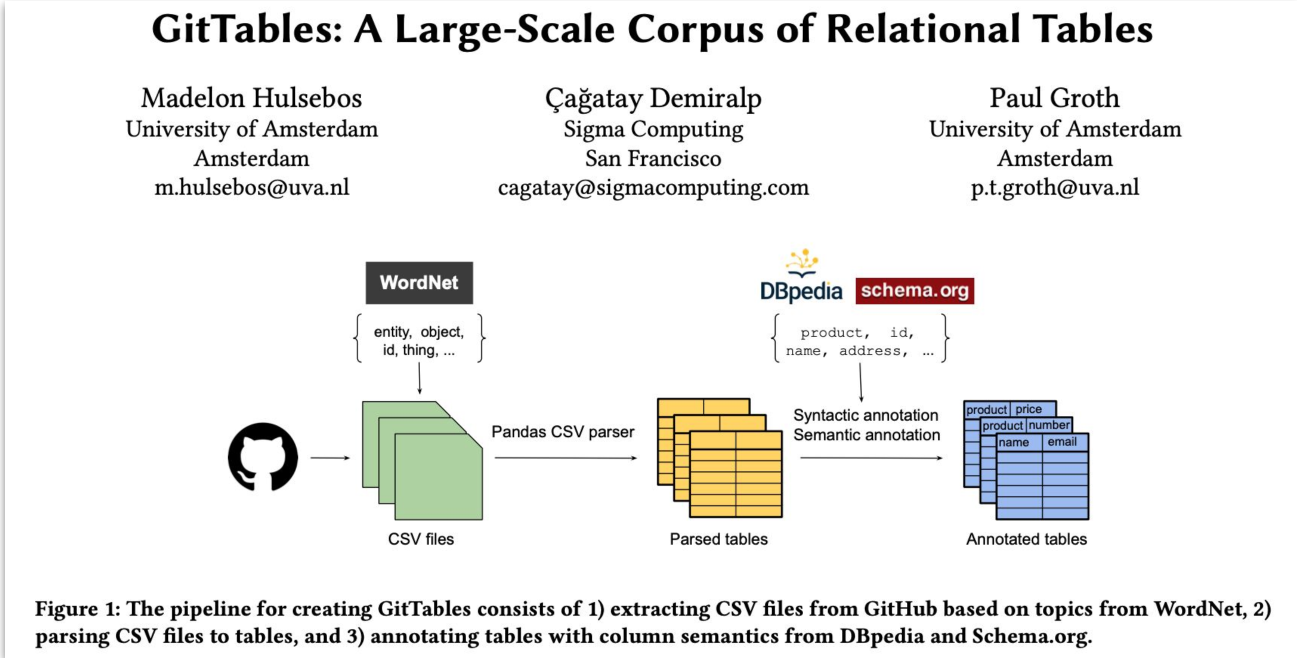
GitTables

A large corpus of relational tables

Madelon Hulsebos
University of Amsterdam
Sigma Computing

CWI
4 February 2022



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.

In this talk: **why**, **how**, **what**?

Why do we need table corpora?

- From understanding images, natural language, and code → **understanding tables**

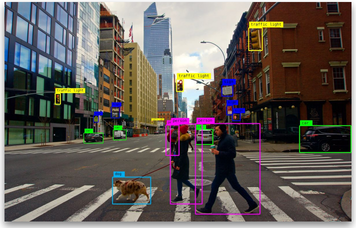
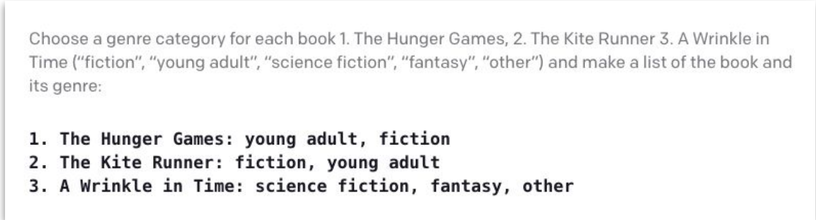


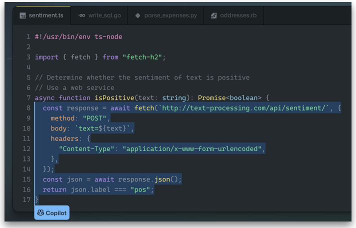
Figure 1: Image object detection



Choose a genre category for each book 1. The Hunger Games, 2. The Kite Runner 3. A Wrinkle in Time ("fiction", "young adult", "science fiction", "fantasy", "other") and make a list of the book and its genre:

```
1. The Hunger Games: young adult, fiction
2. The Kite Runner: fiction, young adult
3. A Wrinkle in Time: science fiction, fantasy, other
```

Figure 2: Language understanding with GPT-3



```
#!/usr/bin/env ts-node

import { fetch } from "fetch-h2";

// Determine whether the sentiment of text is positive
// Use a web service
async function isPositive(text: string): Promise<boolean> {
  const response = await fetch(`http://text-processing.com/api/sentiment/`, {
    method: "POST",
    body: `text=${text}`,
    headers: {
      "Content-Type": "application/x-www-form-urlencoded",
    },
  });
  const json = await response.json();
  return json.label === "pos";
}
```

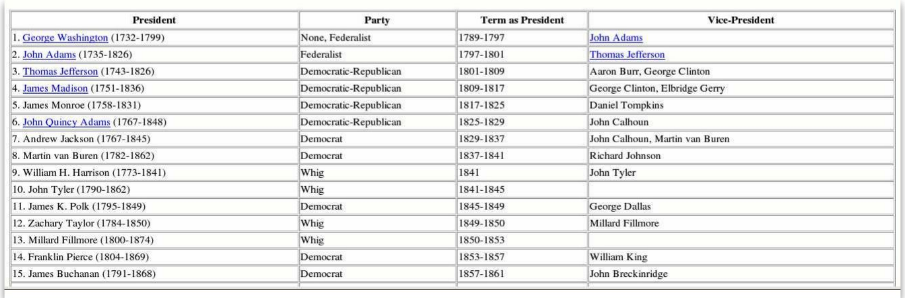
Figure 3: Code completion

- Applications:
  - Data search, integration, validation.
  - Query optimization, validation and recommendation.
- **Web**Tables [Cafarella et al., VLDB '08], **Wiki**Tables [Bhagavatula et al., KDD '13]:
  - Large corpora, generally relevant knowledge.

Why are we not satisfied?

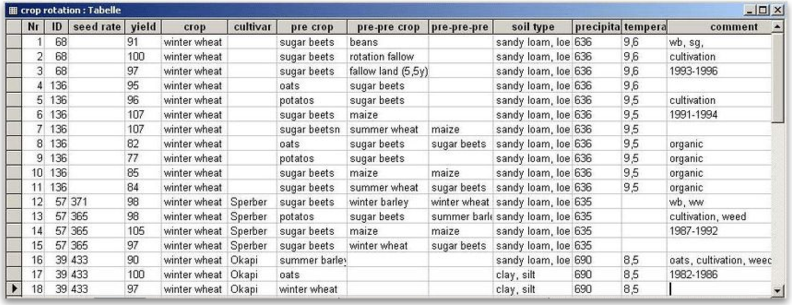
- WebTables → Web applications. Data management with offline tables?
- Web tables ≈ DB tables?
  - ○ Feedback on Sherlock trained on Web tables [Hulsebos et al., KDD '19]: different types, different data.
  - ○ Low transferability: different semantics and data characteristics [Langenecker et al., BTW '21].

Table 1: Table from a Web page about US presidents [Cafarella et al., VLDB '08].



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 2: Table with crop data, first result "example database table".



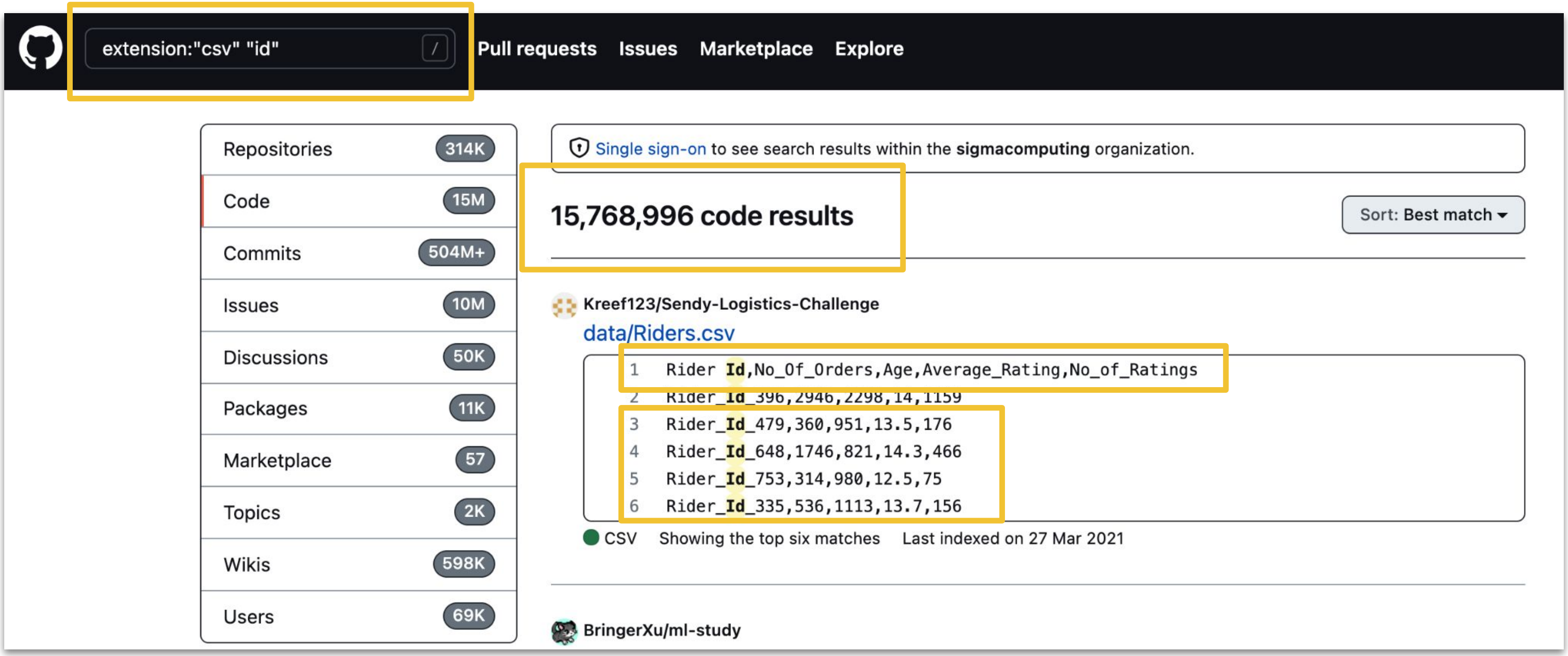
crop rotation : Tabelle

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 barley			sandy loam, loe	690	8,5	oats, cultivation, weed
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	

What do we need from a table corpus?

- Database-like table content and structure (semantics, data types, size).
- Large-scale to facilitate table representation models.
- Broad coverage to generalize to a diversity of domains.
- Table semantics (e.g. column types).

Can we use CSVs from GitHub?



Repositories	314K
Code	15M
Commits	504M+
Issues	10M
Discussions	50K
Packages	11K
Marketplace	57
Topics	2K
Wikis	598K
Users	69K

Single sign-on to see search results within the **sigmacomputing** organization.

15,768,996 code results

Kreef123/Sendy-Logistics-Challenge

data/Riders.csv

```
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
```

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

BringerXu/ml-study

Figure 4: Result from GitHub code search when querying for CSV files containing “id”.

How we built GitTables.

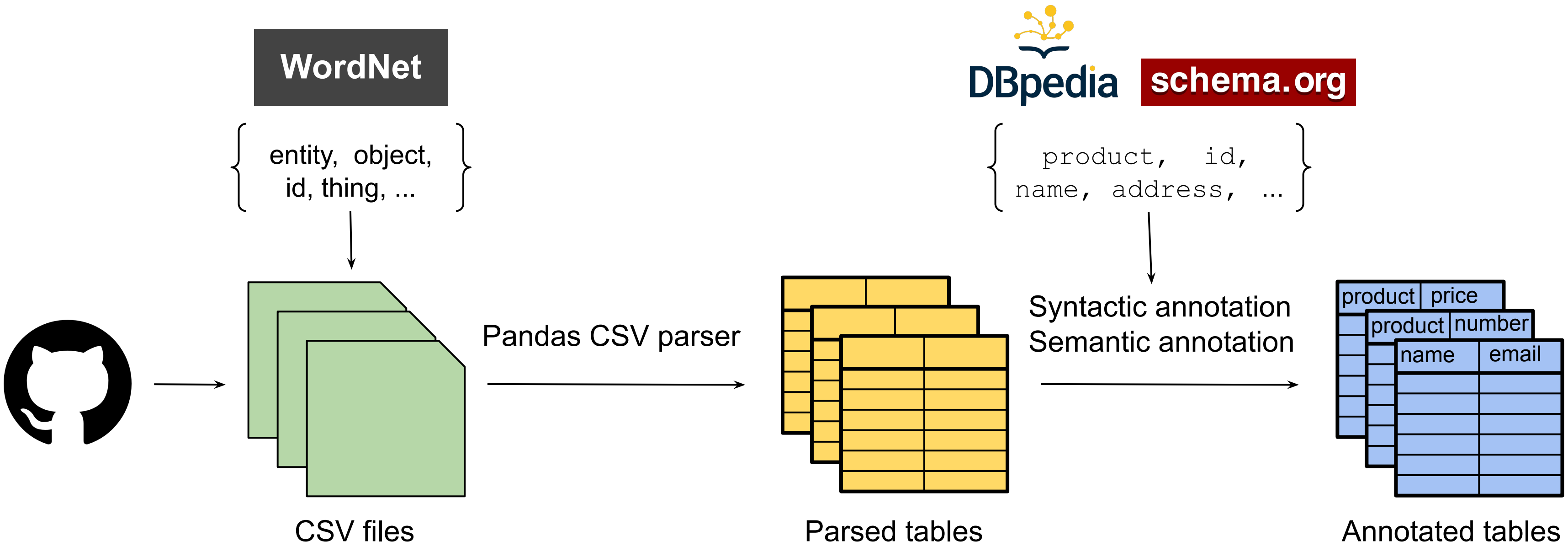


Figure 5: High-level pipeline for constructing GitTables from CSV extraction, to table curation and column annotation.

CSV extraction: get as many CSVs as possible.

- Query CSV files from GitHub by WordNet topic (e.g."id","population").
- Segment query using initial query size and file size:

```
q="id" extension:csv size:50..100
```

- Filter CSVs on permissive repo license (+/- 25%).

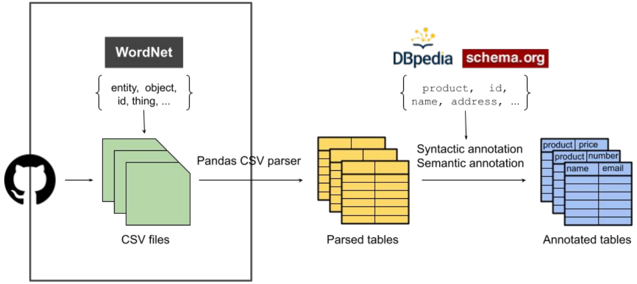


Table curation: collect quality tables to publish safely.

- Parse CSVs to tables assuming header is on first row.
- Filter out tables with social media data.
- Substitute potential Personal Identifiable Information (PII) using Faker.

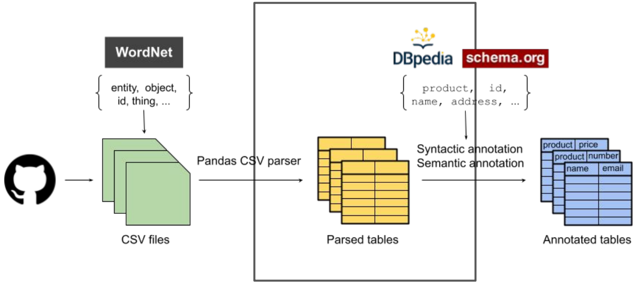


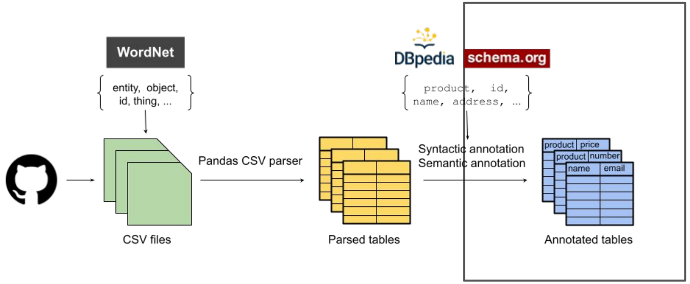
Table 3: Percentage of detected PII columns.

Semantic type	Percentage columns	Faker class
name	2.202%	faker.name
address	0.163%	faker.address
person	0.068%	faker.name
email	0.042%	faker.email
birth date	0.017%	faker.date
home location	0.008%	faker.city
birth place	0.003%	faker.postcode
postal code	0.003%	faker.city

Column annotation: table semantics for e.g. data integration.

- Types from DBpedia, Schema.org → KB lookups/augmentation.
- Types come with hierarchical relations, data types, etc.
- Basic syntactic and semantic matching: column name ↔ type.

  Syntactic matching (“Email” → `email`) = high quality due to human source of data.



Corpus statistics

Published datasets:

Table 4: Table and annotation statistics of the published datasets (tables from 10 query topics).

	# tables	**# syntactic annotated tables (dbpedia, schema)**	**# semantic types (dbpedia, schema)**
GitTables	1.7M	1.0M, 1.5M	1218, 924
GitTables: semantic type detection benchmark dataset	1101	1101	121, 58

Ongoing extraction process: currently ~7M.

Corpus analysis

Table structure and content:

- Avg: 25 cols, 209 rows (WebTables: 3 cols, 12 rows).
- 9% small tables → WIP: further curation.
- Data shift VizNet (mostly WebTables) vs GitTables.

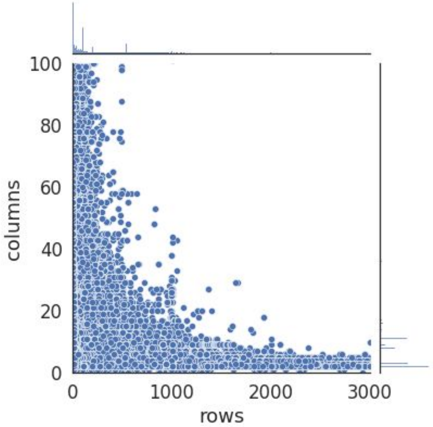


Figure 6: Table size distributions.

Topical coverage:

- 40% overlap top-10 DBpedia types WebTables/GitTables.
- Most common type in GitTables: `id`.
- Most common type in WebTables: `name` (`id` > #20).

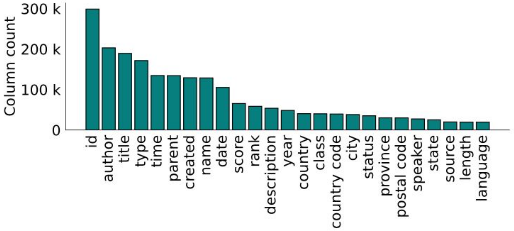
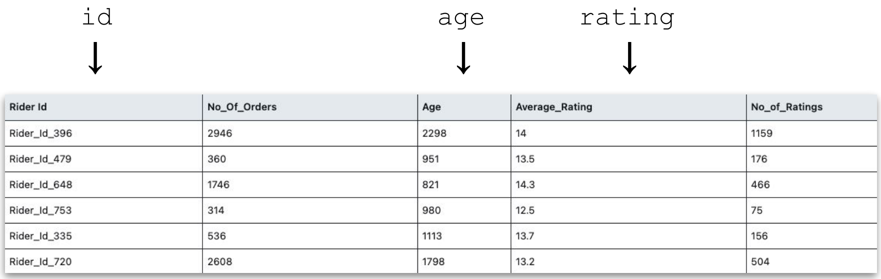


Figure 7: Distribution of the number of annotations per semantic type (DBpedia).

Use-case: semantic column type detection

Task:



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

Approach:

- 10K columns, 5 types, VizNet & GitTables.
- Extract 1K+ features from columns.
- Train RF classifier with default settings.

Results:

Table 5: Model performance when trained on source corpus and evaluated on target corpus. GitTables is complementary and difficult.

Source corpus	Target corpus	F1-score
VizNet	VizNet	0.90
GitTables	GitTables	0.82
VizNet	GitTables	0.62

Use-case: header autocompletion

Task:

[id, company, ?]

↓

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

Approach:

- 16K unique headers from GitTables.
- USE representations of column names.
- Closest header distance based on prefix.

Results:

Table 6: Suggested headers based on initial set of attributes. This simple method informed by GitTables makes sensible suggestions.

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

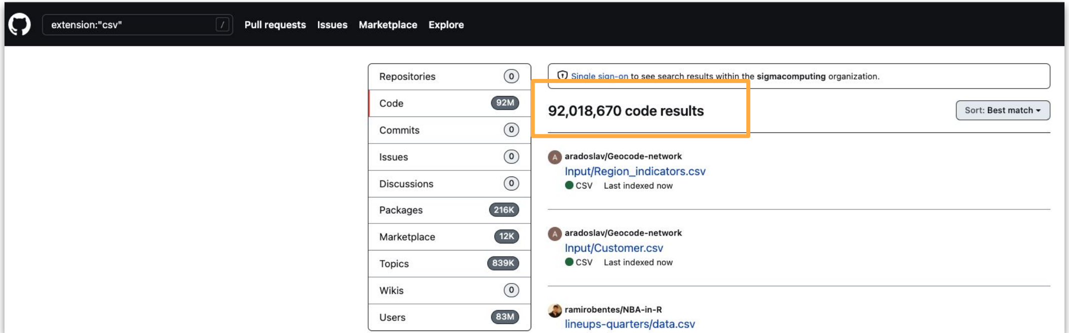
Early impact of GitTables

Between June ‘21 and January ‘22:

- Already 1.2TB over 532 downloads.
- Benchmark dataset featured in SemTab, challenge for “Table to KG matching”.
- Inspired GitDBSchemas [Döhmen et al, ‘22]: table schemas from SQL files.
- Ongoing work on benchmarking data discovery methods [UMich, TU Delft].
- We use GitTables to train table models, for e.g. data search.

Opportunities

- How many tables can we get? GitHub has **92M+** CSVs. **10M+** aim for GitTables.



Repositories	0
Code	92M
Commits	0
Issues	0
Discussions	0
Packages	216K
Marketplace	12K
Topics	839K
Wikis	0
Users	83M

Single sign-on to see search results within the **sigmacomputing** organization.

92,018,670 code results

aradoslav/Geocode-network
Input/Region_indicators.csv
CSV Last indexed now

aradoslav/Geocode-network
Input/Customer.csv
CSV Last indexed now

ramirobentes/NBA-in-R
lineups-quarters/data.csv

Figure 8: Result from GitHub code search when querying for CSV files.

- Can we annotate GitTables with enterprise ontologies? Or infer an ontology?
- Can we enhance KBs [Weikum, VLDB '21] with GitTables?
- What other use-cases can benefit from this corpus?

Resources

- Paper: https://arxiv.org/abs/2106.07258
- Website: https://gittables.github.io
- GitTables dataset: https://zenodo.org/record/4943312
- GitTables type detection benchmark: https://zenodo.org/record/5706316

Summary

- GitTables is a large-scale repository of relational tables.
- GitTables better resembles typical database tables.
- GitTables is effective for tasks like header autocompletion.
- GitHub is a rich data source for the community.
- Opportunities: benchmarks, enhancing KBs, table models for e.g. data integration.

Reach out: m.hulsebos@uva.nl