Combining Knowledge Graphs and Language Models to Answer Questions over Tables

Judith Knoblach[1,†], Nikhil Acharya[1,†], Bhavya Koranemkattil[1,†], Andreas Both[2,3] and Diego Collarana[1,4,*]

[1]*Fraunhofer Institute for Intelligent Analysis and Information Systems, Dresden, Germany*
[2]*Leipzig University of Applied Sciences, Leipzig, Germany*
[3]*DATEV eG, Nuremberg, Germany*
[4]*Universidad Privada Boliviana, Cochabamba, Bolivia*

Abstract

Tables remain a primary modality for organizing and presenting information to people. We interact every day with Excel sheets, CSV files, tables in PDF documents, and web tables. Providing a natural language interface to query table information is paramount for several use cases. This demo shows a solution to query semantically described tables using natural-language questions. Our solution employs knowledge graphs as a medium to integrate tables coming from heterogeneous sources. Then, a transformer-based language model analyzes a user's question and finds the answer in the semantically represented tables. During the demo session, we will show a use case developed in collaboration with DATEV eG, where tax consultants can efficiently query information from financial tables. Attendees will experience how a natural-language interface speeds up the information retrieval process from tables. They will also be allowed to ask their questions to a prepared dataset, showing the scalability of our solution. The video demo is available at https://owncloud.fraunhofer.de/index.php/s/uXFmUfzCta70rqN.

Keywords

Knowledge Graphs, Transformers, Language Models

1. Introduction

Companies still use tables as the main modality to present information to employees. For example, DATEV eG is a German company mainly providing large-scale business software (e.g., accounting). These services are widely used by tax consultants, lawyers, auditors, small and medium-sized enterprises, municipalities, start-ups, and many more. More than two million German companies use financial accounting programs from DATEV, interacting with hundreds of tables daily. To continue this success story and remain competitive in the market, DATEV relies on employees who are experts in their field and on state-of-the-art software solutions to accelerate internal processes even in environments that become increasingly data-dependent.

SEMANTICS 2022 EU: 18th International Conference on Semantic Systems, September 13-15, 2022, Vienna, Austria
*Corresponding author.
†These authors contributed equally.
judith.knoblach@iais.fraunhofer.de (J. Knoblach); nikhil.acharya@iais.fraunhofer.de (N. Acharya); bhavya.koranemkattil@iais.fraunhofer.de (B. Koranemkattil); andreas.both@datev.de (A. Both); diegocollarana@upb.edu (D. Collarana)
0000-0002-9177-5463 (A. Both); 0000-0002-2583-0778 (D. Collarana)

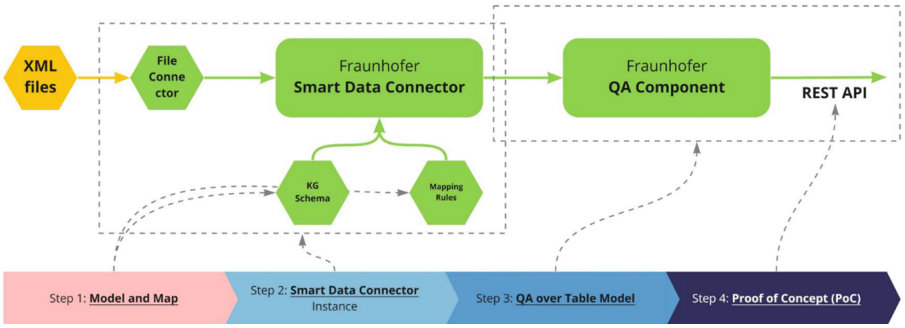


Figure 1: Question Answering over Tables – A Knowledge Graph Transformer based approach

DATEV uses AI solutions that facilitate the development and relieve the employees' workload. One of the most relevant aspects is data management and information retrieval. DATEV employees handle a wide range of information on different domains, e.g., taxes in European countries, specific software versions for wire transfers, or (strict) deadlines for storing personal, sensitive data. Often the data is presented as tables, either as web tables or CSV files coming from internal, external, or official data providers. This data-intensive environment makes it challenging for users to find information efficiently.

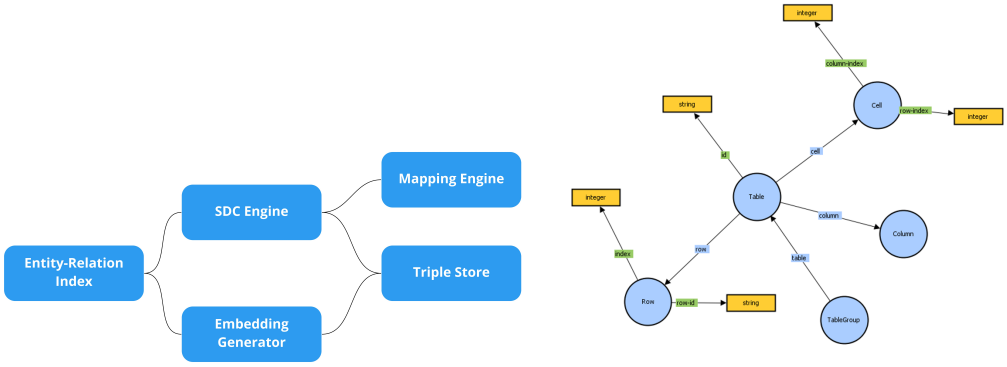
In the scope of the SPEAKER[1] project, Fraunhofer IAIS and DATEV eG have teamed up to provide an application to query tables with natural language, i.e., Question Answering (QA) over tables. Our approach allows non-technical users to express what they want in the table more naturally in text form. Moreover, our solution integrates tables represented in various formats, e.g., CSV, Web Tables, or even tables encoded in XML, as in DATEV's use case. Figure 1 depicts the structure of the overall application. It consists of two proprietary components – the Fraunhofer Smart Data Connector (SDC) and the Fraunhofer QA Component. The SDC is a solution to create knowledge graphs by transforming heterogeneous enterprise data into actionable knowledge. The QA component allows answering questions expressed in natural language over the knowledge graph. The following section presents an overview of our approach that combines the SDC and QA components to provide a solution to the problem of answering questions over tables. The last section describes the demonstration of the use case in detail.

2. Architecture

2.1. Smart Data Connector

The SDC is a generic component to transform and store enterprise data sources into a knowledge graph. Figure 2a depicts a general overview of the SDC components and their interactions. Following a mapping approach [1], the SDC uses mapping rules to transform tables into RDF triples following the CSVW vocabulary. In DATEV's use case, we define a mapping rule to transform tables encoded in XML files to RDF. However, our component is generic enough to handle other scenarios, e.g., transforming CSV tables into RDF. At runtime, the SDC Engine

[1] cf. https://www.speaker.fraunhofer.de/en



(a) Components of the Smart Data Connector

(b) The CSVW vocabulary

Figure 2: Main elements of our solution

provides a service to upload XML files. Then, the Mapping Engine takes the file and the mapping rules and creates the semantic representation of the table using CSVW vocabulary. Figure 2b shows the main concepts from CSVW that we are using in this application. Once the tables are transformed into RDF, the SDC offers different services to the QA component. The SDC Entity-Relation Index provides a search service for entities and relations (including their labels, types, and synonyms), and it can bed used for Entity Linking tasks by the QA component. The SDC Engine provides a SPARQL endpoint used by the QA component to query the tables. Finally, the SDC Embeddings Generator offers advanced services, e.g., entity similarity based on embeddings. The SDC Embeddings Generator uses PyKEEN [2] to generate embeddings of the entities and relations of the graph with different models.

2.2. QA component

Our QA component is inspired by the TaPas model [3]. In our approach, the model is extended to answer questions from tables semantically described in knowledge graphs. TaPas is a deep learning model based on BERT's encoder architecture [4] and is specifically designed for question answering over tabular data. The TaPas model is built through two stages, pre-training, and fine-tuning. The pre-training has been done over millions of tables and related text segments crawled from Wikipedia and this is a crucial reason behind the performance of the model [3].

The fine-tuning process of TaPas has been done in a supervised fashion in multiple public datasets such as WIKISQL, WIKITQ, and SQA. Figure 3 depicts the architecture of the TaPas model. In addition to the BERT's encoder embeddings [4], the table data and structure are also encoded as inputs for the TaPas model. A table is flattened to a string format where the column headers and cells are concatenated as string tokens. Then question tokens are appended to the sequence. TaPas, trained in the weakly supervised setting, achieves close to state-of-the-art performance for WIKISQL (accuracy: 83.6). For SQA, TaPas leads to substantial improvements on all metrics: improving all metrics by at least 11 points. For WIKITQ [5] the model trained only from the original training data reaches an accuracy of 42.6, which surpasses similar approaches [6].

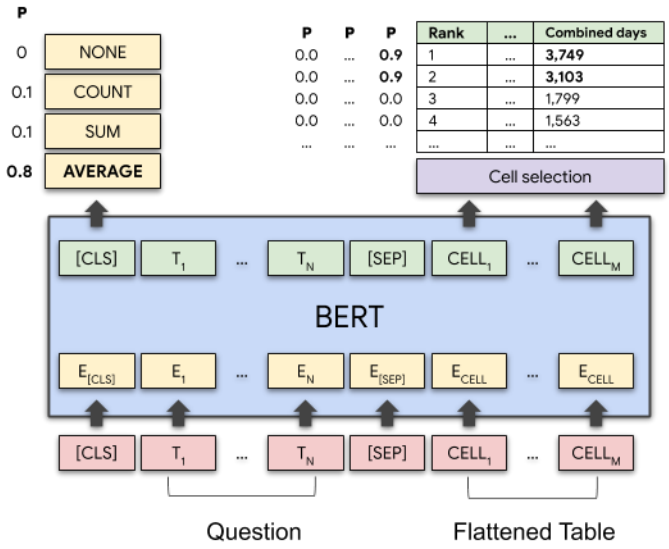


Figure 3: Overview of TaPas Architecture [3]

3. Demonstration

This demo shows the two-fold approach of our solution:

Building a knowledge graph from domain- and format-independent tables. The biggest challenge for the DATEV's use case is to map tables from different domains and formats without the additional effort of creating a complex ontology, keeping the mapping rules to a minimum. We select to reuse the CSV on the Web[2] (CSVW) vocabulary for having a generic semantic representation of the tables. Figure 2b shows the main concepts we use, i.e., "Table" and connects it with the concepts "Row", "Cell" and "Column". In the demo video, XML files from DATEV (document storage period, VAT rate in European countries, and payment initiation versions for the SEPA formats) are mapped to the CSVW with the rule "transform-tables". Using the query service in the SDC dashboard, the structure of the populated knowledge graph can be explored in more detail. The knowledge graph contains triples that connect the table URI with the corresponding columns and rows. There are also triples that link the content of a cell to the expected row via the row URI.

Answering questions over knowledge graph. The table URI is the key element for connecting with the "QA component". The user sends both the URI of the table and a question in natural language, e.g., "What is the intermediate VAT rate in Belgium?". Thanks to the use of a shared vocabulary for all tables, i.e., CSVW, the QA component can send a standard SPARQL to retrieve the table's columns and rows based on the URI. Then the Cells and Columns instances are pre-processed as embeddings used to represent the table in TaPas, i.e., column embeddings which indicate the column the token belongs to, row embedding, which indicates the row the token belongs to, rank embeddings which indicate the rank of a particular cell according to the

[2]https://www.w3.org/ns/csvw

column it belongs. Thus, our QA component can combine tables stored in knowledge graphs and the TaPas model with this information. It is also possible to answer questions requiring cell aggregation, e.g., "How many European countries have a standard VAT rate of 20 percent?" (COUNT), "What is the average standard VAT rate?" (AVERAGE), "Can you tell me the total years I need to keep all invoices?" (SUM). As a weakly supervised deep learning model, TaPas can represent the relationships between columns and values in tables and has an excellent semantic understanding of natural language queries. We can deploy TaPas for multiple use cases in multiple domains since the model has the ability for cross-domain [3].

4. Conclusions

The application to the DATEV use case is just one of many possible applications of our question answering over tables solution. This demo emphasizes the combination of RDF knowledge graphs with Language Models to solve the problem of answering questions over heterogeneous tables. In future work, we will explore improving the TaPas model by taking advantage of the semantically described tables. Moreover, we want to extend the TaPas model to the German language. Also, we plan to add more aggregation operators, like MINIMUM and MAXIMUM, into the TaPas architecture. Finally, the QA component can be improved to automatically identify the right table and answer without explicitly specifying the table URI. This way, it would be possible to answer questions from a pool of tables.

Acknowledgements: We acknowledge the support of the EU H2020 Projects Opertus Mundi (GA 870228), and the Federal Ministry for Economic Affairs and Energy (BMWi) project SPEAKER (FKZ 01MK20011A).

References

[1] A. Dimou, M. V. Sande, P. Colpaert, R. Verborgh, E. Mannens, R. V. de Walle, RML: A generic language for integrated RDF mappings of heterogeneous data, in: WWW, Seoul, Korea, volume 1184 of *CEUR Workshop Proceedings*, 2014.

[2] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, J. Lehmann, PyKEEN 1.0: A Python library for training and evaluating knowledge graph embeddings, J. Mach. Learn. Res. 22 (2021) 82:1–82:6.

[3] J. Herzig, P. K. Nowak, T. Müller, F. Piccinno, J. M. Eisenschlos, TaPas: Weakly supervised table parsing via pre-training, in: ACL, Online, 2020, pp. 4320–4333.

[4] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, 2019, pp. 4171–4186.

[5] Q. Liu, B. Chen, J. Guo, Z. Lin, J.-g. Lou, TaPEx: Table pre-training via learning a neural SQL executor, arXiv preprint arXiv:2107.07653 (2021).

[6] A. Neelakantan, Q. V. Le, M. Abadi, A. McCallum, D. Amodei, Learning a natural language interface with neural programmer, in: ICLR, Toulon, France, 2017.