Optimising Coverage, Freshness and Diversity in Live Exploration-based Linked Data Queries

Steven Lynden, Makoto Yui, Akiyoshi Matono,
Akihito Nakamura, Hirotaka Ogawa, Isao Kojima

Artificial Intelligence Research Center, AIST

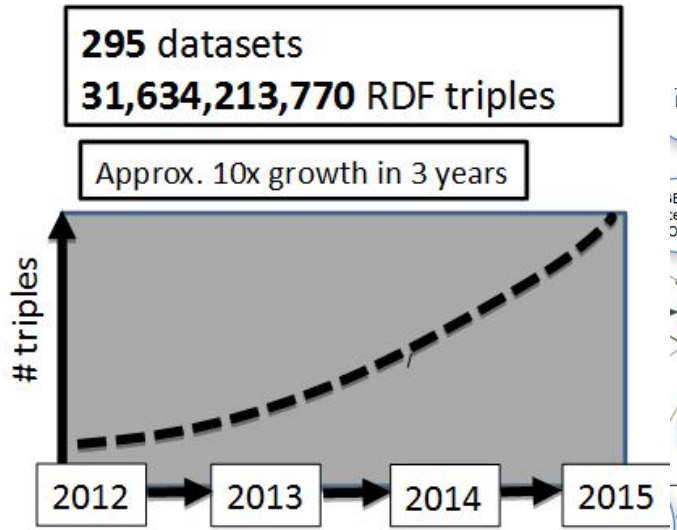
Overview

- **Introduction**
  - Linked data
  - Query processing over Linked Open Data
- **Motivation**
  - Information Retrieval + Linked Data Query Answering
  - Best-effort query answering
- **Challenges** of live exploration-based linked data query processing
- **Our approach**
  - IRI selection strategy based on similarity measures
- **Experimental investigation** over the Web of Data
- **Conclusions**

Linked Data

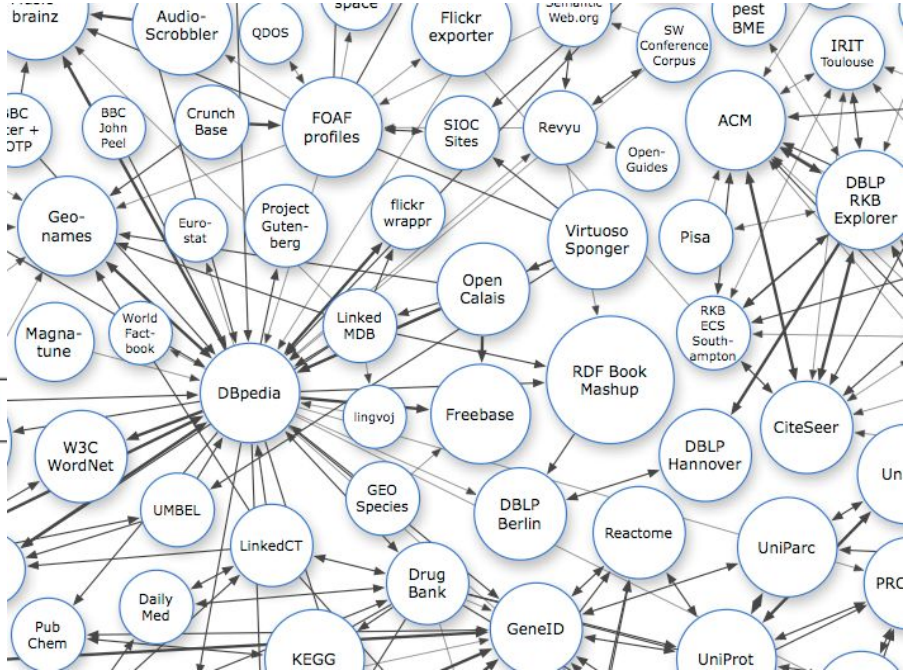
- Use URIs/IRIs to identify things
- Use HTTP IRIs
  - So that things can be looked up (dereferenced)
- Provide useful information about resource being identified
  - Using standards such as RDF.
- Refer (link) to other resources using HTTP IRI-based names when publishing data on the Web

Linked Open Data (LOD) “Cloud”



Sources:
http://webdatacommons.org
http://lod-cloud.net

http://sparqles.ai.wu.ac.at/ SPARQL Endpoint Monitoring (549) endpoints



Querying: general approaches (1)

- SPARQL endpoint provided by a single data publisher e.g. DBpedia/Wikidata

✓ Fast, up-to-date, results

✗ Misses out on the potential of inter-links

Querying: general approaches (2)

- Data warehouse approaches (copying data into a central repository).

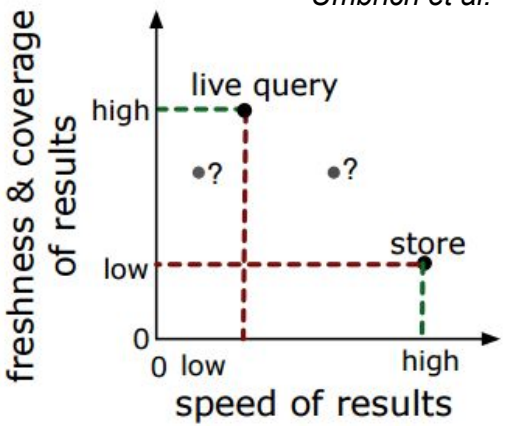
✓ Fast results

✗ Set-up/maintenance of repository

✗ Results may not be fresh

  - Dynamic Linked Data Observatory
    - http://swse.deri.org/dyldo/
    - ~ 20% of data may be dynamic

From Hybrid SPARQL Queries: Fresh vs. Fast Results *Jurgen Umbrich et al.*



Querying: general approaches (3)

- Federated query processing over multiple endpoints (inspired by relational federated query processors).

✓ Fresh results

✗ Limited coverage (some data no exposed as endpoints)

✗ Optimisation difficult, high response times

Querying: general approaches (4)

- Linked data query execution (rely on the linked data query execution principles).

✓ Fresh results

✓ No bounds on coverage

Many open challenges in terms of producing results within usable time-frames for most applications.

Linked data query processing

- Use data links and dereferencing IRIs during query execution.
- Generally called “live-exploration” or “link traversal”
- Number of variants, common properties:
  - Recursive IRI dereferencing process.
  - Dereferencing IRIs provides two purposes, providing more IRIs to dereference and proving the RDF data necessary to answer the query.
  - Possibility of using no, or almost no a priori information about data sources (depending on the query).

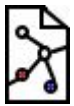
Live Exploration-based Query Processing

SPARQL query

```
SELECT DISTINCT *
WHERE {
        ?paper <http://data.semanticweb.org/ns/swc/ontology#isPartOf>
                <http://data.semanticweb.org/conference/iswc/2008/proceedings> .
        ?paper <http://swrc.ontology.org/ontology#author> ?p .
        ?p rdfs:label ?n .
}
```

Initial dereferencing (seed IRIs)

e.g. http://data.semanticweb.org/conference/iswc/2008/proceedings is dereferenced and RDF data obtained.



Triple pattern matching

Contains RDF matched against triple patterns, used to answer the query.

e.g. http://conference:iswc/2008/paper/37 (subject)

isPartOf (predicate)

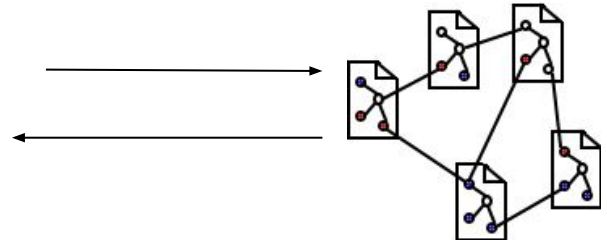
http://data.semanticweb.org/conference/iswc/2008/proceedings (object)

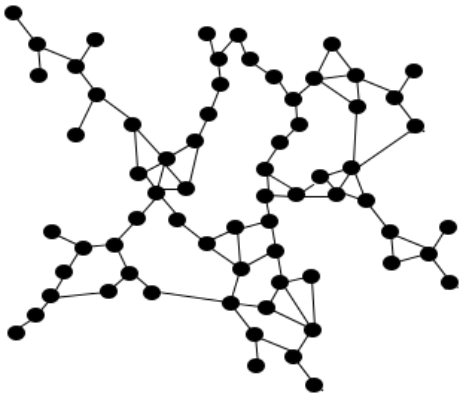
Matches the triple pattern: ?paper <http://data.semanticweb.org/ns/swc/ontology#isPartOf> <http://data.semanticweb.org/conference/iswc/2008/proceedings> .

Live Exploration-based Query Processing (2)

Iterative dereferencing

IRIs are repeatedly selected, dereferenced and matching triples added to a local graph for the set of incrementally discovered linked documents.





Graph grows until some termination condition.

Query evaluation

The local graph is used to obtain an answer to the initial SPARQL query.

Variants

- Computation of query result
  - Two phased execution
    - First phase just to deference IRIs and obtain RDF triples.
    - Second phase evaluates the query.
  - Integrated execution
    - Query result is generated incrementally as IRIs are dereference, results potentially streamed.
- Dereferencing strategy
  - None, e.g. dereference IRIs in the order they are obtained.
  - Optimised (use indexes, query history etc.).

Enhancements

- Data source selection (required where no seed-IRIs exist) and ranking.
  - Indexing
  - Hybrid approaches
- The work in our paper focuses on IRI selection strategies for live exploration, however indexes/hybrid approaches can complement the approach.

Challenges - scale

- Potentially the number of IRIs to dereference can be large
  - Possibly cannot retrieve everything due to memory/processing constraints
  - No guarantees of termination
    - Results outside certain time constraints may be useless
  - Many repeated HTTP requests may have consequences
    - Web crawling-like “politeness” policy required.
- In the absence of a priori information, need to implement IRI selection policies

Challenges - IRI selection

- Can something be gained by looking at IRI structure alone?
  - May seem counter-intuitive, however
    - Various work on it exists:

      Eda Baykan , Monika Henzinger , Ludmila Marian , Ingmar Weber, Purely URL-based topic classification, Proceedings of the 18th international conference on World wide web, April 20-24, 2009, Madrid, Spain

      Min-Yen Kan , Hoang Oanh Nguyen Thi, Fast webpage classification using URL features, Proceedings of the 14th ACM international conference on Information and knowledge management, October 31-November 05, 2005, Bremen, Germany

      Inma Hernández , Carlos R. Rivero , David Ruiz , Rafael Corchuelo, CALA: An unsupervised URL-based web page classification system, Knowledge-Based Systems, 57, p.168-180, February, 2014

    - The growing use of Semantic URLs (clean URLs, RESTful URLs, user-friendly URLs, search engine-friendly URLs)
      - "Slug": part of the URL with human-readable keywords reflecting content

Challenges - query construction

- If little (or no) a priori information about the linked open data retrievable
  - writing a query itself may become challenging.
- Best practices exist, but
  - different linked data sources may use different ontologies.

- Users may desire certain properties of results that are more abstract, e.g. diversity.

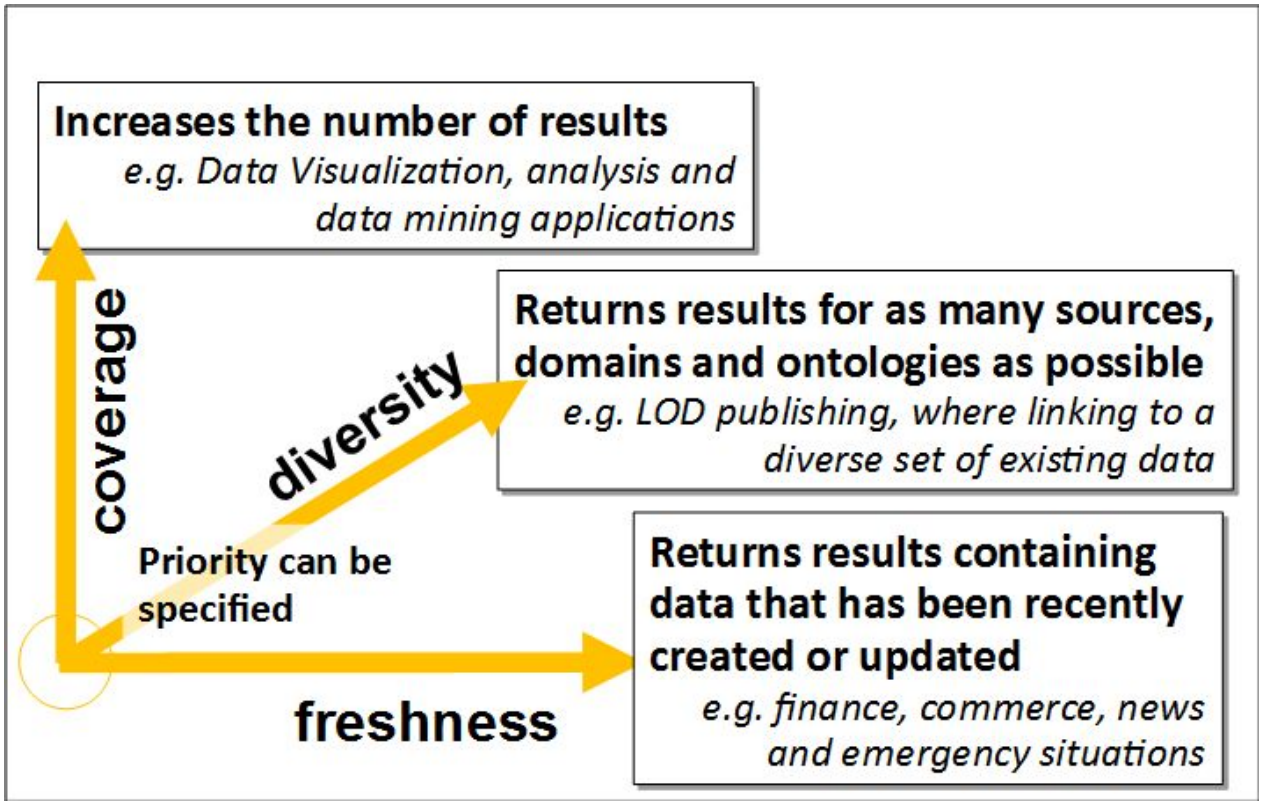
Our approach (1) - best-effort

- Provide results within useful time frame
  - Results may be incomplete, but:
    - Important for practical applications
- Allow the user to specify termination time
  - Alternative to LIMIT

```
SELECT DISTINCT *
WHERE {
  ?paper <http://data
    <http://data.sema
  ?paper <http://swr
  ?p rdfs:label ?n .
} WITHIN 5 seconds
```

- Objective is to optimise IRI selection within the given time-frame

Our approach (2) - optimise user criteria



Assumptions

- Able to function with or without indexes, data source metadata: any a priori information.
- Suitable for any execution paradigm
  - two-phase, integrated, hybrid, ...
- Optimisation should be done on-the-fly within a single query
  - Query execution history not assumed to be available.

Similarity-based IRI prioritisation

- Assume that IRIs in some way similar to those dereferenced previously have similar characteristics
  - Provide similar coverage, diversity, or freshness
- Test if this is true for IRIs that are
  - Representationally similar
  - Semantically similar

IRI similarity - example

```
SELECT * WHERE {
   ?r sw:relatedToEvent
     <http://data.semanticweb.org/conference/
  eswc/2010> .
  ?r sw:editor ?n
}
```

Query to find all editors of material related to ESWC 2010

A linked data query processor may try to answer this by initially dereferencing the IRI:

```
<http://data.semanticweb.org/conference/eswc/2010>
```

which will result in a number of bindings for the variable ?r of the form:

```
http://data.semanticweb.org/conference/...,
```

In this case IRIs representing other conferences have editors, providing more results.

and a number of bindings of the form:

```
http://data.semanticweb.org/person/...
```

IRIs starting with this substring unlikely to match triple patterns

Similarity measures

Representational (string) similarity

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Levenshtein (edit) distance

Semantic connectivity

$$scs(i_1, i_2) = \frac{1}{1 + (\sum_{l=1}^{r} d^l * \mid paths_l(i_1, i_2) \mid)}$$

Combining a co-occurrence-based and a semantic measure for entity linking
Bernardo Pereira Nunes et al.

Ranking functions

- Given a number of IRIs the can be dereferenced, rank them based on:
  - String similarity

$$strRank(iri) = \sum_{i \in I} [utility(i) * (1 - distance(i, iri))]$$

  - Semantic connectivity

$$scsRank(iri) = \sum_{i \in I} [utility(i) * (1 - scs(i, iri))]$$

Utility functions

- Based on the criteria being optimised:
  - Coverage
  - Freshness
  - Diversity

Coverage utility function

Coverage of a document *i* is based on the contribution to the local graph (triples that match patterns in the query)

$$matches(i) = \mid localGraph_{t1} \mid - \mid localGraph_{t0} \mid$$

Freshness utility function

```
dcterms:#modifed
dcterms:modifed
dcterms:date
dc:date
dcterms:created
dcterms:issued
lj:dateCreated
swivt:#creationDate
lj:dateLastUpdated
wiki:Attribute3ANRHP
_certification_date
date 0.18 53
tl:timeline.owl#start
tl:timeline.owl#end
bio:date
po:schedule
date
swrc:ontology#value
cordis:endDate
nl:currentLocationDateStart
po:start_of_media_availability
liteco:dateTime
```

Freshness of a document *i* is based on the existence of RDF triples indicating something about when it was created/updated

$$freshness(i) = \mid tPreds(i)_{day} \mid + (\mid tPreds(i)_{month} \mid \times 0.1) + (\mid tPreds(i)_{year} \mid \times 0.01)$$

Diversity utility function

Dispersion

- The number of distinct pay-level-domains (PLD)s from which IRIs are dereferenced.

Divergence

Jenson Shannon Divergence

$$JSD(P\|Q) = \frac{1}{2}D(P\|M) + \frac{1}{2}D(Q\|M)$$

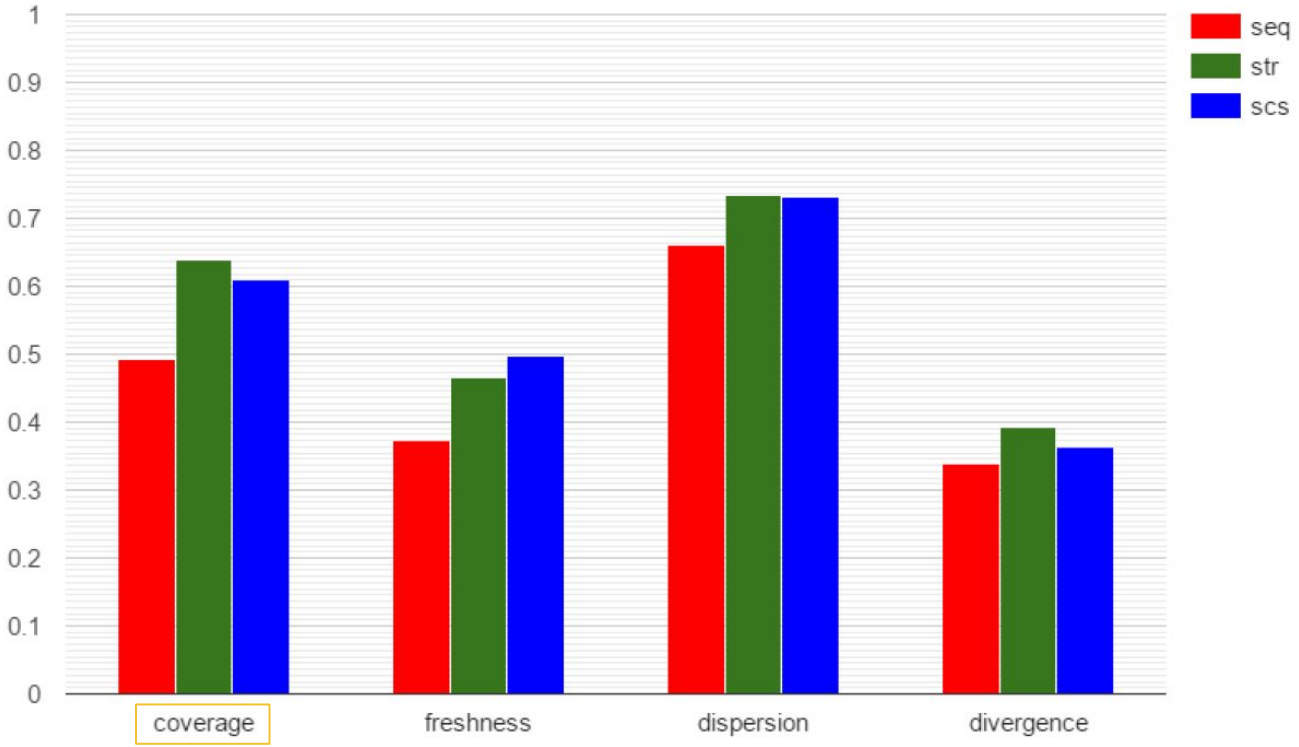
Where $M = \frac{1}{2}(P+Q)$ and $D(P\|Q)$
Is the K-L divergence: $\sum_i P(i) ln \frac{P(i)}{Q(i)}$.

Evaluation

- Performed experiments over the real, live Web of Data using 64 test queries.
- Compared representational similarity (str) and semantic similarity (scs), with sequential selection (seq) of IRIs
- Constraints
  - Politeness policy of 0.5 second delay between HTTP requests to the same pay-level-domain (PLD)
  - Maximum of 200 PLDs per query.
  - Maximum of 300 requests to any PLD.

Results

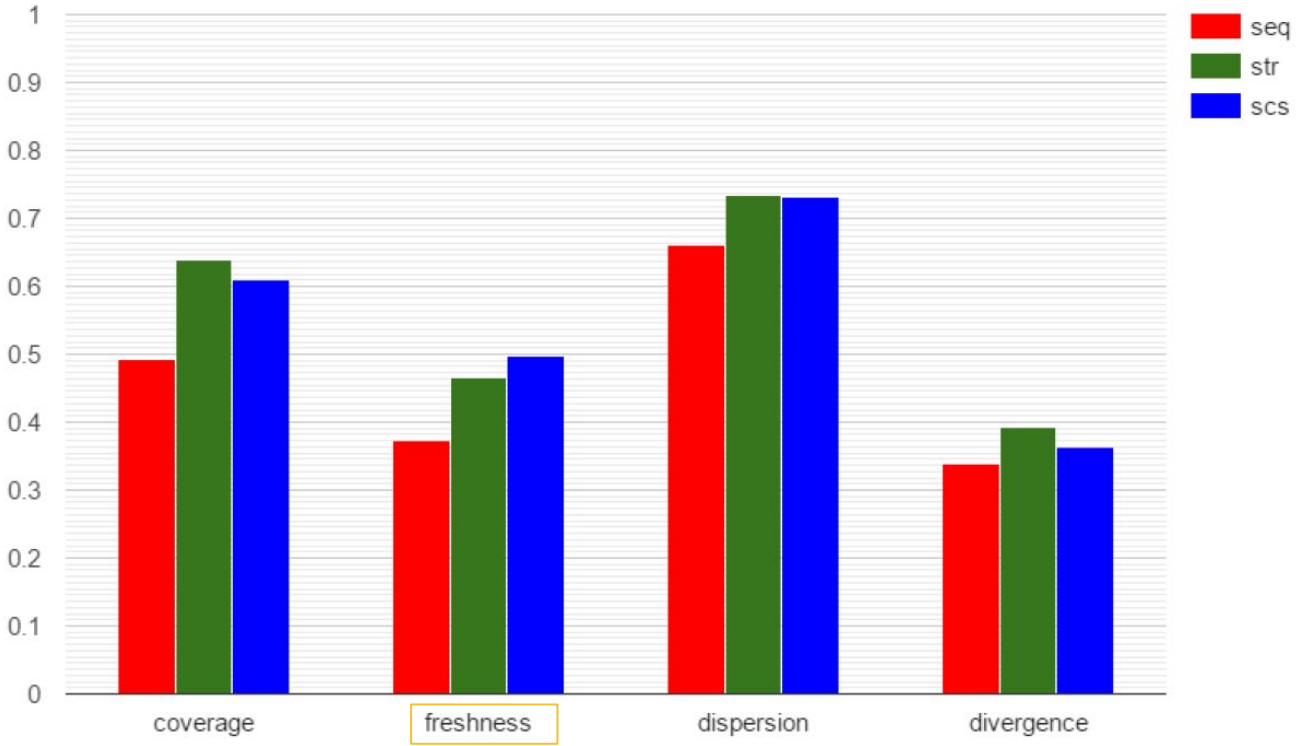
Normalised averages
[0.1] using rescaling



- For coverage, sequential selection was outperformed by 29.4% (str) and 23.7% (scs)

Results

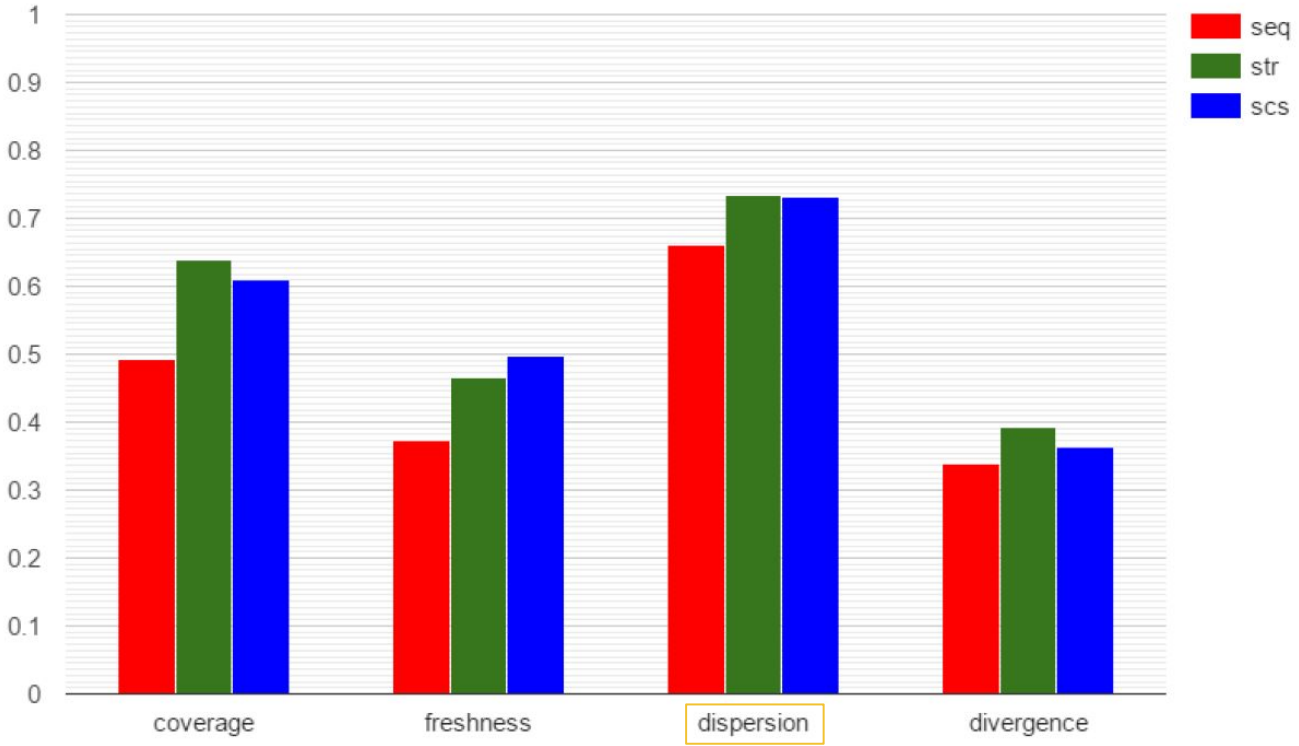
Normalised averages
[0.1] using rescaling



- For freshness, sequential selection was outperformed by 25.2% (str) and 33.9% (scs)

Results

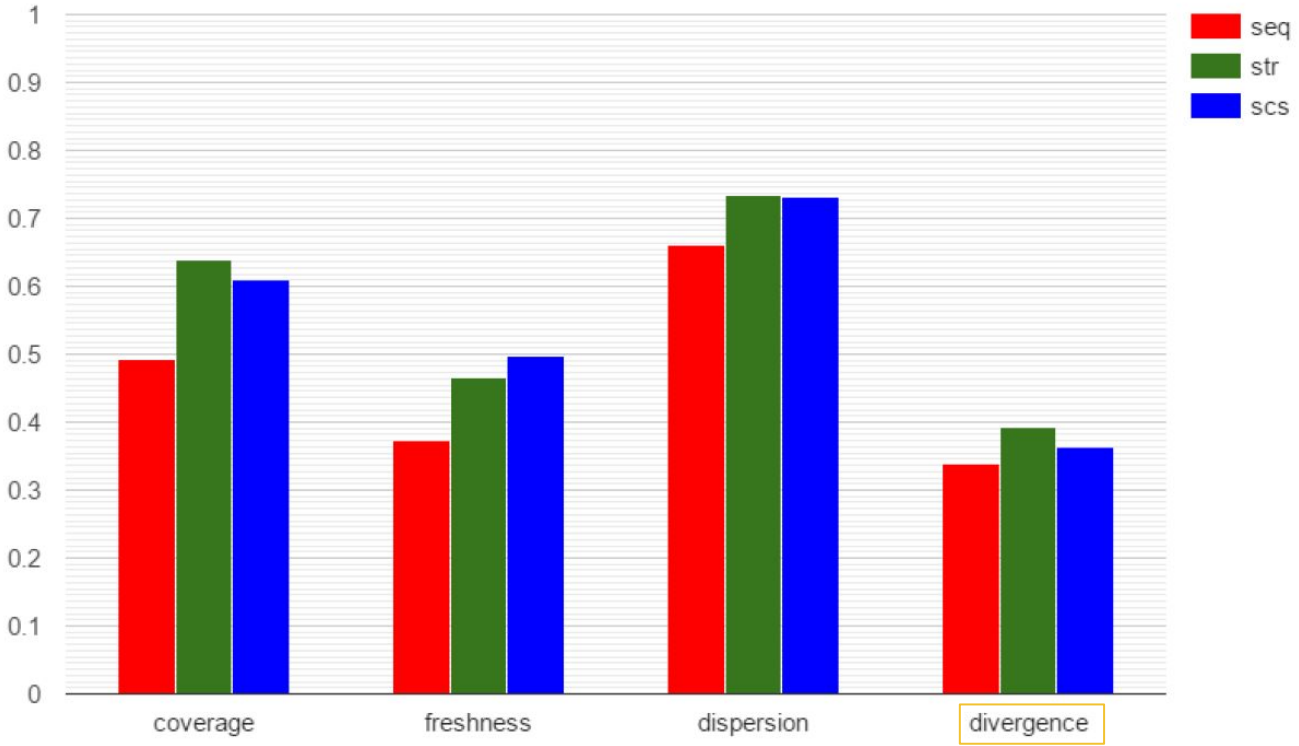
Normalised averages
[0.1] using rescaling



- For dispersion, sequential selection was outperformed by about 10% by both strategies.

Results

Normalised averages
[0.1] using rescaling



- For divergence, sequential selection was outperformed by 16.2% (str) and 7% (scs)

Conclusions

- Similarity-based selection of IRIs for optimising user-criteria during best-effort, live exploration-based query processing.
  - **30% better** than sequential selection when optimising for **coverage** and **freshness**.
  - **~10%** improvements for "**diversity**" (**dispersion/divergence**), which is more difficult to define.

- Can be combined with parallel querying of SPARQL endpoints, and techniques such as indexing
  - Towards practical Linked Data query answering.
- Utilising query execution history may improve results.