STaR: Space and Time-aware Statistic Query Answering

Oana Balalau, Simon Ebel, Helena Galhardas, Théo Galizzi, Ioana Manolescu

STaR: Space and Time-aware Statistic Query Answering

Oana Balalau
Inria & Institut Polytechnique de Paris
Palaiseau, France

Simon Ebel
Inria & Institut Polytechnique de Paris
Palaiseau, France

Helena Galhardas
INESC-ID & IST, Universidade Lisboa
Lisbon, Portugal

Théo Galizzi
Inria & Institut Polytechnique de Paris
Palaiseau, France

Ioana Manolescu
Inria & Institut Polytechnique de Paris
Palaiseau, France

Abstract

High-quality data is essential for informed public debate. High-quality statistical data sources provide valuable reference information for verifying claims. To assist journalists and fact-checkers, user queries about specific claims should be automatically answered using statistical tables. However, the large number and variety of these sources make this task challenging.

We propose to demonstrate STaR, a novel method for *S*pace and *T*ime-*a*ware *STa*tistic *R*etrieval, based on a user natural language query. STaR is deployed within our system StatCheck, which we developed and shared with fact-checking journalists. STaR improves the quality of statistic fact retrieval by treating space and time separately from the other parts of the statistics dataset. Specifically, we use them as dimensions of the data (and the query), and focus the linguistic part of our dataset search on the rich, varied language present in the data. Our demonstration uses statistic datasets from France, Europe, and a few beyond, allowing users to query and explore along space and time dimensions.

CCS Concepts

• **Information systems** → **Information retrieval**; **Spatial-temporal systems**.

Keywords

Statistic fact-checking; table search; table querying

ACM Reference Format:
Oana Balalau, Simon Ebel, Helena Galhardas, Théo Galizzi, and Ioana Manolescu. 2024. STaR: Space and Time-aware Statistic Query Answering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3627673.3679209

1 Introduction

Public debates often involve metrics about society, countries, or regions, such as (un)employment, inflation, revenue, and health statistics. Many such metrics are measured by national organizations, such as France's INSEE (Institute for Economic Studies and Statistics), or the ministries of Justice or Education, among others. International bodies, such as Eurostat, the International Monetary Fund, and the UN, also gather statistics for many countries. Smaller entities, such as trade unions and NGOs, also collect specialized statistics, such as wildlife presence or pollutants in an area.

Statistics can ground public debate, but they are not easily usable. First, statistical data is published in a *variety of formats* such as CSV, Excel variants, HTML tables, RDF graphs, or specific standard formats such as SDMX [3], an XML vocabulary for statistic (multi-dimensional) data and metadata exchange. *A statistic file may be huge*, with two consequences: the publisher may prefer to share it *compressed*, thus search engines cannot index its content, and it is hard for users to look for one statistic point in a file of millions of rows (such as we found in Eurostat).

To make statistics data more accessible to use, *question answering methods over statistic datasets* are needed. The scientific literature comprises a set of methods that, given a set of tables and a user query (a phrase or set of keywords), find the most relevant table(s), and may also extract answers from the tables, e.g., [5, 10, 11] (see Sec. 2). We have been collaborating with RadioFrance journalists on the StatCheck system [4], focused specifically on answering search queries on *statistic* tables, different from the *relational* tables considered in other works. Even if laid out in two-dimensional layouts (HTML tables, CSV or Excel files, etc.), statistic tables are conceptually *multidimensional*, leading to a different semantics, where each fact (a number) is characterized by several dimensions and their values. Among statistic data dimensions, *time* and *space* are omnipresent, and frequent also in user queries.

Inspired by this, we develop a new *S*pace- and *T*ime-aware *STa*tistic *R*etrieval method, extracting time and space indications from both the statistics and the query, and processing them separately from the rest of the content. This understanding of the data leads to *higher-quality results* than those from the state of the art. Making the time and space dimensions explicit, furthermore: (*i*) provides a natural basis for query *relaxation* when a user searches yields no result, e.g., "high schools in Paris", but a similar search, e.g., "high schools in Île-de-France", has results (Île-de-France encloses Paris); (*i*) provide *facets* for inspecting a set of statistics , e.g., see which metrics are measured for which world regions, or how the metrics evolve over time, etc.

2 Related Work

Existing works can be organized in three groups. **1. NLP question answering (QA) over a given table** has been studied for *relational tables*, having a single header row, followed by data rows. Solutions either translate the question into SQL queries answered by a database holding the tables (see the survey [10]), or apply neural methods applied directly on the table, e.g., TaPaS [8]. Given a query and a set of tables, **2. table search** returns a list of ranked tables, most pertinent for query. Putting it all together, **3. end-to-end**

CIKM '24, October 21–25, 2024, Boise, ID, USA

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679209

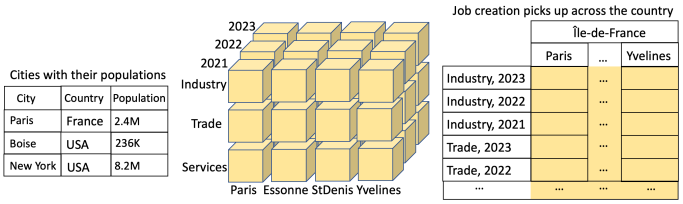

Figure 1: Relational table (left); multidimensional statistic data (center), and typical table representation (right).

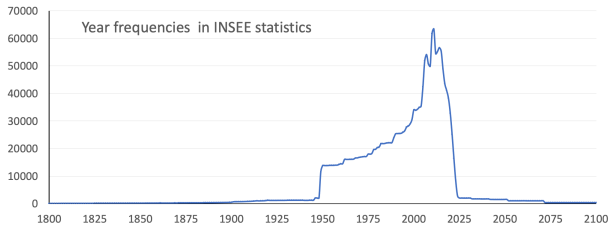

Figure 2: Year distribution within statistics from INSEE.

QA aims to answer a question, given a large set of tables. This is typically implemented by table search (also called *retrieval*), which restricts the scope of the search, followed by *(re)-ranking*, with a higher quality but more expensive model.

Among **table search** methods, [18] builds syntactic and semantic representations of queries and tables, and combines them with other features into a *learning-to-rank* framework. They show that *semantic* representations improve retrieval quality, compared to only using statistic-based methods (such as BM25 [13]), which fails to detect *synonyms* or semantically close words as relevant to those in the query. In [15], for the *retrieval*, the authors transform the query, and each table (caption and headers) into bag-of-words (BoW), then use BM25. For the *ranking*, they introduce a set of *non-neural* features and *neural* features, representing the query and the table, based on RNNs; then, they train a forest of decision trees to learn feature weights. They show that ranking using both neural and non-neural features is best. In [14], several *modalities* of each table (e.g., description, schema, each row, column) are independently embedded, then a joint representation thereof is learned using Gated Multimodal Units.

The first approach to leverage Large Language Models (in particular, BERT) for table retrieval is [6]. Because only limited-length strings can be embedded, they propose *content selectors* that extract, from each table, bounded-length fragments to represent it. This method fails to represent *most* of a large table's content, leading to potentially many missed results. In [7], the retrieval method is based on computing a table representation as in [8], a similar one for the query, and retrieving the tables closest to the query.

[7], [4] and [17] are examples of **end-to-end QA**. [7] uses a reader model to extract the answer for a given question using previously retrieved candidate tables. The model scores each candidate and extracts a suitable answer span from the table. Tables and questions are jointly encoded using TaPaS. From well-structured, relational tables, SOLO [17] automatically generates SQL queries whose answer are (known) in the table, then uses an LLM to get natural language (NL) phrasings from the SQL query, thus obtaining (NL query, table, response) triples. SOLO's relevance model ranks question/table relevance by using a feature vector encoded by the pretrained QA model.

Unlike the above systems, we work with *statistic (multidimensional) data* tables that have headers (possibly nested) on both the top and left of the data cells. Thus, several dimensions may be encoded in row or column headers. Understanding the semantics of these tables is crucial for effective retrieval. Additionally, our datasets consistently include time and geographical space. **The novelty of this work compared to StatCheck** [4] is the separate handling of time and space, enabling more accurate table retrieval.

3 Indexing and retrieving statistics

We introduce statistic datasets, and in particular their space and time dimensions. Then, we explain how we leverage them to answer natural language questions over a large corpus of statistic data.

3.1 Statistic datasets, space and time

A **statistic dataset** consists of: a *title*; a set of *data cells*, holding numeric data (integer or real values); a set of *header cells*, which may state the metric (what is measured in a dataset), and values of its dimensions; and *comments*, i.e., a text published next to the data, which interprets and makes observations about it. Time and space are frequent dimensions, e.g., a data cell holds a number of births in a French department in a given year. However, *the set of possible dimensions (and their values) is extremely large*: combustion-engine vs. electric cars; age intervals in which workers, new mothers, or public servants are grouped; crop planted per year and country, etc.

In our statistic corpus (see Sec. 5 for details), time appears at different *granularities*: marriages are counted per *day* (of a given year), many economic metric per *month*, *quarter* or *year*, and a few macro-economic parameters are measured for several *year intervals*, one of which could be "1990-2000". Similarly, the *space* dimension is present: at the level of the smallest *settlement* (think small village) or sectors of large cities, then the *commune*, *city*, and various *regional divisions*, up to *countries* and *country groups*, e.g., "15-states EU", "27-states EU", etc. We call *temporal value* (or **tval**, in short) any mention of time found in the data, at any granularity; similarly, we call *spatial value* (or **sval**, in short) any individual spatial unit. tvals are organized in a natural *containment hierarchy*, and the same holds for svals: each tval or sval, seen as a *node*, has one or more *parents* which include (subsume) it, and may have *children* which it includes in turn [9]. The spatial hierarchy is not a tree, but a directed acyclic graph (**DAG**). An sval with multiple parents is "France", which is part of "15-states EU" and of "27-states EU".

3.2 Dataset space and time scopes

To anchor our statistic datasets, as well as queries, to geographic space, we built a **geographic reference**, containing all the locations that may be encountered in the statistic data. As we work with French and EU statistics, our starting points were: NUTS [2], a geographic dictionary of Eurostat, covering all Europe (including country groups, etc.), also down to the level of *departments*; and COG [1], an INSEE reference covering only France, but at all granularity levels. Thus, NUTS is wider, but covers only the high level, while COG is deeper and narrower, specific to France. To obtain a single spatial reference, we fused them on French departments (the nodes described in both), leading to about 40.000 distinct nodes in the hierarchy. Geographical units *evolve* over time in NUTS/COG, i.e., some svals were renamed or fused, etc. In our DAG, we keep *all*

successive versions of each unit, as a standalone sval. Each sval is thus characterized by: an ID, a *level* (in the DAG), a *validity interval* (a start year and an end year), together with a *name* and possibly some *alternate names*. For what concerns the **time**, we encountered dates from 1800 (sic!) to 2100 (a few statistic predictions), with a much higher coverage starting from the 1950's (presumably due to the development of digital databases), and a sharp decrease up to the present year (Fig. 2).

We **extract sval and tval values** from each title and header cell of a statistic dataset (data cells only contain numbers, and have been distinguished from header cells, as explained in [4]). For sval, we rely on exact matching with the name (or an alternate name) from the geographic reference. We also tried using trained language models (LM) to extract geographic locations, but they performed poorly on header cells (the vast majority of texts). This is because (*i*) these are very short strings, thus the LM lacks context; and (*ii*) some location names are confused with homonym common nouns, e.g., *Homme* (French for "man") and *Aucun* ("none").

For tval, we rely on a *pattern-based approach* to recognize various date formats, combined with a *threshold*, as follows: if at least $t\%$ of the values in a given header column (or row) are dates, we consider all of them to be dates. We call *spatial*, res. *temporal scope* of a dataset d, denoted σ_d, respectively, τ_d, the sets of tval, respectively, sval values it contains. If we find a sval label shared by two different spatial reference nodes, we include in the dataset's spatial scope all such sval nodes. We simplify temporal scopes at the *year* level, i.e., "Jan 2023", "2023" and "01/03/2023" each lead to the {2023}. In the dataset at right in Figure 1, the scope is {2021, 2022, 2023}.

3.3 Representing the datasets' linguistic content

The **textual** components of a dataset, i.e., the title, comments, and any text found in headers, encode the semantic richness of a statistic database. A proper representation and indexing of these components is crucial for both the retrieval and re-ranking phases. We experimented with: **(a) BM25**, used in prior work; **(b) Word2Vec**-based representations, as in StatCheck's prior method [4, 5]; and **(c) SBert** [12] embeddings, popular for their efficiency and accuracy.

For each dataset d, we compute such representations for *a set of strings derived from d*. This set contains: its title $d.t$; each header cell individually, denoted $d.h_{i,j}$; the concatenation of all row header cells, denoted $d.rh$; and the concatenation of all column headers, denoted $d.ch$. For instance, in the statistic table in Fig. 1, $d.t$ is "Job creation picks up across the country", headers cell include "Industry, 2023" and all the cells below, as well as "Île-de-France", "Paris", etc. up to "Yvelines"; $d.ch$ is a single string starting with "Île-de-France", and ending with "Yvelines"; we use \n as separator. The short texts in individual header cells are interesting because they are likely to match user-specified query terms. The concatenation of several header cells has been used in prior work, e.g., [6, 14].

To handle space and time separately from the rest of the text, for each string s_d derived from a dataset d as above, we compute $\overline{s_d}$, the *stripped version* of s_d, by *removing all the space and time* (sval and tval) extracted from s_d as shown in Sec. 3.2), while *keeping* $\overline{s_d}$ *as natural as possible* (from a linguistic perspective). For instance, stripping "premature deaths in France in 2022 per maternal age" yields "premature deaths per maternal age": we remove "France", "2022", and also their leading "in", with the help of a syntactic analysis. A stripped string may be empty, if it only consisted of geographic or space values; this is the case for the geographical header cells in Fig. 1. We build all the above representations over all the non-empty stripped strings derived from d as well.

Name	string	ϕ	Indices	$\oplus$
A	s_d	Word2Vec	I_{ft}, I_{fh}, I_{fcom}	Weighted sum [5]
B	$\overline{s_d}$	Word2Vec	I_{ft}, I_{fh}, I_{fcom}	Weighted sum [5]
C	$\overline{s_d}$	SBert	I_{ft}, I_{ch}, I_{rh}	max
D	$\overline{s_d}$	SBert	I_{ft}, I_{ch}, I_{rh}	sum
E	$\overline{s_d}$	SBert	I_{ft}, I_h	max
F	$\overline{s_d}$	SBert	I_{ft}, I_h	sum

Table 1: Table retrieval methods.

To enable efficiently looking up datasets for a given query (Sec. 3.4), we **index** within QDrant, a popular multidimensional index, entries of the form $(\phi(s_d), s_d, d.id, \sigma_d, \tau_d)$, and $(\phi(\overline{s_d}), \overline{s_d}, d.id, \sigma_d, \tau_d)$, where $\phi(\cdot)$ denotes one of the representations (a), (b), or (c) above. Thus, to each representation, we associate the original string, the dataset ID, as well as the spatial and temporal scopes of the dataset.

We build a separate index, denoted I^{ϕ}_{f}, for each method ϕ and **family (or group)** f **of strings** derived from the datasets the family *ft* contains all dataset titles, the family *fch* contains all column header concatenations, *frh* all the row header concatenations, *fcom* all the comments that may be associated to the dataset, and *fh* all individual header cells. The indexes are used to retrieve tables relevant for a query, as we discuss below.

3.4 Table retrieval and question answering

A **table retrieval method** is determined by: a choice of *original* or *stripped* strings; a linguistic representation method ϕ; a set of indexes $\{I^{\phi}_{f1}, \ldots I^{\phi}_{fp}\}$, each of which holds the ϕ representation of a string family f_i, original or stripped, according to this method's choice; and a *score aggregator*. When ϕ is SBert, $\otimes \in \{max, sum\}$; for Word2Vec, the aggregation method is a weighted sum as described in [5]. Given a query q, the method proceeds as follows.

(1) Extract the spatial and temporal scopes of q: σ_q and τ_q.
(2) Compute the vector $l = \phi(q)$ or $l = \phi(\overline{q})$.
(3) Issue a lookup to each index $I^{\phi}_{f1}, \ldots I^{\phi}_{fp}$ asking for their N entries closest to l, and whose dataset's temporal and spatial scopes intersect σ_q and τ_q. This leads to $p \times N$ entries; each entry is about a string s_i from one of the families $f1, \ldots, fp$, such that s_i is at distance δ_i from l (δ_i is returend by the lookup).
(4) Compute the score of each dataset d present these entries for q, as: $\otimes(\delta_1, \ldots, \delta_{n \times P})$.

Note that our primary search criterium is the representation of the query's linguistic core, and we only filter the results based on the spatial and temporal scopes. This is because (*i*) space and/or time may be missing from the query; (*ii*) we view the stripped query as expressing the core of the user query, while time and space are associated dimensions, in our multidimensional statistic tables.

On the list of retrieved datasets, we use the technique previously built in StatCheck [4] to identify answers at the *data cell*, *row*, *column*, or *table* level, which we prioritize in this order.

Query relaxation Some queries may have no results, e.g., "Unemployment in France in 2024", because this has not been measured

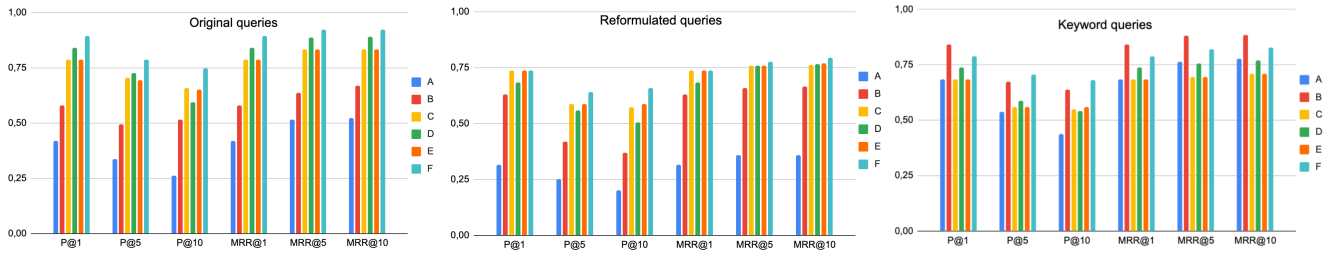

Figure 3: Precision (P@k) and Mean Reciprocal Rank (MRR@k) for different table retrieval methods, $k \in \{1, 5, 10\}$.

yet. In such cases, we rank the retrieved datasets by the smallest distance between their temporal scope and the query's (the smaller the distance, the better). If the space condition is not met, we suggest to the user either the parent, or one of the children of the original sval, taken from the geographical reference. If the query's space and time scope cannot be met, we compute both ranking orders (by space, resp., time proximity) and allow users to explicitly rank by one or another. Discussing with journalists, we found that when their searches do not succeed exactly, they prefer having the control over the approximations we introduce.

4 Evaluation

To see whether the separate treatment of space and time improves the precision of table retrieval, lacking search benchmarks on multidimensional statistic tables, we built a **benchmark** as follows.

We manually selected 20 statistic datasets from INSEE, and asked Llama3 [16] to generate (question, answer) pairs from each table, where the answer exists in the dataset (a cell, line, column, or whole table). We asked for *three query variants*: the *original* one, which tends to be quite close to the dataset vocabulary; a *reformulated* one which uses different words and may omit some details, i.e., some dimensions; and a *keyword* version where just the core concepts of the question are mentioned. We selected the tables to cover a diversity of topics; we asked Llama for three times more queries than we used, and picked for each dataset, the query (with its three versions) that we estimated of the best linguistic quality.

We experimented with many **table retrieval methods**. We could see that setting ϕ to BM25 performed poorly, which can be attributed to relatively little text in the datasets, and BM25's insensitivity to reformulations. The **six methods** which we found best performing and most interesting for our study, denoted A to E, are shown in Table 1. Method A is the previous one used in StatCheck [4], while the otehrs are novel, and rely on stripped strings, specific to our separate treatment of space and time. We asked each method in Table 1 for their top 20 results, leading to a **pool of 2123 datasets**, each potentially relevant to one query (in at least one of its variants). We manually scored each dataset, for the respective query, using $\{0, 1, 2\}$ labels (irrelevant, partially relevant, very relevant). Our benchmark with its annotations is online, as a resource for further research in this area.

We ran each method and show its precision P@k and Mean Reciprocal Rank MRR@k in Fig. 3. We make several observations: (i) *The Word2Vec-based method A using original (unstripped) strings s_d is almost always the worst*, in particular when the queries are full phrases. Indeed, this method was devised for keyword queries, and treats each keyword independently; in full phrases, this method does not distinguish the relative importance of each keyword, unlike phrase embeddings computed with SBert.

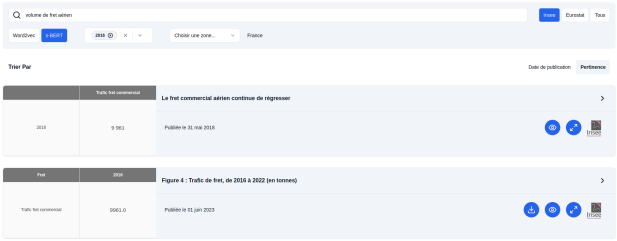

Figure 4: Demonstration screen shot.

(ii) *Method B systematically outperforms A (strongly in some cases).* This shows that looking up just with the core keywords, and *filtering* result by spatial and temporal scope as we do in STaR, improves performance even for this method. (iii) *Methods based on SBert (C to F) clearly outperform Word2Vec ones (A and B) on phrase queries*, demonstrating their advantage for such interactions with the data. Their good performance also shows that our stripping method (Sec. 3.3) succeeds in preserving natural-looking texts, whose SBert embeddings are close to queries asking for them. On the contrary, method B (Word2Vec on stripped keywords) is the best for keyword queries. (iv) *Among SBert methods, the* sum *aggregator performs better than* max: D generally outperforms C, and F always outperforms E (the difference between each of these two is the aggregator function $\oplus$). When more than one header cell(s) and/or the dataset title match the query, *sum* sends a strong signal that the dataset is relevant. (v) Overall, F is the best Bert-based method.

5 Demonstration Scenarios

We will demonstrate STaR within StatCheck (Fig. 4), based on the following statistic corpus: **194K datasets** extracted from more than 37K statistic publications from INSEE and Eurostat. An INSEE publication is an HTML page including HTML and/or CSV or XLS tables. An Eurostat publication consists of a title, and the data laid out as a CSV table, which can be (very) large. We split large tables into several smaller ones, primarily to have a more manageable unit of data returned to the user; in each small table, the entries agree on all dimension values except one. To these, we add **5K statistics datasets** crawled from official web sites from France, the US, Japan and Qatar (all but the French one are in English). INSEE publications are in French, Eurostat ones in French and English.

Users can **ask their own queries (or use pre-prepared ones)** and inspect the results using the table evaluation methods discussed. They can select time and space scopes via **explicit menus (year interval, space hierarchy)**; also, they can focus the query's spatial search, and get proposed frequent statistic keywords for that area, through a **zoomable world map**, to which we attach each dataset d by its geographical scope. **Demo video: link.**

References

[1] 2024. COG. https://www.insee.fr/fr/information/7766585.

[2] 2024. NUTS. https://ec.europa.eu/eurostat/documents/345175/629341/NUTS2021-NUTS2024.xlsx/2b35915f-9c14-6841-8197-353408c4522d?t=1702990824080.

[3] 2024. SDMX Technical Specifications. https://sdmx.org/?page_id=5008.

[4] Oana Balalau, Simon Ebel, Théo Galizzi, Ioana Manolescu, Quentin Massonnat, Antoine Deiana, Emilie Gautreau, Antoine Krempf, Thomas Pontillon, Gérald Roux, and Joanna Yakin. 2022. Fact-checking multidimensional statistic claims in French. In *Truth and Trust Online Conference*. https://truthandtrustonline.com/wp-content/uploads/2022/10/TTO_2022_paper_4.pdf

[5] Tien Duc Cao, Ioana Manolescu, and Xavier Tannier. 2018. Searching for Truth in a Database of Statistics. In *Proceedings of the 21st International Workshop on the Web and Databases, Houston, TX, USA, June 10, 2018*. ACM, 4:1–4:6. https://doi.org/10.1145/3201463.3201467

[6] Zhiyu Chen, Mohamed Trabelsi, Jeff Heflin, Yinan Xu, and Brian D. Davison. 2020. Table Search Using a Deep Contextualized Language Model. In *SIGIR*. https://doi.org/10.1145/3397271.3401044

[7] Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. Open Domain Question Answering over Tables via Dense Retrieval. In *ACL/HLT*. https://doi.org/10.18653/v1/2021.naacl-main.43

[8] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly Supervised Table Parsing via Pre-training. In *ACL*. Online. https://doi.org/10.18653/v1/2020.acl-main.398

[9] C.S. Jensen and R.T. Snodgrass. 1999. Temporal data management. *IEEE TKDE* 11, 1 (1999). https://doi.org/10.1109/69.755613

[10] George Katsogiannis-Meimarakis and Georgia Koutrika. 2023. A survey on deep learning approaches for text-to-SQL. *VLDB J.* 32, 4 (2023). https://doi.org/10.1007/S00778-022-00776-8

[11] Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir Radev. 2022. FeTaQA: Free-form Table Question Answering. *TACL* 10 (2022). https://doi.org/10.1162/tacl_a_00446

[12] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP-IJCNLP*. Hong Kong, China, 3982–3992. https://doi.org/10.18653/v1/D19-1410

[13] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3, 4 (2009). https://doi.org/10.1561/1500000019

[14] Roee Shraga, Haggai Roitman, Guy Feigenblat, and Mustafa Cannim. 2020. Web Table Retrieval using Multimodal Deep Learning. In *SIGIR* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1399–1408. https://doi.org/10.1145/3397271.3401120

[15] Yibo Sun, Zhao Yan, Duyu Tang, Nan Duan, and Bing Qin. 2019. Content-based table retrieval for web queries. *Neurocomput.* 349, C (jul 2019), 183–189. https://doi.org/10.1016/j.neucom.2018.10.033

[16] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR* abs/2302.13971 (2023). https://doi.org/10.48550/ARXIV.2302.13971 arXiv:2302.13971

[17] Qiming Wang and Raul Castro Fernandez. 2023. Solo: Data Discovery Using Natural Language Questions Via A Self-Supervised Approach. *PACMOD* 1, 4, Article 262 (dec 2023), 27 pages. https://doi.org/10.1145/3626756

[18] Shuo Zhang and Krisztian Balog. 2018. Ad Hoc Table Retrieval using Semantic Similarity. In *WebConf* (Lyon, France) *(WWW '18)*. 10 pages. https://doi.org/10.1145/3178876.3186067