Practical Linked Data Access via SPARQL: The Case of Wikidata

Adrian Bielefeldt Julius Gonsior Markus Krötzsch

Knowledge-Based Systems
TU Dresden

Also reporting on joint work with
Stas Malyshev (Wikimedia) and Larry Gonzalez (TU Dresden)

Research supported by the Wikimedia Foundation

For the eponymous LDOW 2018 paper, see https://iccl.inf.tu-dresden.de/web/Inproceedings3196/en

+

= ?

Wikidata, the knowledge graph of Wikipedia,
uses SPARQL as its main query API

- Who is using this?
- What are those SPARQL queries like?
- What can we learn from them?

Wait! – Wikidata uses RDF?!

Louis Néel (Q155781)

French physicist

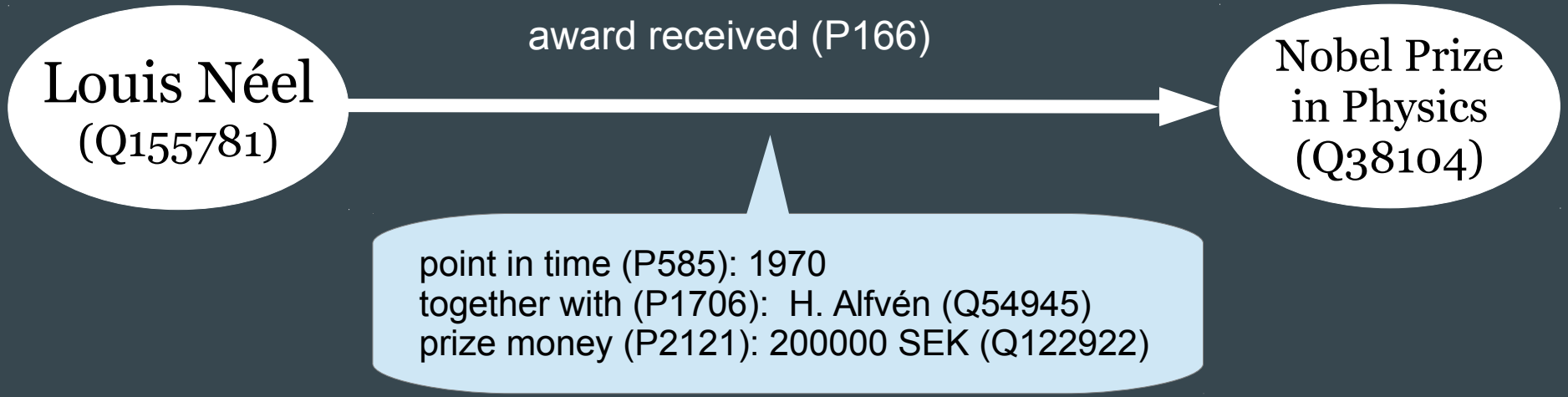
Louis Neel | Louis Eugène Felix Néel

award received	Nobel Prize in Physics	edit
	point in time	1970
	together with	Hannes Alfvén
	prize money	200,000 Swedish krona

2 references

copy

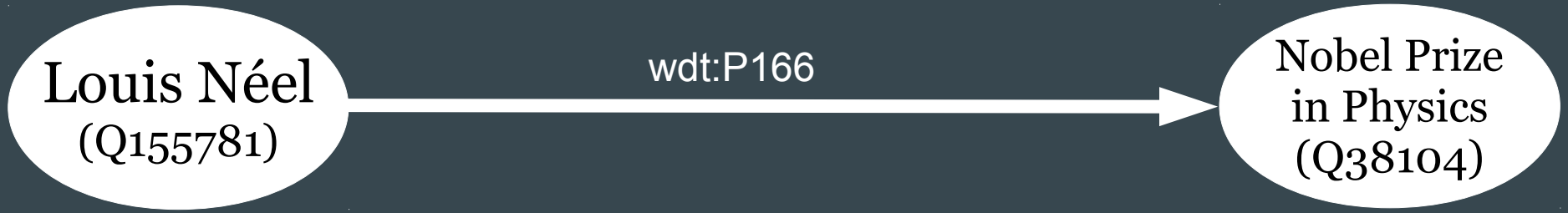
Wait! – Wikidata uses RDF?!



How does Wikidata's rich graph model relate to RDF?

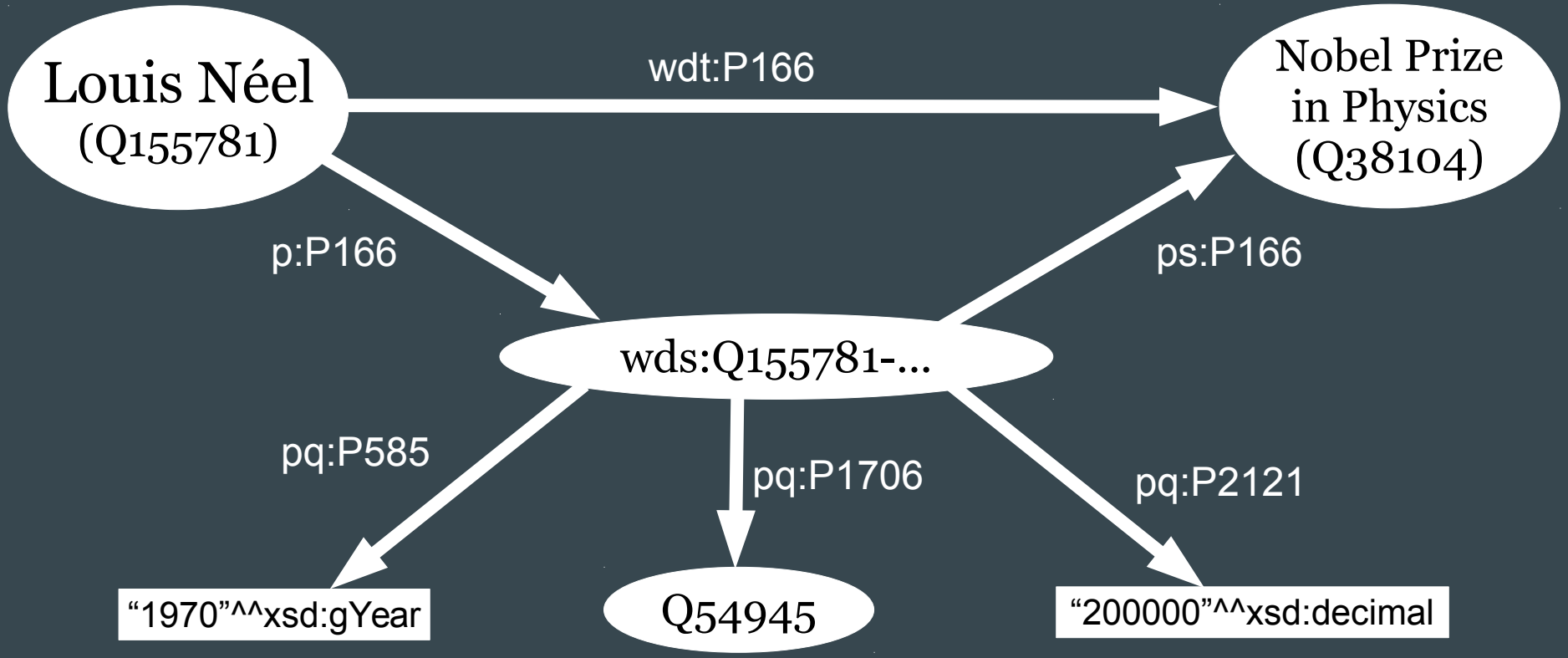
Wait! – Wikidata uses RDF?!

Official RDF version follows Erxleben et al. [ISWC 2014]:



Wait! – Wikidata uses RDF?!

Official RDF version follows Erxleben et al. [ISWC 2014]:



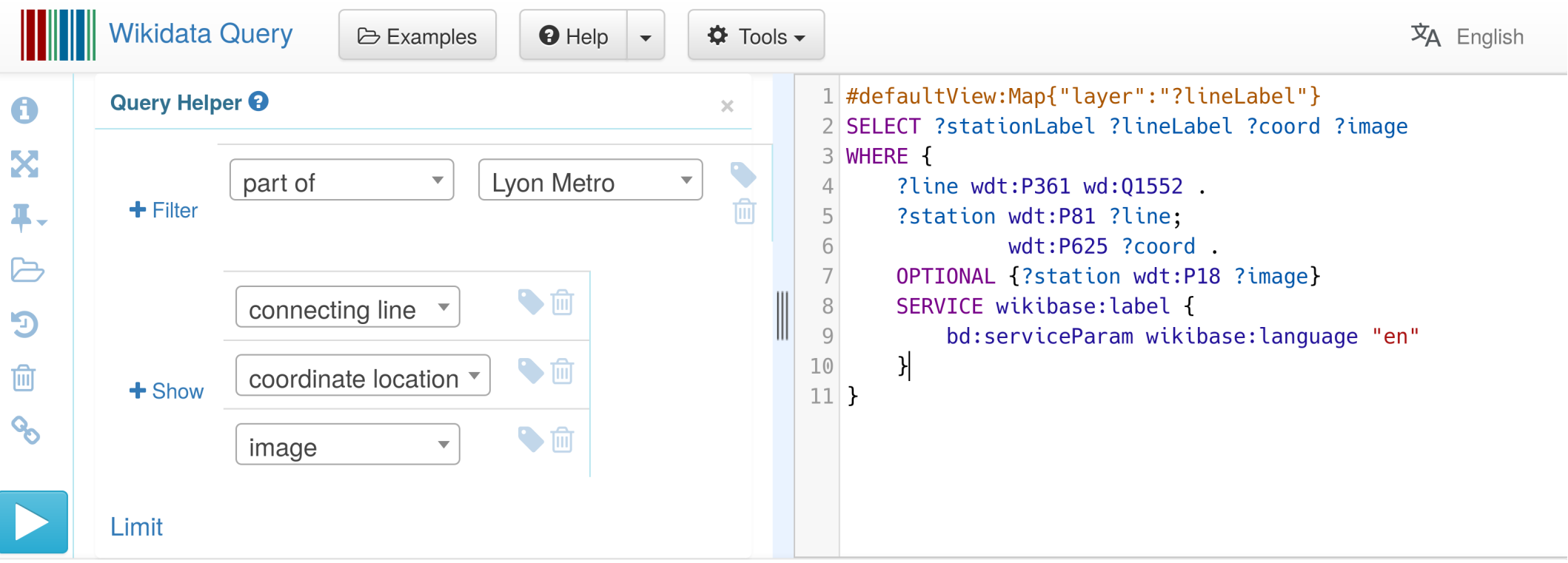
RDF for Wikidata

- Wikidata offers **all of its content** in RDF
  - Linked data live exports
    (Example: https://www.wikidata.org/wiki/Special:EntityData/Q42.nt)
  - Weekly dumps
    (See https://dumps.wikimedia.org/wikidatawiki/entities/)
- Currently **4.9B triples** (as of April 2018)
  - >415M Wikidata Statements
  - 4.5K Wikidata properties → >48K RDF properties
  - >1.5B labels/descriptions/aliases
  - >63M links to Wikipedia and friends

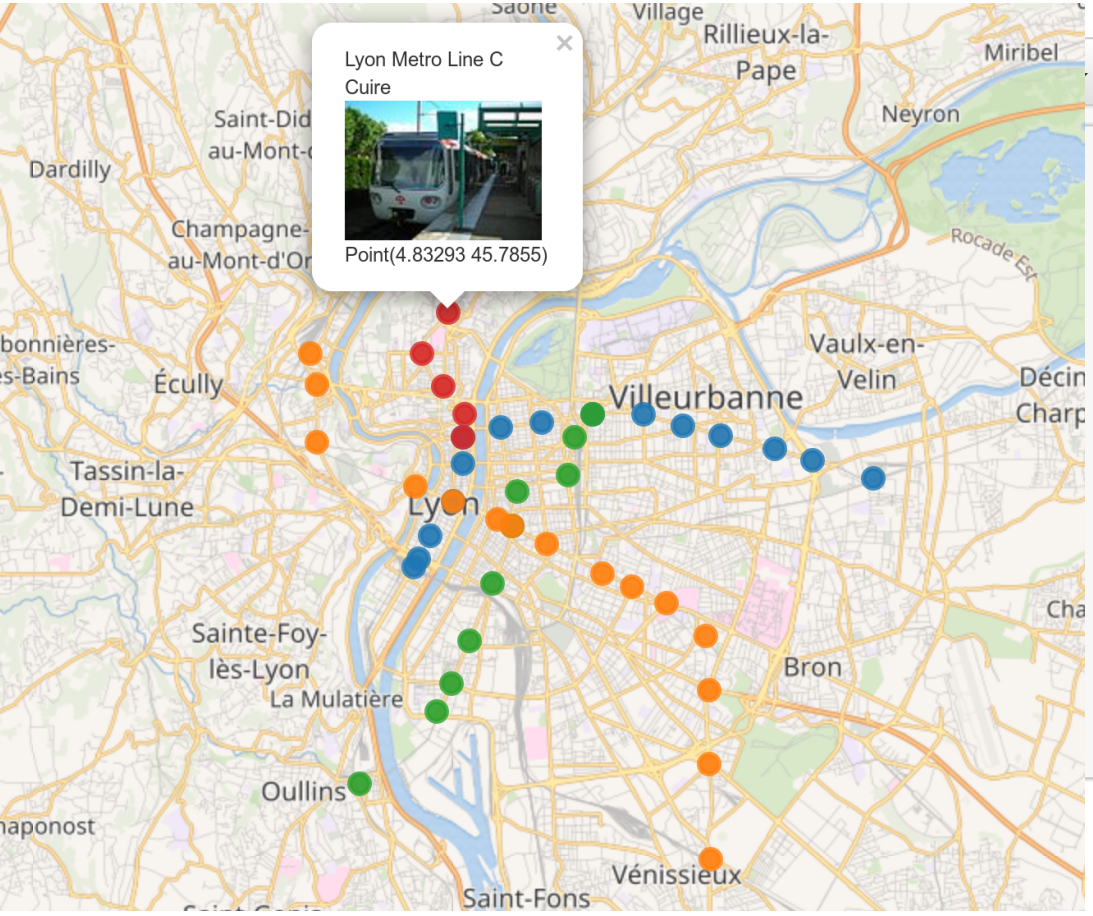
Wikidata SPARQL Query Service

- Official query service since mid 2015
  - User interface at https://query.wikidata.org/
- All the data (4.9B triples), live (latency<60s)
- No limits (well, almost):
  - 60sec timeout
  - No limit on result size (!)
  - No limit on query numbers per IP
  - Clients might be paused after too many parallel requests

A simple SPARQL query



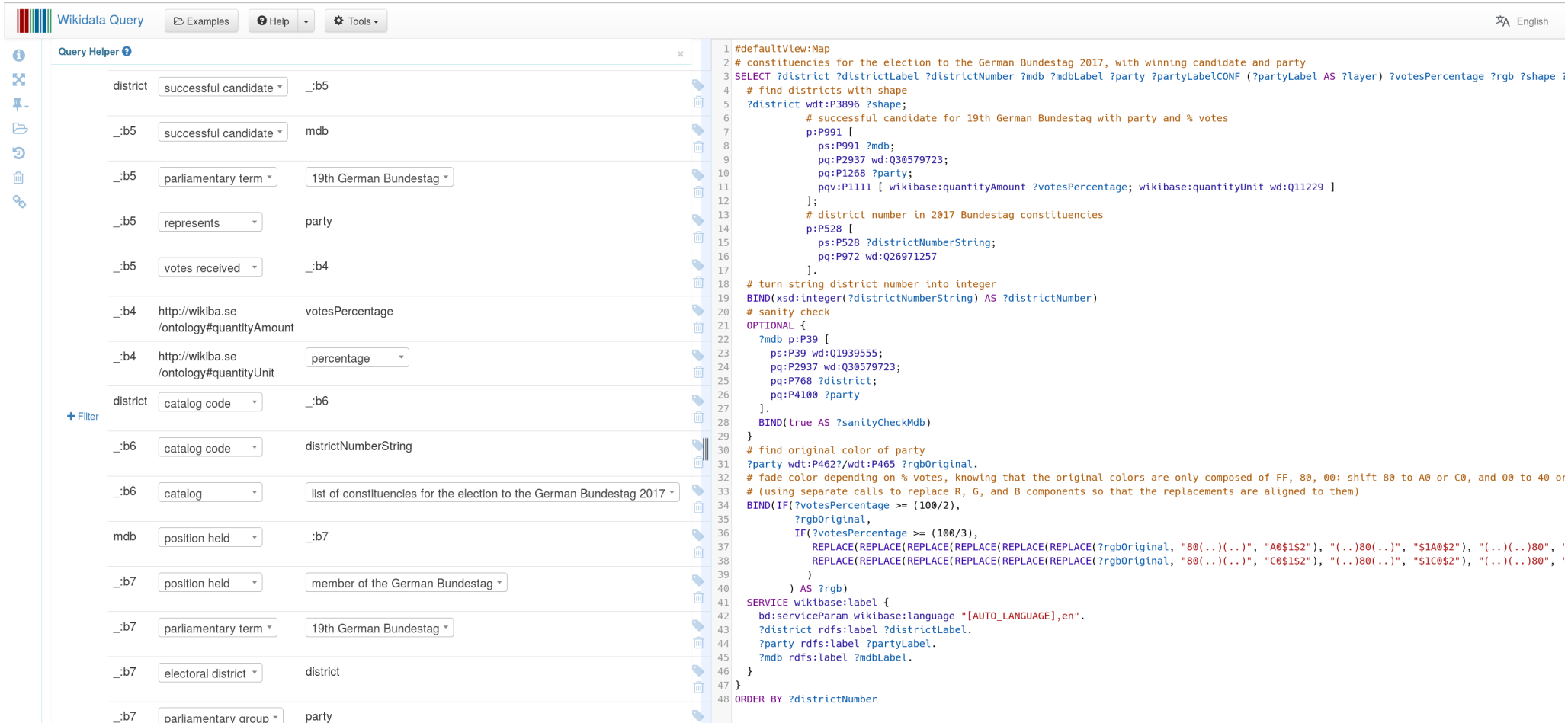
A simple SPARQL query



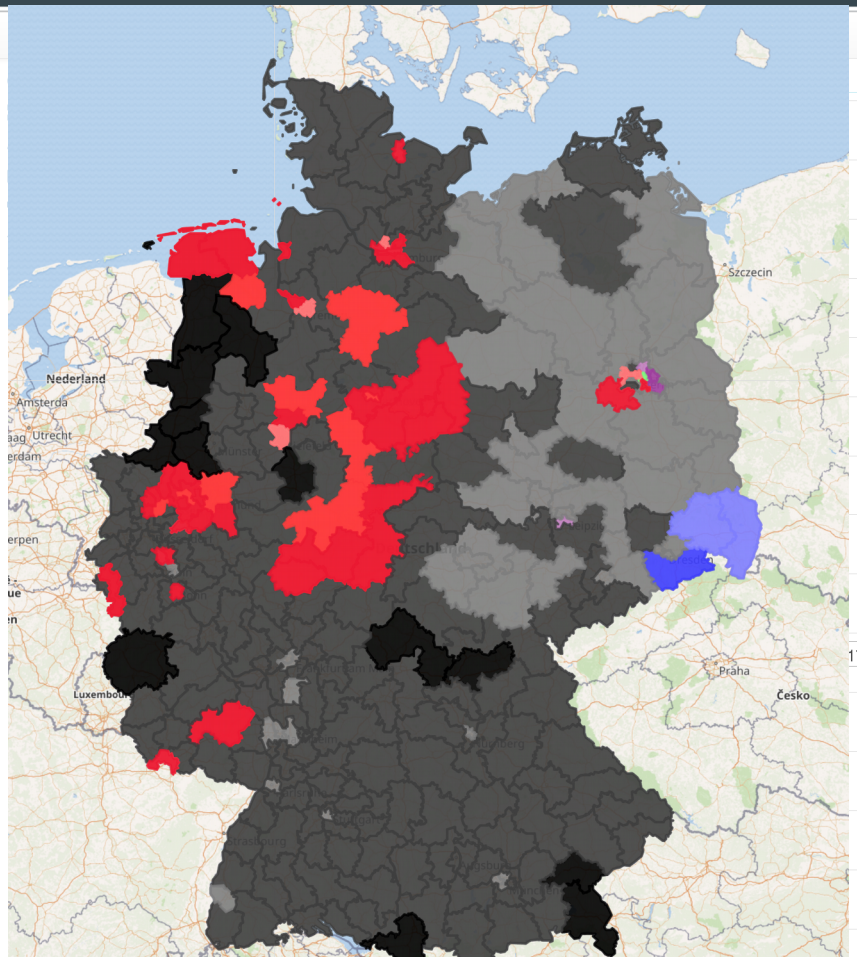
English

```
#defaultView:Map{"layer":"?lineLabel"}
SELECT ?stationLabel ?lineLabel ?coord ?image
WHERE {
    ?line wdt:P361 wd:Q1552 .
    ?station wdt:P81 ?line;
             wdt:P625 ?coord .
    OPTIONAL {?station wdt:P18 ?image}
    SERVICE wikibase:label {
        bd:serviceParam wikibase:language "en"
    }
}
```

A not-so-simple SPARQL query



A not-so-simple SPARQL query



English

```
#defaultView:Map
# constituencies for the election to the German Bundestag 2017, with winning candidate and party
SELECT ?district ?districtLabel ?districtNumber ?mdb ?mdbLabel ?party ?partyLabelCONF (?partyLabel AS ?layer) ?votesPercentage ?rgb ?shape ?
  # find districts with shape
  ?district wdt:P3896 ?shape;
            # successful candidate for 19th German Bundestag with party and % votes
            p:P991 [
              ps:P991 ?mdb;
              pq:P2937 wd:Q30579723;
              pq:P1268 ?party;
              pqv:P1111 [ wikibase:quantityAmount ?votesPercentage; wikibase:quantityUnit wd:Q11229 ]
            ];
            # district number in 2017 Bundestag constituencies
            p:P528 [
              ps:P528 ?districtNumberString;
              pq:P972 wd:Q26971257
            ].
  # turn string district number into integer
  BIND(xsd:integer(?districtNumberString) AS ?districtNumber)
  # sanity check
  OPTIONAL {
    ?mdb p:P39 [
      ps:P39 wd:Q1939555;
      pq:P2937 wd:Q30579723;
      pq:P768 ?district;
      pq:P4100 ?party
    ].
    BIND(true AS ?sanityCheckMdb)
  }
  # find original color of party
  ?party wdt:P462?/wdt:P465 ?rgbOriginal.
  # fade color depending on % votes, knowing that the original colors are only composed of FF, 80, 00: shift 80 to A0 or C0, and 00 to 40 or
  # (using separate calls to replace R, G, and B components so that the replacements are aligned to them)
  BIND(IF(?votesPercentage >= (100/2),
          ?rgbOriginal,
          IF(?votesPercentage >= (100/3),
             REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(?rgbOriginal, "80(..)(..)", "A0$1$2"), "(..)80(..)", "$1A0$2"), "(..)(..)80", "
             REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(REPLACE(?rgbOriginal, "80(..)(..)", "C0$1$2"), "(..)80(..)", "$1C0$2"), "(..)(..)80", "
             )
          ) AS ?rgb)
  SERVICE wikibase:label {
    bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en".
    ?district rdfs:label ?districtLabel.
    ?party rdfs:label ?partyLabel.
    ?mdb rdfs:label ?mdbLabel.
  }
}
ORDER BY ?districtNumber
```

Some metrics

- Running on BlazeGraph database engine
  - 3 servers (+3 as backup) Intel Xeon E5-2620 8 core/128G mem/800G SSD
  - Standard caching (Varnish) and load balancing (LVS)
  - Some custom tools, extension and tunings

  All available online: https://github.com/wikimedia/wikidata-query-rdf

Some metrics

- Running on BlazeGraph database engine
  - 3 servers (+3 as backup) Intel Xeon E5-2620 8 core/128G mem/800G SSD
  - Standard caching (Varnish) and load balancing (LVS)
  - Some custom tools, extension and tunings

    All available online: https://github.com/wikimedia/wikidata-query-rdf
- Serving >100M requests/month (3.8M/day)
  - 50% of queries answered in <40ms (95% in <440ms; 99% in <40s)
  - Less than 0.05% of queries time out
  - Service has never been down so far

Analysing SPARQL logs: The Bot Problem

Analysing SPARQL logs: The Bot Problem

- Query traffic is **ruled** by a few bots

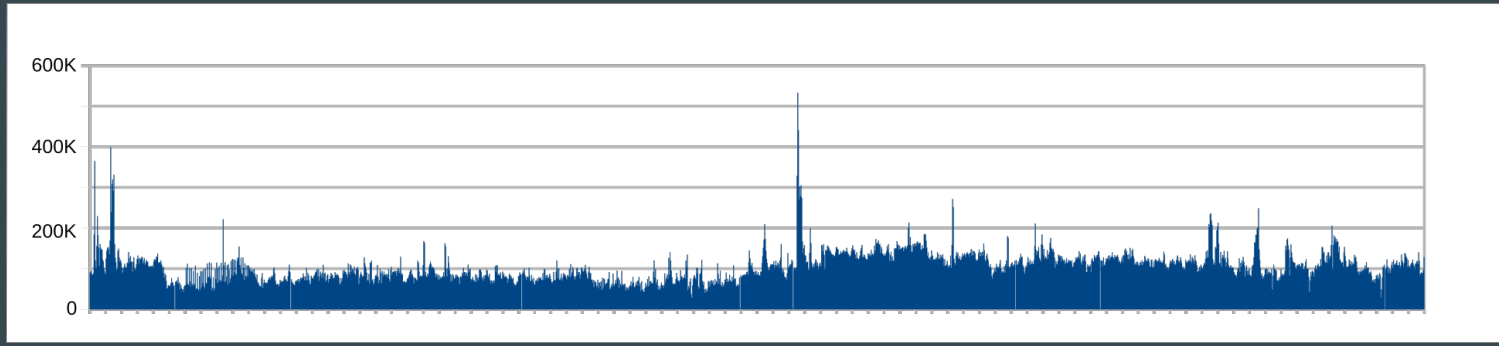


Fig.: Wikidata SPARQL traffic Jun-Sep 2017

Analysing SPARQL logs: The Bot Problem

- Query traffic is **ruled** by a few bots

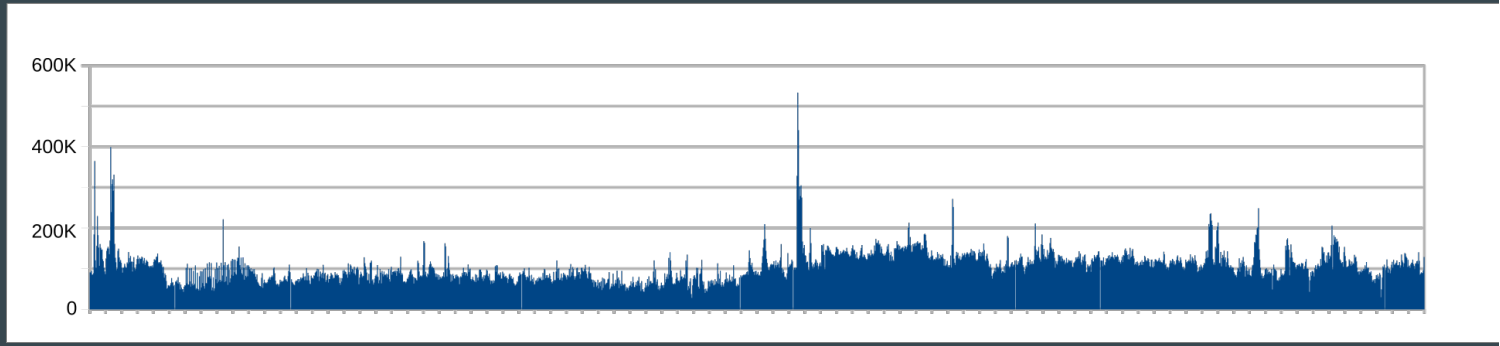


Fig.: Wikidata SPARQL traffic Jun-Sep 2017

- 41% of all Wikidata query traffic from June – September 2017 caused by one super-power user (Magnus Manske)

Analysing SPARQL logs: The Bot Problem

- Query traffic is **ruled** by a few bots

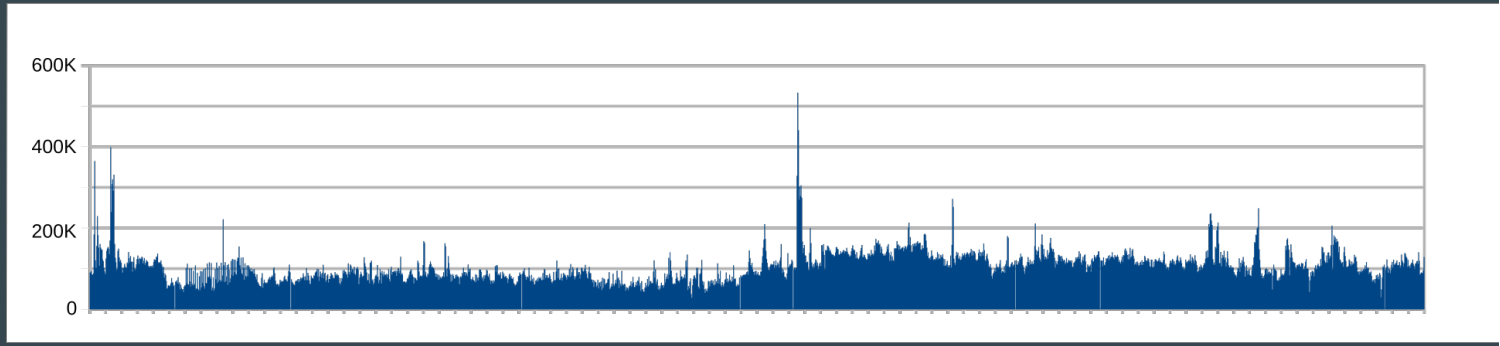


Fig.: Wikidata SPARQL traffic Jun-Sep 2017

- 41% of all Wikidata query traffic from June – September 2017 caused by one super-power user (Magnus Manske)
- The effect does **not** average out, and it affects other sites too

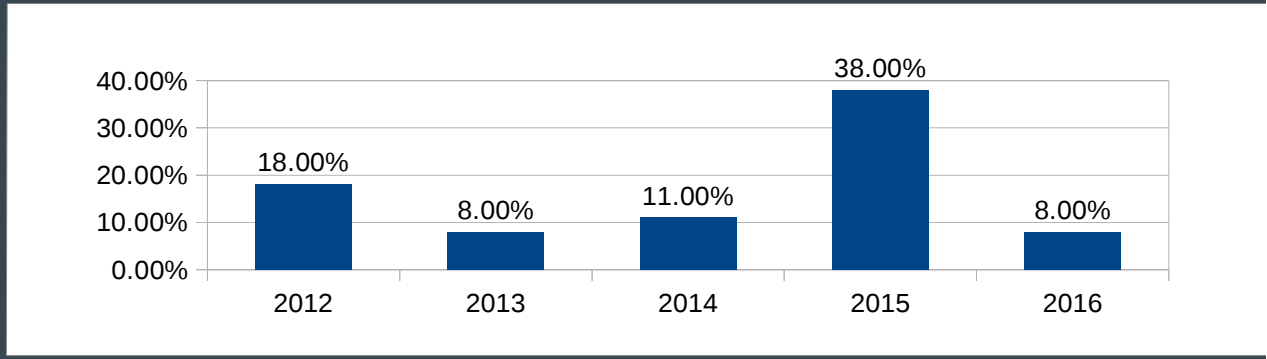


Fig.: Usage of DISTINCT on DBpedia [Bonifati et al. 2017]

Analysing SPARQL logs: The Bot Probl

- Query traffic is **ruled** by a few bots

Fig.: Wikidata SPARQL traffic Jun-Sep 2017

- 41%

une – September 2017
(Magnus Manske)

sites too

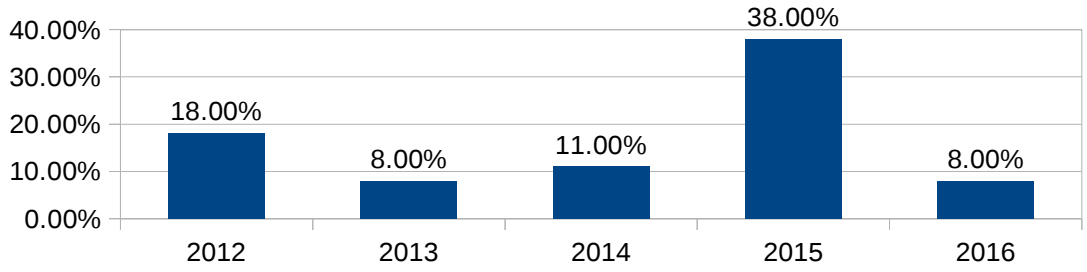


Fig.: Usage of DISTINCT on DBpedia [Bonifati et al. 2017]

No trends!
No predictability!
No insights!

Are SPARQL queries interesting after all?

- Observation: Robotic traffic dominates
  - May not represent any real interest
  - Governed by very few sources
  - Random changes – not uniform on any observed scale

Are SPARQL queries interesting after all?

- Observation: Robotic traffic dominates
  - May not represent any real interest
  - Governed by very few sources
  - Random changes – not uniform on any observed scale
- Hypothesis: Organic traffic also exists
  - Representing human information need during some interaction
  - Composed of many diverse sources
  - Continuous change over months

Note: “Organic” ≠ “hand-written SPARQL” (user apps might use SPARQL to get user-requested data without users actually writing queries)

Extracting organic traffic

- Main signal: User Agents
  - Assumption: organic traffic generally from browser-like agents

Extracting organic traffic

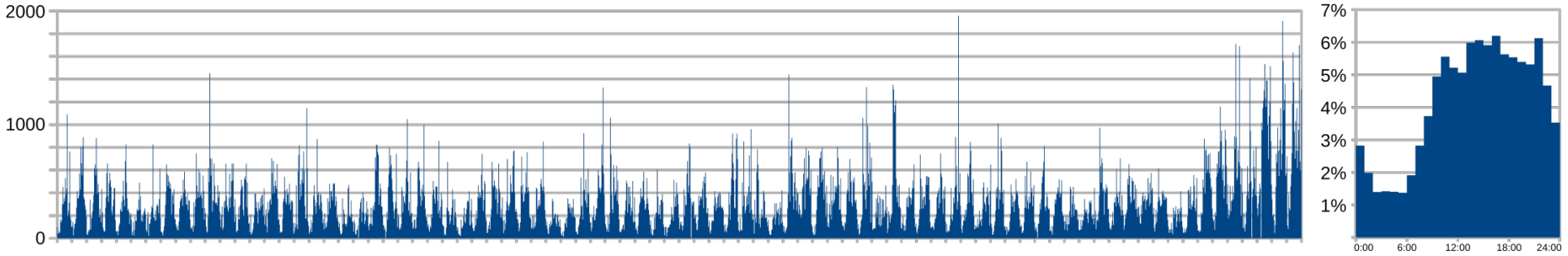
- Main signal: User Agents
  - Assumption: organic traffic generally from browser-like agents
- 2nd signal: query comments
  - Some browser-based tools mark queries using comments
- 3rd signal: activity spikes
  - Group queries by query pattern (following [Raghuveer, USEWOD'12])
  - Find agent-pattern pairs that spike (>2K requests/month)
  - Manually inspect these queries to decide if organic or robotic

→ About 300 further browser-based sources classified "robotic"

Results: Organic component

- Jun–Sep 2017: 658,890 queries (<0.5%)
- More triples
  organic 17%: 1, 97%: ≤11 vs. robotic 57%: 1, 96%: ≤7
- More varied (vocabulary, SPARQL features)

Temporal distribution of organic queries (12 weeks / time of day)



Insights on SPARQL Usage

- General: more features than reported elsewhere
- Typically organic: LIMIT, DISTINCT, OPTIONAL, ORDER BY, subqueries, aggregates, services
- Typically robotic: BIND, UNION, VALUES
- Conjunctive regular path queries with converse (C2RPQs)
  - Main query fragment for robotic queries (75% when allowing VALUES)
- OPTIONAL:
  - Important mostly for organic queries
  - Recent data (2018) also shows shift to C2RPQ+OPTIONAL (up to 82%)

Insights on Wikidata Usage

- Robotic traffic:
  - Mainly information integration bots (comparing database contents)
  - Potentially also selective data download (spider-like)
  - Most queries from a few dominant bots (>60% from top-three bots)
- Organic traffic:
  - Data browsers (often general-purpose)
  - Mobile apps (often topical)
  - Most queries from of unidentified “small” sources
- Reified statements in 4%–10% of queries

Conclusion and Outlook

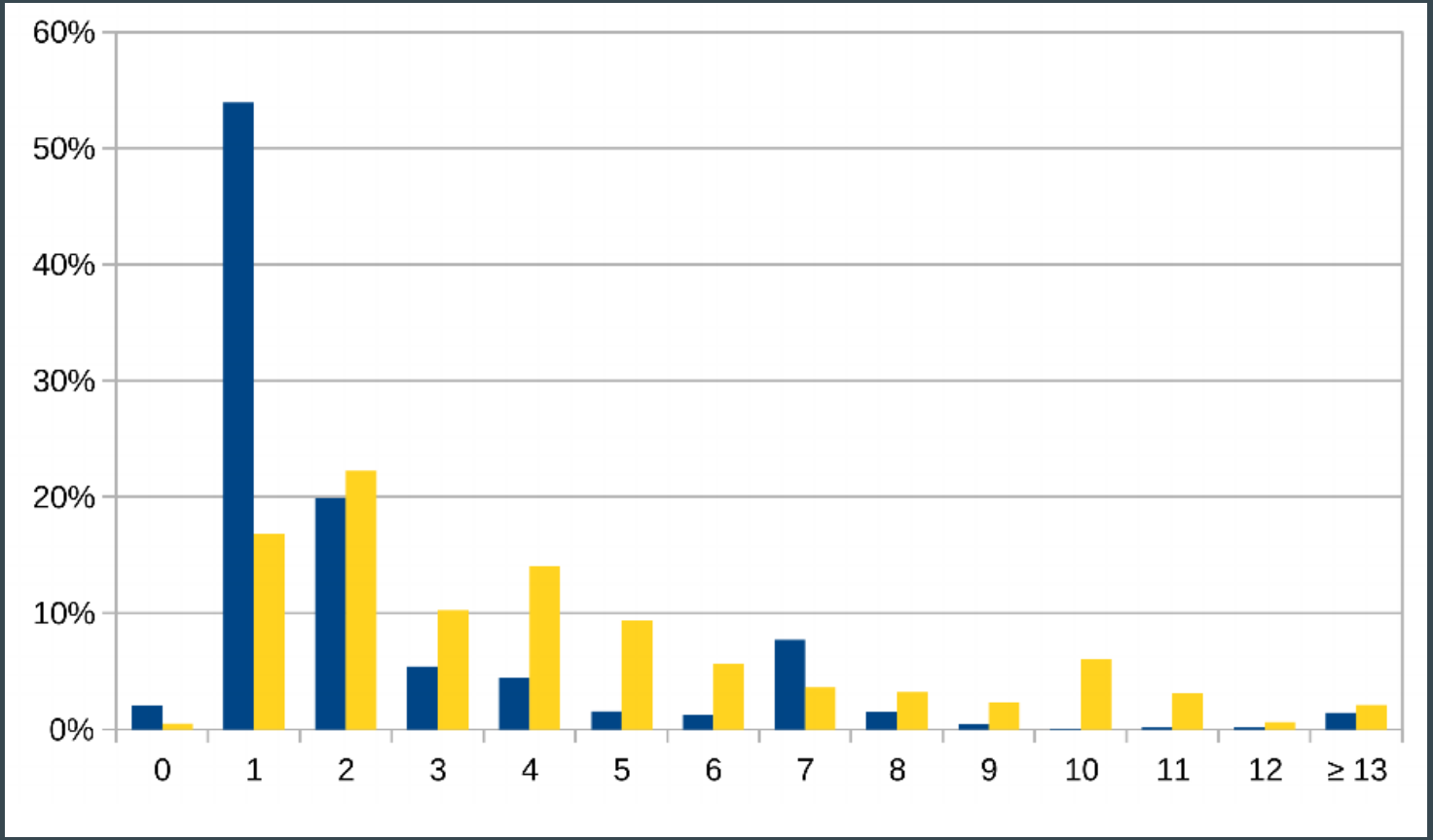
Wikidata relies on RDF and SPARQL for some of its core features – a fascinating use case!

- **Conclusions**
  - SPARQL log analysis is **methodologically difficult**
  - **Organic traffic** can be extracted based on User Agent and timestamps
  - SPARQL queries are **more varied and more complex** than reported elsewhere
  - After Joins, **path queries** are the second most important feature
- **Outlook**
  - Publishing anonymised datasets: under review; stay tuned
  - Documenting Wikidata's SPARQL deployment insights
  - Wikidata will expand further … (Dictionary content! Media meta-data!)

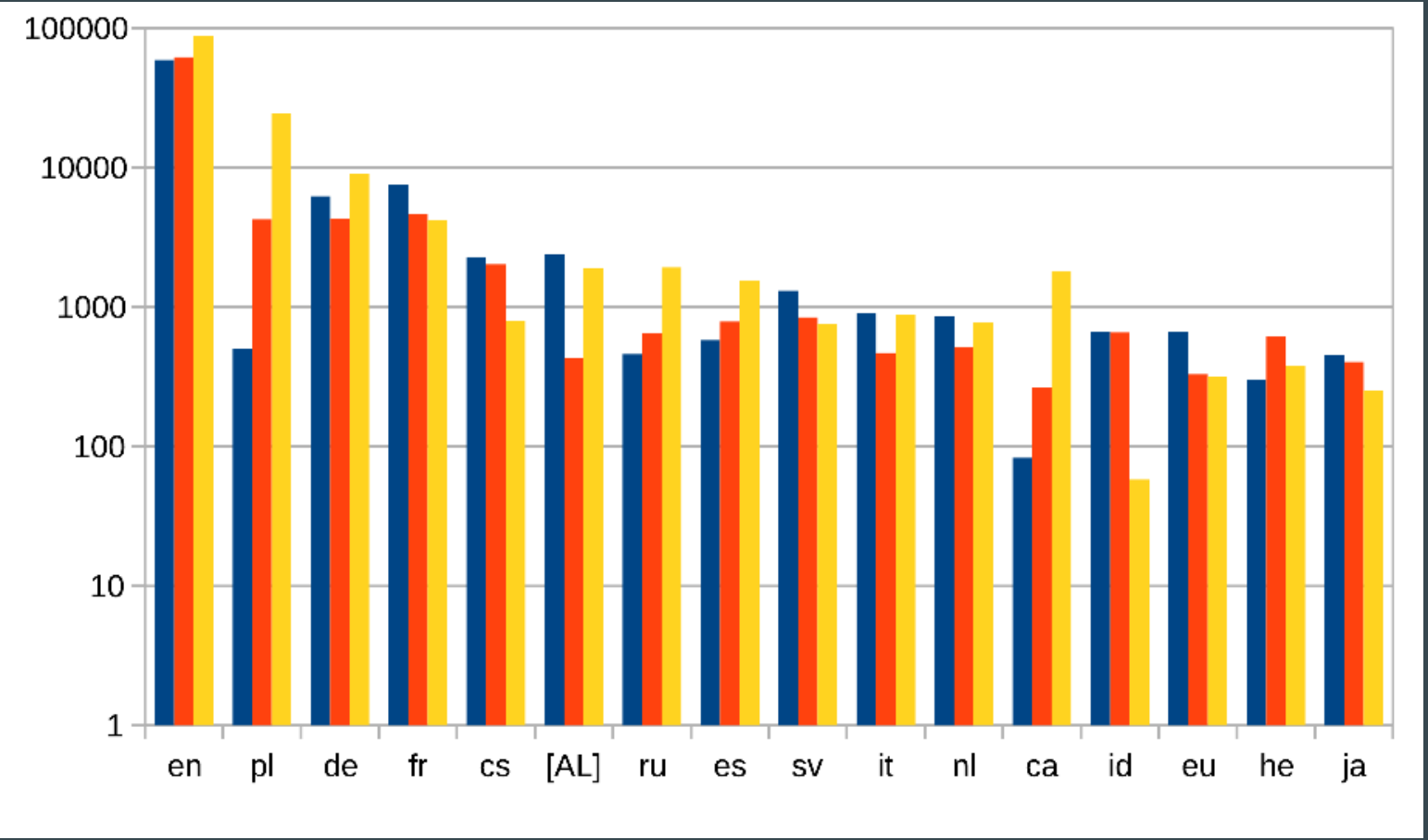
SPARQL Feature Distribution (2017/2018)

	organic						robotic					
	Jun 2017	Jul 2017	Aug 2017	Jan 2018	Feb 2018	Mar 2018	Jun 2017	Jul 2017	Aug 2017	Jan 2018	Feb 2018	Mar 2018
Limit	31.08%	39.55%	46.56%	52.31%	51.23%	36.87%	21.12%	16.86%	17.42%	20.38%	11.47%	15.17%
Distinct	26.50%	31.40%	19.05%	59.30%	60.42%	63.78%	15.84%	5.48%	4.27%	4.32%	7.54%	12.25%
Order By	17.29%	14.75%	13.22%	46.89%	46.99%	34.53%	12.97%	8.01%	6.78%	8.76%	7.68%	17.46%
Offset	0.40%	2.92%	0.37%	0.09%	0.08%	0.06%	7.73%	6.07%	6.29%	0.10%	0.07%	0.10%
Join	87.59%	87.82%	89.76%	82.50%	91.70%	87.02%	88.48%	78.53%	67.41%	73.26%	61.39%	70.19%
Optional	42.36%	46.24%	55.92%	50.90%	41.30%	41.15%	25.08%	11.63%	11.45%	12.73%	15.41%	30.71%
Filter	25.89%	29.12%	22.24%	12.59%	11.76%	11.76%	21.64%	17.92%	13.79%	14.70%	16.83%	29.02%
Path with *	15.02%	15.59%	12.88%	40.92%	32.43%	30.34%	16.43%	19.19%	14.80%	20.56%	17.26%	24.81%
Subquery	13.09%	15.30%	12.79%	6.45%	5.07%	5.39%	0.34%	0.28%	0.33%	0.09%	0.13%	0.11%
Bind	9.85%	9.23%	8.68%	4.72%	3.99%	4.15%	16.29%	12.07%	9.60%	11.94%	13.79%	24.03%
Union	5.10%	5.76%	12.62%	2.56%	2.07%	3.39%	11.26%	8.63%	7.61%	13.96%	13.05%	18.57%
Values	4.44%	3.07%	10.88%	3.29%	3.23%	3.20%	35.72%	30.74%	28.92%	29.82%	23.80%	11.90%
Not Exists	3.31%	3.37%	2.46%	1.24%	0.94%	0.69%	0.19%	0.21%	0.19%	0.27%	0.29%	0.35%
Minus	2.04%	2.91%	1.60%	0.82%	0.57%	0.71%	0.53%	0.92%	1.07%	1.46%	1.26%	1.78%
Service (lang)	44.63%	42.09%	54.78%	50.88%	41.71%	42.95%	10.40%	6.15%	4.27%	7.15%	7.91%	8.90%
Service (other)	11.49%	10.53%	10.32%	7.30%	13.14%	2.31%	4.51%	0.19%	1.16%	0.17%	0.18%	0.51%
Group By	17.12%	19.93%	13.04%	7.00%	5.40%	5.07%	0.41%	0.37%	0.48%	0.22%	0.23%	0.39%
Sample	8.85%	10.93%	4.60%	1.61%	1.63%	1.69%	0.04%	0.04%	0.06%	0.05%	0.04%	0.10%
Count	7.55%	7.60%	8.15%	5.22%	3.88%	3.73%	1.15%	4.30%	0.30%	1.52%	0.65%	0.89%
GroupConcat	1.80%	2.79%	1.17%	0.86%	0.86%	0.74%	0.06%	0.09%	0.02%	0.03%	0.02%	0.28%
Having	1.17%	1.14%	0.72%	0.65%	0.26%	0.33%	0.01%	0.01%	0.00%	0.00%	0.00%	0.04%

Triples per query: organic (blue)/robotic (yellow)



Languages of labels in organic queries



SPARQL feature co-occurrence

							organic		robotic	
J	**F**	**O**	**U**	**P**	**V**	**S**	I1–I3	I4–I6	I1–I3	I4–I6
		(none)					8.04	9.22	19.67	27.67
J							13.29	31.35	11.26	10.09
	F						1.10	0.98	1.92	1.31
J	F						6.68	2.39	2.61	1.68
J				P			2.98	1.62	13.50	13.94
J	F			P			2.48	0.58	0.39	0.07
J					V		0.39	2.01	30.42	17.47
		O					1.26	1.64	0.11	0.63
J		O					22.32	7.04	1.86	1.95
J		O		P			2.07	29.10	0.35	0.05

							organic		robotic	
J	**F**	**O**	**U**	**P**	**V**	**S**	I1–I3	I4–I6	I1–I3	I4–I6
J	F	O					2.66	1.32	2.13	1.18
J		O	U				3.49	0.25	0.02	0.01
J		O			V		3.38	0.41	0.11	0.43
J		O		P	V		1.01	0.06	0.16	0.07
J						S	2.76	1.41	0.06	0.01
J		O				S	4.78	0.62	0.00	0.01
J	F					S	3.19	2.28	0.03	0.01
J	F	O				S	1.02	0.13	0.00	0.00
J	F	O		P			0.79	0.31	0.64	1.58
J			U	P	V		0.01	0.02	0.05	1.92