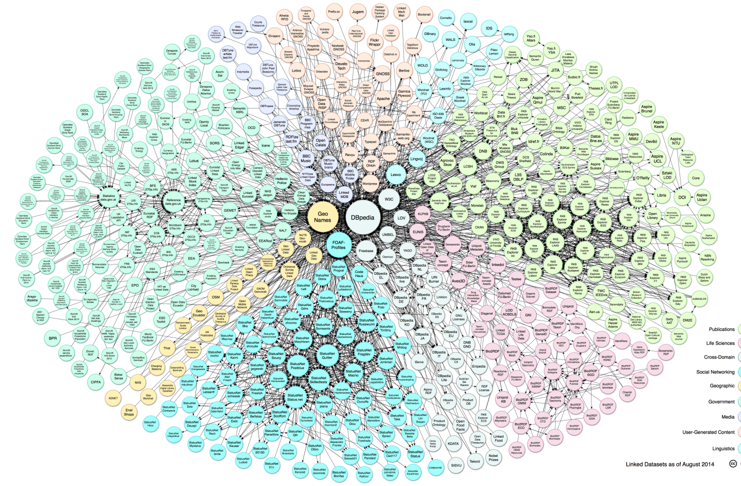
FORK: Feedback-aware ObjectRank-based Keyword Search over Linked Data

Takahiro Komamizu, Sayami Okumura, Toshiyuki Amagasa, Hiroyuki Kitagawa

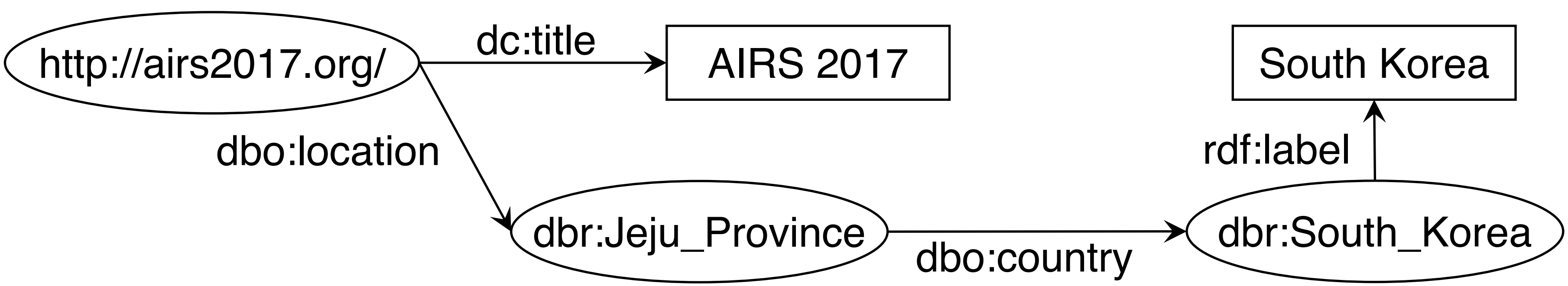
University of Tsukuba, Japan

Linked Data (LD)



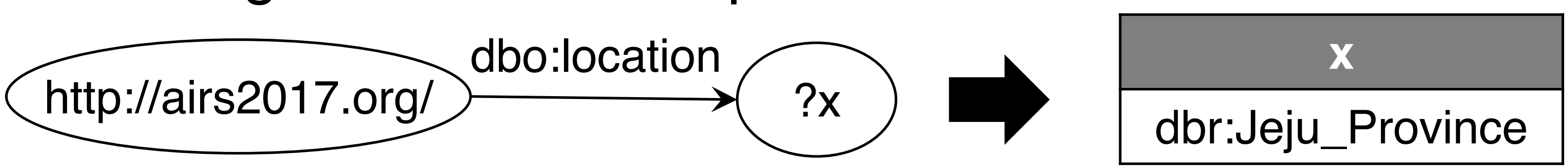
Linked Open Data cloud diagram (2014-08)

- Open data paradigm
- Linking facts in open data
  - RDF (Resource Description Framework)
  - e.g.,
    - <http://airs2017.org/> dc:title "AIRS 2017" .
<http://airs2017.org/> dbo:location dbr:Jeju_Province .
dbr:Jeju_Province dbo:country dbr:South_Korea .
dbr:South_Korea rdf:label "South Korea" .



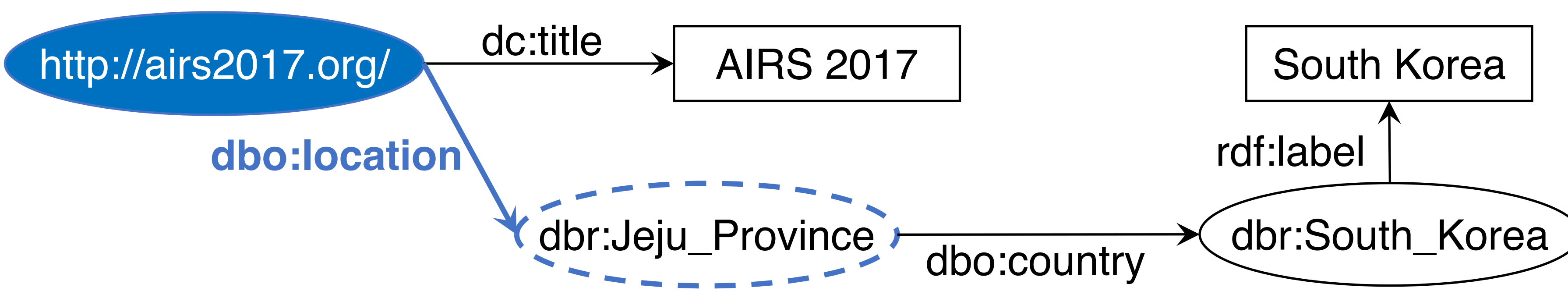
Search over LD

- Finding facts in LD data
- Standardized method: SPARQL query
  - Graph pattern-based requirement representation
  - Bindings to variables in patterns are results.



Graph pattern query

Results

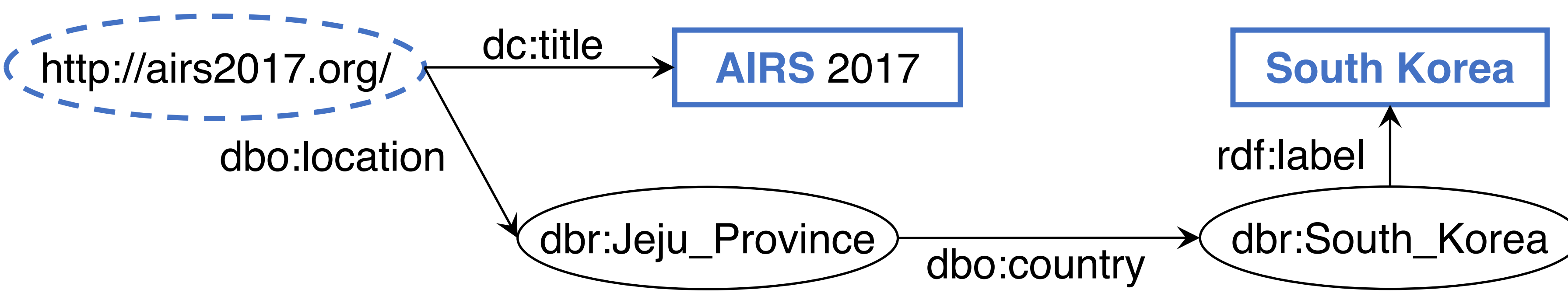


Keyword Search over LD

- User-friendly method: keyword search
  - Keyword-based representation
  - Facts **related** with query are results.
    - e.g., related means common ancestor node

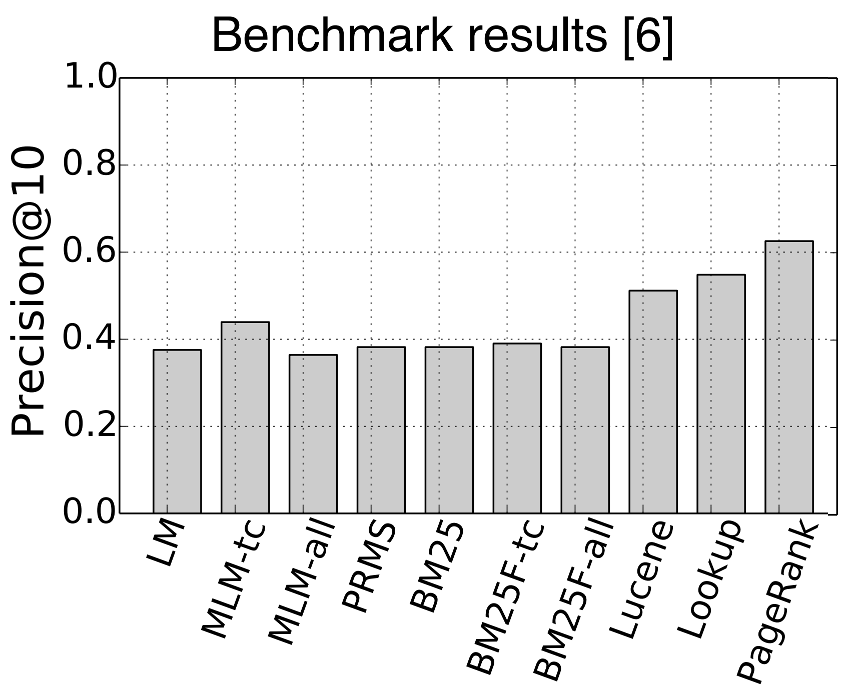
(“AIRS”, “South Korea”) → http://airs2017.org/

Keyword query — Results



Ranking is still challenging.

- IR-based techniques are < 0.6.
- Graph analysis-based techniques are still < 0.65.



[6] Balog, K., Neumayer, R.: A Test Collection for Entity Search in DBpedia. In: SIGIR 2013. pp. 737–740 (2013)

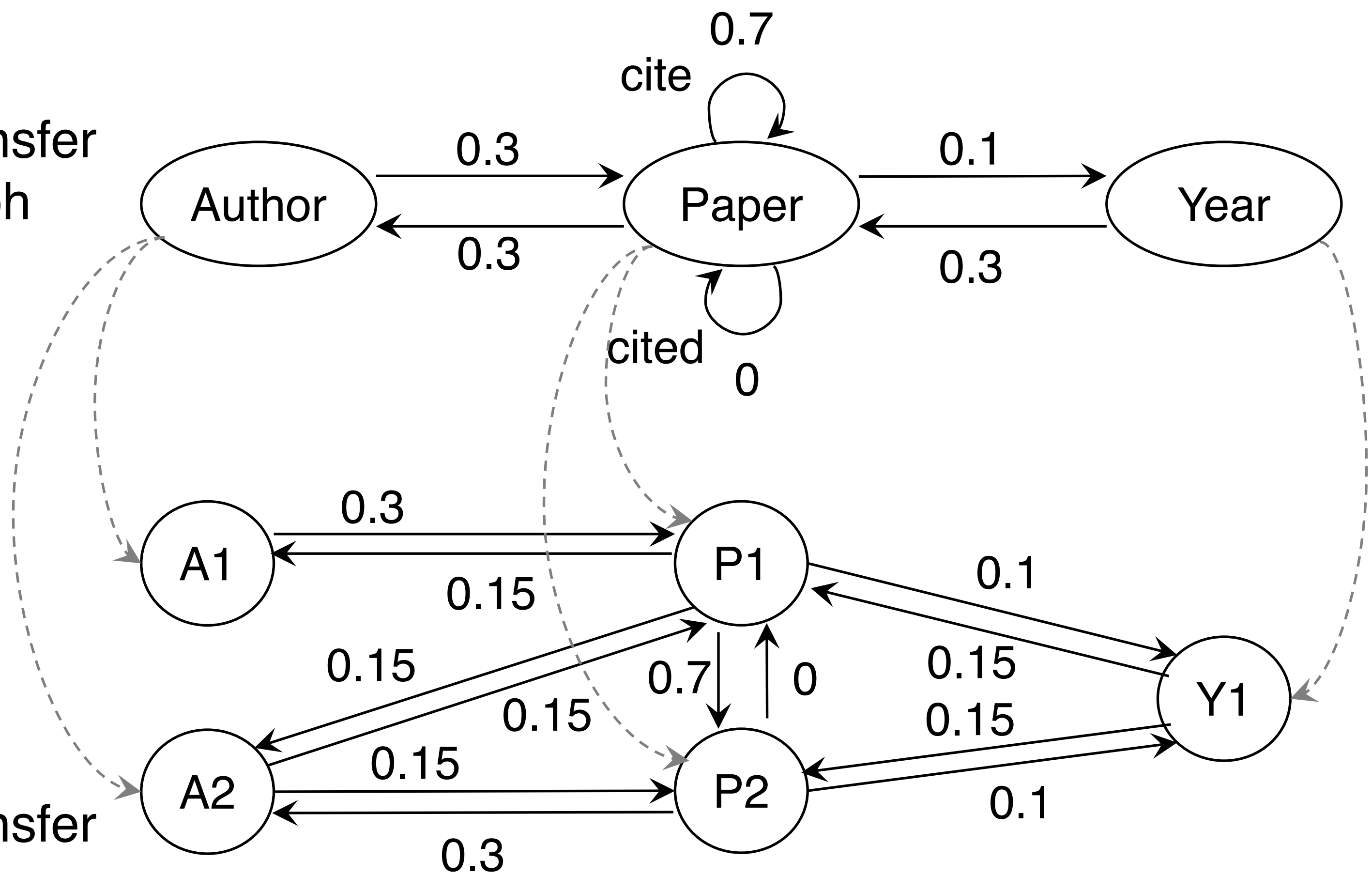
Objective and Approach

- Objective: Ranking quality improvement
- Approach
  - ObjectRank-based ranking [4]
    - Heterogeneous kinds of entities in LD
      - e.g., Locations, Events, Person, etc.
    - More flexible than PageRank
      - Different relationships b/w entity types can have different authority transfer rates.
      - Appropriate rates lead good ranking results [4].
  - Issue
    - Appropriate setting of authority transfer rates.

[4] Balmin, A., Hristidis, V., Papakonstantinou, Y.: ObjectRank: Authority-Based Keyword Search in Databases. In: VLDB 2004. pp. 564–575 (2004)

Graphs in ObjectRank [4]

Authority transfer <u>schema</u> graph



Authority transfer <u>data</u> graph

[4] Balmin, A., Hristidis, V., Papakonstantinou, Y.: ObjectRank: Authority-Based Keyword Search in Databases. In: VLDB 2004. pp. 564–575 (2004)

Calculation in ObjectRank [4]

Global ObjectRank
(Precomputed)

$$\mathbf{r}_g^{(t+1)} = dA\mathbf{r}_g^{(t)} + \frac{1-d}{|O|}\mathbf{e}$$

Query-specific ObjectRank
(Compute when query comes)

$$\mathbf{r}_q^{(t+1)} = dA\mathbf{r}_q^{(t)} + \frac{1-d}{|S(q)|}\mathbf{s}$$

Overall ObjectRank scores for given query *q*
(u is weighing parameter)

$$\mathbf{r} = \mathbf{r}_g \circ (\mathbf{r}_q)^u$$

[4] Balmin, A., Hristidis, V., Papakonstantinou, Y.: ObjectRank: Authority-Based Keyword Search in Databases. In: VLDB 2004. pp. 564–575 (2004)

How to apply ObjectRank?

1. Schema graph construction

```
SELECT distinct ?class
WHERE{?s rdf:type ?class}
```

Vertices

```
ASK{?s ?predicate ?d.
  ?s rdf:type <c1>. ?d rdf:type <c2>.}
```

Edges

2. Data graph construction

```
SELECT distinct ?s ?d
WHERE{?s ?predicate ?d. ?s rdf:type <c1>. ?d rdf:type <c2>.}
```

Vertices and edges

3. Calculate ObjectRank scores

Weights on the Graphs

- No evidential design principle
  - Design principle of graphs in ObjectRank is highly dependent on application scenario.
  - No research has clear principle for design principle for **keyword search on LD**.
- Ideal weights are hardly defined.

➔ Heuristic determination is reasonable.

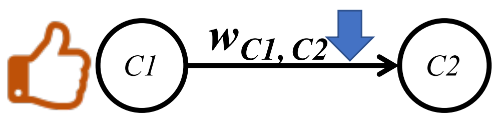
Proposed Weight Learning

Idea: employing human judgements on search results

- Approach: relevance feedback
  - Input: relevance judgements on (top-k) search results
  - Output: modified edge weights on schema graph
    - Afterward, ObjectRank scores are re-calculated.
- Process
  1. Map judgements to classes of result entities.
     - Entities of same classes are also relevant.
  2. Modify weights (authority transfer rates) according to the judgements

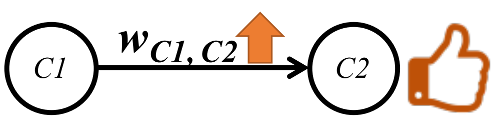
Weight Modification

Relevant classes



Decrease weights for outgoing edges.

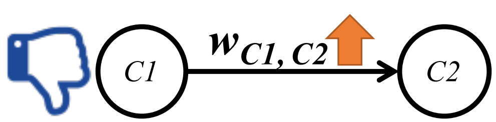
➔ leak less authority



Increase weights for incoming edges.

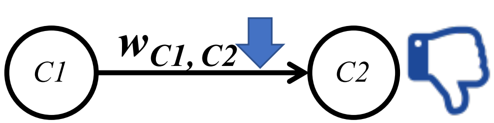
➔ gain more authority

Non-relevant classes



Increase weights for outgoing edges.

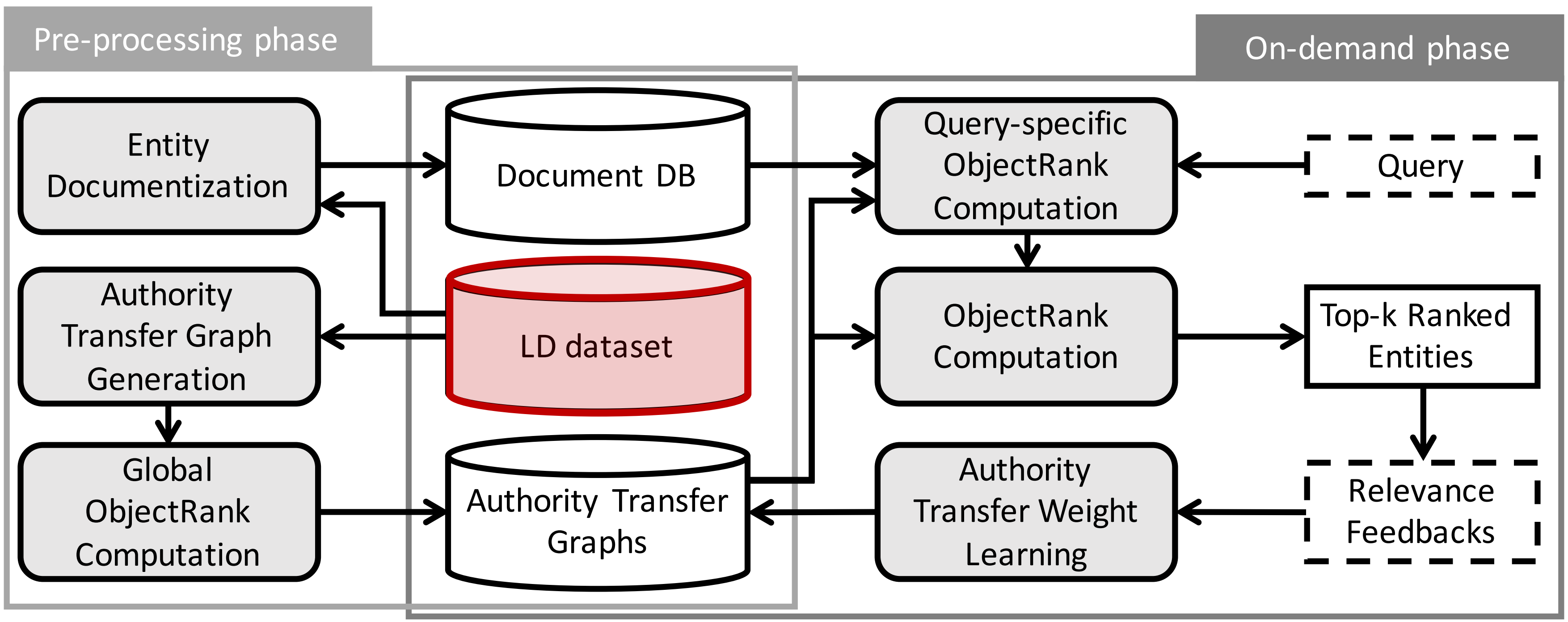
➔ leak more authority



Decrease weights for incoming edges.

➔ gain less authority

FORK: overall architecture



- Entity Documentization: prepare for keyword matching

```
SELECT ?value
WHERE { <entity> ?predicate ?value.
        FILTER(isLiteral(?value)) }
```

Experimental Evaluation

- Objective
  - Check if FORK successfully learns the weights.
  - Compare ranking quality with existing works.
- Dataset
  - Data: DBpedia 3.9
  - Entity search benchmark [6]
    - 61 keyword search queries are selected.
    - Answer entity list for each query.
- Measurement: Precision@10
  - Comparable with the benchmark results

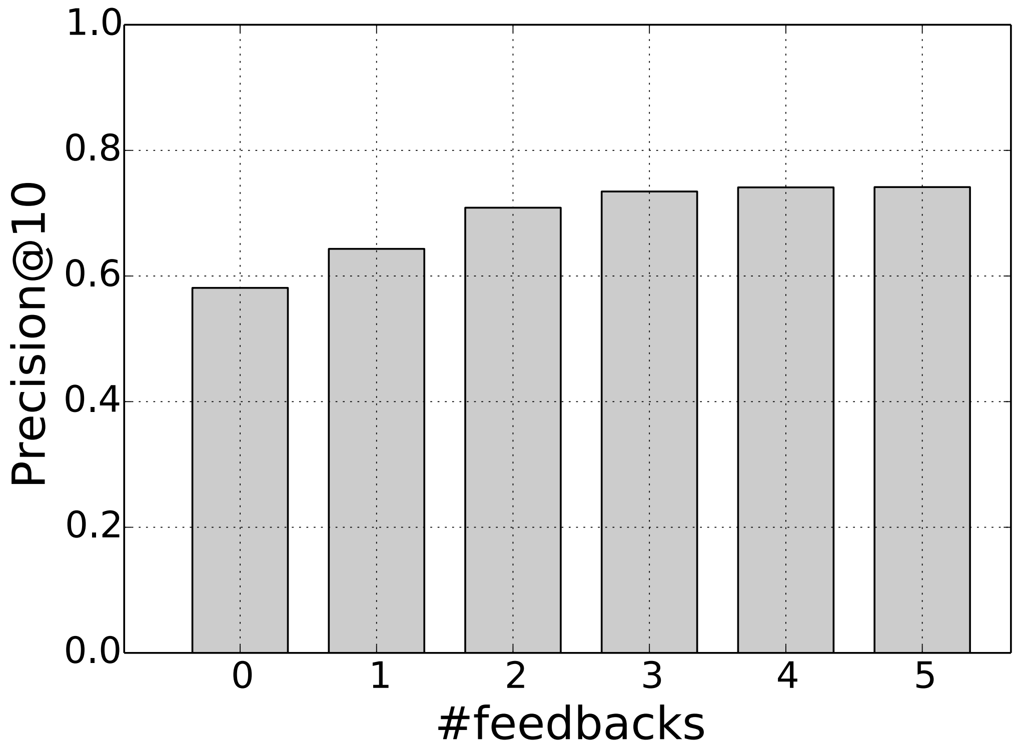
[6] Balog, K., Neumayer, R.: A Test Collection for Entity Search in DBpedia. In: SIGIR 2013. pp. 737–740 (2013)

Simulated Relevance Feedback

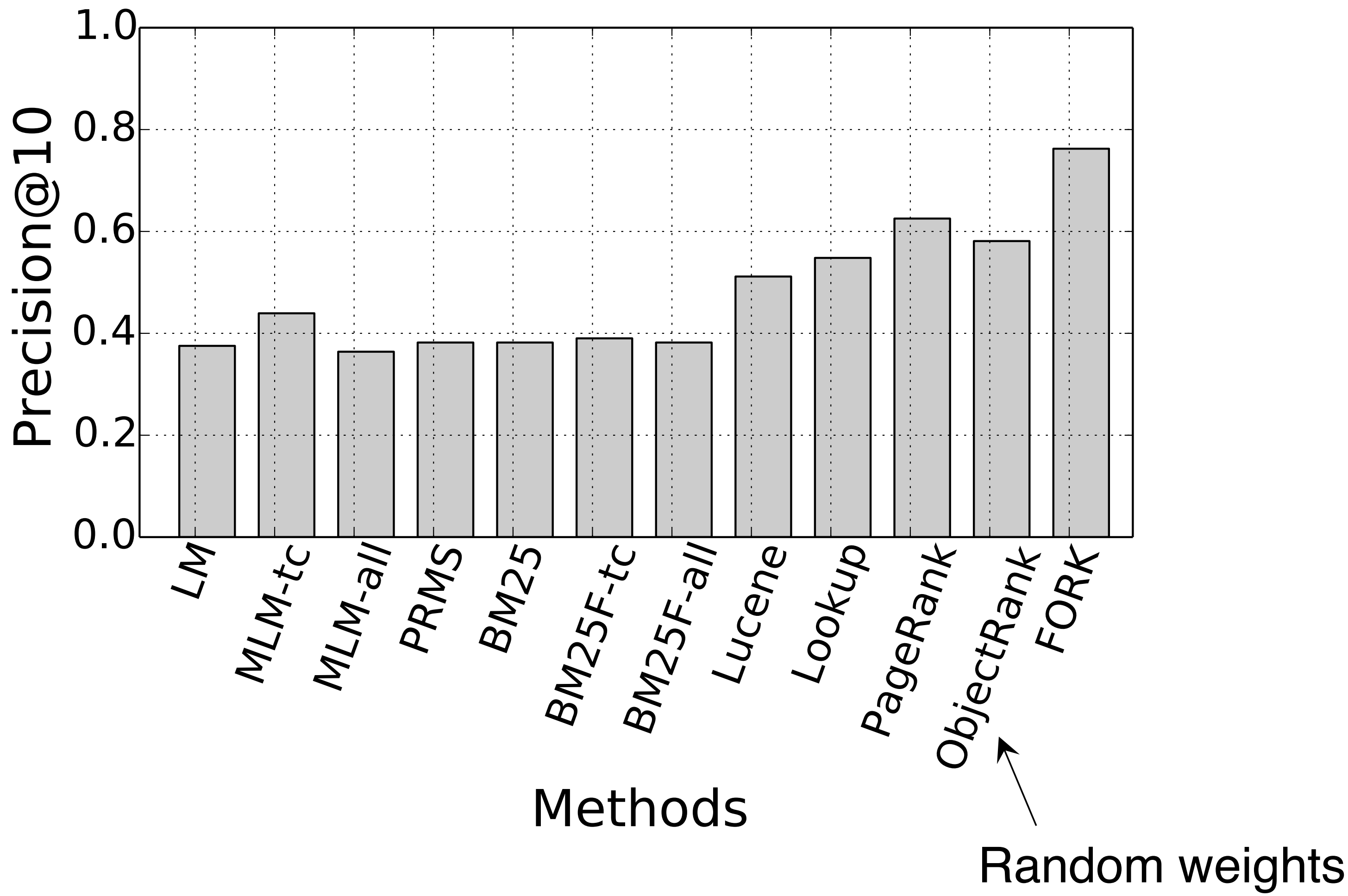
- Assumption
  - During a query, users do not change their mind.
- Procedure
  1. Given a query, FORK provides top-k answer list.
  2. Correct answers in the list are set to relevant, non-relevant otherwise.
  3. FORK learns weights and re-calculate the top-k answer list.
  4. Continue 2-3.

FORK improves Ranking.

- Observe accuracy change over feedbacks.



Best-learnt ObjectRank is the best.



Conclusion and Future Work

- FORK
  - ObjectRank-based keyword search over LD
  - Relevance feedback-based authority transfer weights learning
- Experiments
  - Ensure weights are learnt properly.
  - Best-learnt ObjectRank achieves the best accuracy.
- Future work
  - Employing keyword-based relevance feedback [20] for further improvement.

]20] Varadarajan, R., Hristidis, V., Raschid, L.: Explaining and Reformulating Authority Flow Queries. In: ICDE 2008. pp. 883–892 (2008)