An Analytical Study of Large SPARQL Query Logs

Angela Bonifati

Wim Martens Thomas Timm

Lyon 1 University

University of Bayreuth

Motivation

We want to understand SPARQL queries in practice

Which keywords and operators do SPARQL queries use?

What is the (graph-)structure of queries?

How are advanced features (such as property paths) used?

Do we see sequences of similar queries?

How do we get there?

Get our hands on **query logs** and analyze them

We collected:

Repository with ~**180 million** queries

~**56 million** are **well-formed and unique**

We'll look at these

Queries for

Biological
Geographical
Museum
Semantic Web

} databases

from 2009 to 2017

Let's analyze

Basic Types of Queries

	Absolute	Relative
Select	~49.4M	88 %
Ask	~2.8M	5 %
Describe	~2.5M	4.5 %
Construct	~1.4M	2.5 %

Today: **Select / Ask queries**

Select / Ask are the main "bread and butter" queries

Keyword and Feature Usage

We have a bunch of data on:

- Which % of queries uses which keyword?
  (e.g., distinct, limit, filter, exists, count, ...)

- Which % of queries uses which combination of operators?
  (e.g., how many use only and/optional/filter)

- Which % uses subqueries? What about projection?

Analysis similar to [Picalausa, Vansummeren SWIM'11]

Size of Queries

Measured by counting "number of triples"

```
SELECT ?item
WHERE {
  ?item wdt:composer wd:Heitor_Villa-Lobos.
  ?item wdt:catalog_code ?catalog_code.
  ?item wdt:publication_date ?publication_date
}
```

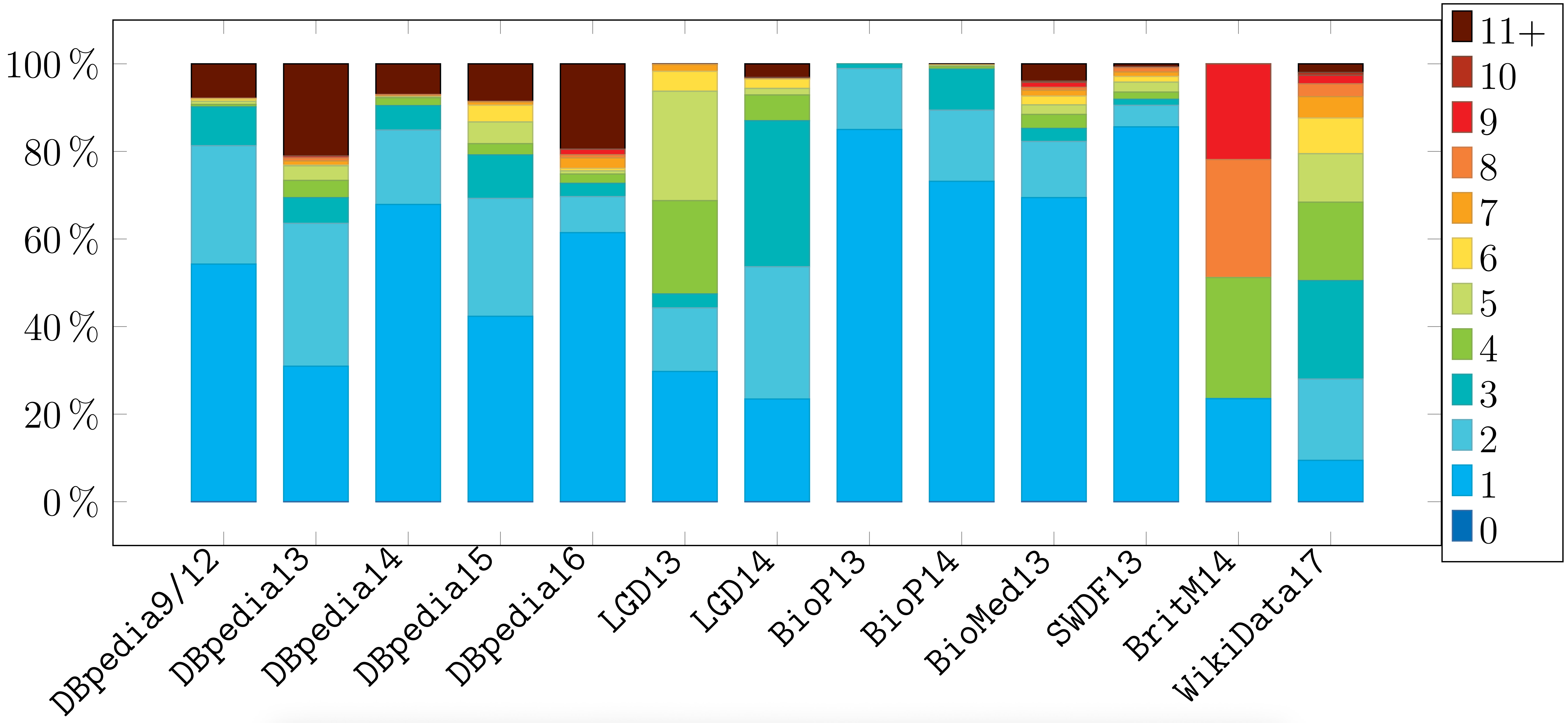
3 triples

Pieces composed by
Heitor Villa-Lobos

*1887 Rio de Janeiro
† 1959 Rio de Janeiro

(Image: public domain)

Triple Count



We see:

Lots of blue (queries use few triples)

Triple Count

Select / Ask queries in the logs:

56% have only 1 triple
91% have at most 6 triples
99% have at most 12 triples

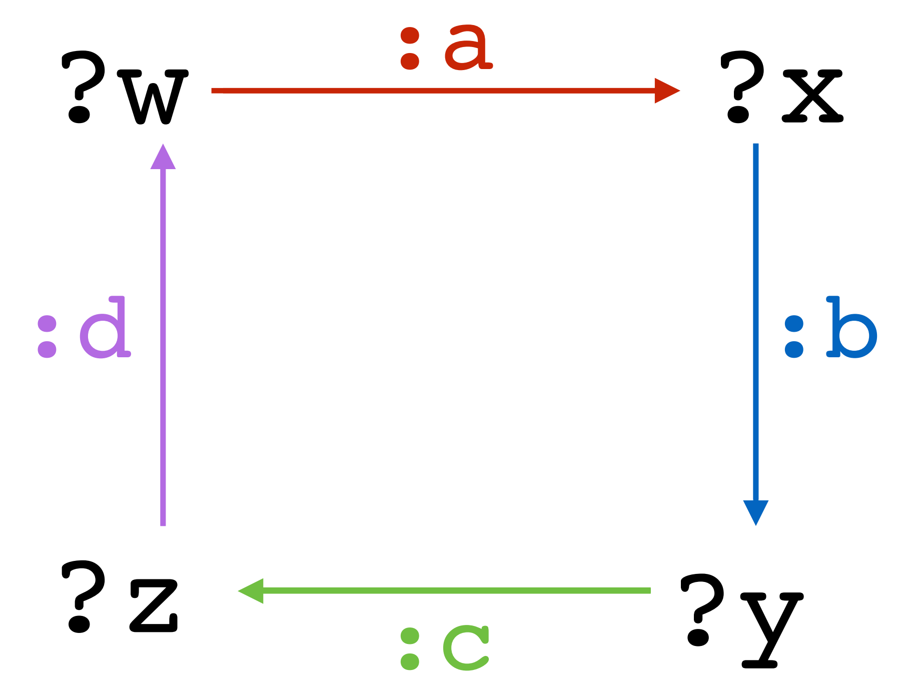
This has a significant impact on later analysis (structure of queries)

Shape Analysis

Shape Analysis

Some Queries are like **graph patterns**

```
SELECT *
WHERE
{
    ?w :a ?x .
    ?x :b ?y .
    ?y :c ?z .
    ?z :d ?w filter(?w < 30)
}
```



For many queries,
undirected graph structure ~ complexity of evaluation

e.g. k-clique

Shape Analysis

For some kinds of queries, shape $\sim$ complexity

We take queries only using

and filter optional

only unary

with care

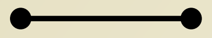
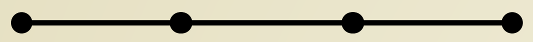
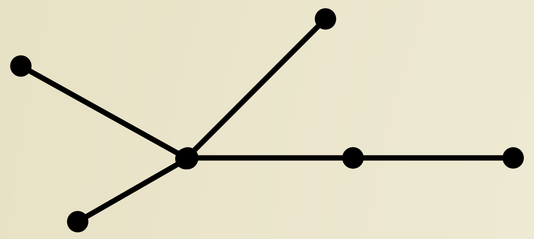
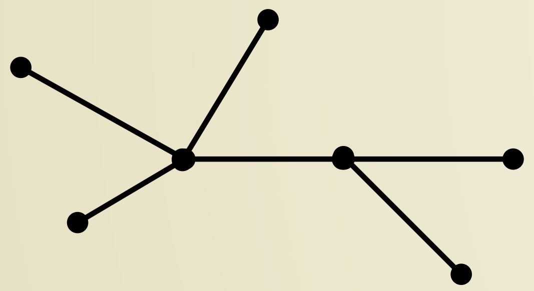
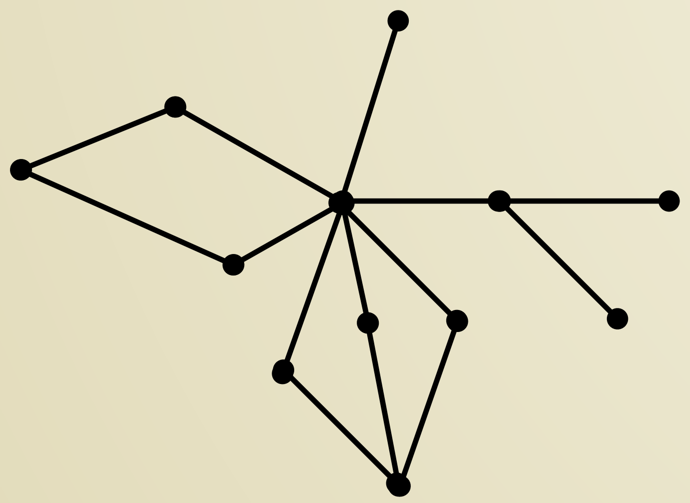
following [Barceló, Pichler, Skritek PODS15]

(well-designed, interface width 1)

How many?

56% of the Select/Ask queries (29.7M queries)

Shape Analysis

		relative to suitable and/filter/optional queries
	"single edge"	72.3%
	"chain"	90.5%
	"star"	99.5%
	"tree"	99.63%
	"flower"	99.68%

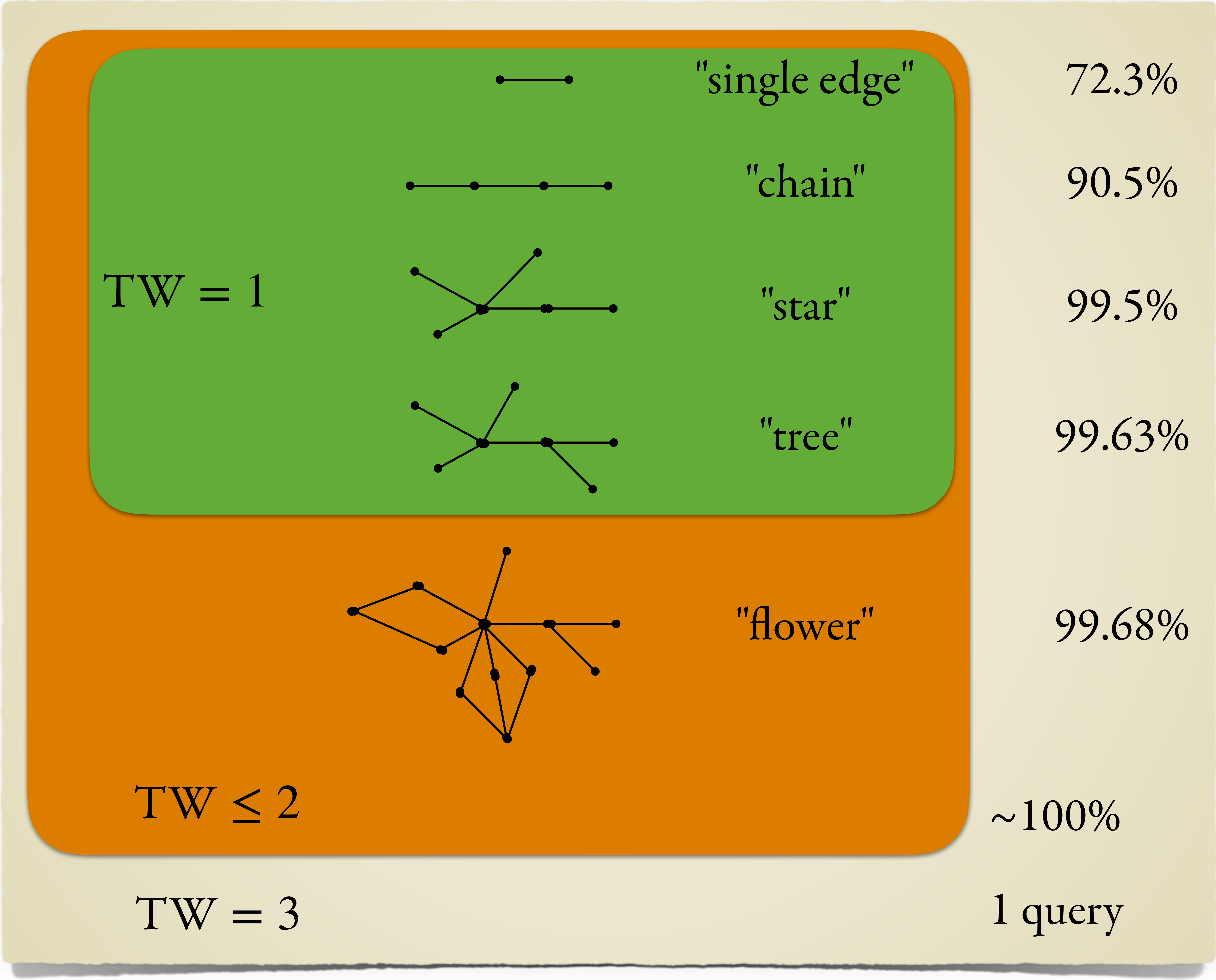
cumulative

Left over: ~42,500 queries

Treewidth

Treewidth

...measures how closely a graph resembles a tree



"single edge" 72.3%

"chain" 90.5%

TW = 1

"star" 99.5%

"tree" 99.63%

"flower" 99.68%

TW ≤ 2 ~100%

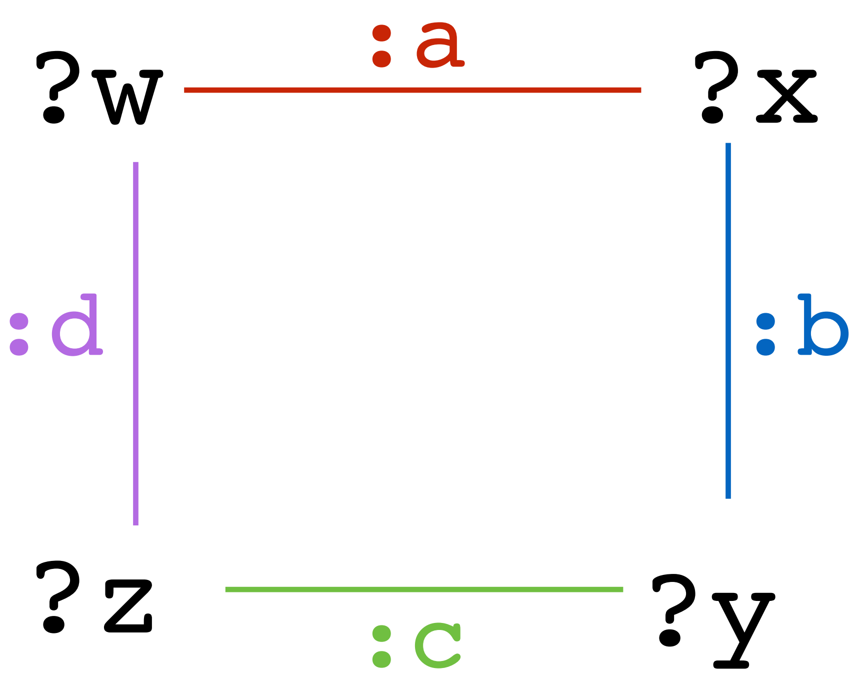
TW = 3 1 query

Shape Analysis

including more queries

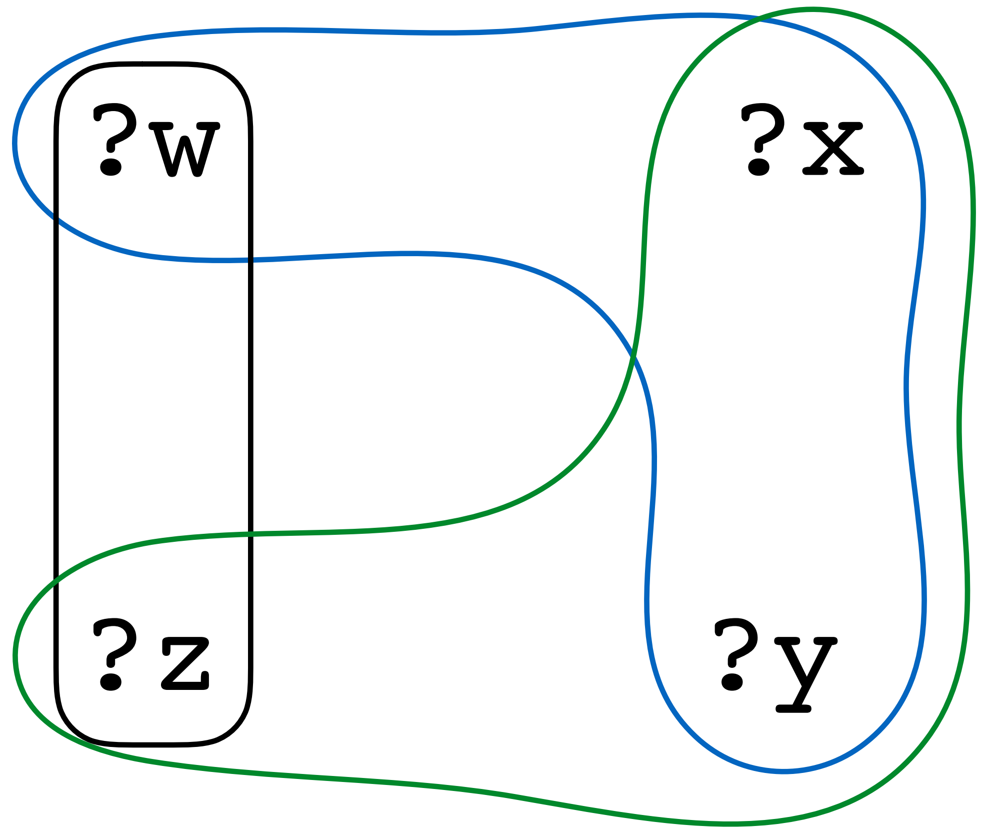
Some Queries are like Graphs

```
SELECT ?x ?y
WHERE
{
    ?w :a ?x .
    ?x :b ?y .
    ?y :c ?z .
    ?z :d ?w filter(?w < 30)
}
```



Some Queries need Hypergraphs

```
SELECT ?x ?y
WHERE
{
    ?w ?x ?y .
    ?x ?y ?z .
    ?z :d ?w filter(...)
}
```



"suitable for graph- or hypergraph analysis"

69% of the Select/Ask queries (36.7M queries)

Hypertreewidth

relevant for ~36.7 M queries
~70% of the Select/Ask queries

HTW = 1 (acyclic)	~36.65 M queries	99.84 %
HTW = 2	57,453 queries	0.16 %
HTW = 3	9 queries	0.00...02%

Take-Away

Queries have low (hyper)treewidth

Star-like shapes are very common

Property Paths

aka regular expressions
regular path queries (RPQs)

Property Paths

Standardized in SPARQL 1.1 since 2013

Overall Use

only 247.404 property paths in entire corpus

but their use highly depends on the data!

Wikidata

92 out of 308 queries (~30%)

Larger logs of Wikidata queries also have ~25 - 30% of property paths
[Bielefeldt et al. LDOW'18, Malyshev et al. ISWC'18]

Property Paths

~250K in total

63K property paths are !a (follow an edge not labeled a)

This leaves us with 184K remaining property paths

Property Paths

~250K in total

The remaining 184K property paths:

Expression Type	Relative
$(a_1 \mid \ldots \mid a_k)^*$	39.12 %
a^*	26.42 %
$a_1/\ldots/a_k$	11.65 %
a^*/b	10.39 %
$a_1 \mid \ldots \mid a_k$	8.72 %
a^+	2.07 %
$a_1?/\ldots/a_k?$	1.55 %
$a(b_1/\ldots/b_k)$	0.02 %
$a_1/a_2?/\ldots/a_k?$	0.02 %
$(a/b^*) \mid c$	0.01 %

Expression Type	Relative
$a^*/b?$	0.01 %
$a/b/c^*$	0.01 %
$(a_1 \mid \ldots \mid a_k)^+$	0.01 %
$(a_1 \mid \ldots \mid a_k)(a_1 \mid \ldots \mid a_k)$	5
$a? \mid b$	2
$a^* \mid b$	2
$(a \mid b)?$	2
$a \mid b^+$	1
$a^+ \mid b^+$	1
$(a/b)^*$	1

Observation

These are quite simple,

considering that PPs can be arbitrary regular expressions

Property Paths

~250K in total

The remaining 184K property paths:

Expression Type	Relative
$(a_1 \mid \ldots \mid a_k)^*$	39.12 %
a^*	26.42 %
$a_1/\ldots/a_k$	11.65 %
a^*/b	10.39 %
$a_1 \mid \ldots \mid a_k$	8.72 %
a^+	2.07 %
$a_1?/\ldots/a_k?$	1.55 %
$a(b_1/\ldots/b_k)$	0.02 %
$a_1/a_2?/\ldots/a_k?$	0.02 %
$(a/b^*) \mid c$	0.01 %

Expression Type	Relative
$a^*/b?$	0.01 %
$a/b/c^*$	0.01 %
$(a_1 \mid \ldots \mid a_k)^+$	0.01 %
$(a_1 \mid \ldots \mid a_k)(a_1 \mid \ldots \mid a_k)$	5
$a? \mid b$	2
$a^* \mid b$	2
$(a \mid b)?$	2
$a \mid b^+$	1
$a^+ \mid b^+$	1
$(a/b)^*$	1

Almost all expressions

do some local navigation (optionally) followed by a transitive step

"Simple transitive expressions" [M. and Trautner, ICDT 2018]

Property Paths

~250K in total

The remaining 184K property paths:

Expression Type	Relative
$(a_1 \mid \ldots \mid a_k)^*$	39.12 %
a^*	26.42 %
$a_1/\ldots/a_k$	11.65 %
a^*/b	10.39 %
$a_1 \mid \ldots \mid a_k$	8.72 %
a^+	2.07 %
$a_1?/\ldots/a_k?$	1.55 %
$a(b_1/\ldots/b_k)$	0.02 %
$a_1/a_2?/\ldots/a_k?$	0.02 %
$(a/b^*) \mid c$	0.01 %

Expression Type	Relative
$a^*/b?$	0.01 %
$a/b/c^*$	0.01 %
$(a_1 \mid \ldots \mid a_k)^+$	0.01 %
$(a_1 \mid \ldots \mid a_k)(a_1 \mid \ldots \mid a_k)$	5
$a? \mid b$	2
$a^* \mid b$	2
$(a \mid b)?$	2
$a \mid b^+$	1
$a^+ \mid b^+$	1
$(a/b)^*$	1

"This one looks a bit strange"

Almost all expressions

do some local navigation (optionally) followed by a transitive step

"Simple transitive expressions" [M. and Trautner, ICDT 2018]

Wrapping Up

Interpreting Our Results

What do query logs say about "what users want"?

Query Logs from SPARQL Endpoints Have Bias

- Many different users
- Many simple queries ("getting started")
- Slow engine / time-outs generate bias

Simple queries are overrepresented

If some class of queries is prominent in the logs, it's OK to conclude that it's an important class

But if something doesn't appear a lot in the logs, it doesn't mean it's not important

Main Findings

"Things you can cite"

In the logs we investigated...

...most queries are small

...most queries are conjunctive

...most queries are patterns

...most queries are acyclic

...most queries have low (hyper)treewidth

...property paths (regular expressions, RPQs) are usually simple

...queries appear in streaks (sequences of similar queries)