A Large Scale Test Corpus for Semantic Table Search

Aristotelis Leventidis
leventidis.a@northeastern.edu
Northeastern University
Boston, USA

Martin Pekár Christensen
mpch@cs.aau.dk
Aalborg University
Aalborg, Denmark

Matteo Lissandrini*
matteo.lissandrini@univr.it
University of Verona
Verona, Italy

Laura Di Rocco
la.dirocco@northeastern.edu
Northeastern University
Boston, USA

Katja Hose†
katja.hose@tuwien.ac.at
Technische Universität Wien
Vienna, Austria

Renée J. Miller
miller@northeastern.edu
Northeastern University
Boston, USA

ABSTRACT

Table search aims to answer a query with a ranked list of tables. Unfortunately, current test corpora have focused mostly on needle-in-the-haystack tasks, where only a few tables are expected to exactly match the query intent. Instead, table search tasks often arise in response to the need for retrieving new datasets or augmenting existing ones, e.g., for data augmentation within data science or machine learning pipelines. Existing table repositories and benchmarks are limited in their ability to test retrieval methods for table search tasks. Thus, to close this gap, we introduce a novel dataset for *query-by-example* Semantic Table Search. This novel dataset consists of two snapshots of the large-scale Wikipedia tables collection from 2013 and 2019 with two important additions: (1) a page and topic aware ground truth relevance judgment and (2) a large-scale DBpedia entity linking annotation. Moreover, we generate a novel set of entity-centric queries that allows testing existing methods under a novel search scenario: *semantic exploratory search.* The resulting resource consists of 9,296 novel queries, 610,553 query-table relevance annotations, and 238,038 entity-linked tables from the 2013 snapshot. Similarly, on the 2019 snapshot, the resource consists of 2,560 queries, 958,214 relevance annotations, and 457,714 total tables. This makes our resource the largest annotated table-search corpus to date (97 times more queries and 956 times more annotated tables than any existing benchmark). We perform a user study among domain experts and prove that these annotators agree with the automatically generated relevance annotations. As a result, we can re-evaluate some basic assumptions behind existing table search approaches identifying their shortcomings along with promising novel research directions.

CCS CONCEPTS

• **Information systems** → **Information retrieval**.

KEYWORDS

table search, semantic search, benchmark, query-by-example

ACM Reference Format:
Aristotelis Leventidis, Martin Pekár Christensen, Matteo Lissandrini, Laura Di Rocco, Katja Hose, and Renée J. Miller. 2024. A Large Scale Test Corpus for Semantic Table Search. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3626772.3657877

1 INTRODUCTION

Tables are one of the most used models for organizing data in almost any domain [1, 5, 10]. Numerous approaches have been proposed to retrieve tables within large tabular data corpora, i.e., *the table search task* [36, 38]. This includes Web tables, such as Wikipedia tables [2], tabular data represented within data lakes, and Open Data repositories [28].

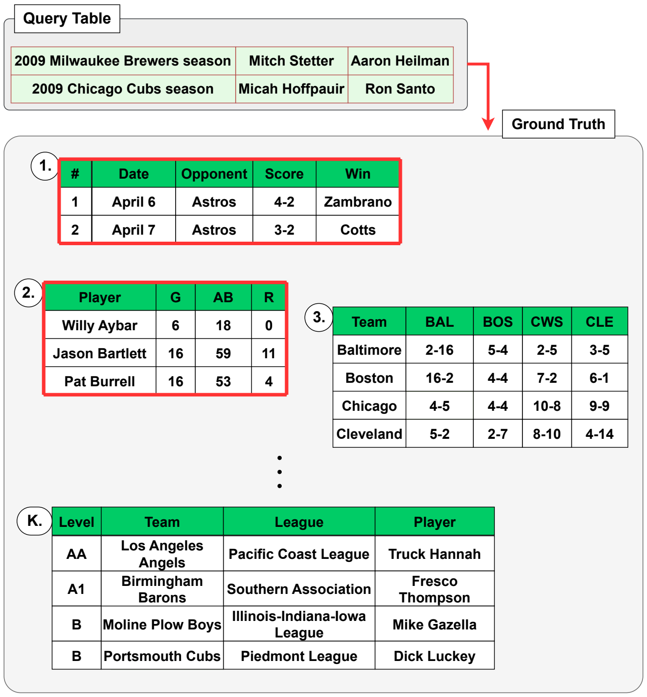
There are two common variants of table search: the first expects the *information need* to be expressed as a keyword query, and the second accepts an existing *query* table, i.e., "*query-by-example-table*" [38] (or QbE for short). For keyword queries, the expectation is that a user will consider the first few (highest ranked) answers, analogous to Web page search. QbE is instead often required by users who need to identify new datasets or to expand an initial dataset they have at hand, i.e., example-based exploratory search [25, 38]. Hence, in the latter case, the search engine may retrieve a larger set of tables (all qualifying tables), which could be relevant even when their content does not exactly overlap with the content of the query. For example, in dataset augmentation, it is important to identify all tables that can provide additional features or samples that do not appear in the query table [27, 38].

Therefore, the relevance of a candidate table extends beyond simple content matching and requires an understanding of *the semantics* of the query table [15, 25, 36, 38]. This task is referred to as *Semantic Table Search (STS)* [25, 36]. Thus, in some proposals, an STS engine can also exploit a reference knowledge graph (KG) to enable entity-centric similarity measurements of KG entities which thereby allows to rank tables by semantic relevance [4, 12, 13, 19].

Example 1.1. Consider a betting company analyzing baseball teams and lead players to cross-reference their performance. Given some baseball teams of interest, an initial query table would contain some players from two teams of interest, as in Figure 1. A data scientist within the betting company then executes this query in

[*] Also with Aalborg University, matteo@cs.aau.dk.
[†] Also with Aalborg University, khose@cs.aau.dk.

SIGIR '24, July 14–18, 2024, Washington, DC, USA
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657877


Query Table
Ground Truth
1.
2.
3.
K.

2009 Milwaukee Brewers season	Mitch Stetter	Aaron Heilman
2009 Chicago Cubs season	Micah Hoffpauir	Ron Santo

#	Date	Opponent	Score	Win
1	April 6	Astros	4-2	Zambrano
2	April 7	Astros	3-2	Cotts

Player	G	AB	R
Willy Aybar	6	18	0
Jason Bartlett	16	59	11
Pat Burrell	16	53	4

Team	BAL	BOS	CWS	CLE
Baltimore	2-16	5-4	2-5	3-5
Boston	16-2	4-4	7-2	6-1
Chicago	4-5	4-4	10-8	9-9
Cleveland	5-2	2-7	8-10	4-14

Level	Team	League	Player
AA	Los Angeles Angels	Pacific Coast League	Truck Hannah
A1	Birmingham Barons	Southern Association	Fresco Thompson
B	Moline Plow Boys	Illinois-Indiana-Iowa League	Mike Gazella
B	Portsmouth Cubs	Piedmont League	Dick Luckey

Figure 1: An example query table containing baseball teams and players and a set ground truth semantically relevant tables. Notice that a keyword-based approach will be unable to retrieve tables such as 1 and 2 even though they are highly relevant to the query.

an Open Data portal for other datasets to cross-reference their results. The STS engine then retrieves tables recording similar data for other baseball teams or players, as well as player transfers between teams and team results in different games as context. Yet, the STS engine should also recognize when information is not so relevant or likely irrelevant, e.g., a list of teams and player names but from different sports, even if the teams are from the same cities as the one in the query table. Figure 1 illustrates a set of tables that are semantically relevant to a query table about lead baseball players playing in two different teams in the same season. With this information, the betting company is able to understand the performance of players and teams and is therefore better prepared to offer bets to its customers. For example, the betting company can learn from Table 1 in Figure 1 that baseball teams with Zambrano and Cotts perform well against the Astros and that Cleveland is the weakest team according to Table 3.

Example 1.1 and Figure 1 describe a sample of top-relevant tables that are relevant even though they do not mention any baseball players or seasons from the query. Moreover, tables ranked as top-1 and -2 (marked in the figure) cannot be retrieved by any keyword search method, since they do not share any keyword match, despite being the most relevant in the set. However, all the tables in that example output are semantically relevant given that they contain information describing baseball teams, baseball players, and various statistics and relations involving them.

To enable a better understanding of the semantics of a dataset, we have witnessed an increasing interest in matching the content of a table with entities from an existing Knowledge Graph (KG) [2, 6, 16, 22]. KGs present information as a semantic graph connecting both entities and concepts with semantic relationships.

Thus, given the importance of studying advanced methods to support this type of search, it is necessary to compare existing and future solutions across a fair and extensive set of benchmarks. Recently, a few benchmark datasets have been proposed for the task of table search [2, 6, 16, 17, 22, 36]. Among those, the most relevant datasets are a dataset proposed to evaluate ad-hoc table retrieval methods [6] and the WTR (its extension) developed also for the task of Web table retrieval. Unfortunately, *these datasets suffer from a variety of limitations in terms of scope and quality:*

(1) They do not provide out-of-the-box linking to an existing KG for entities both in queries and tables and thus they are not suited to properly test semantic QbE search methods.
(2) Furthermore, the datasets that come with relevance annotations [6, 17, 36] offer a ground truth that is biased towards exact-content-match methods due to the pre-filtering of the results in their ground truth.
(3) Finally, they only provide a handful of queries for which only a few results are marked as relevant, hence limiting the opportunity to train and test machine learning models that need substantial amounts of training data.

Therefore, *we propose a new dataset, the Semantic Table Search corpus* (STSD)[1] *that overcomes these limitations.* In particular, we provide the following contributions:

(1) We extend the commonly adopted WikiTable corpus from 2013 [2] and 2019 [3] with automatically extracted links to DBpedia, resulting in 238K and 457K tables, respectively. For the first time, we additionally construct a large set of 9K and 2K query tables from the 2013 and 2019 WikiTable corpus, respectively for QbE table search.
(2) We extract ground truth relevance annotations for our set (see Figure 1) using Wikipedia categories and navigational links as relevance signals. Our ground truth relevance annotations comprise 610K relevance scores across all 9K queries for the 2013 WikiTable corpus and 958K relevance scores across all 2K queries for the 2019 WikiTable corpus. The reliability of our relevance annotations has been further verified via a crowdsourced manual annotation process.
(3) We furthermore share the RDF2Vec embeddings (on Zenodo [8]) for all entities in the DBpedia snapshot we used.

To ease access and interoperability of our dataset, we publish our STSD repository on GitHub, and we maintain the same table file names as the original datasets from which it is derived. This allows us to compare our STSD to other works that use a subset of the same tables and to retrieve the original files as well. Our dataset is *the first that can enable an evaluation of existing semantic table search methods that is not biased towards content overlap between queries and tables* (since we do not perform any content-based prefiltering) and *allows testing their scalability to large-scale repositories of tables*

[1] https://github.com/EDAO-Project/SemanticTableSearchDataset

(with 2 to 3 orders of magnitude more queries and tables than other existing datasets).

In this paper, we first present a more in-depth analysis of the use cases for our dataset (Section 2), and then we discuss the limitations of existing benchmarks (Section 3). Furthermore, we describe how we built our test collection (Section 4). Finally, we evaluate our test collection over two baselines designed to exploit the semantic annotations present in our dataset and discuss new promising research directions (Sections 5). We consider how our dataset can shed more light into the successes and failures of semantic table search methods sparking new insights that may help to advance this important field. We conclude our paper in Section 6.

2 TASK DEFINITION

In semantic table search (STS), the task is to rank a set of top-k tables in a corpus given an input query [36, 38]. The task that this resource aims to benchmark is example-driven top-k semantic ranking. This means that input queries are given as small tables containing information of interest organized in rows and columns. For example, if a user wants to find tables describing baseball players, teams, games, and statistics, the user can pose a table query containing example baseball players and teams, such as in Figure 1. Therefore, the ranking of tables is based not only on exact matches between the query table and the corpus tables, but also on the semantic similarities between entities. This necessitates access to semantic information which can be achieved by referencing entities in KGs.

A KG is a directed labeled graph G=$\langle\mathcal{N},\mathcal{E},\lambda\rangle$, where nodes $\mathcal{N}$ consist of entities, concepts, and their attributes and edges $\mathcal{E}$ are labeled relationships between nodes. The nodes and edges are usually annotated with literals $\mathcal{L}$ by a mapping function $\lambda : \mathcal{N}{\cup}\mathcal{E}{\mapsto}\mathcal{L}$. Given a reference KG, tables can be ranked based on the semantic similarity of entities within tables. For example, the entities `Mitch Stetter` and `Micah Hoffpauir` are not exact matches but have a high semantic similarity when considering their set of entity types, their attributes, or the distance from each other in the KG.

We can now define STS as follows:

Problem 1 (Semantic Table Search). *Given a table corpus C and a table query Q as input, both with mappings to entities from a knowledge graph G, the Semantic Table Search task requires extracting from C a top-k ranked list of tables that are semantically relevant to Q according to a semantic relevance scoring function* $\textsc{SemRel}_G(Q,T)$.

Therefore, the answer to an STS example query table Q consists of a top-k ranking of tables from the corpus C. The query table can be seen as a set of tuples. For this ranking to be effective, each entity mention m in a table $T \in C$ is linked to a corresponding entity $e \in G$, where G is an instance of a reference KG. This entity linking can be defined as the mapping function $\Phi{:}M{\mapsto}\mathcal{N}$, where M is the set of entity mentions across C and Q. Note that the entity links in tables also allow our dataset to be used to benchmark the performance of entity linkers.

Moreover, there are also parallels between our semantic table search and document search. Standard document search frameworks, such as those based on BM25 like Lucene, can be extended with semantic weighing of query terms [30]. In this type of semantic document search, however, only documents containing exact matching are retrievable. Alternatively, another option is to identify the concepts in the query and in the documents and then compute the concept overlap to rank documents [11]. Also in this case, the search paradigm tries to identify documents containing exact matches of the keywords or entities present in the query.

3 LIMITATIONS OF EXISTING BENCHMARKS

Table search methods tackle a variety of problems, such as discovery of joinable tables [40, 41], unionable tables [29], related tables [4, 33], and augmentation search based on textual matches [35, 38]. In this section, we survey existing methods and existing datasets to test these methods. Our analysis reveals that, while recent work [15, 38] has highlighted the need for semantic table search, existing methods disregarded the importance of semantic information provided by KGs, as they focus on exact attribute/value overlap and use only taxonomic KG relationships or metadata. Other methods focus on matching tables to text queries relying on textual features in tables. Thus, *existing test collections are not sufficient* to fully evaluate new solutions for the challenging task of semantic table search.

3.1 Information Retrieval Methods

In IR, there exists a wide array of approaches designed for *table search and augmentation explicitly designed for Web tables* [37] that focus on matching the content of a table to a *text query*, e.g., keyword queries describing the topics of interest. Thus, the relevance score for a table is often based on table context (e.g., text in the same web page, headings, captions), table content (i.e., overlaps among cell values [40, 41]), and sometimes also on semantic relatedness based on the taxonomy of column names [24]. Recently, the task of ad-hoc table search [35, 37, 38] has also been considered, where the provided query can be a small Web table or a subset of it. In this case, the table query is treated as a complex text object, and text embedding methods are used to estimate relevance. In practice, they represent queries and tables in multiple vector spaces (both discrete sparse and continuous dense vector representations) that they refer to as semantic but *are based on textual features only*, while completely *ignoring actual semantic information* provided by either ontologies or KGs. Moreover, their performance is evaluated on a needle-in-the-haystack setting, where a very specific user intent is provided and a narrow list (less than 5) of relevant tables exists. Furthermore, these methods always assume a strong presence of textual information, which can be absent in many real cases.

Therefore, we find an important gap, highlighting the need to test a new task where the query is an example of the data of interest, and the goal is to find data that is *semantically* relevant but also substantially extend or provide context to the query.

3.2 Data Management Methods

The database literature has also seen a lot of interest in the task of table search. Existing approaches focused on Web tables [5] and identified two types of related tables [33]: (1) Entity Complement, where two tables are the result of two different sets of selection predicates on the same source table, these two tables are hence *unionable*; and (2) Schema Complement, where two tables are the result of two distinct project operations on the same source table,

Table 1: Characteristics of related test datasets. Type of tables: (Wt) Wikipedia Web tables, (P) tables from other Web pages, or (O) Open Data; if it links to an open KG; the number (# Q) of queries contained, the number of ground truth (# GT) relevance annotations for each query; Average size of the annotated tables; if it supports Query-by-Example (QbE).

Dataset	Type	KG	# Q	# Tables	# GT	Avg Size	QbE
WikiTables [2]	Wt	(✔)	-	1,652,771	-	10.9	-
WDC-EN [22]	P+Wt	(✔)	-	50,820,165	-	5.5	-
GitTables [16]	O	(✔)	-	871,411	-	136.8	-
NTCIR-16 [17]	O	✗	96	93,367	2,030	*115,586.5	✗
SemSearch [36]	Wt	(✔)	60	2,932	3,120	15.4	✗
SemSearch'21 [38]	Wt	(✔)	50	2,932	2,855	26.3	(✔)
WTR [6]	P+Wt	(✔)	60	6,629	6,949	6.1	✗
STSD WT'13 (ours)	Wt	✔	9,296	238,038	640,467	25.1	✔
STSD WT'19 (ours)	Wt	✔	2,560	457,714	958,214	24.9	✔

these two tables are hence *joinable*. This distinction uses *entity-centric* definitions, i.e., they assume a source table describing some set of entities (products, countries, customers), and thus relevance is determined by their ability to share exact values in a specific column (for joining them) or they feature the same set of attributes (for their union) [4, 7, 13].

More recently QbE approaches for joinability [40, 41] and unionability [29] that are not entity-centric have been developed to exploit approximate value overlap along with KGs and natural language models. The use of these technologies allows for the extraction of semantic similarities of attributes. These have also been generalized to consider relationship semantics and table semantics using large language models [12, 19].

Different works on table search [23, 29, 42] also use some taxonomic information (e.g., WebIsA and Freebase types) as reference knowledge to determine if two sets of entities come from the same set, or when attributes with different names are equivalent. *Yet, they do not fully exploit semantic resources, i.e., KGs. In particular, a KG contains not only taxonomic information (instance, type, and subclass relations), but it also contains attributes and inter-entity relationship information* that can help to determine relatedness in a broader sense. *Other approaches for related table search exploit instead meta-data derived by static analysis of the usage of specific tables in different programs (e.g., python notebooks) [39]. Kumar et al. [21] proposed a set of rules to determine if avoiding performing a join would be safe in a relational context, while in Shah et al. [34] the rules of Kumar et al. [21] are applied to high-capacity classifiers to test their validity. Other approaches to augmentation return a set of related tables [4] where a relatedness search in data lakes is performed, that identifies joins between tables containing overlapping sets of entities [7, 13].*

A framework called ARDA [7] has been proposed to also evaluate the quality of the information obtained through augmentation. ARDA works as a two-step algorithm. Firstly, it searches for joinable tables and then prunes out irrelevant features using feature selection algorithms. In particular, given a specific predictive model, it takes as input a dataset and a data repository and outputs an augmented dataset such that training the predictive model on this augmented dataset results in improved performance. Thus, *these approaches mostly exploit value overlaps and co-occurrences* while using taxonomies and natural language models to extract semantic similarities of attributes for schema alignment.

Linking tables to KGs also allows the consideration of non-taxonomic relationships to estimate the relatedness of two tables. Therefore, *a benchmark dataset for the STS task should allow the testing of semantic relatedness of generic tables in a data lake as required in exploratory use cases.* Yet, existing datasets do not offer ground truth relevance judgments for such pairs of tables.

Therefore, our proposed STSD corpus supports the study of methods that match relatedness *beyond the concepts of joinable and unionable tables*, since tables can be relevant even when they do not share any part of their data, e.g., none of the ground truth tables in Figure 1 are joinable or unionable with the query table.

3.3 Existing Test Collections

Multiple table collections have been proposed to test different information retrieval and data management use cases. In Table 1, we present their statistics, including the type of data they contain and whether they provide ground truth relevance that can be useful to test table search solutions. Some are large Web table collections that are designed to test entity linking approaches, where entities are usually linked directly to Wikipedia pages or a selected set of DBpedia entities. This is the case of the WikiTables [2] dataset and the tables extracted from the English WebDataCorpus (WDC-EN) [22]. The more recent GitTables [16] are extracted from tabular data published on GitHub, but is not linked to entities, instead some columns are linked to entity types or relationships. Overall, these datasets contain a large number of tables but no queries and no ground truth relevance judgments. Moreover, their entity-linking annotation is generally sparse, incomplete, and/or outdated.

Other datasets have been specifically designed for information retrieval tasks. This is the case of the NTCIR-16 [17], the SemSearch [36], and the WTR [6] datasets. Yet, for these datasets, relevance judgments are limited to keyword queries and queries are not linked to entities. Moreover, the relevance judgments have been obtained through human annotation. For this reason, while they are derived from a large set of tables (e.g., 3M for WTR), the actual annotated tables and relevance judgments are limited to a few thousand (6,629 in WTR), and the tables that are actually annotated are usually small. Moreover, almost all of them do not support the QbE use case, i.e., when the input is a portion of a table instead of a keyword query [6, 17, 36]. That is, users were not tasked to provide a relevance judgment of a table compared to another table as QbE requires. Furthermore, the entity linking in these tables is very sparse (we estimate that in WTR more than half of the tables have less than 10 entities while in our STSD corpus, all tables have at least 10). Therefore, most of the semantic information in WTR comes from table headers and captions. Recently, the work of SemSearch has been expanded to also include some tables as queries (SemSearch'21 [38]). However, this dataset is not suitable for training and testing advanced methods for the STS task, as its queries are not linked to KG entities. Furthermore, only 50 queries and few ground truth tables were annotated. These query tables have an average entity coverage of only 25.4%, i.e., an average of 29.3 entities per table among an average of 116.2 query table cells.

Another important limitation of WTR, SemSearch, and SemSearch'21 is that many *queries have few or no answer tables marked as relevant.* On average, 80.1% of the annotated tables in SemSearch'21 have been marked as irrelevant (i.e., assigned a ground truth score of 0). For keyword queries in SemSearch, this number is 72.1% and 67.9% in WTR. Finally, *both SemSearch [36] and WTR [6]* dataset annotations have been extracted *through a biased pooling* mechanism, which used the original keyword queries with BM25 to retrieve a subset of tables, and only those tables have been evaluated by human annotators for relevance. As we discuss in the next sections, this favors text-based methods and fails to collect relevance judgment on many other relevant tables. This means that any search method tested on these collections will not be able to verify the ability to encompass more expressive semantics. As an example, a query table in SemSearch'21 describes ferry boats of the East Frisian Islands. This query table has 1 annotated table with a non-zero relevance score in SemSearch'21 while it has 27 tables marked as relevant in our proposed STSD (using Wikipedia categories), among which the one table annotated as relevant in SemSearch'21 is also found. The annotated table in SemSearch'21 lists East Frisian Islands and sand flanks, and it has been assigned a low relevance score in both ground truths. In the STSD ground truth, however, the tables contain semantically relevant information, including a list of the largest ferries in Europe and the Caledonian MacBrayne fleet. Similarly, another query table describes athletes from Montana, USA. This table has 7 annotated tables with a non-zero relevance score in SemSearch'21 and 27 in STSD, with three tables appearing in both ground truths, and in both being assigned the highest relevance. All of the relevance annotated tables in SemSearch'21 are assigned the highest relevance score. Among the relevance annotated tables in our STSD dataset are tables describing information such as famous people from Montana, USA, athletes from Georgia Institute of Technology, and medalists from USA. Among these, only one table has also been assigned a non-zero relevance score in both datasets which is the highest relevance score. Finally, these two SemSearch'21 queries are the only queries that can be mapped to our queries in STSD. The remaining 48 SemSearch'21 queries have instead very few entity mentions appearing in them, resulting in their exclusion from our corpus. By selecting only those rows and columns containing entity links, the queries become too small to be sufficiently descriptive.

This comparison shows how important it is to offer a complementary set of tables and ground-truth relevance annotations since all existing datasets do not offer a sufficiently large, diverse, and annotated table corpus.

4 SEMANTIC TABLE SEARCH TEST CORPUS

Table 2: Benchmark statistics: # of tables (T), mean # of rows (R), mean # of columns (C), mean # of ground truth tables per query using WP categories (GT), and mean entity link coverage (Cov).

	Queries				Data Lake Tables			
	T	R	C	GT	T	R	C	Cov
WT'13	9,296	25.1	3.4	65.7	238,038	35.1	5.8	27.7%
WT'19	2,560	24.9	2.4	374.4	457,714	23.9	6.3	18.2%

The STSD corpus is extracted and annotated from tables in Wikipedia pages (WP) from 2013 and 2019 [2, 3] via the following steps: (1) linking tables cells that contain links to a corresponding WP to DBpedia using the `isPrimaryTopicOf` property, (2) filtering of tables to ensure a wide representation of entities and table sizes, (3) extraction of a large and heterogeneous set of query tables, and (4) construction of a ground truth relevance score for each query table using ground truth meta-data from the original Wikipedia pages.

4.1 Table Corpus

The STSD corpus is comprised of Wikipedia tables (WT) and is annotated by linking cell values to DBpedia entities. Since many cell values in the WT corpus contain links to WP pages (e.g., when a cell contains the value "U.S.A." that is linked to the corresponding WP page), and given that DBpedia entities are linked to WP pages via the `isPrimaryTopicOf` property, we match the cell to the corresponding DBpedia entity via those links. Note that, if no WP links is present, a state-of-the-art entity linker can be applied for this task. Then, our STSD corpus is obtained by selecting from the 1.6*M* WT tables in the original corpus (from a 2013 snapshot) those with at least 10 unique DBpedia entities across their cell values. Performing this procedure results in 238,038 tables. In addition, for each table, we also extract the set of Wikipedia categories and navigation links from the WP page containing the table. *Note that this information is missing from the original WT corpus. We crawled it from the current online Wikipedia.* These are used to obtain our ground truth relevance assessment as explained below. Following the same procedure, we extracted tables and annotations from a 2019 snapshot [3] of 714,632 WT tables resulting in an additional 457,714 tables. Figures 2a and 2b show the table size distribution across the 2013 and 2019 corpora. Table 2 summarizes the characteristics of our two STSD corpora from 2013 and 2019, as well as the entity link coverage. Both WT tables from 2013 and 2019 show similar characteristics regarding rows and columns. However, WT tables from 2013 contain on average more DBpedia entity annotations and are therefore semantically slightly more descriptive.

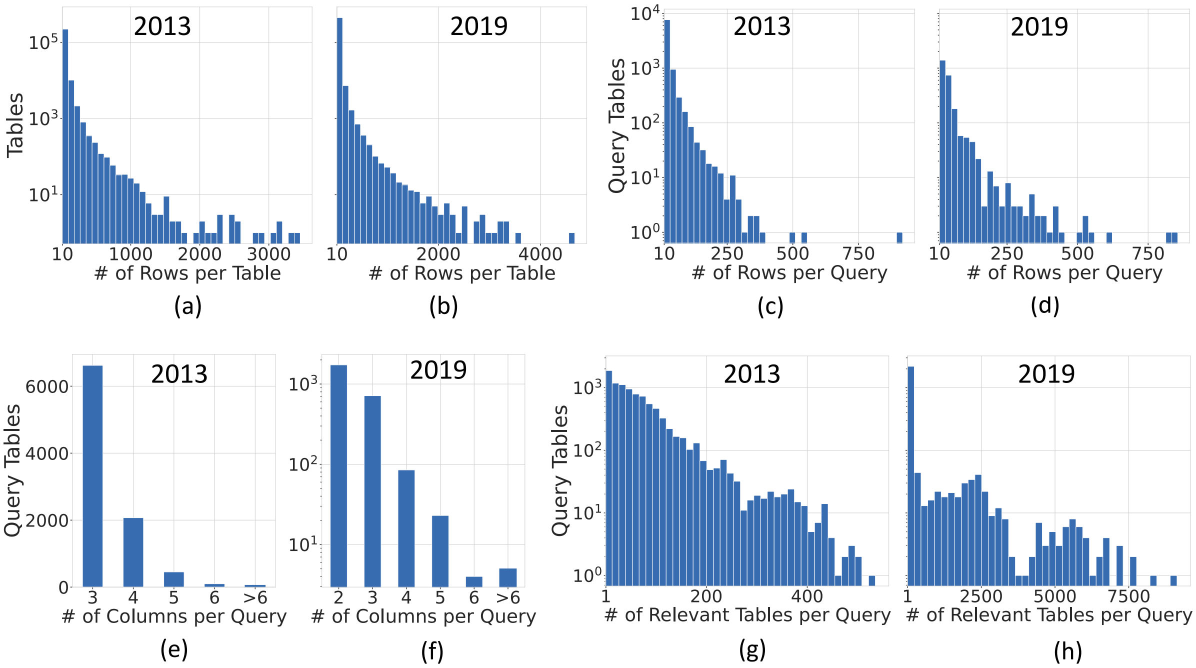


Figure 2: Distribution of rows across tables in the corpus and rows, columns, and relevant tables across all query tables in STSD.

4.2 Query Corpus

Our set of query tables is generated by selecting one table from each WP page. If a WP page contains more than one table, the table with the largest number of entity mappings within a single row (called horizontal entity mapping) is chosen. We keep only query tables with a minimum horizontal entity mapping of at least 3 entities to capture a sufficiently large schema. As an additional restriction to improve the quality of our queries, we keep only query tables with at least 10 rows. Furthermore, we keep only query tables with at least one relevant (see below how relevance is established) table from a different WP page in the ground truth than the page of the query table itself. These restrictions ensure that the query tables can be used to find other *non-trivial* relevant tables. Applying these restrictions provides 9,296 query tables on our 2013 STSD corpus and 2,560 query tables on our 2019 STSD corpus. Focusing on the entities, the query tables can also be considered as a list of entity tuples. Figures 2c and 2d show the distribution of the number of rows per query, while Figures 2e and 2f show the distribution of the number of columns (i.e., width) per query table.

4.3 Relevance Assessment

Our dataset is the first to provide a large-scale set of ground truth relevance annotations for the query-by-example semantic table search task. Our ground truth annotations are designed specifically for the semantic exploratory search task and establish the relevance of two tables that appear on two pages by comparing the set of Wikipedia categories or navigation links (i.e., links to related WP pages) those pages share. Thus, for each query-table pair, we provide two relevance judgments: the Jaccard similarity of Wikipedia categories and the Jaccard similarity of navigational links of pages from which the two tables were extracted. More specifically, we provide continuous relevance scores ranging between zero (irrelevant) and one (highest relevance possible): if the Jaccard similarity comparing either the pages or links of the query table and a given table is above zero, then that table is considered as somehow relevant to the query. Notice that the relevance score of a query table to a table from the same WP page will always be one, since tables from the same WP page share the same categories and navigation links. To further improve the quality of the relevance assessments, generic Wikipedia categories and navigation links (e.g., "Pages Under Construction", or "Living People") were excluded as they do not contain semantically useful information. Figures 2g and 2h show the distribution of the number of tables annotated as relevant per query table. It should be noted that WPs in a few cases suffer from generic or small and specific sets of categories. For example, the only WP category of *108th United States Congress* is "`108th United States Congress`" and the WP *2014 in amusement parks* has the generic types "`Amusement parks by year`" and "`2014-related lists`".

We evaluate the trustworthiness of our automatically established ground truth using Wikipedia categories and navigation links by performing a user study over 25 randomly chosen 5-tuples queries (query table containing five tuples). For each query, we select 20 tables for human annotation which are split into four groups of five tables: (1) tables retrieved by both semantic relevance search and BM25 keyword search and annotated as relevant in our ground truth, (2) tables retrieved only by semantic relevance search and annotated as relevant in our ground truth, (3) tables retrieved by semantic relevance search but annotated as not relevant in our ground truth, and (4) random tables not annotated as relevant in our ground truth. Semantic relevance search is implemented as a table search approach finding semantically relevant tables by comparing the entities within the query and the tables. Pairs of entities in the query and the table are compared in two alternative ways: either (i) computing the Jaccard similarity of entity types or (ii) computing the cosine similarity of entity embeddings. This search method is described in greater detail in the Section 5. More complex methods could also be designed, but we favored a simple and easily explainable method for this experiment.

Given the query and answers described above, we performed the user study on 12 users, with a background in data science and data management. The users were tasked to annotate each query-table pair with a relevance score in $[1, 4]$, where 1 is "irrelevant" and 4 is "very relevant". We ensured each table is annotated by three different participants and the final table annotation used is determined by averaging the votes of the three annotators. We computed the Kendall-W coefficient of concordance [14] of the human annotators to measure the inter-annotator agreement. We used Kendall-W instead of Kendall Tau [18] since it is more appropriate when comparing multiple annotators. We then obtained an average of 0.87 per query, which shows a high agreement between the human annotators. However, this measure requires the annotations to be unique rankings of the tables per query. As this is not the case in our user study, i.e., tables with the same annotation score for a query are randomly assigned a ranking, Kendall-W can provide an underestimation of the real agreement. Therefore, we also compute Krippendorff's Alpha [26], which can handle ordinal input containing duplicates and multiple annotators. This score is 0.62, which once again shows a high inter-annotator agreement.

Once retrieved human relevance annotations for our sample, we compared the ranking of the 20 tables for each query to the ground truth ranking of the same tables. We employed Kendall Tau [18] to measure the agreement between the automatically established ground truth and the human annotators. Since the relevance scores we produced in our automatic relevance computation are in the range 0-1, we quantized these values into the 1-4 range of relevance scores obtained in our user study. The quantization was obtained by computing thresholds for the scores as the 1st quartile, the median, and the 3rd quartile. Thus, comparing the quantized relevance scores computed in our dataset to the annotators' relevance scores, we obtained an average Kendall Tau value of 0.47, which shows a high agreement between the human annotators and the automatically established relevance scores.

5 SEMANTIC TABLE SEARCH EVALUATION

In this section, we show how the STSD corpus allows testing the limitations of table search approaches for the QbE search task. Specifically, using STSD 2013, we evaluate the recall of 5 baselines: one based on BM25, one based on dense representation learning, one based on a large language model for table union search, and two based on semantic information that are also purely content-based and do not use any other table information other than the cell values.

BM25 [32] is a well-established keyword search algorithm in information retrieval that has also been used as a baseline method in numerous table search works [35, 36, 38]. To use BM25 to search over our corpus, we first convert our query tables into keyword queries. This is done by extracting the text field from each cell and using that as the keyword (e.g., the entity "dbpedia:Boston" from a query table will be converted into the keyword "Boston"). Thus, every query and every candidate table is then treated as a text document composed of all its cell values. Metadata is left out, and tables are indexed for full-text search with BM25 to answer keyword queries in a purely content-based manner.

TURL [9] is a deep-learning model for Table Representation which we adapt for the dense table search task. That is, queries and tables are represented as dense vectors. Specifically, we aggregate all contextualized vector representations in each table to construct an embedding for each query and table using the pre-trained model of TURL. To rank the tables, we use cosine similarity between the aggregated query and table representations.

Starmie [12] is a representation learning approach applying a large language model to perform semantic matching of column representations for table union search. Starmie is able to capture rich contextual semantic information within tables using trained column encoders.

Jaccard of Entity Types (STST) performs Semantic Table Search by measuring the similarity between pairs of entities in the query and the table. Given two entities (e.g., from DBpedia), we define their relevance score as the Jaccard similarity between the two sets of their types (e.g., mapped via `rdf:type` in DBpedia). Then, to estimate the similarity between the query table and the candidate table, we aggregate the Jaccard similarity across all their entities. To do so, we first align each column in the query tuple to the column in the candidate table maximizing the Jaccard similarity of their types in that column. We generate such an alignment using the Hungarian algorithm [20] so that the Jaccard of their DBpedia types is maximized per column. Once the best alignment is identified the relevance score for a candidate table can be computed by averaging the Jaccard similarities of all query tuples with each row in the candidate table.

Embedding Similarity (STSE) is an alternative to the above Jaccard of types, that tries to better exploit the semantic information encoded in the structure of the knowledge graph. We use RDF2Vec [31] to construct 200-dimensional node embedding vectors for each entity (which we also make available on Zenodo [8]). The entity similarity is then the normalized (shifted in $[0, 1]$) cosine similarity, between the query entities and entities in the candidate table. Then, we align the query and target table columns as done with STST to compute an aggregate score.

Experimental Results. We sample 50 query tables from our STSD corpus, consisting of 1 and 5 tuples. We choose this subset of 50 queries, as evaluating the complete set of 9K queries would be too time-consuming. The queries have an average of 65.7 and 374.4 relevant tables in the ground truth in the 2013 and 2019 snapshots (some more than 400), respectively, as listed in Table 2. Table 3 shows recall across our baselines at two different query sizes (i.e., number of tuples) for k=100 and 200.

For computing recall, we compute the median ground truth relevance score and use that as the relevance threshold. Then, when

Table 3: Average recall at top-100 and -200 for STST, STSE, BM25, TURL, and Starmie.

	STST	STSE	BM25	TURL	Starmie
Top-100					
1-Tuple	0.263	0.267	0.278	0.002	0.033
5-Tuples	0.159	0.145	0.166	0.002	0.042
Top-200					
1-Tuple	0.292	0.331	0.320	0.006	0.033
5-Tuples	0.099	0.095	0.105	0.003	0.042

Table 4: Average NDCG at top-10 for STST, STSE, BM25, TURL, and Starmie.

Top-10	STST	STSE	BM25	TURL	Starmie
1-Tuple	0.534	0.543	0.573	0.005	0.102
5-Tuples	0.595	0.628	0.660	0.004	0.126

Table 5: Average NDCG at top-10 for STST and STSE on the 2019 WT snapshot.

Top-10	STST	STSE
1-Tuple	0.546	0.549
5-Tuples	0.612	0.617

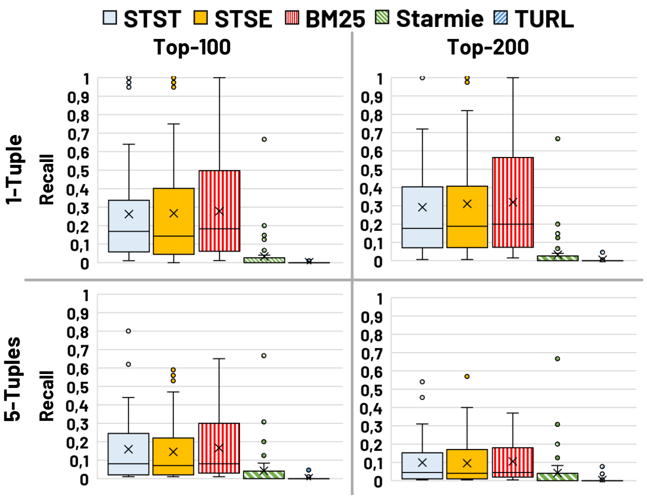


Figure 3: Recall at top-100 and -200 for STST, STSE, BM25, TURL, and Starmie.

measuring recall, we consider only tables with a ground truth score above that value as relevant. This offers a conservative estimation of the performance of these baselines and also shows the versatility of our dataset.

We observe that STST, STSE, and BM25 generally achieve similar recall (Table 3). However, STST and STSE retrieve almost completely disjoint sets of tables compared to BM25. Specifically, for k=100, we measure that the result sets of the two methods differ by a median of 66 and up to 100 results. This is because BM25 is based on keyword search and is therefore not able to retrieve ground truth tables that do not contain any keyword matches to the query. Therefore, BM25 leaves out a portion of tables that are semantically relevant to the query. *This also highlights how biased the existing text collections are*, since their relevance judgment is given only for top-100 queries returned by BM25 (called pooling [6]) resulting in the omission of a large set of tables that are relevant but not included in their test collections and thus their ground truth. However, a fraction of our ground truth relevant tables do contain keyword matches and are therefore retrievable with BM25, and hence, BM25 can compete with STS if the fraction is large enough. The performance of BM25 correlates to this fraction, and its performance will therefore decrease as this fraction decreases. We similarly plot the recall in Figure 3. Although, the average recall is quite low, as reported in Table 3, STST, STSE, and BM25 are all able to perform well on some queries. Even Starmie performs similarly to STST, STSE, and BM25 in a few instances. Interestingly, even though 5-tuples queries can be more informative as they contain more entities overall, STST, STSE, and BM25 perform worse on recall compared to the 1-tuple queries. This can be partly attributed due to the overall smaller number of tables retrieved as relevant since alignment between a query with more tuples and a candidate table can lead to overspecialization, blocking out some tables that would have been relevant.

Table 4 and Figure 4 present the ranking quality across our baselines using NDCG (Normalized Discounted Cumulative Gain) for top-10 results. TURL performs poorly in both recall (Table 3) and

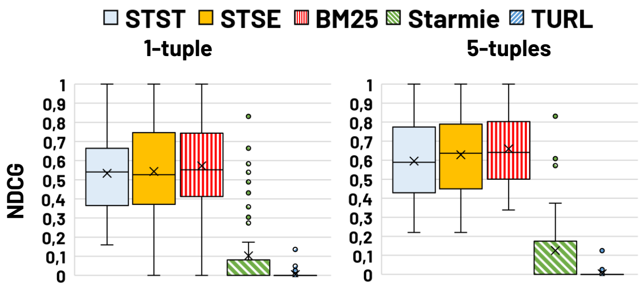


Figure 4: NDCG at top-10 for STST, STSE, BM25, TURL, and Starmie.

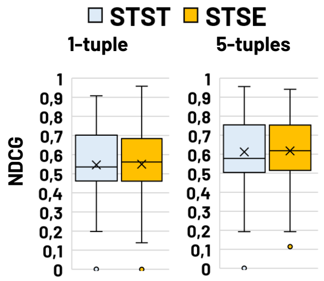


Figure 5: NDCG at top-10 for STST and STSE on the WT'19.

NDCG (Table 4) using 1- and 5-tuples queries. This is due to TURL not being designed for semantic table search but rather as a table understanding approach. However, when TURL is given the entire source table, i.e., the entire WT from which 1- and 5-tuples queries are extracted, as input query, it can reach an NDCG of 0.488. Starmie also does not perform well in both recall and NDCG. This is due to Starmie being a union search approach, meaning that it is designed to only find a subset of tables that are relevant in semantic tables search: namely, those tables that are unionable with the query table. Figure 4 shows a varying performance of STST, STSE, and BM25, and once again Starmie can only compete with these baselines in a few instances. We also perform an experiment to verify the quality of the 2019 WT snapshot. We present the ranking quality of STST and STSE on this snapshot in Table 5 and Figure 5. Although this snapshot has a comparatively lower entity-link coverage among the tables (18.2% compared to 27.7% for the 2013 snapshot), semantic table search with STST and STSE still performs nearly identically to the WT tables from the 2013 snapshot. This shows that the more recent snapshot of the WT tables from the 2019 snapshot is also of high semantic quality, despite its lower entity link coverage.

6 CONCLUSION

While table search tasks have attracted substantial attention, existing benchmark datasets are still limited in quality and dimension of the set of tables, queries, and relevance annotations they offer. The STSD corpus we propose provides instead queries and ground truth relevance judgments that go beyond simple keyword matching. The high quality of this resource is assured by the presence of Wikipedia links and annotations which are assigned by human editors. Furthermore, we performed a user study that shows that the established ground truth aligns well with human relevance annotations. Thanks to this dataset, we perform an exploratory evaluation of a few table search methods and compare them to two naïve baselines that try to exploit the information encoded in a KG. Our experimental results point towards an unexplored potential hidden in the semantic annotations when tables are linked (even just partially) to a reference KG. This shows the necessity to study methods for table search that can better take into account *semantic similarity*. While in this work we only present very simple baselines, in the future, we foresee the need for and plan to design advanced tables search algorithms that can complement the effectiveness of keyword search approaches based on content to more advanced semantic-aware techniques that are also equally scalable as keyword search. Furthermore, we plan to design a similar resource that goes beyond Web tables and extends our corpus to a test collection that includes also larger Open Data tables.

Availability. We have published the following resources in our STSD GitHub repository[2]:

- **Scripts**, we release scripts to reproduce our results and extract the Wikipedia categories and navigation links.
- **Table Corpus and Queries**, we release our 2013 corpus of 238K tables and 9K table-queries (represented as both entity tuples and text blobs) along with the entity linking to a recent snapshot of DBpedia, as well as the Wikipedia categories and navigation links extracted for each page. We also publish our 2019 corpus of 457K tables and 2K table queries, including the Wikipedia categories and navigation links. We maintain the same WT file names, so it is possible to retrieve the original files as well.
- **Relevance Assessment**, we release the relevance assessments we collected for each query table based on both categories and navigation links from Wikipedia comprising 610K relevance scores across all queries and corpus tables from the 2013 snapshot, as well as 958K relevance scores on the 2019 snapshot.
- **User Study**, we release our user study of our automatically established relevance assessments for a total of 500 query-table pairs from the 2013 snapshot.
- **RDF2Vec Embeddings over DBpedia**, along with this resource we also share the KG graph embedding [8] for all entities in the DBpedia snapshot we used.

Acknowledgement

This research was partially funded by the Danish Council for Independent Research (DFF) under grant agreement no. DFF-8048-00051B, the EU's H2020 research and innovation programme under grant agreement No. 838216, and the Poul Due Jensen Foundation. Additional support from NSF awards IIS-2107248, IIS-1956096, and IIS-2325632.

REFERENCES

[1] Omar Benjelloun, Shiyu Chen, and Natasha F. Noy. 2020. Google Dataset Search by the Numbers. In *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 12507)*. Springer, 667–682. https://doi.org/10.1007/978-3-030-62466-8_41

[2] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. TabEL: Entity Linking in Web Tables. In *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 9366)*. Springer, 425–441. https://doi.org/10.1007/978-3-319-25007-6_25

[3] Tobias Bleifuß, Leon Bornemann, Dmitri V. Kalashnikov, Felix Naumann, and Divesh Srivastava. 2021. The Secret Life of Wikipedia Tables. In *Proceedings of the 2nd Workshop on Search, Exploration, and Analysis in Heterogeneous Datastores (SEA-Data 2021) co-located with 47th International Conference on Very Large Data Bases (VLDB 2021), Copenhagen, Denmark, August 20, 2021 (CEUR Workshop Proceedings, Vol. 2929)*. CEUR-WS.org, 20–26. https://ceur-ws.org/Vol-2929/paper4.pdf

[4] Alex Bogatu, Alvaro A. A. Fernandes, Norman W. Paton, and Nikolaos Konstantinou. 2020. Dataset Discovery in Data Lakes. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*. IEEE, 709–720. https://doi.org/10.1109/ICDE48307.2020.00067

[5] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. 2020. Dataset search: a survey. *VLDB J.* 29, 1 (2020), 251–272. https://doi.org/10.1007/S00778-019-00564-X

[6] Zhiyu Chen, Shuo Zhang, and Brian D. Davison. 2021. WTR: A Test Collection for Web Table Retrieval. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 2514–2520. https://doi.org/10.1145/3404835.3463260

[7] Nadiia Chepurko, Ryan Marcus, Emanuel Zgraggen, Raul Castro Fernandez, Tim Kraska, and David R. Karger. 2020. ARDA: Automatic Relational Data Augmentation for Machine Learning. *Proc. VLDB Endow.* 13, 9 (2020), 1373–1387. https://doi.org/10.14778/3397230.3397235

[8] Martin Pekár Christensen, Matteo Lissandrini, and Katja Hose. 2022. *DBpedia RDF2Vec Graph Embeddings.* https://doi.org/10.5281/zenodo.6384728

[9] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2020. TURL: Table Understanding through Representation Learning. *Proc. VLDB Endow.* 14, 3 (2020), 307–319. https://doi.org/10.5555/3430915.3442430

[10] Yuyang Dong, Kunihiro Takeoka, Chuan Xiao, and Masafumi Oyamada. 2021. Efficient Joinable Table Discovery in Data Lakes: A High-Dimensional Similarity-Based Approach. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*. IEEE, 456–467. https://doi.org/10.1109/ICDE51399.2021.00046

[11] Faezeh Ensan and Ebrahim Bagheri. 2017. Document Retrieval Model Through Semantic Linking. In *Proceedings of the Tenth ACM International Conference*

[2] https://github.com/EDAO-Project/SemanticTableSearchDataset

on Web Search and Data Mining (Cambridge, United Kingdom) *(WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/3018661.3018692

[12] Grace Fan, Jin Wang, Yuliang Li, Dan Zhang, and Renée J. Miller. 2023. Semantics-aware Dataset Discovery from Data Lakes with Contextualized Column-based Representation Learning. *Proc. VLDB Endow.* 16, 7 (2023), 1726–1739. https://doi.org/10.14778/3587136.3587146

[13] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A Data Discovery System. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018.* IEEE Computer Society, 1001–1012. https://doi.org/10.1109/ICDE.2018.00094

[14] Andy P Field. 2005. Kendall's coefficient of concordance. *Encyclopedia of Statistics in Behavioral Science* 2 (2005), 1010–11.

[15] Sainyam Galhotra and Udayan Khurana. 2020. Semantic Search over Structured Data. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020.* ACM, 3381–3384. https://doi.org/10.1145/3340531.3417426

[16] Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2023. GitTables: A Large-Scale Corpus of Relational Tables. *Proc. ACM Manag. Data* 1, 1 (2023), 30:1–30:17. https://doi.org/10.1145/3588710

[17] Makoto P. Kato, Hiroaki Ohshima, Ying-Hsang Liu, and Hsin-liang Oliver Chen. 2021. A Test Collection for Ad-hoc Dataset Retrieval. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021.* ACM, 2450–2456. https://doi.org/10.1145/3404835.3463261

[18] M. G. Kendall. 1938. A New Measure of Rank Correlation. *Biometrika* 30, 1/2 (1938), 81–93. http://www.jstor.org/stable/2332226

[19] Aamod Khatiwada, Grace Fan, Roee Shraga, Zixuan Chen, Wolfgang Gatterbauer, Renée J. Miller, and Mirek Riedewald. 2023. SANTOS: Relationship-based Semantic Table Union Search. *Proc. ACM Manag. Data* 1, 1 (2023), 9:1–9:25. https://doi.org/10.1145/3588689

[20] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

[21] Arun Kumar, Jeffrey F. Naughton, Jignesh M. Patel, and Xiaojin Zhu. 2016. To Join or Not to Join?: Thinking Twice about Joins before Feature Selection. In *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016.* ACM, 19–34. https://doi.org/10.1145/2882903.2882952

[22] Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A Large Public Corpus of Web Tables containing Time and Context Metadata. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume.* 75–76. https://doi.org/10.1145/2872518.2889386

[23] Oliver Lehmberg, Dominique Ritze, Petar Ristoski, Robert Meusel, Heiko Paulheim, and Christian Bizer. 2015. The Mannheim Search Join Engine. *J. Web Semant.* 35 (2015), 159–166. https://doi.org/10.1016/J.WEBSEM.2015.05.001

[24] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. 2010. Annotating and Searching Web Tables Using Entities, Types and Relationships. *Proc. VLDB Endow.* 3, 1 (2010), 1338–1347. https://doi.org/10.14778/1920841.1921005

[25] Matteo Lissandrini, Davide Mottin, Themis Palpanas, and Yannis Velegrakis. 2018. *Data Exploration Using Example-Based Methods.* Morgan & Claypool Publishers. https://doi.org/10.2200/S00881ED1V01Y201810DTM053

[26] Giacomo Marzi, Marco Balzano, and Davide Marchiori. 2024. K-Alpha Calculator–Krippendorff's Alpha Calculator: A user-friendly tool for computing Krippendorff's Alpha inter-rater reliability coefficient. *MethodsX* 12 (2024), 102545. https://doi.org/10.1016/j.mex.2023.102545

[27] Renée J. Miller. 2018. Open Data Integration. *Proc. VLDB Endow.* 11, 12 (2018), 2130–2139. https://doi.org/10.14778/3229863.3240491

[28] Fatemeh Nargesian, Erkang Zhu, Renée J. Miller, Ken Q. Pu, and Patricia C. Arocena. 2019. Data Lake Management: Challenges and Opportunities. *Proc. VLDB Endow.* 12, 12 (2019), 1986–1989. https://doi.org/10.14778/3352063.3352116

[29] Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. 2018. Table Union Search on Open Data. *Proc. VLDB Endow.* 11, 7 (2018), 813–825. https://doi.org/10.14778/3192965.3192973

[30] José R. Pérez-Agüera, Javier Arroyo, Jane Greenberg, Joaquin Perez Iglesias, and Victor Fresno. 2010. Using BM25F for semantic search. In *Proceedings of the 3rd International Semantic Search Workshop* (Raleigh, North Carolina, USA) *(SEMSEARCH '10)*. Association for Computing Machinery, New York, NY, USA, Article 2, 8 pages. https://doi.org/10.1145/1863879.1863881

[31] Jan Portisch, Michael Hladik, and Heiko Paulheim. 2020. RDF2Vec Light - A Lightweight Approach for Knowledge Graph Embeddings. *CoRR* abs/2009.07659 (2020). arXiv:2009.07659 https://arxiv.org/abs/2009.07659

[32] Stephen E. Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[33] Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Y. Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. 2012. Finding related tables. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale, AZ, USA, May 20-24, 2012.* ACM, 817–828. https://doi.org/10.1145/2213836.2213962

[34] Vraj Shah, Arun Kumar, and Xiaojin Zhu. 2017. Are Key-Foreign Key Joins Safe to Avoid when Learning High-Capacity Classifiers? *Proc. VLDB Endow.* 11, 3 (2017), 366–379. https://doi.org/10.14778/3157794.3157804

[35] Mohamed Trabelsi, Zhiyu Chen, Shuo Zhang, Brian D. Davison, and Jeff Heflin. 2022. StruBERT: Structure-aware BERT for Table Search and Matching. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022.* ACM, 442–451. https://doi.org/10.1145/3485447.3511972

[36] Shuo Zhang and Krisztian Balog. 2018. Ad Hoc Table Retrieval using Semantic Similarity. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018.* ACM, 1553–1562. https://doi.org/10.1145/3178876.3186067

[37] Shuo Zhang and Krisztian Balog. 2020. Web Table Extraction, Retrieval, and Augmentation: A Survey. *ACM Trans. Intell. Syst. Technol.* 11, 2 (2020), 13:1–13:35. https://doi.org/10.1145/3372117

[38] Shuo Zhang and Krisztian Balog. 2021. Semantic Table Retrieval Using Keyword and Table Queries. *ACM Trans. Web* 15, 3 (2021), 11:1–11:33. https://doi.org/10.1145/3441690

[39] Yi Zhang and Zachary G. Ives. 2020. Finding Related Tables in Data Lakes for Interactive Data Science. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020.* ACM, 1951–1966. https://doi.org/10.1145/3318464.3389726

[40] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J. Miller. 2019. JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019.* ACM, 847–864. https://doi.org/10.1145/3299869.3300065

[41] Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée J. Miller. 2016. LSH Ensemble: Internet-Scale Domain Search. *Proc. VLDB Endow.* 9, 12 (2016), 1185–1196. https://doi.org/10.14778/2994509.2994534

[42] Erkang Zhu, Ken Q. Pu, Fatemeh Nargesian, and Renée J. Miller. 2017. Interactive Navigation of Open Data Linkages. *Proc. VLDB Endow.* 10, 12 (2017), 1837–1840. https://doi.org/10.14778/3137765.3137788