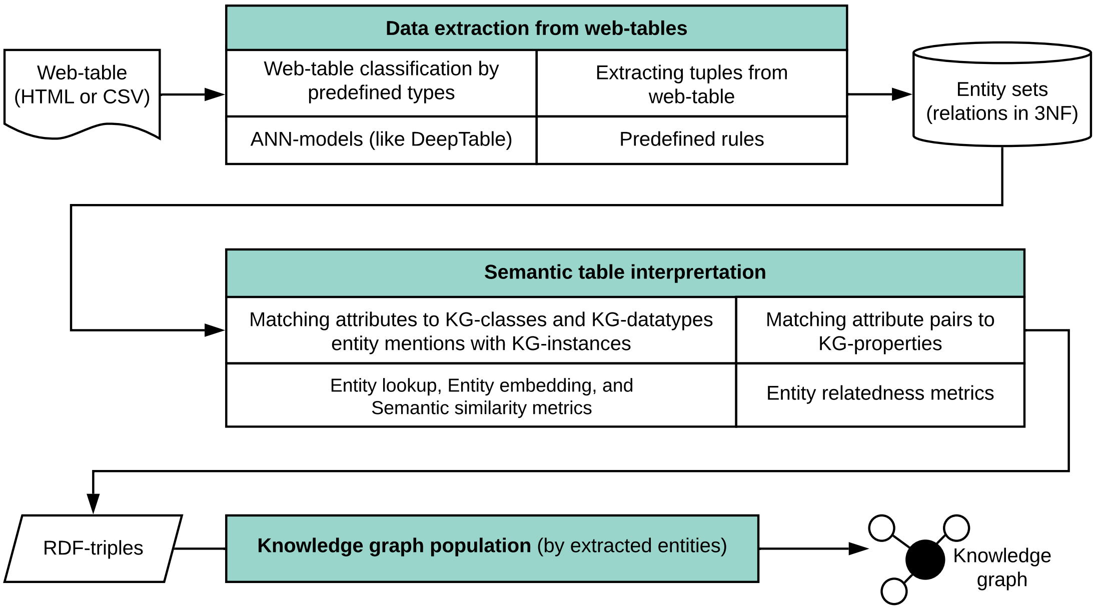

Web-table
(HTML or CSV)
Data extraction from web-tables
Web-table classification by predefined types
Extracting tuples from web-table
ANN-models (like DeepTable)
Predefined rules
Entity sets
(relations in 3NF)
Semantic table interprertation
Matching attributes to KG-classes and KG-datatypes entity mentions with KG-instances
Matching attribute pairs to KG-properties
Entity lookup, Entity embedding, and Semantic similarity metrics
Entity relatedness metrics
RDF-triples
Knowledge graph population (by extracted entities)
Knowledge graph

**Figure 1:** Steps for the knowledge graph population with entities extracted from web-tables.

decades, hundreds of papers devoted to its issues were published [6, 7, 10, 11]. The literature survey shows that this topic continues to rapidly develop in several communities such as document understanding, semantic web, and end-user programming. The last 3 years were marked by an extraordinary growth of proposals based on a novel apparatus, namely, deep learning, word and entity embedding, and knowledge graphs. The two highly-rated conferences, "*Int. Semantic Web Conf.*" and "*Int. Conf. Document Analysis and Recognition*", recently conducted competitions related to this problem.

The complexity of the problem is determined by two factors: (i) a wide variety of tricks to present table layout, formatting, and content; (ii) limited representation formats (such as Excel or HTML) that do not provide all semantics needed for data interpretation. Generally, the solution requires all stages of the table understanding: (i) table detection or discrimination; (ii) table structure recognition and cleaning; (iii) role and structural analysis (i.e. extracting interrelated data and metadata values from the content); (iv) semantic interpretation (i.e. matching the semantic table structure with an external dictionary).

There several are challenges for the expert community. One of them is to develop of a common theoretical and technological basis applicable to various digital environments and formats for representing tabular data in the Web (such as print-oriented documents, spreadsheets, and web-pages). Our approach is addressed to this challenge, namely the extraction and semantic interpretation of data from web-tables represented in HTML-format (Fig. 1).

The recent surveys of the thematic literature [6, 7, 10, 11] note that the problem of table understanding remains open. The review [6] revealed that the majority of the works focuses mainly on the tasks of discrimination and semantic interpretation of web-tables. Roldán et al. [7] indicated that none of the known solutions is complete. They do not provide all steps of the table understanding. This is also confirmed by Burdick et al. [10]. As reported in [7], many table design properties are not taken into account by the state-of-the-art solutions. This often hinders their practical application.

The existing models of table representation do not completely reflect the complexity of the structure of real tables. One of the commonly-used assumptions is "*all cell values are atomic*". They assume that any non-blank cell contains only one functional data item. To the best of our knowledge, all competitive solutions follow this simplification. However, a real cell can have several data items with the same or different functions. The latter should be taken to account in order to extend the range of cases for table processing.

The novelty of our proposal is established by the following. First, we propose an end-to-end solution covering the stages from extracting data from syntactically tagged tables to their semantic interpretation (i.e. mapping extracted data and metadata to a cross-domain knowledge graph). Second, we take into account structured cells which content should be decomposed into several atomic data items with different functional roles. Third, our proposal can show the applicability of some promising techniques (cell embeddings, contextualized word embeddings, entity embedding) to the tasks of table understanding.

Unlike the general-purpose text mining and web-scraping tools, our solution takes into account the relational nature of the information represented in web-tables. In contrast to similar proposals that target data extraction from web-tables, we cover a wider range of cases by involving the structured content of a cell. Moreover, the competitive techniques are limited either by data extraction stage or the stage of semantic table interpretation, whereas, our approach implies both of them. Therefore, the expected results could be applied in the knowledge base population.

## 2. Data extraction from web-tables

The approach to the data extraction from web-tables includes two stages: (i) classifying web-tables by predefined types; (ii) extracting entity sets from web-tables, using algorithms appropriated to the corresponding types.

### 2.1. Web-table classification

In the last decade, several taxonomies of web-table types were published in the last decade [12, 13, 14]. All taxonomies describe three common types of web-tables with relational data (Fig. 2). Eberius et al. refer to them as "*vertical listing*", "*horizontal listing*", and "*matrix*". This taxonomy is used by the latest proposals for the table type classification [15, 16, 17, 18]. We also rely on this taxonomy. This will allow us to quantitatively compare our results with others.

We plan to develop a deep neural network model for classifying web-tables based on DeepTable[1] [17], the ad-hoc architecture that provides four main blocks: (i) Embedding layer for extracting vector representation of cell tokens; (ii) LSTM (recurrent neural network) for identifying semantic dependencies between tokens in a cell; (iii) MLP (multilayer perceptron) for

[1] https://github.com/marhabibi/deeptable

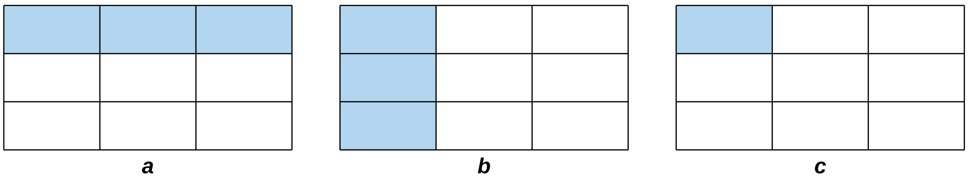


**Figure 2:** Web-tables taxonomy: vertical listing — (*a*), horizontal listing — (*b*), and matrix — (*c*).

identifying non-linear dependencies between all cells in a table; (iv) Softmax as a classification layer.

An open collection of tagged tables extracted from biomedical research papers (PubMed Central[2]) can be used as training data. To select a basic tool for contextualized vector representation of words, we propose to try several variations (ELMo[3] [19], fastText[4] [20], etc.). Some classifiers can be trained for each variation. This allows us to compare their accuracy and choose the best for this task.

## 2.2. Transformation of entity sets from web-tables

We propose to develop algorithms that target three table types of [14]. The algorithms should analyze the logical structure of web-tables by using built-in rules and trained classifiers dealing with these types. It is important to note that web-tables mix data and metadata. Moreover, one cell may contain several values of both data and metadata. The extraction of the logical structure requires: (i) to associate each cell value (data item) with its functional role (data and metadata); (ii) to associate data values with metadata ones; (iii) to group data belonging to one record (entity).

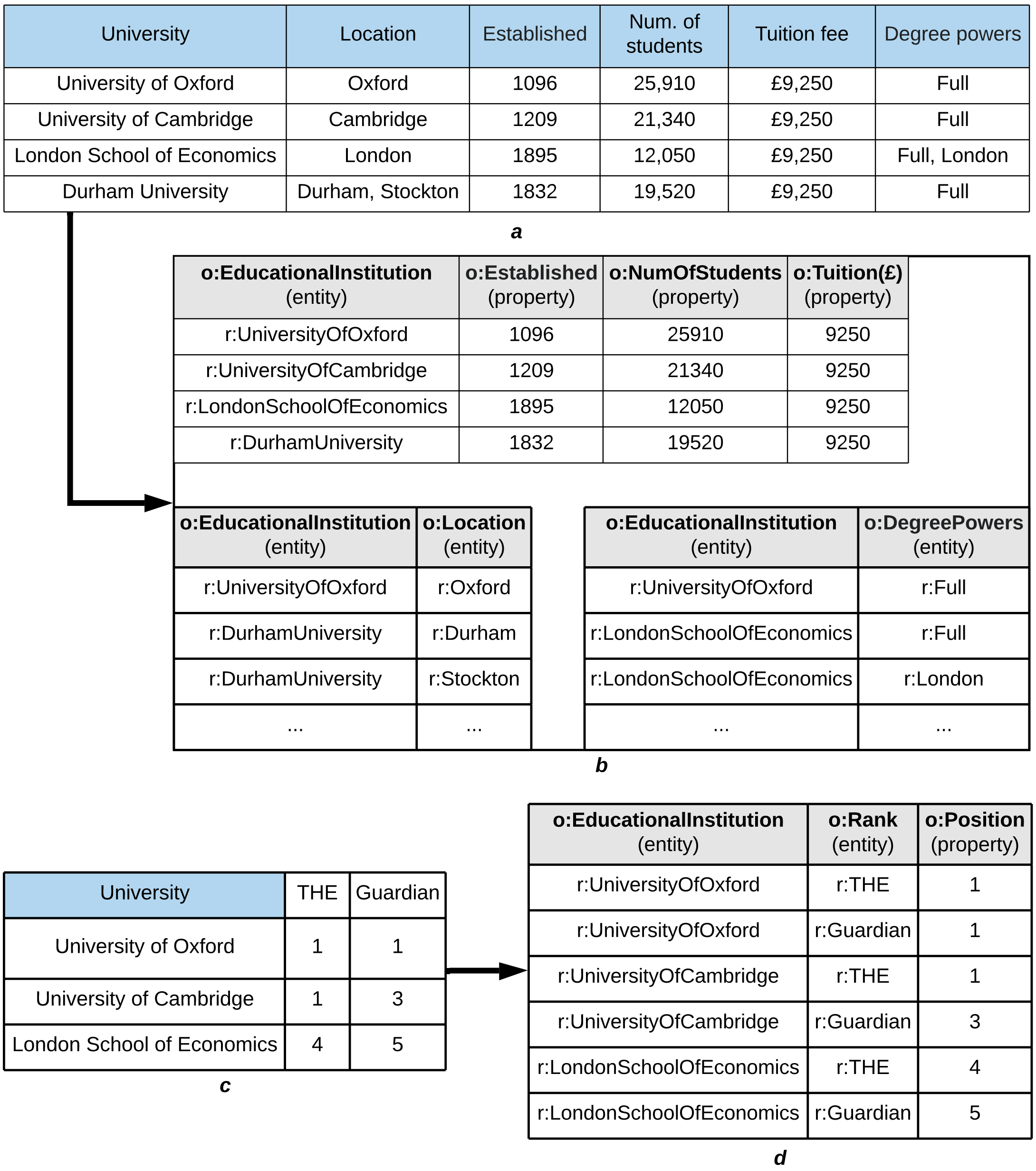
After the table type classification, the extraction of data and metadata values becomes a "cornerstone" step. We plan to implement a classifier based on machine learning algorithms to assign functional roles to data items. A promising approach to encoding cell context in the vector representation named "*cell embedding*" was recently proposed in [21, 22]. In our case, the cell context can be deduced automatically by using the built-in type-specific table structure analysis. This approach should allow cells classification taking into account the properties of their layout and formatting, as well as the semantic similarity of their text content. We plan to reduce the number of possible false-positive and false-negative errors by using some table-specific constraints. To associate data with metadata and group data values with records, we also suggest using rule-based analysis of the table structure. The extracted entity sets can be represented in JSON notation compatible with TOMATE[5] [18], a recently published framework for the performance evaluation of web-table data extraction tools.

[2] https://www.ncbi.nlm.nih.gov/pmc

[3] https://allennlp.org/elmo

[4] https://fasttext.cc

[5] http://tomatera.tdg-seville.info



| University | Location | Established | Num. of students | Tuition fee | Degree powers |
|---|---|---|---|---|---|
| University of Oxford | Oxford | 1096 | 25,910 | £9,250 | Full |
| University of Cambridge | Cambridge | 1209 | 21,340 | £9,250 | Full |
| London School of Economics | London | 1895 | 12,050 | £9,250 | Full, London |
| Durham University | Durham, Stockton | 1832 | 19,520 | £9,250 | Full |

*a*

| **o:EducationalInstitution** (entity) | **o:Established** (property) | **o:NumOfStudents** (property) | **o:Tuition(£)** (property) |
|---|---|---|---|
| r:UniversityOfOxford | 1096 | 25910 | 9250 |
| r:UniversityOfCambridge | 1209 | 21340 | 9250 |
| r:LondonSchoolOfEconomics | 1895 | 12050 | 9250 |
| r:DurhamUniversity | 1832 | 19520 | 9250 |

| **o:EducationalInstitution** (entity) | **o:Location** (entity) |
|---|---|
| r:UniversityOfOxford | r:Oxford |
| r:DurhamUniversity | r:Durham |
| r:DurhamUniversity | r:Stockton |
| ... | ... |

| **o:EducationalInstitution** (entity) | **o:DegreePowers** (entity) |
|---|---|
| r:UniversityOfOxford | r:Full |
| r:LondonSchoolOfEconomics | r:Full |
| r:LondonSchoolOfEconomics | r:London |
| ... | ... |

*b*

| University | THE | Guardian |
|---|---|---|
| University of Oxford | 1 | 1 |
| University of Cambridge | 1 | 3 |
| London School of Economics | 4 | 5 |

*c*

| **o:EducationalInstitution** (entity) | **o:Rank** (entity) | **o:Position** (property) |
|---|---|---|
| r:UniversityOfOxford | r:THE | 1 |
| r:UniversityOfOxford | r:Guardian | 1 |
| r:UniversityOfCambridge | r:THE | 1 |
| r:UniversityOfCambridge | r:Guardian | 3 |
| r:LondonSchoolOfEconomics | r:THE | 4 |
| r:LondonSchoolOfEconomics | r:Guardian | 5 |

*d*

**Figure 3:** Example of the semantic table annotation: origin web-tables (vertical listing —*a* and matrix — *c*) and their normalized and annotated forms (*b*, *d*); the prefix **o** ("*ontology*") denotes a *KG*-class or *KG*-property of entities defined in the terminological component (TBox) while **r** ("*resource*") is a *KG*-instance from the assertion component (ABox).

## 3. Semantic annotation of entity sets

There are 3 main approaches to the semantic table interpretation, namely: (i) ontology matching; (ii) entity lookup and wikification; (iii) vector representations of knowledge graphs (entity embedding). The recent study [23] showed experimentally that a hybrid approach combining lookup services and entity embedding is one of the most efficient ways. We plan to exploit such a hybrid using the available toolset (DBpedia Lookup[6], DBpedia SPARQL Endpoint[7], DBpedia Spotlight[8], RDF2Vec[9], KGloVe[10], and Wikipedia2Vec[11]).

The end-to-end semantic table interpretation includes 3 stages:

- Entity linking — CEA (Cell-Entity Annotation).
- Attribute-concept matching — CTA (Column-Type Annotation).
- Relation extraction — CPA (Column-Property Annotation).

As a result, this enables knowledge graph augmentation. Fig. 3 shows two examples where web-tables from Wikipedia[12][13] (Fig. 3, *a*, *b*) are normalized and enriched by the semantic annotation (Fig. 3, *c*, *d*), i.e. links to a knowledge graph.

## 3.1. Entity linking — CEA

The proposed solution should provide for the following: (i) identifying a subject column containing names of entities listed in a table; (ii) lookup a set of candidate $KG$-instances for each entity; (iii) entity disambiguation in cases when several candidate *KG*-instances are associated with an entity.

The subject column is selected among potential keys that contain entity mentions. We are limited by the trivial case when there is only one candidate subject column. (Note that the general case requires the end-to-end semantic table interpretation).

As the main tool for linking entities, we propose to use the vector representations of subsets from a knowledge graph. The initial lookup of candidate *KG*-instances can be performed by using SPARQL-queries to the knowledge graph. Such queries are composed of surface forms contained in the text of cells. Each *KG*-instance can be encoded as a vector representation of the entity by the existing algorithms, such as RDF2Vec [24], KGloVe [25], or Wikipedia2Vec [26]. The formed vector model should allow us to use some semantic similarity metrics [27] to rank candidate *KG*-instances by relevance to the entity.

The approach to the entity disambiguation relies on the assumption proposed by [28] which implies that that the most relevant *KG*-instances from the candidate sets have the highest semantic similarity values in pairwise matching. This can be explained by the following example from [28]. Let a column contain 3 mentions: "*USA*", "*China*", and "*India*". They should be matched to 3 sets of candidate *KG*-instances respectively: "*USA*" → ["*University of South Alabama (University)*", "*United States of America (Country)*"], "*China*" → ["*People's Rep. of China (Country)", "China (Band)*", "*China, Kagoshima (City)*"], "*India*" → ["*India (Country)", "India (George W. Bush's cat)*", "*India (Xandria album)*"]. Among all pairs of *KG*-instances, "*United*

[6] https://lookup.dbpedia.org

[7] https://dbpedia.org/sparql

[8] https://www.dbpedia-spotlight.org

[9] http://rdf2vec.org

[10] https://datalab.rwth-aachen.de/embedding/kglove

[11] https://wikipedia2vec.github.io/wikipedia2vec

[12] https://en.wikipedia.org/wiki/List_of_universities_in_England

[13] https://en.wikipedia.org/wiki/Rankings_of_universities_in_the_United_Kingdom

*States of America (Country)*", "*People's Rep. of China (Country)*" and "*India (Country)*" would be the most semantically similar (they mean a common concept in the knowledge graph).

Thus, this approach should allow us to rank candidate *KG*-instances and select from them the reference KG-instances for specific mentions. For example, the table showed in Fig. 3, *a* contains the surface form "*London*" that can mean "*Location*" or "*Degree powers*". Obviously, in the context of the column *[Oxford, Cambridge, London, Durham, Stockton]* it should be assigned to the instance of "*Location*" while in the context of the column *[Full, London, Taught]* it corresponds to the instance of "*Degree powers*".

## 3.2. Attribute-concept matching — CTA

In practice, many tables are not accompanied by metadata (named attributes). Generally, to map a column to a *KG*-class, first it is needed to associate the entities listed in the column with the reference *KG*-instances. After that, it is possible to form an index of all candidate *KG*-classes to which the reference *KG*-instances belong. Among them, the *KG*-class which is most relevant to all column values is selected. For example, in Fig. 3, *a* three columns should be matched to *KG*-classes (Fig. 3, *b*) as follows:

```
"University" -> o:EducationalInstitution
"Location" -> o:Location
"Degree powers" -> o:DegreePowers
```

While the rest of columns are corresponded to *KG*-properties of `o:EducationalInstitution` (*KG*-class) as follows:

```
"Established" -> o:Established
"Num. of students" -> o:NumOfStudents
"Tuition fee" -> o:Tuition(£)
```

In the cases when entity linking (CEA-stage) fails, we propose to use ANN-models to predict the *KG*-class of a column based on ColNet algorithms [29, 30]. To map a column of literal values (NUMERIC, DATE, CURRENCY, etc.) to a *KG*-datatype, it is enough to recognize standard named entities. This is reached by using regular expressions and NER-models available in popular NLP-libraries (e.g. Stanford CoreNLP[14], AllenNLP[15]).

## 3.3. Relation extraction — CPA

To map pairs of columns ($< E, P >$, where $E$ is a subject, $P$ is not a subject) with *KG*-properties, we plan to use entity relatedness metrics [31]. It is assumed that these metrics will allow ranking the index of candidate *KG*-properties and choosing the most relevant ones. For example, two web-tables showed in Fig. 3 contain the following relationships:

```
<r:DurhamUniversity, o:located_in, r:Durham>
<r:DurhamUniversity, o:located_in,  r:Stockton>
<<r:UniversityOfOxford, o:ranked_in, r:THE>, o:positioned_at, 1>
<<r:UniversityOfOxford, o:ranked_in, r:Guardian>, o:positioned_at, 1>
```

[14] https://stanfordnlp.github.io/corenlp

[15] https://github.com/allenai/allennlp

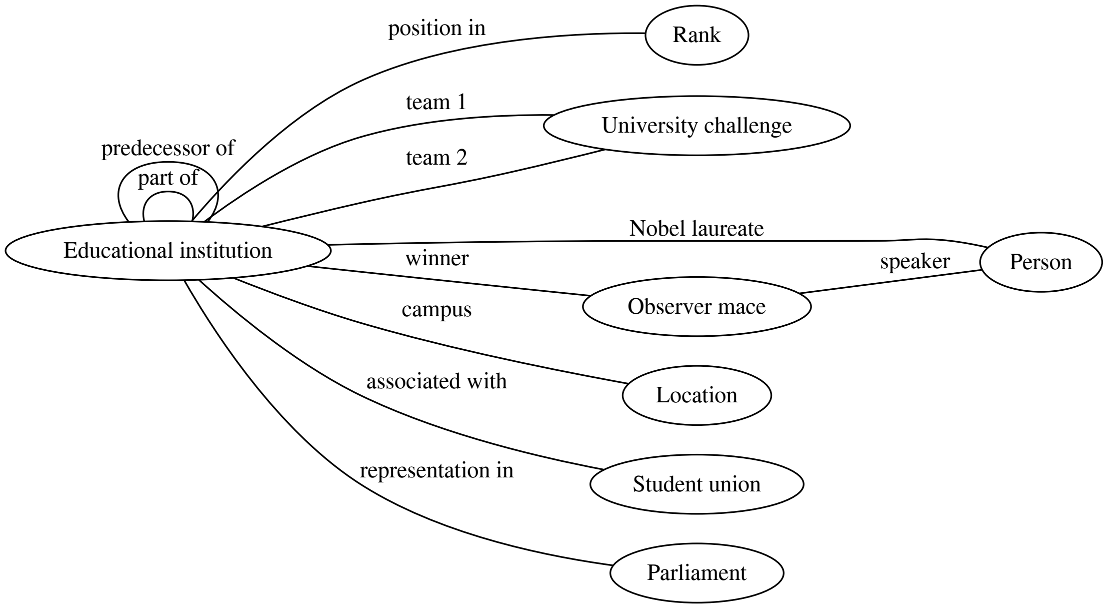


**Figure 4:** An example of the terminological level (TBox) of a knowledge graph constructed from tables of Wikipedia pages in "*Category: Universities in the United Kingdom*".

### 3.4. Knowledge graph augmentation

An entity set represented as linked data (RDF-triples with URI-references to concepts in a knowledge graph) should be suitable for further interpretation. In particular, other facts (RDF-triples) can be inferred from them and asserted to the knowledge graph. Such restored semantics would provide populating the existing knowledge graphs with new entities extracted from web-tables. For example, Fig. 4 shows the terminological level (TBox) of a knowledge graph constructed by using 49 tables scrapped from Wikipedia pages in "*Category: Universities in the United Kingdom*". The facts extracted from these tables can be asserted into the ABox component of the knowledge graph.

We plan to demonstrate the applicability of the proposed solution by an illustrative example of populating a domain-specific knowledge graph (ABox component) in the area of industrial safety expertise. This should cover aligning entity set records with the structure of the knowledge graph (row-to-instance matching) and synthesizing new *KG*-instances and *KG*-properties. Tables extracted from real reports on industrial safety expertise may be used as a source of domain data.

## 4. Conclusions

Our previous work [32, 33, 34] was aimed at data extraction from spreadsheets driven by user-defined rules. We proposed end-user programming as the main approach. This allowed us to support specific tricks of table layout, formatting, and content. However, scaling such solutions may be challenging when there are ambiguous tricks applied within source tables. Nonetheless, a solution intended for the Web should be easily scaled. This is possible when there are pre-defined types of web-tables. The latter is needed to classify them and select type-specific algorithms of analysis and interpretation. Thus, our previous approach is suitable for spreadsheet sources, but not for the Web.

The current proposal aims to fill this gap by the development of a scalable solution for web-tables. The expected results contribute to the following: (i) data extraction, including algorithms for classifying web-tables by types of taxonomy and extracting entity sets from tables of predefined types, (ii) semantic table annotation, including algorithms for mapping entities, attributes, and relations to concepts of an external knowledge graph, (iii) open software for implementing the functionality of the extraction and semantic annotation of tabular data in applications of the knowledge graph population.

To the best of our knowledge, all existing proposals for data extraction from web-tables exploit a specific constraint: "*any cell contains only one atomic data item*". This constraint can be eliminated in the proposed solution. We argue that the structured content of a cell can be decomposed into several data items. Moreover, all proposals implement the semantic table interpretation only for entity sets, not pivots. We plan to study both kinds of tabular data. We think this can expand the range of cases to be processed.

We propose to apply the state-of-the-art methods and tools, including contextualized word embeddings, vector representations of knowledge graphs, entity lookup services, as well as metrics of semantic similarity and entity relatedness. The applicability of some of these tools for the considered issues remains poorly studied. The expected results could demonstrate the promise of the use of these techniques.

The expected results could be useful to intellectualize software for tabular data extraction and integration in scientific and industrial applications. It can be of particular interest in areas with the intensive use of tabular data (e.g., finance, government statistics, and business management) to form linked open data.

## Acknowledgments

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

## References

[1] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, Y. Zhang, WebTables: exploring the power of tables on the web, Proc. VLDB Endowment 1 (2008) 538–549. doi:10.14778/1453856.1453916.

[2] M. J. Cafarella, A. Halevy, D. Z. Wang, U. C. Berkeley, E. Wu, Uncovering the relational web, in: Proc. 11th Int. W. on Web and Databases, 2008.

[3] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.-C. N. Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, 2021. arXiv:2003.02320.

[4] J. L. Martinez-Rodriguez, A. Hogan, I. Lopez-Arevalo, Information extraction meets the semantic web: a survey, Semantic Web 11 (2020) 255–335. doi:10.3233/SW-180333.

[5] M. Cafarella, A. Halevy, H. Lee, J. Madhavan, C. Yu, D. Z. Wang, E. Wu, Ten years of WebTables, Proc. VLDB Endowment 11 (2018) 2140–2149. doi:10.14778/3229863.3240492.

[6] S. Zhang, K. Balog, Web table extraction, retrieval, and augmentation: a survey, ACM Trans. Intell. Syst. Technol. 11 (2020). doi:10.1145/3372117.

[7] J. C. Roldán, P. Jiménez, R. Corchuelo, On extracting data from tables that are encoded using html, Knowledge-Based Systems 190 (2020) 105157. doi:10.1016/j.knosys.2019.105157.

[8] M. F. Hurst, The interpretation of tables in texts, Ph.D. thesis, 2000.

[9] M. Hurst, Layout and language: challenges for table understanding on the web, in: Proc. Int. W. on Web Document Analysis, 2001, pp. 27–30.

[10] D. Burdick, M. Danilevsky, A. V. Evfimievski, Y. Katsis, N. Wang, Table extraction and understanding for scientific and enterprise applications, Proc. VLDB Endow. 13 (2020) 3433–3436. doi:10.14778/3415478.3415563.

[11] S. Bonfitto, E. Casiraghi, M. Mesiti, Table understanding approaches for extracting knowledge from heterogeneous tables, WIREs Data Mining and Knowledge Discovery 11 (2021) e1407. doi:10.1002/widm.1407.

[12] E. Crestan, P. Pantel, Web-scale table census and classification, in: Proc. 4th ACM Int. Conf. on Web Search and Data Mining, 2011, pp. 545–554. doi:10.1145/1935826.1935904.

[13] L. R. Lautert, M. M. Scheidt, C. F. Dorneles, Web table taxonomy and formalization, ACM SIGMOD Record 42 (2013) 28–33. doi:10.1145/2536669.2536674.

[14] J. Eberius, K. Braunschweig, M. Hentsch, M. Thiele, A. Ahmadov, W. Lehner, Building the dresden web table corpus: a classification approach, in: Proc. IEEE/ACM 2nd Int. S. on Big Data Computing, 2015, pp. 41–50. doi:10.1109/BDC.2015.30.

[15] O. Lehmberg, D. Ritze, R. Meusel, C. Bizer, A large public corpus of web tables containing time and context metadata, in: Proc. 25th Int. Conf. on World Wide Web, 2016, pp. 75–76. doi:10.1145/2872518.2889386.

[16] K. Nishida, K. Sadamitsu, R. Higashinaka, Y. Matsuo, Understanding the semantic structures of tables with a hybrid deep neural network architecture, in: Proc. 31st AAAI Conf. on Artificial Intelligence, 2017, pp. 168–174.

[17] M. Habibi, J. Starlinger, U. Leser, Deeptable: a permutation invariant neural network for table orientation classification, Data Mining and Knowledge Discovery 34 (2020) 1963–1983. doi:10.1007/s10618-020-00711-x.

[18] J. C. Roldán, P. Jiménez, P. Szekely, R. Corchuelo, Tomate: A heuristic-based approach to extract data from html tables, Information Sciences (2021). doi:10.1016/j.ins.2021.04.087.

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proc. of NAACL, 2018.

[20] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, 2017. arXiv:1607.04606.

[21] M. Ghasemi-Gol, J. Pujara, P. Szekely, Tabular cell classification using pre-trained cell embeddings, in: 2019 IEEE International Conference on Data Mining (ICDM), 2019, pp.

230–239. doi:10.1109/ICDM.2019.00033.

[22] M. Ghasemi-Gol, J. Pujara, P. Szekely, Deeptable: a permutation invariant neural network for table orientation classification, Data Mining and Knowledge Discovery 63 (2021) 39–64. doi:10.1007/s10115-020-01508-6.

[23] V. Efthymiou, O. Hassanzadeh, M. Rodriguez-Muro, V. Christophides, Matching web tables with knowledge base entities: from entity lookups to entity embeddings, in: The Semantic Web – ISWC 2017, volume 10587 LNCS, 2017, pp. 260–277. URL: http://link.springer.com/10.1007/978-3-319-68288-4{_}16. doi:10.1007/978-3-319-68288-4_16.

[24] P. Ristoski, H. Paulheim, Rdf2vec: Rdf graph embeddings for data mining, in: The Semantic Web – ISWC 2016, 2016, pp. 498–514. doi:10.1007/978-3-319-46523-4_30.

[25] M. Cochez, P. Ristoski, S. P. Ponzetto, H. Paulheim, Global rdf vector space embeddings, in: The Semantic Web – ISWC 2017, 2017, pp. 190–207. doi:10.1007/978-3-319-68288-4_12.

[26] I. Yamada, A. Asai, J. Sakuma, H. Shindo, H. Takeda, Y. Takefuji, Y. Matsumoto, Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia, in: Proc. 2020 Conf. Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, 2020, pp. 23–30.

[27] G. Zhu, C. A. Iglesias, Exploiting semantic similarity for named entity disambiguation in knowledge graphs, Expert Systems with Applications 101 (2018) 8–24. doi:10.1016/j.eswa.2018.02.011.

[28] S. Zwicklbauer, C. Seifert, M. Granitzer, Doser – a knowledge-base-agnostic framework for entity disambiguation using semantic embeddings, in: The Semantic Web. Latest Advances and New Domains, 2016, pp. 182–198. doi:10.1007/978-3-319-34129-3_12.

[29] J. Chen, E. Jiménez-Ruiz, I. Horrocks, C. Sutton, Colnet: Embedding the semantics of web tables for column type prediction, Proc. AAAI Conf. Artificial Intelligence 33 (2019) 29–36. doi:10.1609/aaai.v33i01.330129.

[30] J. Chen, E. Jimenez-Ruiz, I. Horrocks, C. Sutton, Learning semantic annotations for tabular data, in: Proc. 28th Int. Joint Conf. Artificial Intelligence, 2019, pp. 2088–2094. doi:10.24963/ijcai.2019/289.

[31] M. Ponza, P. Ferragina, S. Chakrabarti, On computing entity relatedness in wikipedia, with applications, Knowledge-Based Systems 188 (2020) 105051. doi:10.1016/j.knosys.2019.105051.

[32] A. Shigarov, Table understanding using a rule engine, Expert Syst. Appl. 42 (2015). doi:10.1016/j.eswa.2014.08.045.

[33] A. Shigarov, A. Mikhailov, Rule-based spreadsheet data transformation from arbitrary to relational tables, Inform. Syst. 71 (2017) 123–136. doi:10.1016/j.is.2017.08.004.

[34] A. Shigarov, V. Khristyuk, A. Mikhailov, TabbyXL: software platform for rule-based spreadsheet data extraction and transformation, SoftwareX 10 (2019) 100270. doi:10.1016/j.softx.2019.100270.