From Liberating to Questioning Tabular Data in Documents Using Knowledge Graphs

Kautuk Raj[1,2,3] and Pierre Maret[1]

[1] Université de Lyon, CNRS UMR 5516 Laboratoire Hubert Curien, France
{kautuk.raj,pierre.maret}@univ-st-etienne.fr
[2] York University, Canada
[3] IIIT Bangalore, India

Abstract. Tables, a primary modality for organizing and presenting information for human comprehension, are ubiquitously found in documents. Their design poses significant challenges for systems, including large language models, when it comes to processing and understanding tabular data. We propose a novel method to free the tabular data encumbered inside documents (PDFs, HTML pages, Word documents, etc.) and perform question-answer (QA) on this data via natural language interaction. Our method stresses on its domain-agnostic and "open"-QA-oriented abilities, beating LLMs like ChatGPT in several situations. We achieve this using a combination of table extraction tools, followed by the creation of a knowledge graph using the tabular data sources and employing QAnswer[4], a QA system generator. A video demonstration showcases our tool's capabilities on United Nations (UN) disability documents and webpages.

Keywords: Question Answering · Knowledge Graph · Semantic Web

1 Introduction

Tables efficiently summarize information in a visually organized framework to facilitate comprehension and comparison across dimensions. They are commonly incorporated in scientific, business, and financial documents to present data.

Interpreting tabular information using standard language representations becomes challenging as table complexity increases [11]. PDF documents prioritize layout for human readers, which results in poor machine readability, requiring complex algorithms to recognize text, tables, and retrieve tabular structure [7].

Extensive research exists to extract and leverage valuable tabular data for natural language processing (NLP) tasks, including table QA which comprehends and reasons with tables to provide accurate answers to user questions.

Our approach involves extracting tables contained in a document using several open-source table extraction tools to convert them to comma-separated values (CSV) format. This CSV is then transformed into a resource description

[4] https://www.qanswer.eu/

format (RDF) file to create knowledge graphs corresponding to all tables in the document. We further enrich these knowledge graphs and leverage them as an additional, complementary data source to the original document text. The two data sources are integrated with QAnswer, a QA system generator, which is tuned to simultaneously query documents and knowledge graphs.

The main contributions of this work are:

- A domain-independent approach, unlike pre-trained transformers, enabling application to specialized and unobserved domain data.
- Capable of handling complex table layouts and hierarchies while preserving structure; not flattening tables thus retaining contextual information.
- Performs "open"-QA which retrieves relevant documents and extracts answers, essential for real-world user queries across extensive documents, unlike impractical "closed"-QA which unrealistically provides documents alongside questions.
- Advances over LLM-based methods by addressing challenges of inadequate table interpretation, token limits on large tables, hallucination and erroneous symbolic operations.

2 Related Work

Transformer-based methods such as [1], [5] excel and set benchmarks on open-domain datasets like WikiTableQuestions[5] and Natural Questions[6]. Tables appearing in such datasets exhibit simpler structures, lacking row headers and having a single, non-hierarchical column header [9]. Experiments [6] show that even advanced pre-trained transformers struggle with domain-specific table layouts.

Works like [1], [9] reduce tables to text, however, text-focused representations are sub-optimal for tables as they neglect special cell relationships [11].

Closest to our work, [8] utilizes knowledge graphs for table QA, but unlike ours, it demands table URIs for "closed"-QA and flattens tables to text strings post Wikipedia pre-training.

[2] evaluates LLMs on table QA datasets, noting their failures with "huge" tables due to token limits and doubts their ability to supplant symbolic methods. Evaluation of 14 LLMs in [10] reveals their imperfect factual knowledge grasp, particularly for non-popular entities.

3 Methodology

Table extraction is challenging since tables lack semantics, have varied layouts and span pages with repeating headers/footers requiring analysis of context and format for accurate association. Real-life tables appear embedded in text, necessitating abilities to handle multi-content QA.

[5] https://ppasupat.github.io/WikiTableQuestions/

[6] https://ai.google.com/research/NaturalQuestions/

The first step in our methodology extracts tables from documents using both deterministic and non-deterministic methods via open-source tools like Tabula[7], Camelot[8], and pdfplumber[9]. Discernible cell boundaries are parsed with OpenCV-based transformations while tables with whitespace separators are processed by detecting tabular areas, guessing column structure, and geometrically matching words to cells. Our method integrates the strengths of each tool, optimizing for context-specific performance. In the QA phase, it leverages the best-extracted output to formulate answers.

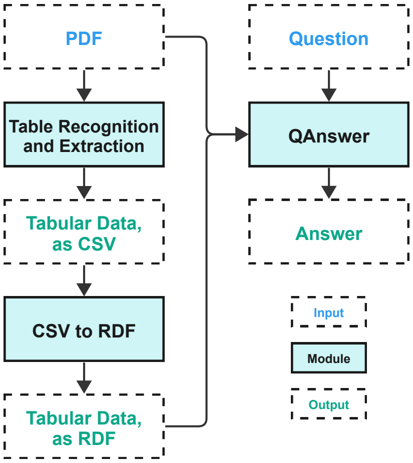

Fig. 1. Flow for the Proposed Methodology

The second step saves the tool outputs as a CSV file, enabling FAIR data principles compliance through openness and interoperability via a non-proprietary format.

The third step converts CSV data to RDF, the backbone for knowledge graphs, using the CSV on the Web (CSVW)[10] vocabulary which enables uniform semantic representation of tabular structures through RDF triples, streamlining alignment of diverse domain and format tables while avoiding elaborate ontology construction, rather maintaining a simplified set of mapping guidelines. The RDF format facilitates knowledge graph creation and increases data openness.

The final step enables combining QA systems over text (like PDFs) [4] and over knowledge graphs (RDF) [3] on the QAnswer platform. User queries are sent to both QA systems, which have been made combinable, so they execute the queries and compare results based on calculated confidence levels to determine the best answer to render to the user. Thus, we created one QA system with RDF data extracted from tables and another QA system with the original document texts from which tables were extracted. These systems have been integrated into

[7] https://tabula.technology/
[8] https://github.com/camelot-dev/camelot
[9] https://github.com/jsvine/pdfplumber
[10] https://www.w3.org/TR/tabular-data-primer/

a combined QA system that leverages both structured and unstructured data sources to execute queries and get better results.

4 Demonstration

A demonstration is presented to exemplify the capabilities of our tool on a corpus comprising publicly accessible United Nations (UN) documents and webpages sourced from the disability domain. An end-to-end presentation of the method along with test runs and comparisons can be found in a video accessible at https://youtu.be/ve6xCwP1LHs.

5 Conclusions and Future Work

We present a pioneering approach to liberate tabular data within documents and perform QA via NLP interaction, surpassing LLMs and existing baselines in initial tests. We intend to evaluate the approach on a popular benchmark dataset. Future plans include numerical and symbolic reasoning, multilingual table querying, and user interface (UI) enhancements for combined QA systems.

Acknowledgements This research occurs in the context of the Disability Wiki project[11], a collaboration between Laboratoire Hubert Curien, York University (Canada) and The QA Company[12].

References

1. Alberti, C., et al.: A BERT baseline for the Natural Questions (2019), http://arxiv.org/abs/1901.08634
2. Chen, W.: Large Language Models are few(1)-shot table reasoners. In: EACL 2023
3. Diefenbach, D., et al.: Qanswer KG: Designing a Portable Question Answering System over RDF Data. In: ESWC 2020
4. Guo, K., et al.: QAnswer: Towards Question Answering Search over Websites. In: Web Conference 2022. ACM
5. Herzig, J., et al.: TaPas: Weakly Supervised Table Parsing via Pre-training. In: ACL 2020
6. Katsis, Y., et al.: AIT-QA: Question Answering Dataset over Complex Tables in the Airline Industry. In: NAACL 2022. ACL
7. Khusro, S., et al.: On methods and tools of table detection, extraction and annotation in PDF documents. Journal of Information Science
8. Knoblach, J., et al.: Combining Knowledge Graphs and Language Models to Answer Questions over Tables. 18th SEMANTiCS 2022
9. Oğuz, B., et al.: UNIK-QA: Unified representations of structured and unstructured knowledge for Open-Domain question answering. NAACL 2022
10. Sun, K., et al.: Head-to-Tail: How Knowledgeable are Large Language Models (LLM)? (2023), https://arxiv.org/abs/2308.10168
11. Zayats, V., et al.: Representations for Question Answering from Documents with Tables and Text. In: EACL 2023

[11] https://www.christoelmorr.ca/ai-disability-advocacy.html
[12] https://the-qa-company.com/