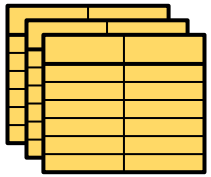


Towards “Large Table Models”

for enterprise data management

Madelon Hulsebos

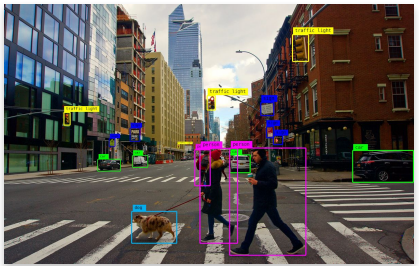
TU Darmstadt
3 June 2022

sigma

UNIVERSITY
OF AMSTERDAM

Plethora of Large <something> Models

We achieved artificial "understanding" of images, natural language, and code.

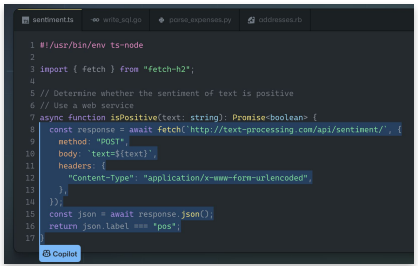


Large **Image** Models
for detecting objects in videos

Input: What's the difference between a zebra and an umbrella? One is a striped animal related to horses, another is a device you use to stop the rain from falling on you.

Model Output: This joke is an anti-joke. The joke is that the answer is obvious, and the joke is that you were expecting a funny answer.

Large **Language** Models
for explaining (anti-)jokes



Large **Code** Models
for completing code

We need Large **Table** Models

Relational tables dominate the enterprise data landscape.

Rank May 2022	Rank Apr 2022	Rank May 2021	DBMS	Database Model
1.	1.	1.	Oracle	Relational, Multi-model
2.	2.	2.	MySQL	Relational, Multi-model
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model
4.	4.	4.	PostgreSQL	Relational, Multi-model
5.	5.	5.	MongoDB	Document, Multi-model

Category	Number of datasets	% of total	Sample formats
Tables	7,822K	37%	CSV, XLS
Structured	6,312K	30%	JSON, XML, OWL, RDF
Documents	2,277K	11%	PDF, DOC, HTML
Images	1,027K	5%	JPEG, PNG, TIFF
Archives	659K	3%	ZIP, TAR, RAR
Text	623K	3%	TXT, ASCII

And the "modern data stack" would benefit from some "intelligence":

- data ingestion, validation, search, integration, preparation, analysis.
- query optimization, validation, recommendation.

Sherlock: A Deep Learning Approach to Semantic Data Type Detection

Madelon Hulsebos
MIT Media Lab
madelonhulsebos@gmail.com

Kevin Hu
MIT Media Lab
kzh@mit.edu

Michiel Bakker
MIT Media Lab
bakker@mit.edu

Emanuel Zgraggen
MIT CSAIL
emzg@mit.edu

Arvind Satyanarayan
MIT CSAIL
arvindsatya@mit.edu

Tim Kraska
MIT CSAIL
kraska@mit.edu

Çağatay Demiralp
Megagon Labs
cagatay@megagon.ai

César Hidalgo
MIT Media Lab
hidalgo@mit.edu

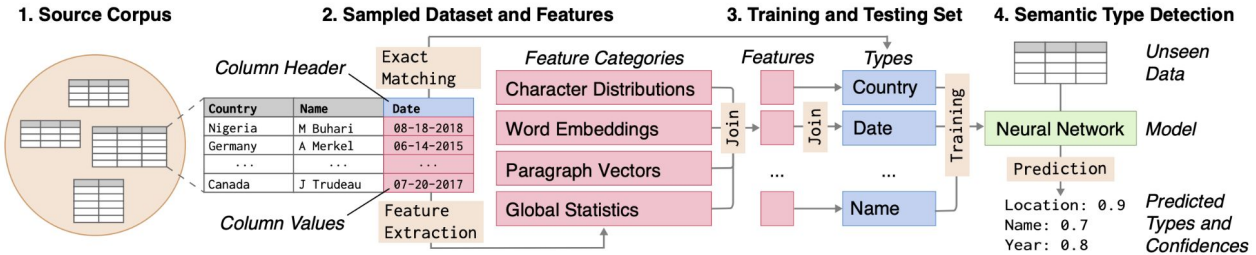


Figure 1: Data processing and analysis flow, starting from (1) a corpus of real-world datasets, proceeding to (2) feature extraction, (3) mapping extracted features to ground truth semantic types, and (4) model training and prediction.

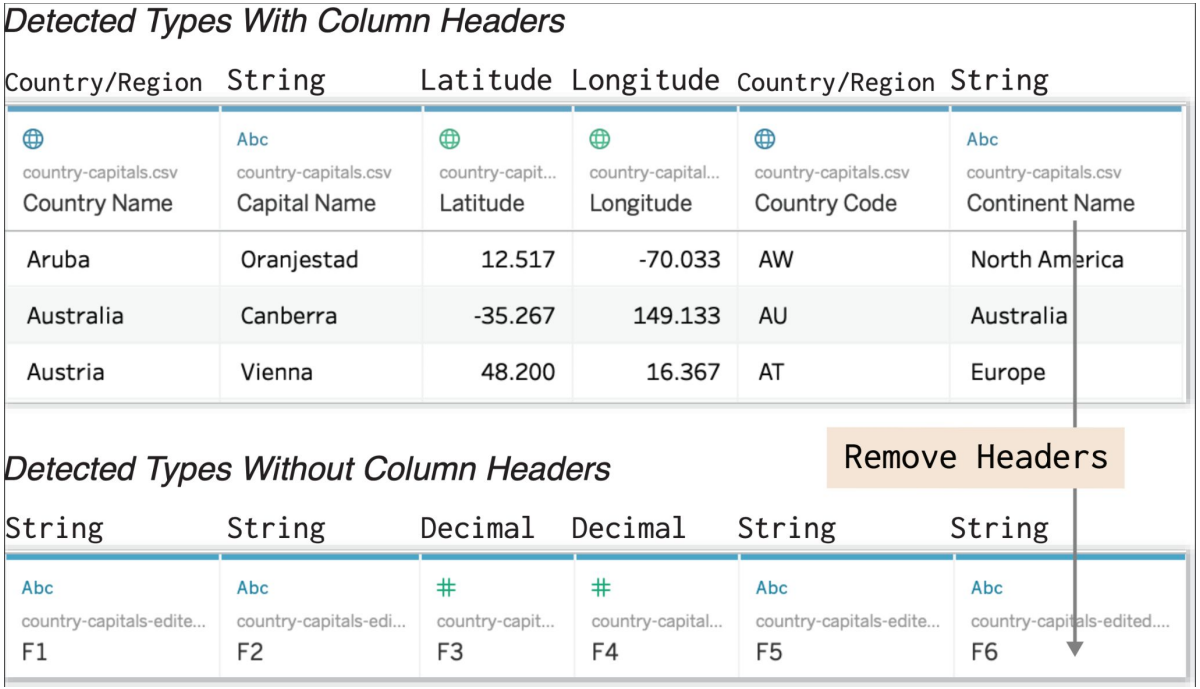
mit media lab

CSAIL

Megagon Labs

The need for semantic column types

Similar to "object detection", but in tables…



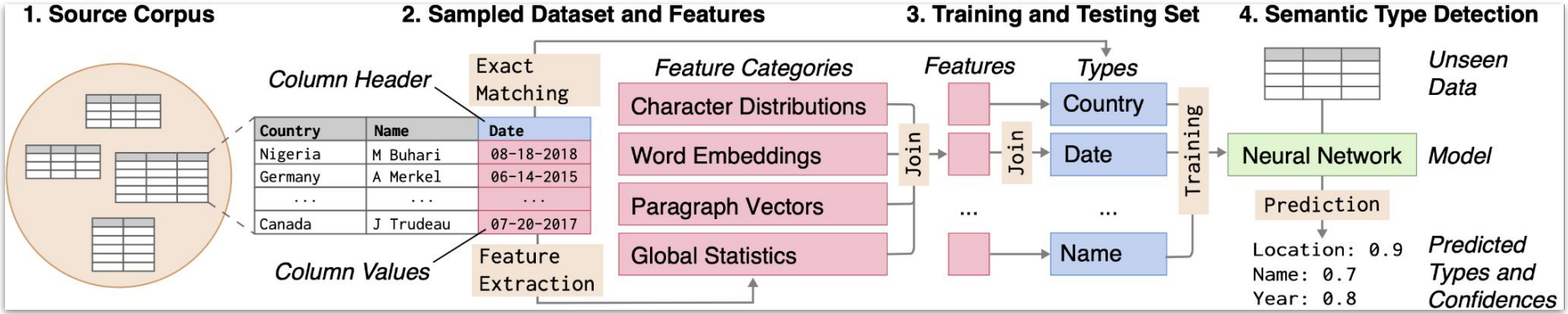
Systems have no idea!

- How to validate?
- How to clean?
- How to visualize?

Approaches to semantic type detection

Prior work Rule-based, KB lookups, basic ML, and pattern matching.

Sherlock Deep Learning to capture diversity and generalize.



Results and benchmarks

Our results from 2019…

Method	F_1 Score	Runtime (s)	Size (Mb)
Machine Learning			
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
Matching-based			
Dictionary	0.16	0.01 (±0.03)	0.5
Regular expression	0.04	0.01 (±0.03)	0.01
Crowdsourced Annotations			
Consensus	0.32 (±0.02)	33.74 (±0.86)	–

still challenges Large Table Models [1]!

Method	Column type C			Pairwise column relation $\mathcal{R}$		
	Acc.	F1-weighted	Cohen's kappa κ	Acc.	F1-weighted	Cohen's kappa κ
TABLE2VEC	.832	.820	.763	.822	.810	.772
TABERT	.908	.861	.834	.877	.870	.846
TURL	.914	.877	.876	.890	.889	.838
HNN	.916	.883	.869	.848	.843	.794
SHERLOCK	.922	.895	.863	.831	.818	.802
TCN-intra	.911	.881	.873	.893	.894	.869
TCN-$\mathcal{N}_v$	.939 (+3.1%)	.916 (+4.0%)	.897 (+2.8%)	.920 (+3.0%)	.920 (+2.9%)	.898 (+3.3%)
TCN-$\mathcal{N}_s$	.934 (+2.5%)	.908 (+3.1%)	.894 (+2.4%)	.908 (+1.7%)	.912 (+2.0%)	.881 (+1.4%)
TCN-$\mathcal{N}_p$	.923 (+1.3%)	.890 (+1.0%)	.880 (+0.8%)	.906 (+1.4%)	.904 (+1.1%)	.875 (+0.7%)
TCN	**.958 (+5.2%)**	**.938 (+6.5%)**	**.913 (+4.6%)**	**.934 (+4.6%)**	**.925 (+3.5%)**	**.905 (+4.1%)**

[1] Wang, Daheng, et al. "TCN: Table Convolutional Network for Web Table Interpretation." The *Web Conference, 2021*

Fastforward 2 years:

- Feedback on Sherlock: useful, but not all types, different data, not adaptive.
- Large Table Models arrived: TaBERT, TURL, TUTA, Tabbie, RPT…
- All Large Table Models are pretrained on tables from the Web!

GitTables: A Large-Scale Corpus of Relational Tables

Madelon Hulsebos
University of Amsterdam
Amsterdam
m.hulsebos@uva.nl

Çağatay Demiralp
Sigma Computing
San Francisco
cagatay@sigmacomputing.com

Paul Groth
University of Amsterdam
Amsterdam
p.t.groth@uva.nl

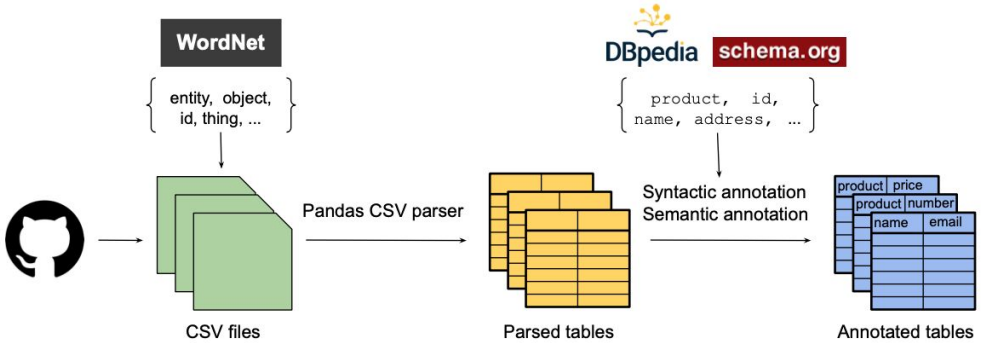


Figure 1: The pipeline for creating GitTables consists of 1) extracting CSV files from GitHub based on topics from WordNet, 2) parsing CSV files to tables, and 3) annotating tables with column semantics from DBpedia and Schema.org.

UNIVERSITY OF AMSTERDAM

sigma

Why are we not satisfied?

- WebTables [Cafarella et al., VLDB '08], WikiTables [Bhagavatula et al., KDD '13]:
- Web tables → Web applications. Data management with offline tables?
- Web tables ≈ DB tables?

Table from a Web page about US presidents.

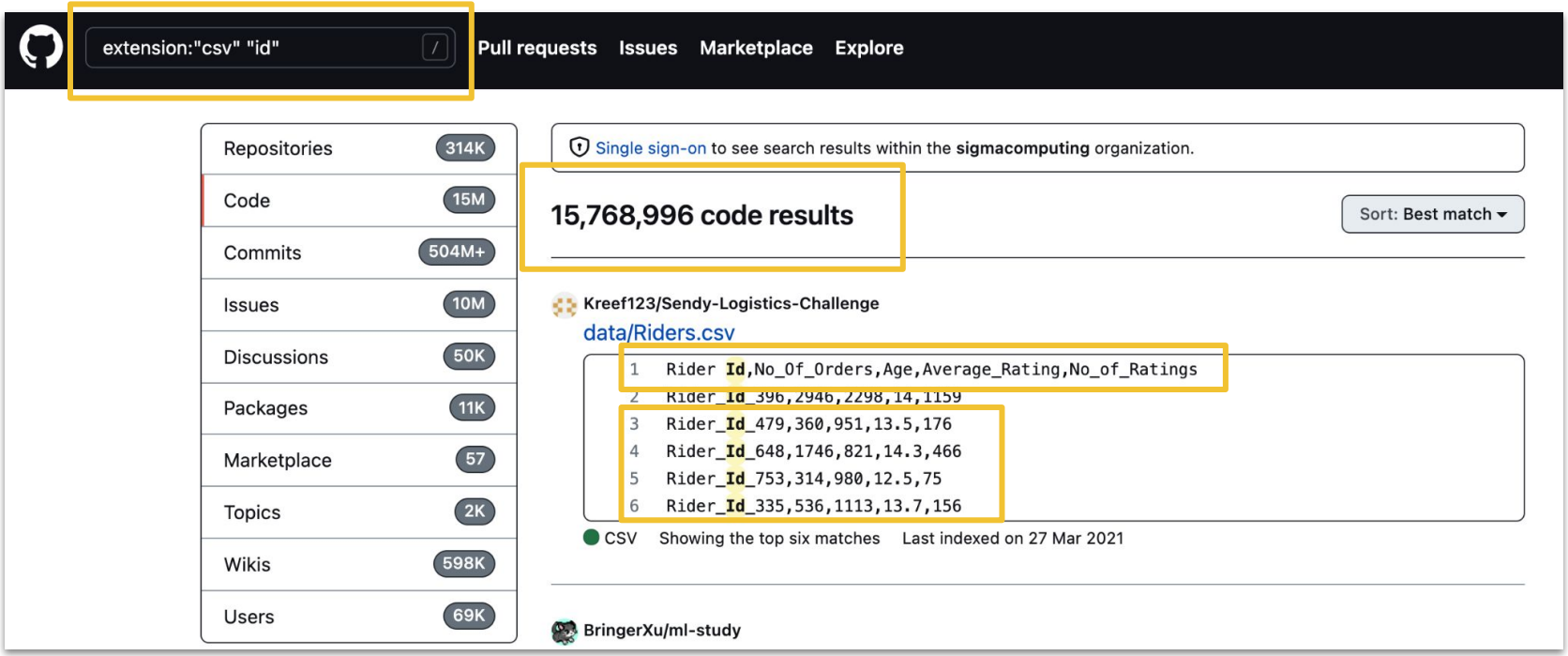
President	Party	Term as President	Vice-President
1. George Washington (1732-1799)	None, Federalist	1789-1797	John Adams
2. John Adams (1735-1826)	Federalist	1797-1801	Thomas Jefferson
3. Thomas Jefferson (1743-1826)	Democratic-Republican	1801-1809	Aaron Burr, George Clinton
4. James Madison (1751-1836)	Democratic-Republican	1809-1817	George Clinton, Elbridge Gerry
5. James Monroe (1758-1831)	Democratic-Republican	1817-1825	Daniel Tompkins
6. John Quincy Adams (1767-1848)	Democratic-Republican	1825-1829	John Calhoun
7. Andrew Jackson (1767-1845)	Democrat	1829-1837	John Calhoun, Martin van Buren
8. Martin van Buren (1782-1862)	Democrat	1837-1841	Richard Johnson
9. William H. Harrison (1773-1841)	Whig	1841	John Tyler
10. John Tyler (1790-1862)	Whig	1841-1845	
11. James K. Polk (1795-1849)	Democrat	1845-1849	George Dallas
12. Zachary Taylor (1784-1850)	Whig	1849-1850	Millard Fillmore
13. Millard Fillmore (1800-1874)	Whig	1850-1853	
14. Franklin Pierce (1804-1869)	Democrat	1853-1857	William King
15. James Buchanan (1791-1868)	Democrat	1857-1861	John Breckinridge

Table from google: "example database table" about crops.

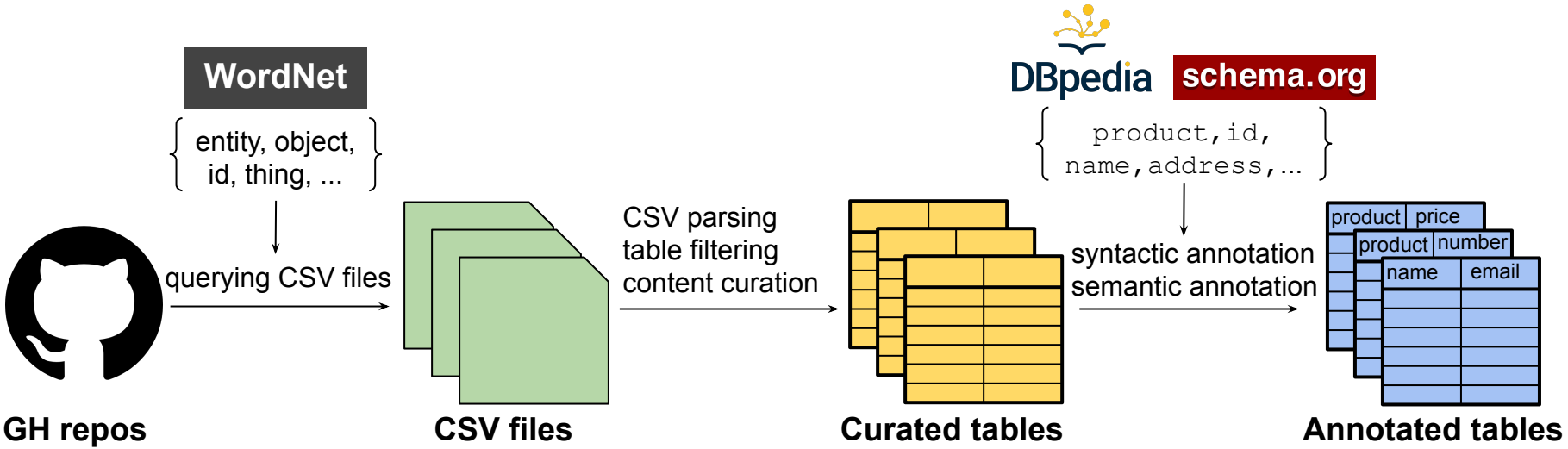
crop rotation : Tabelle

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 barl	sandy loam, loe	635		cultivation, weed
14	57	365	105	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 barle			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	

Can we use CSVs from GitHub?



How we built GitTables

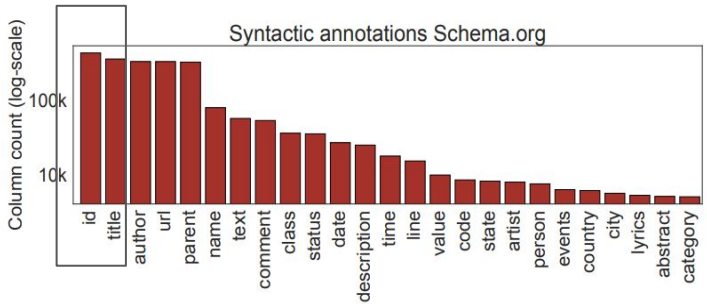


Corpus statistics

GitTables complements related corpora like WebTables, VizNet, etc.

Name	Table source	# relational tables	Avg # rows	Avg # cols
WDC WebTables [26]	HTML pages	90M	11	4
Dresden Web Table Corpus [17]	HTML pages	59M	17	6
WikiTables [3]	Wikipedia tables	2M	15	6
Open Data Portal Watch [29]	CSVs from Open Data portals	107K	365	14
VizNet [20]	WebTables, Plotly, i.a.	31M	17	3
GitTables	**CSVs from GitHub**	**1M**	**142**	**12**

Data type	GitTables	WDC WebTables
Numeric	57.9%	51.4%
String	41.6%	47.4%
Other	0.5%	1.2%



most common type in WebTables = “name”, “id” > #20

Use-cases: column types and schema completion

Semantic column type detection

id ↓ age ↓ rating ↓

Rider Id	No_Of_Orders	Age	Average_Rating	No_of_Ratings
Rider_Id_396	2946	2298	14	1159
Rider_Id_479	360	951	13.5	176
Rider_Id_648	1746	821	14.3	466
Rider_Id_753	314	980	12.5	75
Rider_Id_335	536	1113	13.7	156
Rider_Id_720	2608	1798	13.2	504

Train corpus	Evaluation corpus	F1-score (macro)
GitTables	GitTables	0.86
VizNet	VizNet	0.77
VizNet	GitTables	0.66

Schema completion

[id, company, ?]

↓

[id, company, **order id**, **value**]

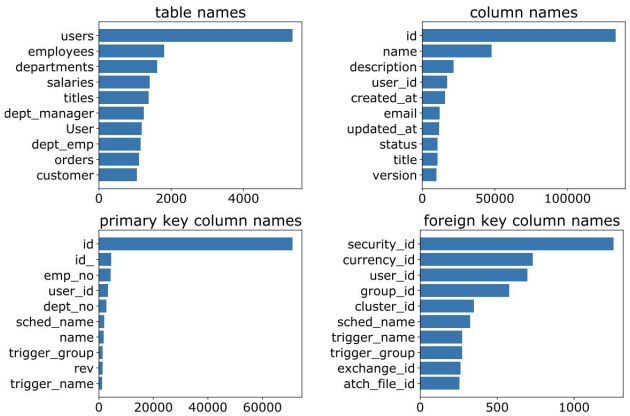
Header prefix	Suggested completion	Similarity
emp_no, birth_date	→ `Title, TitleOfCourtesy, Address, HireDate, City`	0.44
orderNumber, orderDate	→ `ORDER_TRACKING_NUMBER, ORDER_TOTAL, ACCOUNT_ID`	0.50
WorkOrderID, ProductID	→ `productType, inventoryId, articleId`	0.53

Data… Relations? More resources for learning!

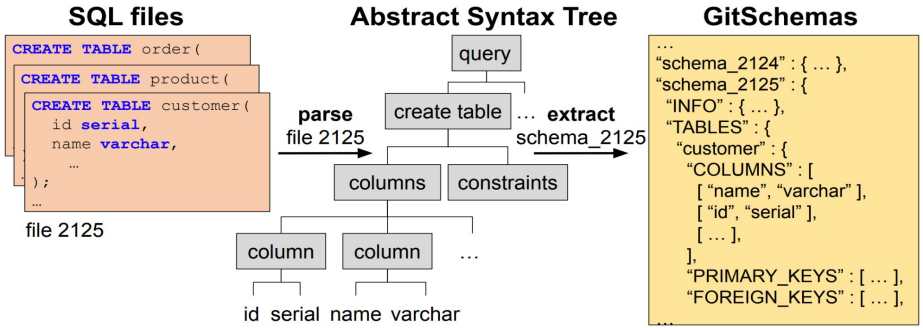
GitSchemas: A Dataset for Automating Relational Data Preparation Tasks

Till Döhmen, *Fraunhofer FIT*, till.doehmen@fit.fraunhofer.de
Madelon Hulsebos, *University of Amsterdam*, m.hulsebos@uva.nl
Christian Beecks, *Fraunhofer FIT & University of Hagen*, christian.beecks@fit.fraunhofer.de
Sebastian Schelter, *University of Amsterdam*, s.schelter@uva.nl

Distribution of schema entities



Extraction process

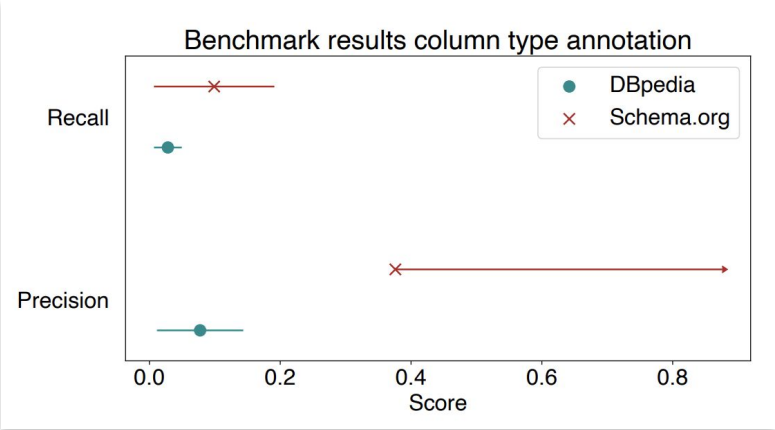


Use-case: data augmentation → AutoML

Data augmentation method	AutoML accuracy (R^2)
No Joins	0.72
Joins by Cupid schema-matching	0.69
Joins by lookup in GitSchemas	**0.85**

Opportunities

- How many tables can we get? GitHub has **92M+** CSVs. **10M+** aim for GitTables.
- Can we annotate GitTables with enterprise ontologies? Or infer an ontology?
- Can we build synthetic databases using GitSchemas and GitTables?
- Can we enhance KBs with GitTables? → checkout SemTab challenge '22!



AdaTyper: Adaptive Semantic Column Type Detection

Madelon Hulsebos*
Sigma Computing
San Francisco, USA
madelon@sigmacomputing.com

Sneha Gathani†
Sigma Computing
San Francisco, USA
sneha@sigmacomputing.com

James Gale
Sigma Computing
San Francisco, USA
jlg@sigmacomputing.com

Isil Dillig
University of Texas
Austin, USA
isil@cs.utexas.edu

Paul Groth
University of Amsterdam
Amsterdam, Netherlands
p.t.groth@uva.nl

Çağatay Demiralp
Sigma Computing
San Francisco, USA
cagatay@sigmacomputing.com

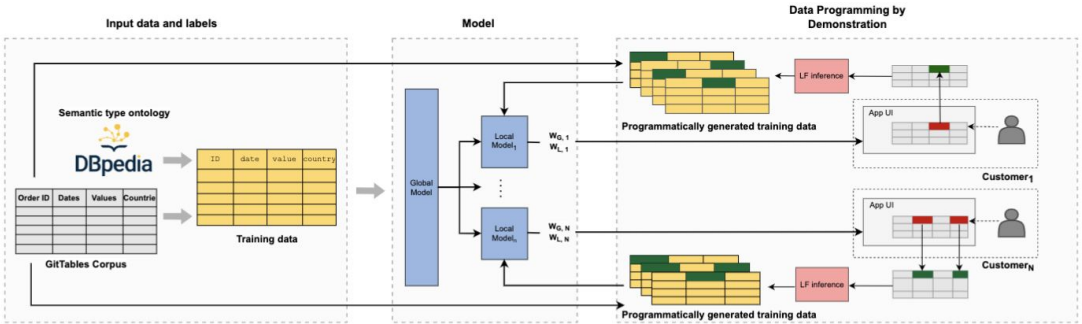


Figure 1: ADATYPER: tables and semantic types are used to pretrain a global model. The local model adapts towards the user's context through data programming by demonstration: based on the user's feedback, labeling functions are inferred and used to generate new training data and function as weak predictors in the local model. The weight of the local model increases over time.

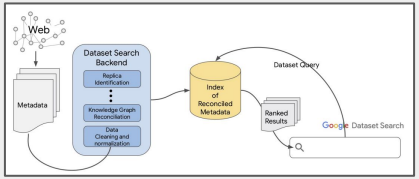
Work in progress!

Deployment of Large Table Models in practice is behind

A gap between **research performance** and **deployment in practice**.

Type detection **Google Dataset Search**

Indexing manual metadata



Type detection **Trifacta**

Email addresses (pattern matching)

- String@String.aaa
- String@String.aaaa

Type detection **Google Data Studio**

Date & Time (pattern matching)

- YYYY/MM/DD-HH:MM:SS
- YYYY-MM-DD [HH:MM:SS[]]
- …

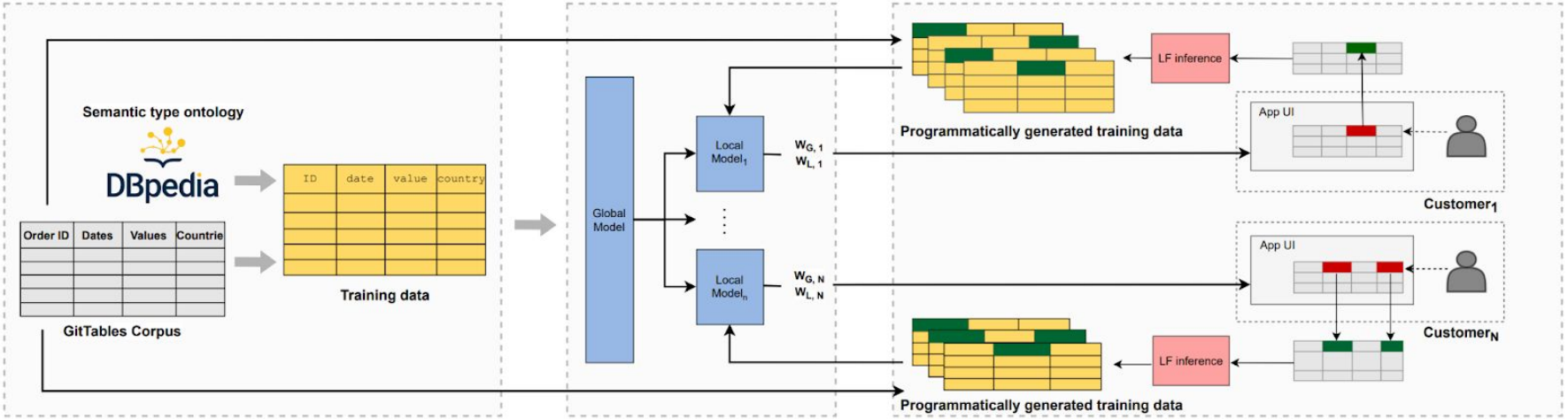
Geo types (lookups)

Name	Canonical Name	Parent ID	Country Code	Target Type
Kabul	Kabul,Kabul,Afghanistan	9075393	AF	City
Luanda	Luanda,Luanda Province,Angola	9070431	AO	City
The Valley	The Valley,Anguilla	2660	AI	City
Philipsburg	Philipsburg,Sint Maarten	2534	SX	City
Willemstad	Willemstad,Curacao	2531	CW	City
Abu Dhabi	Abu Dhabi,Abu Dhabi,United Arab Emirates	9041082	AE	City

How can we adapt to the user's data context?

Scenario unknown type → new type

Approach generate training data from the (type) example → adapt model



Thoughts one/few-shot learning?

Many opportunities, but we need more arms!

We should discuss and explore:

- What are the **opportunities** of Large Table Models in data management?
- **Boundaries** of **natural language**-based Large Table Models in practice?
- How to **encode tabular data** (& other modalities?); values, structure?
- What **pretraining**, **fine-tuning** and **prompting** strategies work, for which **tasks**?

Hopefully, a platform for this discussion will arise soon :-)

Summary

- Tables are a first-class modality, just like images!
- We have initial (adaptive) table representations.
- With GitTables and GitSchemas we have large representative resources.
- Many opportunities of Large Table Models in data management.
- The DB and ML community should join forces to come further!

Reach out:  m.hulsebos@uva.nl  @MadelonHulsebos