Knowledge Graph-driven Tabular Data Discovery from Scientific Documents

Vijay S. Kumar[1,*], Varish Mulwad[2], Jenny Weisenberg Williams[1], Tim Finin[3], Sharad Dixit[1] and Anupam Joshi[3]

[1]GE Research, 1 Research Circle, Niskayuna, NY, USA
[2]GE Research, John F. Welch Technology Center, Whitefield, Bengaluru, India
[3]University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD, USA

Abstract

Synthesizing information from collections of tables embedded within scientific and technical documents is increasingly critical to emerging knowledge-driven applications. Given their structural heterogeneity, highly domain-specific content, and diffuse context, inferring a precise semantic understanding of such tables is traditionally better accomplished through linking tabular content to concepts and entities in reference knowledge graphs. However, existing tabular data discovery systems are not designed to adequately exploit these explicit, human-interpretable semantic linkages. Moreover, given the prevalence of misinformation, the level of confidence in the reliability of tabular information has become an important, often overlooked, factor in discovery over open datasets. We describe a preliminary implementation of a discovery engine that enables table-based semantic search and retrieval of tabular information from a linked knowledge graph of scientific tables. We discuss the viability of semantics-guided tabular data analysis operations, including on-the-fly table generation under reliability constraints, within discovery scenarios motivated by intelligence production from documents.

Keywords

Scientific tables, semantic tabular data discovery, on-the-fly table generation, data fusion

1. Introduction

Tables are ubiquitous across domains in organizing and concisely communicating information in a structured form. Increasing democratization of generative AI techniques and the popularity of their application in the comprehension, creation, and refinement of complex multimodal digital content such as technical documents is triggering a fresh revisit of tables—often, a key structured data component embedded within various kinds of published documents ranging from scientific papers and preprint articles to patents and contractual agreements to intelligence reports and impact assessment statements. We use the term "scientific tables" to denote this sub-category of tabular data objects whose primary role (alongside other structured artifacts like charts) is one of supplementing a document's textual content with vital visual cues, and whose access and interpretation is significantly driven by how they are organized within their containing documents. While extensive research [1, 2, 3, 4] has addressed advanced challenges in discovering, analyzing, and integrating large-scale, diverse tabular data that exists within enterprise data systems or as web tables and open data published on the web, scientific tables pose a unique combination of challenges that necessitate a deeper exploration of such tables to support the information discovery and analysis needs of emerging applications like AI research assistants [5] and intelligence report generation [6].

As part of a broader initiative to help understand and systematically explore scientific tables, we developed an end-to-end framework to: **(i)** harvest tables and associated metadata on-demand from online open-access scientific publications (such as those hosted by PubMed Central [7]), **(ii)** infer the intended meaning of scientific tables via a two-stage semantic table interpretation process that links tabular data to reference knowledge graphs, **(iii)** further contextualize scientific tables with provenance-based estimates of their information reliability, and **(iv)** support rich semantic search capabilities over the ensuing knowledge graph of scientific tables.

While our prior work [8, 9] highlighted the knowledge extraction and knowledge graph (KG) construction aspects of our framework, this paper specifically addresses the relatively under-explored problem of data discovery from a scientific-tables-centric KG, and describes our approach and preliminary implementation of a tabular data discovery system driven by knowledge graph technology.

Joint Workshops at 49th International Conference on Very Large Data Bases (VLDBW'23) — TaDA'23: Tabular Data Analysis Workshop, August 28 - September 1, 2023, Vancouver, Canada

[*]Corresponding author.

v.kumar1@ge.com (V. S. Kumar); varish.mulwad@ge.com (V. Mulwad); weisenje@ge.com (J. W. Williams); finin@umbc.edu (T. Finin); sharad.dixit@ge.com (S. Dixit); joshi@umbc.edu (A. Joshi)
0000-0003-2234-1546 (V. S. Kumar); 0000-0001-9113-5952 (V. Mulwad); 0000-0002-6593-1792 (T. Finin); 0000-0002-8641-3193 (A. Joshi)

2. Scientific Tables

Tables in scientific documents often capture a point-in-time summary view or meta-analysis over a more comprehensive set of information——including over large structured datasets hosted across diverse locations on the web (e.g., open data portals, data markets, research data repositories, etc.). While these latter datasets typically feed data preparation tasks aimed at automating data science and analysis pipelines, we focus more on the former kind of tables to automate technical content generation pipelines. Specifically, we seek to enable technical experts and analysts to (i) efficiently discover useful information in existing scientific tables, (ii) analyze knowledge extracted from across these tables in the wider context of their visual appearance and reliability, and (iii) present their learnings and valuable information (again, in the form of scientific tables) for possible inclusion in new documents or reports. Our work is motivated by how intelligence community standards instruct analysts to "incorporate effective visual presentations" of information (including via tables) to enhance the overall usefulness of intelligence reports [10]. Additionally, we derive inspiration from a recent trend of scientists resorting to AI and conversational search engines as an evolving modern-day lazyweb personification for generating tabular content in lieu of conducting the research themselves.

2.1. Characteristics and Challenges

The very practices seeking to ease human comprehension of scientific tables also introduce challenges for machine-driven table understanding. Scientific tables exhibit certain distinctive characteristics borne out of the general circumstances of their creation:

1. **High structural heterogeneity:** Constrained by 'publication real estate' and desire to place tables in close proximity to any accompanying text, collections of scientific tables display high structural variability, even more so than web tables. Data discovery and integration systems with data and schema matching techniques that assume 'well-structured' or relational tables do not adequately address this complexity.

2. **Domain-specific entities:** Like open datasets, scientific tables typically contain more numerical cell content than text. Where they do contain text, it is usually in the form of literals or idiomatic strings and entities specific to a scientific domain. Data semantics [11] play a key role in disambiguating such content.

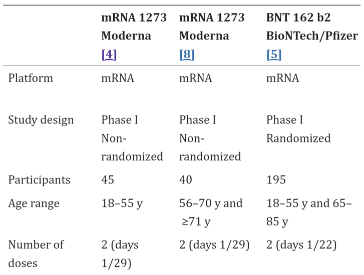
3. As with web tables, scientific tables exhibit **diffuse context** wherein one must draw upon additional contextual information that lies outside an individual table cell (or, even an entire table body) to infer the semantics of its content. Based on how tables are visually formatted to optimize informational content for human consumption, this context may include inferred semantics of other cells in a row or column, table captions or other descriptive text (from within the body of the containing documents) that refers to these tables.

4. In this era of scientific misinformation and non-peer-reviewed preprints, there is a dearth of approaches to tackle the lack of information **reliability of scientific tables**, as is the case with web tables and open datasets.

In [8], we describe in detail our solution to address some of these challenges via extensive structural characterization of over 120,000 tables drawn from scientific publications, and by matching tabular content to reference knowledge graphs with high precision. We also developed a practical entity linker [9], adaptable to different domains, to efficiently match COVID-19-related scientific tables to Wikidata [12].

2.2. Data Discovery from Scientific Tables

As an illustrative example, consider an analyst seeking meta-analyses information about *phase I clinical trials for COVID-19 vaccines developed around the world.* Unless such information is centrally curated, it will be more expeditiously available directly within scientific tables in published documents (e.g., figure 1).



	mRNA 1273 Moderna [4]	**mRNA 1273 Moderna [8]**	**BNT 162 b2 BioNTech/Pfizer [5]**
Platform	mRNA	mRNA	mRNA
Study design	Phase I Non-randomized	Phase I Non-randomized	Phase I Randomized
Participants	45	40	195
Age range	18–55 y	56–70 y and ≥71 y	18–55 y and 65–85 y
Number of doses	2 (days 1/29)	2 (days 1/29)	2 (days 1/22)

Figure 1: Snippet from a real table, (PMC8114590, Table 1)

While some publisher search services [13] can specifically return tables from relevant papers, such matches are not based on tabular data. In reality, most tabular content in scientific documents is not directly accessible even in instances where they are internally maintained as machine-readable (e.g., HTML-formatted) objects. Assuming that analyst queries are best served by synthesizing tabular responses and that the primary source of information for doing so are existing scientific tables, one could potentially adapt current semantic schema matching techniques to help discover ranked lists of tables with matching content. However, as with open datasets, it is unlikely that individual scientific tables will contain all information requested by queries. Instead, relevant "scientific *views*" must be composed on the fly by suitably merging content from multiple scientific tables. We introduce technical challenges with on-the-fly generation of relational scientific tables in response to search requests under contextual constraints and briefly describe our heuristic approach to address them in section 3.3.

3. Technical Approach

Our high-level strategy for tabular data discovery is one of knowledge-based analysis to generate new tables on the fly using information integrated from multiple scientific tables. Our goal is to populate a result table with new knowledge, potentially even one cell at a time.

- We first constructed a KG from scientific tables [8]—Specifically, we perform column type annotation (CTA) and cell entity annotation (CEA) [14]—i.e., header and body cells are automatically annotated with Wikidata concepts and entities respectively (e.g., the cell with content 'Platform' from the table in figure 1 is linked to entity with QID: Q108028785: "Vaccine Platform"). Each table's structural assessments, inferred semantics, and provenance-based estimates of reliability are all encoded as RDF triples in accordance with linked data principles.
- We then designed and implemented a prototype system for ultimately discovering tabular data from this KG of scientific tables. As an initial step towards discovery and on-the-fly generation [15], we first developed foundational capabilities—including a query model and discovery engine—to enable table-based search over our KG under rich contextual constraints. Finally, we extended these foundational capabilities with a preliminary, heuristic approach to identify and fuse the content of *semantically compatible* scientific tables on the fly via union operations. Our overall approach is analogous to the 'reference architecture' approach described in [16] to discover project-join views.

3.1. Table-based Semantic Search

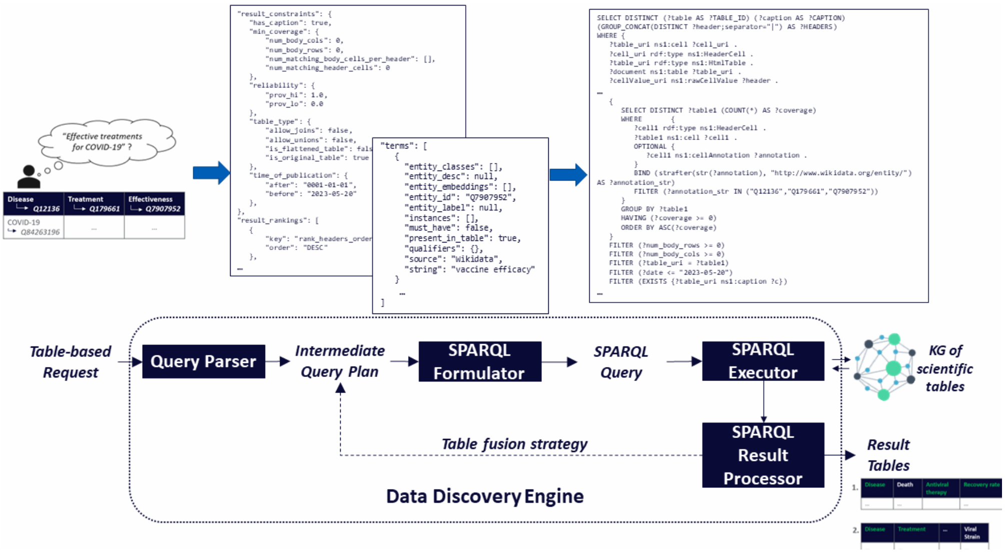
Search requests against our KG can take the form of a keyword list or a (potentially partially-specified query-by-example) [16]) input table−−which can then be semantically resolved to reduce ambiguity—along with any associated contextual constraints. In response to a request, we match the semantics of the query table with inferred semantics of tables in our KG. Since KG queries operate at the granularity of low-level subgraph triple patterns, we elevate scientific tables and relational-style analysis operations on these tables to first-class citizens in our KG. We parse an input table-based search request into an intermediate query plan comprising a set of abstract foundational primitives: SELECT, FILTER, RANK (and FUSE). Akin to relational algebra operators, SELECT logically returns a list of identifiers for all tables that semantically match the query table. FILTER prunes this list by applying one or more temporal, cell coverage-based, or reliability-based constraints on matching tables. RANK orthogonally reorders the list of tables based on some ranking criteria. Any table-based search request can be expressed as a query plan comprising these primitives.

3.2. Tabular Data Discovery Engine

Intermediate query plans are automatically translated into subgraph triple pattern-matching queries for execution against our KG of scientific tables. We built a tabular data discovery engine to incrementally construct SPARQL queries by adding or modifying ad hoc graph patterns corresponding to each primitive instance in a query plan. As depicted in figure 2, the discovery of relevant existing scientific tables without on-the-fly table generation requires a single pass over the engine's components. This engine is also responsible for packaging the results of SPARQL query execution in the form of relational result tables for easy consumption.

By translating query plans into SPARQL as shown, our engine is effectively emulating "database-like" analysis against tables in our KG—where each primitive's implementation is driven by explicit semantic linkages automatically inferred for each table. SELECT compares the set of header-cell QIDs for the query table against those for each KG table. If the two sets of QIDs overlap completely, or if set overlap exceeds some threshold specified in the request, then the candidate table is deemed semantically similar and included in the results. By default, RANK sorts a list of result tables based on this header-cell coverage metric (i.e., tables with maximum QID set overlap are ranked higher). Besides exact comparison of header-cell QIDs, the engine also optionally supports QID similarity based on pre-trained knowledge graph embeddings from Wembedder [17].

While a detailed algorithmic description of query translation is beyond the scope of this paper, in general, given a search request, a Query Parser identifies and assembles (in a specific order) all information needed to formulate a SPARQL SELECT query—including: return variables and limits, (subject, predicate and object) for base triple patterns, entire subgraph patterns (if applicable), variable bindings and clauses for the query constraints, and ranking preferences. A SPARQL Formulator then acts on these inputs one by one in the prescribed order, expanding them into actual triple patterns, nested sub-queries, and clauses (FILTER, HAVING, OPTIONAL, etc.) as required, to produce a functional SPARQL query.



```
"result_constraints": {
   "has_caption": true,
   "min_coverage": {
       "num_body_cols": 0,
       "num_body_rows": 0,
       "num_matching_body_cells_per_header": [],
       "num_matching_header_cells": 0
   },
   "reliability": {
       "prov_hi": 1.0,
       "prov_lo": 0.0
   },
   "table_type": {
       "allow_joins": false,
       "allow_unions": false,
       "is_flattened_table": fals
       "is_original_table": true
   },
   "time_of_publication": {
       "after": "0001-01-01",
       "before": "2023-05-20"
   },
},
"result_rankings": [
   {
       "key": "rank_headers_order
       "order": "DESC"
   },
…
```

```
"terms": [
   {
     "entity_classes": [],
     "entity_desc": null,
     "entity_embeddings": [],
     "entity_id": "Q7907952",
     "entity_label": null,
     "instances": [],
     "must_have": false,
     "present_in_table": true,
     "qualifiers": {},
     "source": "Wikidata",
     "string": "vaccine efficacy"
   }

    …
]
```

```
SELECT DISTINCT (?table AS ?TABLE_ID) (?caption AS ?CAPTION)
(GROUP_CONCAT(DISTINCT ?header;separator="|") AS ?HEADERS)
WHERE {
   ?table_uri ns1:cell ?cell_uri .
   ?cell_uri rdf:type ns1:HeaderCell .
   ?table_uri rdf:type ns1:HtmlTable .
   ?document ns1:table ?table_uri .
   ?cellValue_uri ns1:rawCellValue ?header .
…
   {
      SELECT DISTINCT ?table1 (COUNT(*) AS ?coverage)
      WHERE       {
         ?cell1 rdf:type ns1:HeaderCell .
         ?table1 ns1:cell ?cell1 .
         OPTIONAL {
            ?cell1 ns1:cellAnnotation ?annotation .
         }
         BIND (strafter(str(?annotation), "http://www.wikidata.org/entity/")
AS ?annotation_str)
         FILTER (?annotation_str IN ("Q12136","Q179661","Q7907952"))
      }
      GROUP BY ?table1
      HAVING (?coverage >= 0)
      ORDER BY ASC(?coverage)
   }
   FILTER (?num_body_rows >= 0)
   FILTER (?num_body_cols >= 0)
   FILTER (?table_uri = ?table1)
   FILTER (?date <= "2023-05-20")
   FILTER (EXISTS {?table_uri ns1:caption ?c})
…
```

Figure 2: Discovery Engine: High-level system architecture

3.3. On-the-fly Table Generation

On-the-fly generation of tables or views by fusing the relevant portions of a list of existing tables brings additional complex challenges to knowledge-based analysis, e.g., determining if a pair of candidate tables are compatible for merge operations based on their inferred semantics, determining the order in which a list of candidate tables must be merged to produce an optimal tabular response, assessing the order in which different kinds of merge operations (e.g., union, join, cell-expand) need to be applied on the tables, etc. Moreover, the context needed to effectively merge content across tables may come from other sources like reliability assessments captured in our KG. Knowledge derived from other modalities such as text may act as a 'bridge' between a pair of tables where compatibility may not be established based on semantics of table content alone.

To demonstrate on-the-fly table generation capability, we implemented a preliminary dynamic approach to fuse content of relevant tables via union operations. Ours is a heuristic-based approach that greedily seeks to maximize the amount of populated cells in the result table via new rows while satisfying provenance-based reliability constraints. Our engine breaks up a list of initial candidate tables into groups where tables in each group have identical header-cell set overlap with the query table. It then performs a union of tables within each group to create super-tables. If there is partial set overlap across super-tables, it performs a union across groups in decreasing order of their set overlap sizes. Finally, these merged tables are returned as query results ranked in decreasing order of their number of rows.

4. Conclusions

Supporting tabular data discovery over collections of tables published in scientific documents presents unique challenges that require use of the explicit meaning of scientific tables as inferred by linking them to reference knowledge graphs. We described our approach and implementation of a novel engine that can discover tabular data from knowledge graphs of scientific tables with early support for automatically merging content from multiple tables on the fly via union operations. While preliminary, we believe our approach is foundational and can be highly effective when expanded to cover on-the-fly cell-based table expansion techniques. We believe this work will motivate new research directions in knowledge-guided scientific table generation and analysis at large scales.

Acknowledgments

This research is based on work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via [2021-21022600004]. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government.

References

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

[2] S. Zhang, K. Balog, Web Table Extraction, Retrieval, and Augmentation: A Survey, ACM Trans. Intell. Syst. Technol. 11 (2020). doi:10.1145/3372117.

[3] A. Chapman, E. Simperl, L. Koesten, G. Konstantinidis, L.-D. Ibáñez, E. Kacprzak, P. Groth, Dataset Search: A Survey, The VLDB Journal 29 (2019) 251–272. doi:10.1007/s00778-019-00564-x.

[4] D. Brickley, M. Burgess, N. Noy, Google Dataset Search: Building a Search Engine for Datasets in an Open Web Ecosystem, in: The World Wide Web Conference, WWW '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 1365–1375. doi:10.1145/3308558.3313685.

[5] Elicit: The AI Research Assistant, Ought, 2022. https:elicit.org.

[6] REASON - Rapid Explanation, Analysis and Sourcing ONline, IARPA, 2023. https://www.iarpa.gov/research-programs/reason.

[7] PMC Open Access Subset, National Library of Medicine, Bethesda, MD, 2003. https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/.

[8] V. Mulwad, V. S. Kumar, J. Weisenberg Williams, T. Finin, S. Dixit, A. Joshi, Towards Semantic Exploration of Tables in Scientific Documents, in: ESWC 2023 Workshops and Tutorials Joint Proceedings, volume 3443 of *CEUR Workshop Proceedings*, 2023. URL: https://ceur-ws.org/Vol-3443/ESWC_2023_SemTech4STLD_paper_2.pdf.

[9] V. Mulwad, T. Finin, V. S. Kumar, J. Weisenberg Williams, S. Dixit, A. Joshi, A Practical Entity Linking System for Tables in Scientific Literature, in: Proceedings of the Workshop on Scientific Document Understanding (SDU 2023), colocated with 37th AAAI Conference on Artificial Inteligence (AAAI), 2023.

[10] Army Techniques Publication (ATP) 2-33.4. Intelligence Analysis, 2020. URL: https://irp.fas.org/doddir/army/atp2-33-4.pdf.

[11] U. Khurana, K. Srinivas, S. Galhotra, H. Samulowitz, A Vision for Semantically Enriched Data Science, 2023. arXiv:2303.01378.

[12] D. Vrandečić, M. Krötzsch, Wikidata: A Free Collaborative Knowledgebase, Commun. ACM 57 (2014) 78–85. doi:10.1145/2629489.

[13] The New England Journal of Medicine - Advanced Search, NEJM, 2023. https://www.nejm.org/search?searchType=advancedSearch&allWords=covid19&searchWithin=fullText&objectType=nejm-media&mediaType=Table.

[14] N. Abdelmageed, J. Chen, V. Cutrona, V. Efthymiou, O. Hassanzadeh, M. Hulsebos, E. Jiménez-Ruiz, J. Sequeda, K. Srinivas, Results of SemTab 2022, volume 3320 of *CEUR Workshop Proceedings*, 2022. URL: https://ceur-ws.org/Vol-3320/paper0.pdf.

[15] S. Zhang, K. Balog, On-the-fly Table Generation, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 595–604. doi:10.1145/3209978.3209988.

[16] Y. Gong, Z. Zhu, S. Galhotra, R. C. Fernandez, Ver: View Discovery in the Wild, 2022. arXiv:2106.01543.

[17] F. Årup Nielsen, Wembedder: Wikidata entity embedding web service, 2017. arXiv:1710.04099.