

What Table Representation Learning Brings to Data Systems

Madelon Hulsebos

ETH Zürich
5 December 2024



The Impressive Capabilities of Transformers

Transformer–architecture leveraged for applications over images, text, code:



Generating funny dog images

You
what to include in speech with PhD graduation committee after defense ceremony?

ChatGPT
After successfully defending your PhD dissertation, it's customary to give a speech to express gratitude and appreciation to your graduation committee and others who have supported you throughout your academic journey. Here's what you might want to include in your speech:

- Acknowledgment of the Committee:** Begin by thanking each member of your graduation committee individually, mentioning their contributions to your research and their guidance during the defense process.
- Express Gratitude:** Express your sincere gratitude to your advisor, committee members, mentors, colleagues, friends, and family for their support, encouragement, and assistance throughout your PhD journey.
- Reflect on the Journey:** Reflect briefly on your academic journey, highlighting key milestones, challenges you've overcome, and significant moments of growth and learning.
- Highlight Contributions:** Summarize the main contributions of your research and its potential impact in your field or related areas. This is an opportunity to showcase the significance of your work and its relevance.

Writing graduation speech

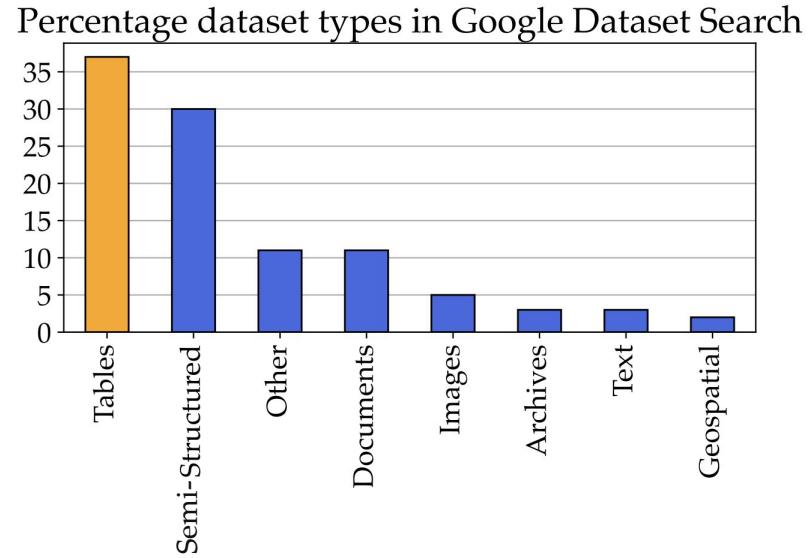
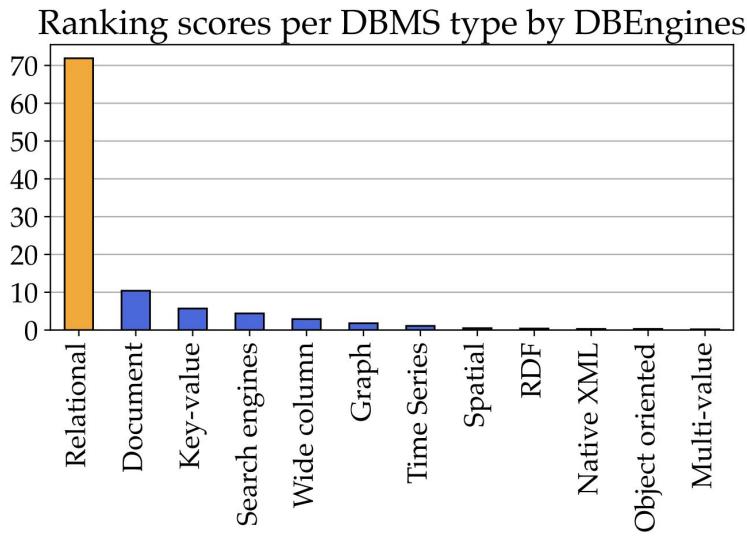
```
sentiment.ts
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10     body: `text=${text}`,
11     headers: {
12       "Content-Type": "application/x-www-form-urlencoded"
13     }
14   });
15   const json = await response.json();
16   return json.label === "pos";
17 }
```

Completing code

What about tables?

We have LLMs... why not analyze docs?

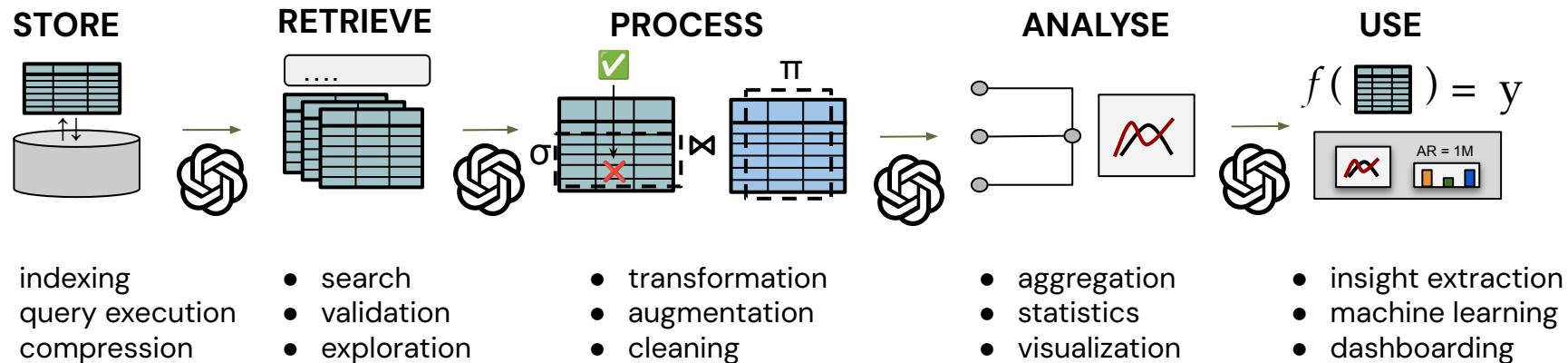
Tables Dominate the Data Landscape



Potential of Table Representation Learning

Available... but also: fresh, structured, domain, data!

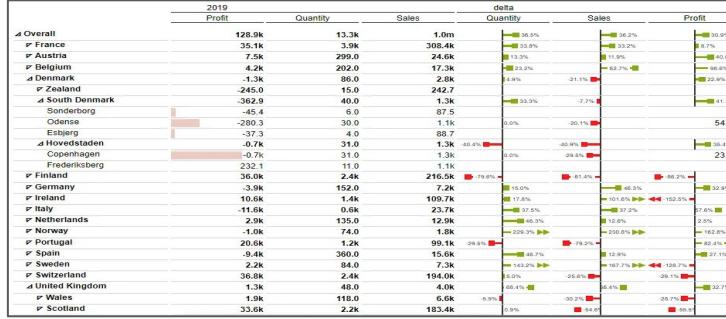
High value use-cases, e.g. *data analysis*: many tables, many tasks!



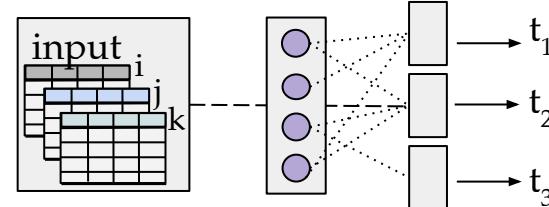
Relational data: rich and challenging

Diverse in dimensions, structure, cleanliness and semantics...

crop rotation : Tabelle												
Nr	ID	seed rate	yield	crop	cultivar	pre crop	pre-pre crop	pre-pre-pre	soil type	precipita	tempera	comment
1	68	91	winter wheat	sugar beets	beans		sandy loam, loe 636	9.6	wb, sg,			
2	68	100	winter wheat	sugar beets	rotation fallow		sandy loam, loe 636	9.6	cultivation			
3	68	97	winter wheat	sugar beets	fallow land (5,5y)		sandy loam, loe 636	9.6	1993-1996			
4	136	95	winter wheat	oats	sugar beets		sandy loam, loe 636	9.6				
5	136	96	winter wheat	potatos	sugar beets		sandy loam, loe 636	9.5	cultivation			
6	136	107	winter wheat	sugar beets	maize		sandy loam, loe 636	9.5	1991-1994			
7	136	107	winter wheat	sugar beetsn	summer wheat	maize	sandy loam, loe 636	9.5				
8	136	82	winter wheat	oats	sugar beets	sugar beets	sandy loam, loe 636	9.5	organic			
9	136	77	winter wheat	potatos	sugar beets		sandy loam, loe 636	9.5	organic			
10	136	85	winter wheat	sugar beets	maize	maize	sandy loam, loe 636	9.5	organic			
11	136	84	winter wheat	sugar beets	summer wheat	sugar beets	sandy loam, loe 636	9.5	organic			
12	57 371	98	winter wheat	Sperber	sugar beets	winter barley	winter wheat	sandy loam, loe 635	wb, ww			
13	57 365	98	winter wheat	Sperber	potatos	sugar beets	summer barley	sandy loam, loe 635	cultivation, weed			
14	57 365	108	winter wheat	Sperber	sugar beets	maize	maize	sandy loam, loe 635	1987-1992			
15	57 365	97	winter wheat	Sperber	sugar beets	winter wheat	sugar beets	sandy loam, loe 635				
16	39 433	90	winter wheat	Okapi	summer barley		sandy loam, loe 690	8.5	oats, cultivation, weec			
17	39 433	100	winter wheat	Okapi	oats		clay, silt	690	8.5	1982-1986		
18	39 433	97	winter wheat	Okapi	winter wheat		clay, silt	690	8.5			



Goal TRL: map tables to some consistent input.
Learn some representation that helps detect
patterns relevant to given task(s).



Today...

- ¹ The power of table and column **semantics**
- ² What we need to **make TRL work**
- ³ Towards **end-to-end data analysis** tools

The power of table and column semantics

Essential understanding
of a table comes
through its columns.

Semantic column types: what and why?

name	salary	country
name	salary	cntr

Looks easy, but....

- Undescriptive header?
- Messy values?
- Diverse data types?

Semantic column types dictate operations *sensible* to perform on them:

name	salary	cntr



naam	status	land

Join tables on "name" and "country" columns

name
Xi
carl
sara

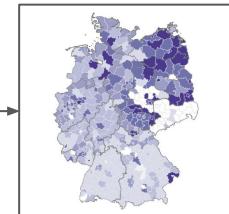


name
Xi
Carl
Sara

Capitalize "name" columns

name	salary	cntr

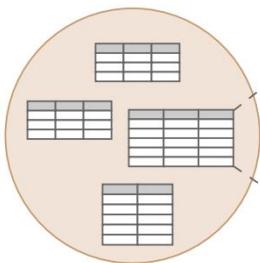
Plot "country" data



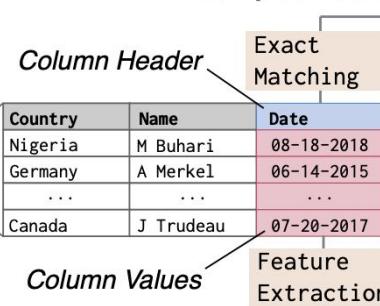
Sherlock: Column Type Detection with DL

Prior: string matching (header/values) w/ **regex** or dict: robust? scale? accurate?

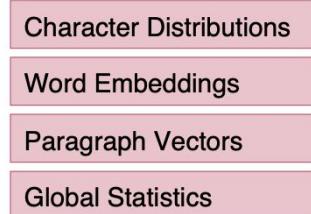
1. Source Corpus



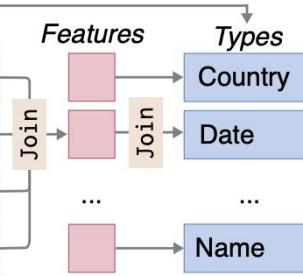
2. Sampled Dataset and Features



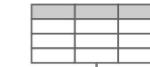
Feature Categories



3. Training and Testing Set



4. Semantic Type Detection



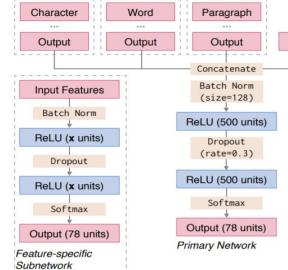
Unseen Data

Model

Prediction

Location: 0.9
Name: 0.7
Year: 0.8

Predicted Types and Confidences



<https://sherlock.media.mit.edu>

How well does Sherlock detect types?

78 semantic types (name, address, etc).

Method	F ₁ Score	Runtime (s)	Size (Mb)
<i>Machine Learning</i>			
Sherlock	0.89	0.42 (± 0.01)	6.2
Decision tree	0.76	0.26 (± 0.01)	59.1
Random forest	0.84	0.26 (± 0.01)	760.4
<i>Matching-based</i>			
Dictionary	0.16	0.01 (± 0.03)	0.5
Regular expression	0.04	0.01 (± 0.03)	0.01
<i>Crowdsourced Annotations</i>			
Consensus	0.32 (± 0.02)	33.74 (± 0.86)	–

Examples of misclassifications.

Examples	True type	Predicted type
<i>Low Precision</i>		
81, 13, 3, 1	Rank	Sales
316, 481, 426, 1, 223	Plays	Sales
\$, \$\$, \$\$\$, \$\$\$\$, \$\$\$\$\$	Symbol	Sales
<i>Low Recall</i>		
#1, #2, #3, #4, #5, #6	Ranking	Rank
3, 6, 21, 34, 29, 36, 54	Ranking	Plays
1st, 2nd, 3rd, 4th, 5th	Ranking	Position

Challenges

- Numeric data
- Non-mutually exclusive types

Don't we have LLMs now?

"Table-tuned" LLM (but not for semantic type detection) [1]:

Zero-Shot		Few-Shot	
GPT-3.5	+table-tune	GPT-3.5	+table-tune
0.332	0.449	0.528	0.538

Sherlock model: ~0.88 F1.

LLM (GPT-3.5) w/ more examples and specific context [2]:

	F ₁ -score	Precision	Recall
DoDuo-VizNet*	0.876	89.4%	87.2%
Sherlock*	0.954	96.2%	94.6%
TaBERT	0.321	32.6%	32.0%
DoDuo-Wiki	0.440	59.2%	45.4%
CHORUS	0.891	91.2%	88.8%

Sure, GPT-x might do better..
but w/ **billions** of params vs **thousands**!

Representation Learning (LM trained on type detection) [3]:

Method	F1	P	R
Sherlock (only entity mention) [17]	78.47	88.40	70.55
TURL + fine-tuning (only entity mention)	88.86	90.54	87.23
TURL + fine-tuning w/o table metadata	94.75	94.95	94.56
only table metadata	93.77	94.80	92.76
	90.24	89.91	90.58

[1] TableGPT: Table-tuned gpt for diverse table tasks. P. Li et al, VLDB, 2024

[2] CHORUS: Foundation Models for Unified Data Discovery and Exploration. Kayali, et al. VLDB, 2024.

[3] TURL: Table understanding through representation learning. Xiang Deng, et al., ACM SIGMOD Record, 2022.

Semantics for optimizing data systems

Example: column semantics -> correlations.

Cardinality estimation

Learned Cardinalities: Estimating Correlated Joins with Deep Learning

Andreas Kipf
Technical University of Munich
kipf@in.tum.de

Viktor Leis
Technical University of Munich
leis@in.tum.de

Thomas Kipf
University of Amsterdam
t.n.kipf@uva.nl

Peter Boncz
Centrum Wiskunde & Informatica
boncz@cwi.nl

Bernhard Radke
Technical University of Munich
radke@in.tum.de

Alfons Kemper
Technical University of Munich
kemper@in.tum.de

Compression

Lightweight Correlation-Aware Table Compression

Mihail Stoian, Alexander van Renen, Jan Kobiolka, Ping-Lin Kuo,
Josif Grabocka, Andreas Kipf
{mihail.stoian, alexander.van.renen, jan.kobiolka, ping-lin.kuo,
josif.grabocka, andreas.kipf}@utn.de
University of Technology Nuremberg

Can Large Language Models Predict Data Correlations from Column Names?

Immanuel Trummer
Cornell Database Group
Ithaca, NY, USA
itrummer@cornell.edu

What we need to make TRL work

As LLM scaling laws
reach their limits:
it is all about the
“quality” of the data,
and the “tricks” we apply.

What Data Do We Need?

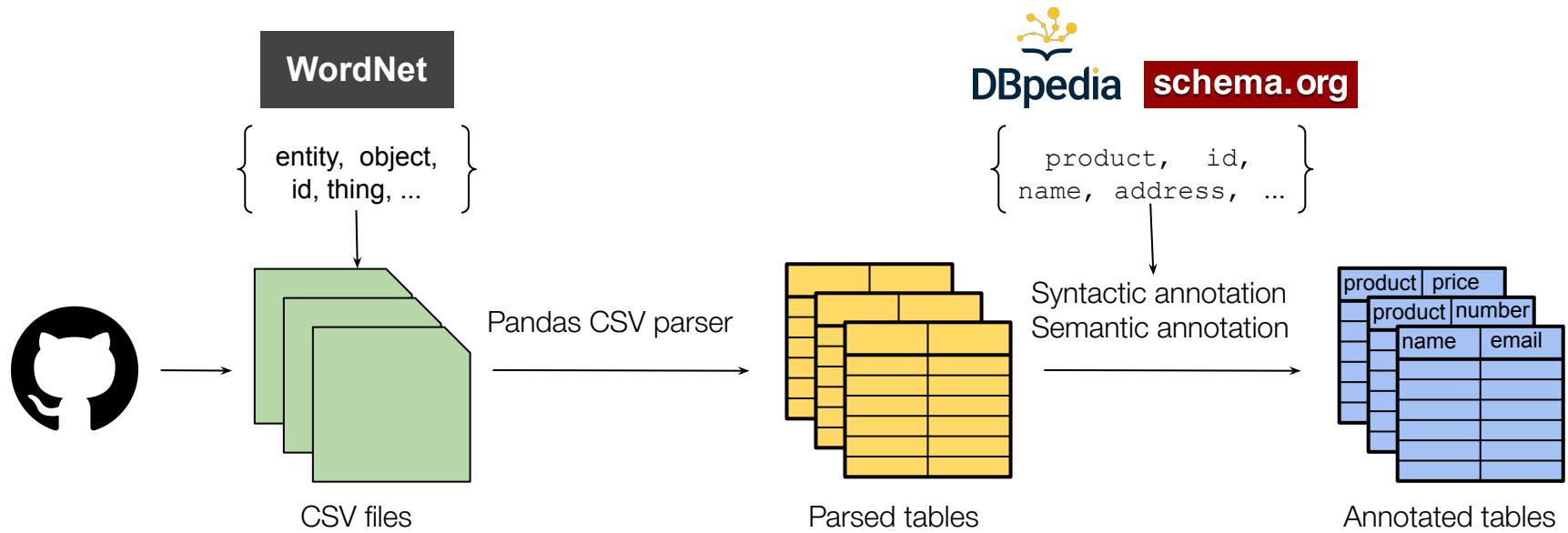
- Web/WikiTables → Web applications. Web tables ≠ DB tables...
- Data tasks on offline tables? GitHub as a data source?

The screenshot shows a GitHub search results page with the query "extension:csv" "id". The results count is 15,768,996 code results. A specific repository, "Kreef123/Sendy-Logistics-Challenge", is highlighted, showing a file named "data/Riders.csv". The file content is displayed as a table:

	Rider Id, No_of_Orders, Age, Average_Rating, No_of_Ratings
1	Rider Id, No_of_Orders, Age, Average_Rating, No_of_Ratings
2	Rider Id 396, 2946, 2298, 14, 1159
3	Rider_Id_479, 360, 951, 13.5, 176
4	Rider_Id_648, 1746, 821, 14.3, 466
5	Rider_Id_753, 314, 980, 12.5, 75
6	Rider_Id_335, 536, 1113, 13.7, 156

CSV Showing the top six matches Last indexed on 27 Mar 2021

GitTables: a new large corpus with tables



<https://gittables.github.io>

Using GitTables

- >1M tables and 800K CSV files.
- Wider+taller, and lots of IDs; more representative.
- Useful for *semantic column type detection and schema completion*:

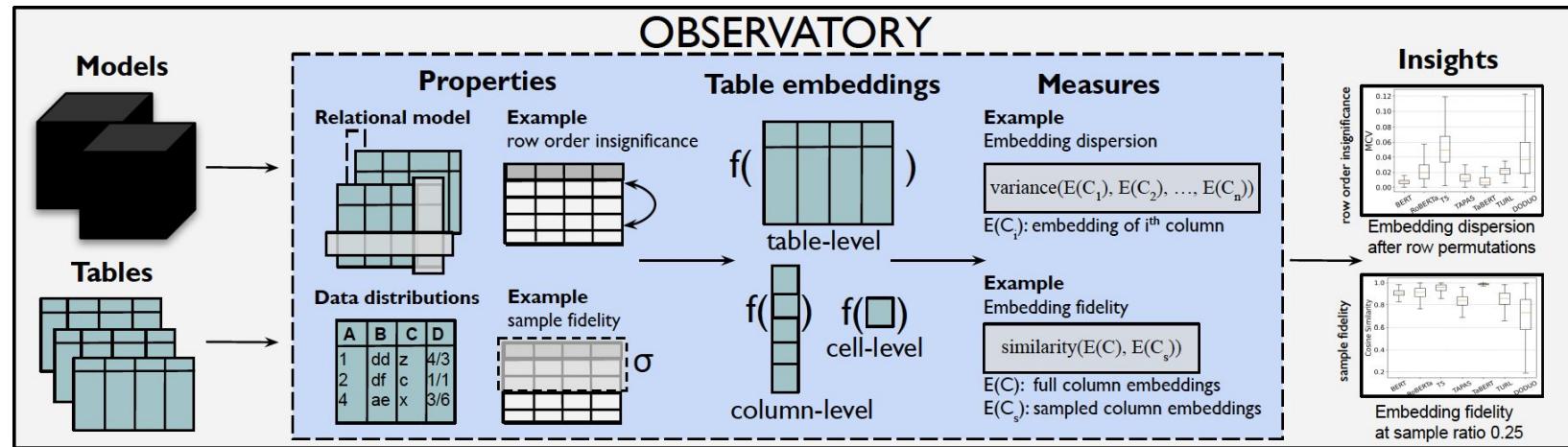
Header prefix	Suggested completion
payment_id, customer_id	review_id, product_id, product_parent, product_title, ...
id, company	ReceivablePaymentHeader, ReceivablePayment, Status, Customer, BankEntity, ...
id, name, location	phone, email, uid, active, ad_organization_id, ...

Used for join discovery, CSV parsing, KG enhancement, retrieval eval, etc.
Other corpora to bridge the “realism gap” e.g. **SchemaPile, BIRD, Spider**.

Do ‘tableLM’ tricks capture relational properties?

Tables ≠ natural language

Studying neural table embeddings through **Codd’s relational model**.

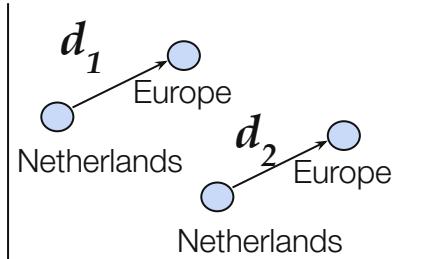


Example Property: Functional Dependencies

Given table with FD: $X=\text{country} \rightarrow Y=\text{continent}$

We argue that:

- FD relations interpretable as *translation* between embeddings $E(\pi X(s))$ and $E(\pi Y(s))$
- Model preserves FD if $d(E(\pi X(s)), E(\pi Y(s))) = d(E(\pi X(t)), E(\pi Y(t)))$ where d preserves magnitude+direction (L1/L2-norm).
- Intuitively:



ID	name	country	continent
1	Kathryn	Netherlands	Europe
2	Oscar	Netherlands	Europe
3	Lee	Canada	North America
4	Roxanne	USA	North America
5	Fern	Netherlands	Europe
6	Raphael	USA	North America
7	Rob	USA	North America
8	Ismail	Canada	North America

Current Architectures Often Fall Short...

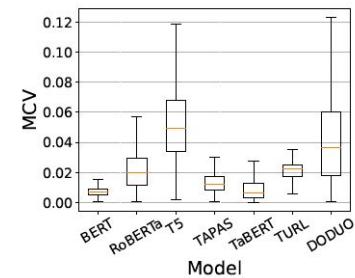
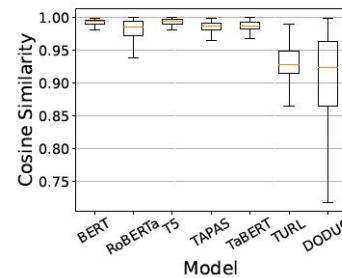
Turns out, most models do not preserve FDs!

RM [4] also has straightforward properties:

A *relation* then consists of a set of tuples, each tuple having the same set of attributes. If the domains are all simple, such a relation has a tabular representation with the following properties.

- (1) There is no duplication of rows (tuples).
- (2) Row order is insignificant.
- (3) Column (attribute) order is insignificant.
- (4) All table entries are atomic values.

Measure by avg cosine similarity of col embeddings across row permutations.



row order robustness

Impact downstream tasks: **row shuffling affects 34% semantic column types!**

Towards end-to-end data analysis systems

End-to-end DS goes far beyond “automl”



2018_Central_Park_Squirrel_Census_-_Squirrel_Data_20240501.csv

Analyze the proportion of adult and juvenile animals in the census data. Are there any spatial patterns in age distribution?

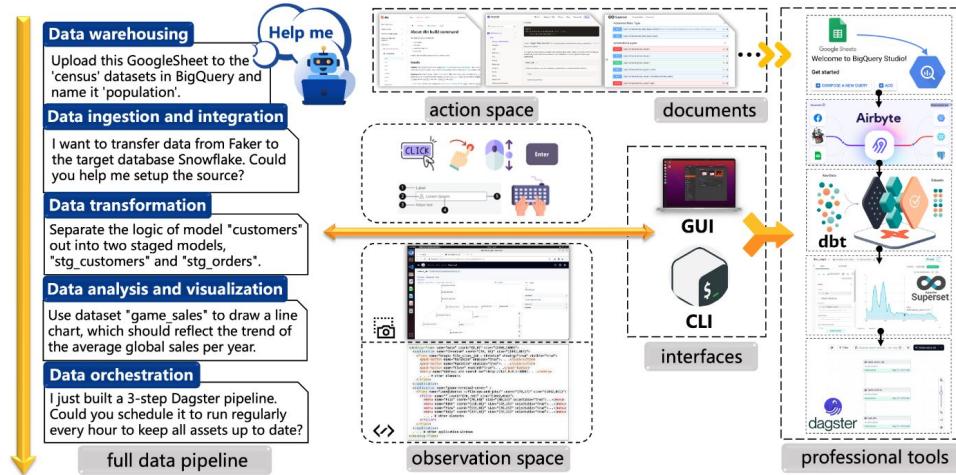
2018_Central_Park_Squirrel_Census_-_Squirrel_Data_20240501.csv

Plan

Typically lots of text-to-code (e.g. SQL) involved!

From DS codegen to DS GUI agents

Spider2-V Framework Infrastructure



What's the right level of abstraction for data agents to operate on, anyway?
Should systems even still focus on human usage?

Sounds cool, how do we get there?

The screenshot shows the Data Science Agent interface. At the top, there's a purple header bar with a lab flask icon, the text "Data Science Agent", and an "Experiment" button. Below the header, there are two tabs: "README.md" and "Playground", with "Playground" being the active tab. A file named "2018_Central_Park_Squirrel_Census_-_Squirrel_Data_20240501.csv" is listed below the tabs. The main area contains several dropdown menus. The first dropdown is labeled "Plan". The second dropdown, which has a blue arrow pointing to it from the left, is labeled "Data Loading". Both dropdowns have a small downward arrow icon at the end.

Who or what is doing data analysis, it will need the right data first.

Finding the right data
for basic questions or
deep analysis
is *still* not easy.

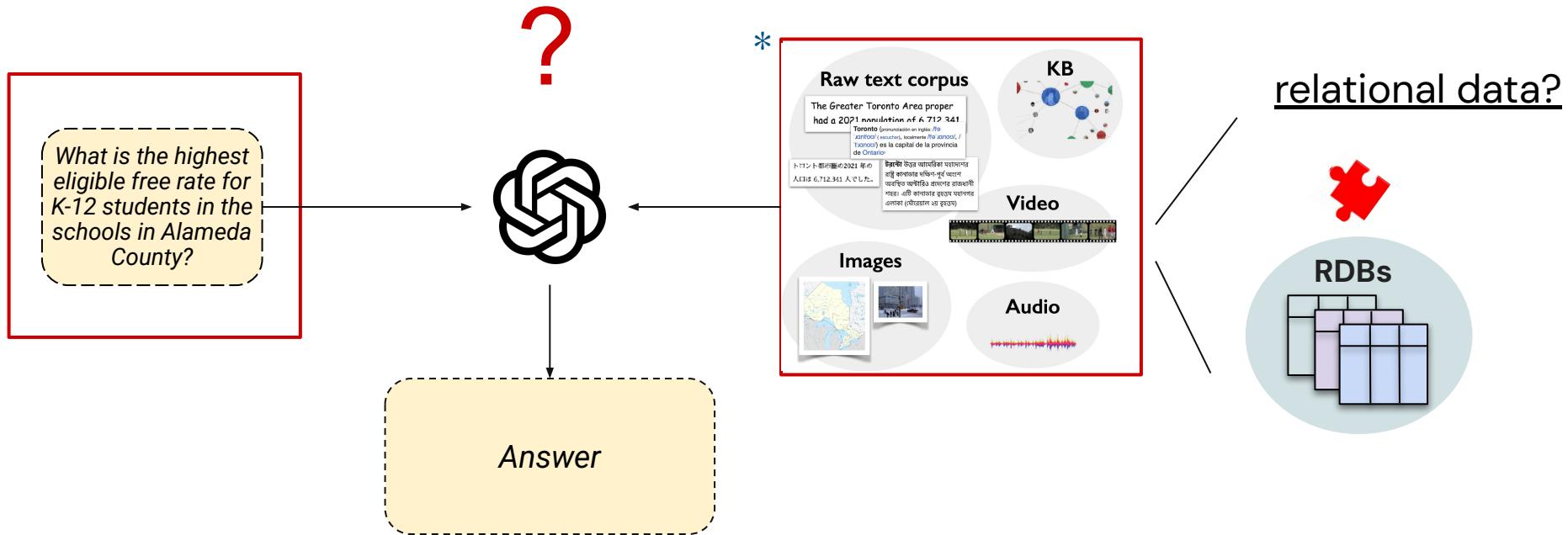
Asking LLMs complex questions

What is the highest eligible free rate for K-12 students in the schools in Alameda County?



.... To determine the highest free rate specifically in Alameda County schools, **you'd generally need data from specific school districts or schools in the area**, as this rate can vary widely depending on the socio-economic demographics of each district. ... *

We need “specific” data to ground LLMs

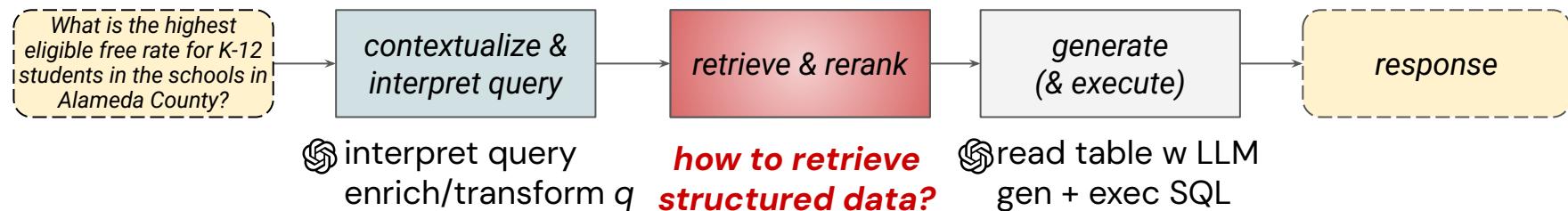


Queries & RAG pipeline

"Which urban Japanese prefecture is not associated with thorny trees?" [table lookup]

"Shane Hall ran a total of 190 races between the year of 1995 – 2008" [aggregate & compare]

"What is the highest eligible free rate for K-12 students in the schools in Alameda County" [aggregate]



Retrieval is difficult, but crucial...

“.. keep in mind that a good RAG system is really hard to build.

If your **retrieval system is mediocre**,
the **retrieval can easily distract LLMs...**

There is still a long way to go.” – Wenhui Chen (Univ of Waterloo)

Methods for table retrieval

① Embed tables in corpus

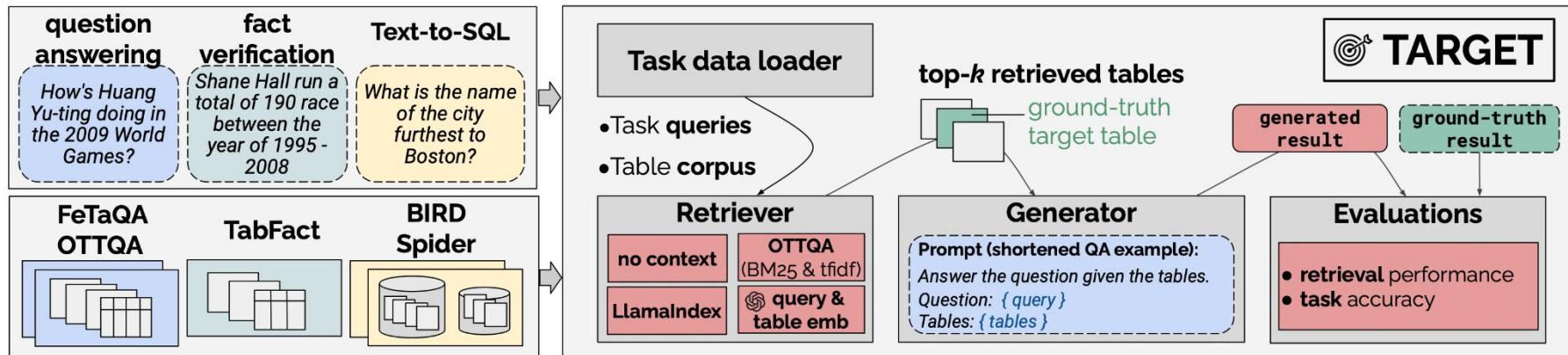
- BM25 / TF-IDF (sparse lexical representations)
- Generate summary/metadata → embed summary + table
- “Naive” embedding of table (header / header+rows) and query

② Embed query

③ Similarity search (e.g. cosine similarity) to identify top- k relevant tables

But how effective are these? How robust across datasets and tasks? No one really knows!

TARGET: Benchmarking Table Retrieval for Generative Tasks



<https://target-benchmark.github.io> (pip install target_benchmark)

TARGET insights

Method	Question Answering						Fact Verification						Text-to-SQL					
	OTTQA			FeTaQA			TabFact			Spider			BIRD					
	R@10	s	SB	R@10	s	SB	R@10	s	P/R/F1	R@1	s	EX	R@1	s	EX			
No context	-	-	0.414	-	-	12.495	-	-	0.578/0.42/0.44	-	-	0	-	-	-	0		
OTT-QA BM25	0.955	0.001	0.606	0.082	0.001	1.631	0.338	0.001	0.75/0.26/0.39	0.635	0.001	0.385	0.709	0.001	0.181			
<i>w/o table title</i>	0.443	0.001	0.529	0.084	0.001	1.555	0.331	0.001	0.75/0.26/0.38	0.5	0.001	0.376	0.535	0.001	0.164			
OTT-QA TF-IDF	0.950	0.001	0.425	0.083	0.001	1.639	0.336	0.001	0.75/0.26/0.38	0.622	0.001	0.474	0.640	0.001	0.227			
<i>w/o table title</i>	0.43	0.001	0.593	0.083	0.001	1.527	0.322	0.001	0.75/0.25/0.37	0.492	0.001	0.376	0.491	0.001	0.164			
LlamaIndex	0.458	0.354	0.507	0.435	0.396	13.745	0.827	0.297	0.73/0.34/0.47	0.735	0.198	0.559	0.937	0.228	0.311			
OpenAI embedding	0.950	0.190	0.599	0.722	0.200	17.64	0.779	0.189	0.76/0.51/0.61	0.768	0.193	0.602	0.926	0.199	0.317			
<i>header only</i>	0.950	0.189	0.61	0.718	0.18	17.66	0.781	0.187	0.75/0.48/0.58	0.833	0.175	0.646	0.958	0.191	0.323			

- BM25/TF-IDF **less effective than for text**, only works with descriptive table name.
- Table **rows can “distract” embeddings**, particularly in RDBs as seen in practice.
- Generating summary/metadata can help, but **not all tables easy to LLM-summarize**.

Still much to explore...

- What is the right input of (meta)data to not “distract” embedding?
- How do we route to proper data source, interpret the task, etc?
- **The reality in practice is much harder:**
 - How do methods perform on more *challenging tasks & datasets*?
 - Closing semantic gap $e(\text{query})$ and $e(\text{table})$; most public datasets relatively “easy” match between query and tables.
 - Relational databases are large → in-DB schema and table retrieval.

Roadmap for TARGET

Take aways...

- **Tables are everywhere**, serving **high-value use-cases** in e.g. gov, health, finance.
- **Tables ≠ natural language**: tables come with **specific properties** (e.g. relational).
- Capabilities of “**foundation**” models should extend to **tables & relational DBs**.
- For this, we need the **right data**, and the **right “tricks”**.
- For any data analysis system, human or agentic; **retrieval is key** (e.g. tables, context).

madelonhulsebos.com, madelon@cwi.nl, @madelonhulsebos  

Hulsebos, M., Hu, K., Bakker M., et al. "Sherlock: A deep learning approach to semantic data type detection." ACM SIGKDD 2019.

Hulsebos, M., Demiralp, C., Groth, P. "GitTables: A large-scale corpus of relational tables." SIGMOD 2023.

Cong, T., Hulsebos M., Groth, P., Jagadish, H. "Observatory: Characterizing embeddings of relational tables." VLDB 2024.

Ji, X., Parameswaran, A., Hulsebos, M. "TARGET: Benchmarking Table Retrieval for Generative Tasks." TRL @ NeurIPS 2024.