# Efficient and Scalable Search for Statistics

## 42nd IEEE International Conference on Data Engineering (ICDE)

**Antoine Gauquier**[1] Simon Ebel[2]
Helena Galhardas[3] Théo Galizzi[2] Ioana Manolescu[2]
Aurélien Peden[2] Pierre Senellart[1]

[1]DI ENS, ENS, CNRS, PSL University & Inria, Paris, France
[2]Inria & Institut Polytechnique de Paris, Palaiseau, France
[3]INESC-ID & IST, Universidade Lisboa, Lisbon, Portugal

May 8, 2026

# Table of Contents

# Statistics in the Public Debate

Public debates rely on metrics: Inflation, growth, (un)employment, education, life expectancy, pollution, etc.

The value of a metric, on a given point of time, and for a given spatial scale, is a useful (and widely used) ingredient.

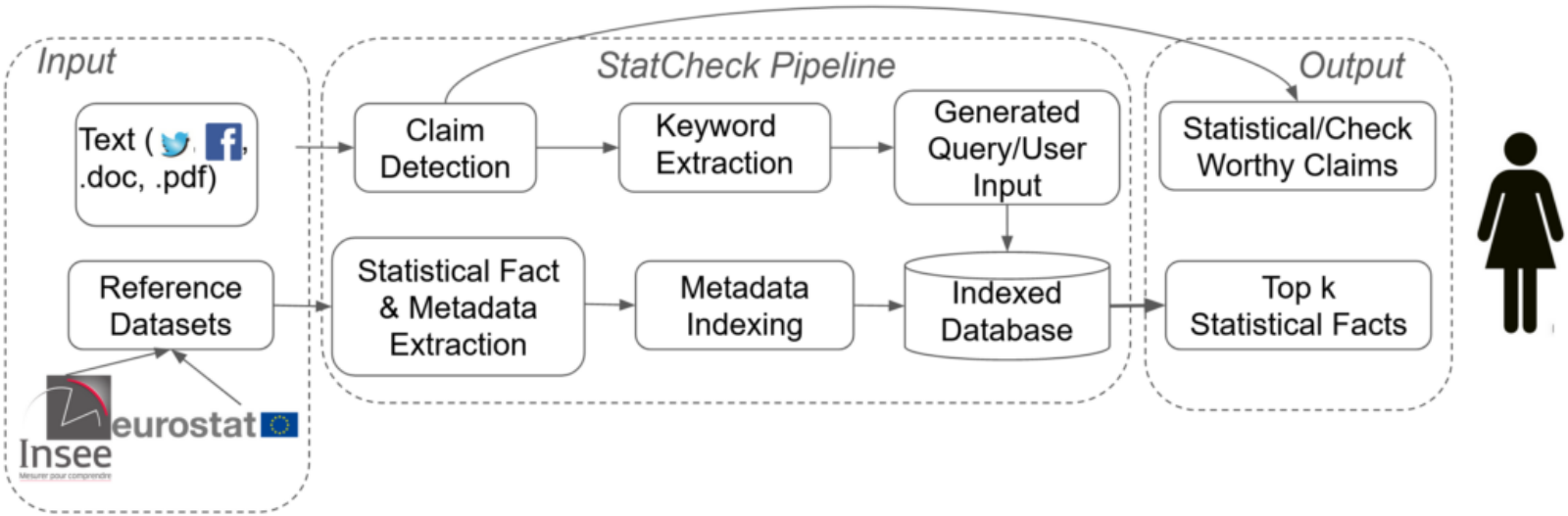
These metrics, generally called statistics, are produced by:

- *National agencies*: INSEE (France), Labor/Census Bureaus (US), ABS (AU)
- *International bodies*: Eurostat, IMF, WB, UN Agencies
- *Smaller entities*: Trade unions, NGOs

# Statistics in the Public Debate

Access to statistical data is crucial:

- To disseminate reliable information to the broad public
- To support journalists, enabling efficient work in an environment flooded with continuous information
- To counter misinformation, especially on social media → enables (semi-)automatic fact-checking



# Challenges in Accessing Statistical Data

Statistics are publicly available, but finding metric values is hard:

- *Heterogeneous formats*: CSV, Excel, HTML tables, JSON, SDMX (XML), etc.
- *Sometimes compressed*, potentially with other content
- Tables with up to *millions of rows* (e.g., in Eurostat)

1. We need automatic ways to query large corpora of statistical tables
2. The targeted audience are broad public/journalists → queries must be in natural language

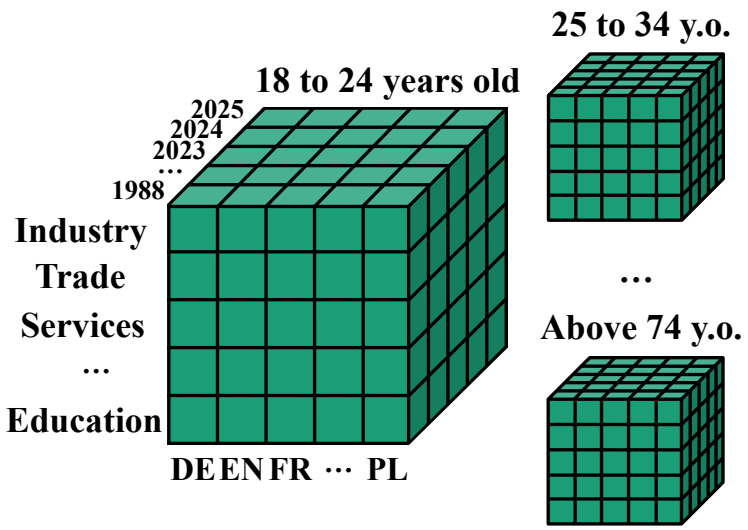
# Table of Contents

# Statistical vs Relational Tables

Statistical tables represent multidimensional data:

- E.g., "New jobs created" by year, country, sector, age group: each statistical value (*metric*) is defined for a set of *d dimension values*.
- Flattened into 2D tables for compatibility with classical tools: serialization.

**Number of new jobs created across the European Union**




| | 1988 | ... | 2024 | 2025 |
|---|---|---|---|---|
| Industry, DE, 24 to 34 y.o. | | | | |
| Trade, DE, 24 to 34 y.o. | | | | |
| Services, DE, 24 to 34 y.o. | | | | |
| ... | ... | ... | ... | ... |
| Trade, EN, 25 to 34 y.o. | | | | |
| ... | ... | ... | ... | ... |
| Education, FR, 25 to 34 y.o. | | | | |
| ... | ... | ... | ... | ... |
| Education, PL, Above 74 y.o. | | | | |

# Statistical vs Relational Tables

Challenges of serialized statistical tables:

- Key semantics are in row or column headers; cells are mostly numerical (no semantic cues).
- High dimensionality leads to very large tables (one row per combination of dimension values).
- Context for dimension values often missing.

Can traditional systems (for relational tables) handle this complexity?

# Table Discovery Methods

Table Discovery (TD) $\rightarrow$ Finding relevant tables within a corpus in response to natural language queries.

Recent methods [WF23, BAW+25, GHM+25] leverage DL and LLMs:

- Annotated data generation
- Table schema synthesis
- Result re-ranking

They are designed for relational tables with well-defined schemas.

Two core challenges:

1. Can they efficiently address the Statistical Table Discovery (STD) problem, i.e., the TD problem over statistical tables?
2. Is it possible to come up with a more frugal method to be used by a broader public (journalists, citizens)?

# Table of Contents

# The STAR Approach

**STAR** = Space- and Time-aware **Statistic** Retrieval

At **indexing time**, STAR:

1. Ignores numerical values → indexes only linguistic elements (titles, row and column headers)
2. Spatio-temporal entities are extracted
3. Aggregates linguistic elements and projects them in a latent space via a LM → store vectors in a multidimensional index
4. Extracted spatio-temporal elements are indexed separately using standardized hierarchies:
   - A Directed-Acyclic Graph for space (through reference sources, e.g., GeoNames)
   - A "natural" time hierarchy (through regular expressions)

   → STAR reasons about spatio-temporal covering

# The STAR Approach

At **query time**, STAR:

1. Takes as input an NL question
2. Question is parsed to extract spatio-temporal entities
3. Computes the subset of tables matching the constraints
4. Tables of the subset are ranked by semantic similarity with the embedding of the question
5. If no tables are found, the spatio-temporal constraints are progressively relaxed until having $k \geqslant 1$ matches.

# Table of Contents

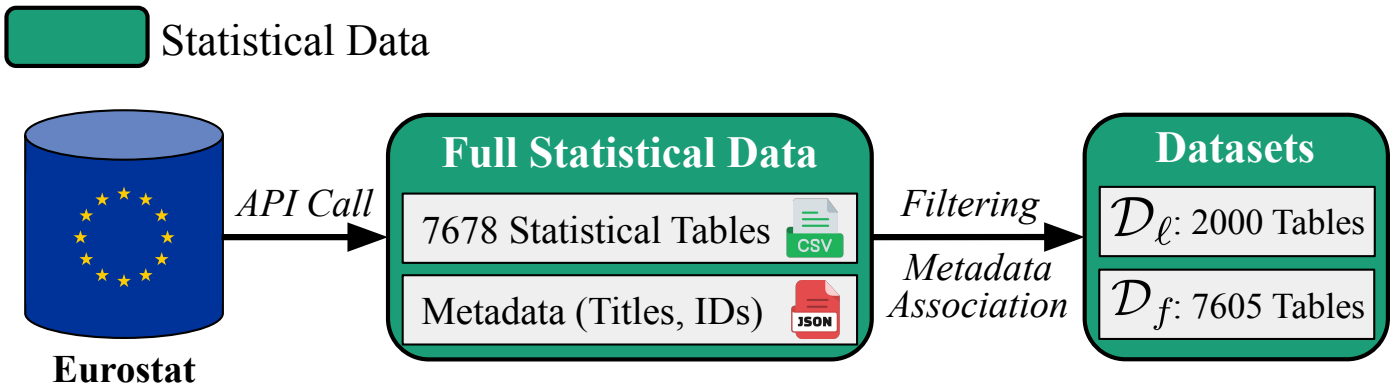
# Experiments: Datasets

Dataset: ∼7 600 Eurostat tables in CSV format ($\mathcal{D}_f$):

- As serialized tables
- With $d \geqslant 1$ dimensions: always includes time, and in most cases space
- Along with their titles (and IDs)

Subset of 2 000 smallest tables used for scalability tests ($\mathcal{D}_\ell$).

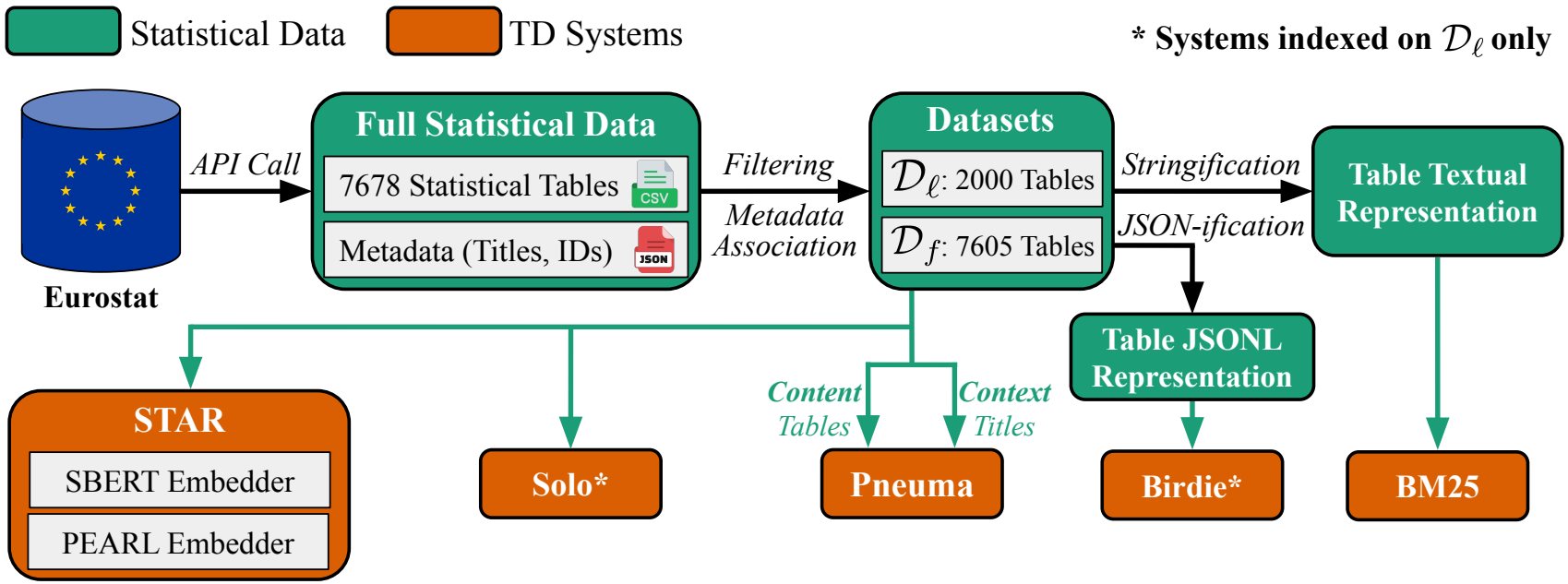
| Dataset | #Tables | Size (MB) | #rows | #columns |
|---|---|---|---|---|
| $\mathcal{D}_\ell$ | 2 000 | 30.3 | 111 (± 101) | 19 (± 23) |
| $\mathcal{D}_f$ | 7 605 | 98 521.7 | 69 077 (± 318 706) | 41 (± 270) |



# Experiments: Systems

We evaluate 7 systems:

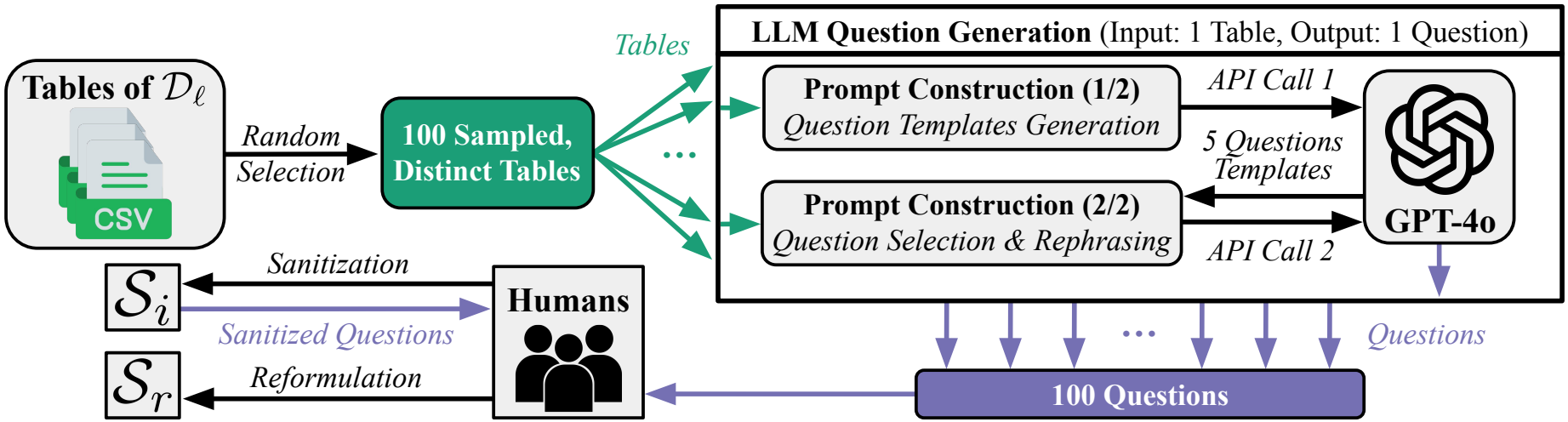
- STAR using standard SBERT embedder [RG19]
- STAR using PEARL [CVS24] → LM for short, context-free phrases
- SoTA TD systems [WF23, BAW+25, GHM+25] (with code fixes)
- BM25 baseline with ElasticSearch, and its variant BM25+Syn.



# Experiments: Question Sets

Two questions sets (for evaluation):

- Initial ($\mathcal{S}_i$): Lexically close to the table content
- Reformulated ($\mathcal{S}_r$): Same semantic meaning as $\mathcal{S}_i$, but with maximal lexical variation




Results in two high-quality, manually sanitized question sets
Evaluating on both lexical and semantic understanding

# Experiments: Metrics

We score each system with 2 metrics:

- **HitRate@$k$**: Proportion of questions from a question set for which a system returned the *golden* table in its top-$k$ ($1 \leqslant k \leqslant 10$).
  $\rightarrow$ **Standard metric in the literature**

- **Relevance@$k$**: measures a system's ability to return semantically relevant tables in its top-$k$ results ($1 \leqslant k \leqslant 5$).
  $\rightarrow$ **More interesting, as there might be $\geqslant 1$ relevant tables**

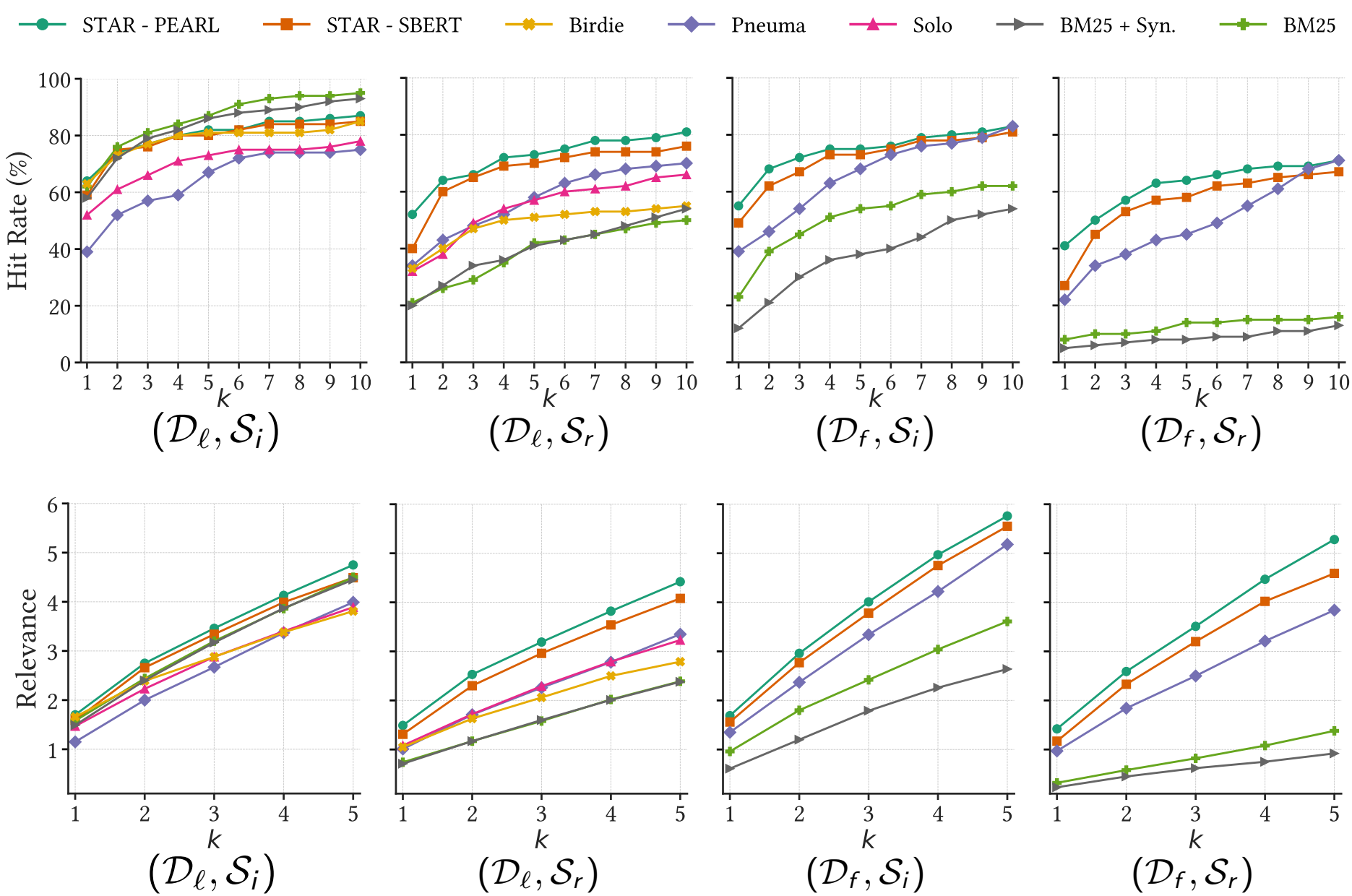
# Experiments: Results – Running Time

Running time of the systems, for indexing and querying:

| Model | Indexing Time (s) | | Query Time (s) | | | |
|---|---|---|---|---|---|---|
| | $\mathcal{D}_\ell$ | $\mathcal{D}_f$ | $(\mathcal{D}_\ell, \mathcal{S}_i)$ | $(\mathcal{D}_\ell, \mathcal{S}_r)$ | $(\mathcal{D}_f, \mathcal{S}_i)$ | $(\mathcal{D}_f, \mathcal{S}_r)$ |
| BM25 | 13 | 7 701 | 0.95 | 0.91 | 41.83 | 45.23 |
| BM25+Syn. | 13 | 7 701 | 3.39 | 3.34 | 47.16 | 51.47 |
| Solo | 81 847 | **OOM** | 1 267.05 | 1 143.06 | **OOM** | **OOM** |
| Pneuma | 45 300 | 532 020 | 3 793.23 | 3 680.90 | 1 770.44 | 1 814.24 |
| Birdie | 24 135 | **OOM** | 20.83 | 20.32 | **OOM** | **OOM** |
| STAR - SBERT | 371 | 45 103 | 36.77 | 37.66 | 186.62 | 190.95 |
| STAR - PEARL | 208 | 44 279 | 29.68 | 34.83 | 106.19 | 116.68 |

- Lexical-only methods scale and are the fastest
- Solo and Birdie do not scale to large datasets (OOM)
- For indexing: STAR is 12× to 393× faster than TD systems
- For querying: Slower than Birdie on $\mathcal{D}_\ell$, but 16× faster than the only scaling TD system (Pneuma) on $\mathcal{D}_f$

# Experiments: Results – Metrics



# Takeaways

1. We formalize the STD problem, and the challenges of statistical data
2. We present STAR, a space and time-aware statistic retriever, operating over query semantics
3. We provide a large novel benchmark for the STD task:
   - 7 600 statistic tables from Eurostat (98 GB),
   - 200 carefully curated questions for lexical and semantic evaluation,
   - nearly 5 000 manually annotated (question, table) pairs
4. STAR outperforms prior TD systems and lexical-only methods:
   - Some competing systems fail to scale (OOM errors)
   - Even when close in score $\rightarrow$ STAR is at least 12$\times$ faster
   - STAR requires no GPU (unlike LLM-based TD systems)

# Thank you for your attention!

# Any questions?

## STAR code

`https://gitlab.inria.fr/cedar/star-statcheck`

## Benchmark & extended version

`https://github.com/AntoineGauquier/efficient_and_scalable_search_for_statistics/`

# Bibliography I


Muhammad Imam Luthfi Balaka, David Alexander, Qiming Wang, Yue Gong, Adila Krisnadhi, and Raul Castro Fernandez.

Pneuma: Leveraging LLMs for Tabular Data Representation and Retrieval.

*Proc. ACM Manag. Data*, 3(3):200:1–200:28, 2025.

Lihu Chen, Gael Varoquaux, and Fabian Suchanek.

Learning High-Quality and General-Purpose Phrase Representations.

In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 983–994, St. Julian's, Malta, 2024. Association for Computational Linguistics.

Yuxiang Guo, Zhonghao Hu, Yuren Mao, Baihua Zheng, Yunjun Gao, and Mingwei Zhou.

Birdie: Natural Language-Driven Table Discovery Using Differentiable Search Index.

*ArXiv preprint*, abs/2504.21282, 2025.

# Bibliography II


Nils Reimers and Iryna Gurevych.

Sentence-BERT: Sentence embeddings using Siamese BERT-networks.

In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics.

Qiming Wang and Raul Castro Fernandez.

Solo: Data Discovery Using Natural Language Questions Via A Self-Supervised Approach.

*Proc. ACM Manag. Data*, 1(4):262:1–262:27, 2023.