Ranking Archived Documents for Structured Queries over Semantic Layers

Vaibhav Kasturia, Pavlos Fafalios, Wolfgang Nejdl

L3S Research Center, University of Hannover, Germany

{kasturia,fafalios,nejdl}@l3s.de

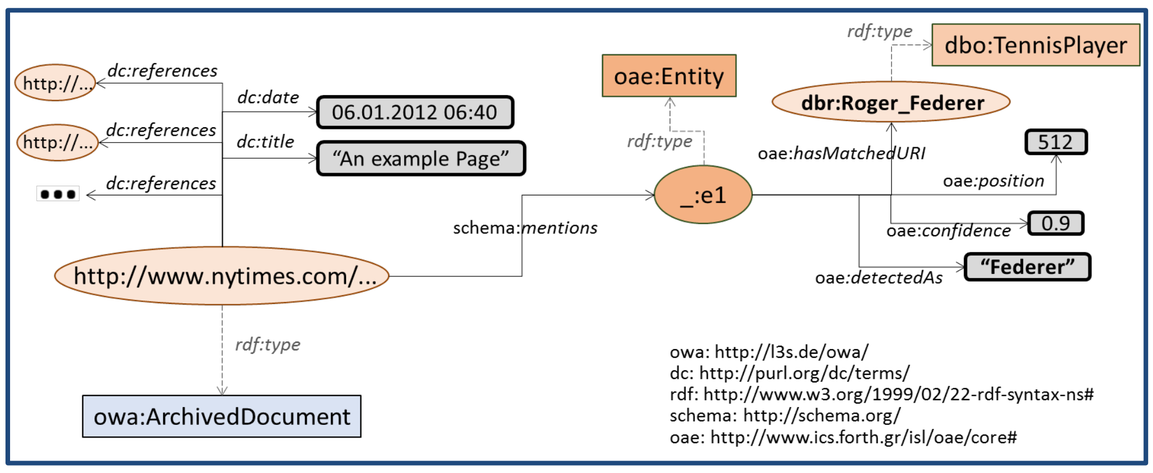
1. Motivation

- How to explore archives in a more **advanced** and **exploratory** way?
  - *Find documents discussing about a specific category of entities (e.g., philanthropists), or about entities sharing some characteristics (e.g., born in Germany before 1960)?*
- How to explore archives by integrating information from existing knowledge bases, like DBpedia?

2. Semantic Layer

- RDF repository describing **metadata** and **annotation** information for a collection of archived documents.
  - Allows running advanced, entity-centric SPARQL queries that combine metadata of the documents (e.g., publication date) and semantic information (e.g., mentioned entities)
  - More at: *Fafalios et al., "Building and Querying Semantic Layers for Web Archives", JCDL'17*

- Example for a **news article**:



- Example **SPARQL queries** over Semantic Layers

```
SELECT DISTINCT ?article WHERE {
  ?article dc:date ?date FILTER(year(?date) = 1990) .
  ?article schema:mentions ?entity1, ?entity2 .
  ?entity1 oae:hasMatchedURI dbr:Nelson_Mandela .
  ?entity2 oae:hasMatchedURI dbr:F._W._de_Klerk }
```

*Retrieve articles of **1990** discussing about **Nelson Mandela** and **F. W. de Klerk***

```
SELECT DISTINCT ?article WHERE {
  ?article dc:date ?date FILTER(year(?date) = 1990) .
  ?article schema:mentions ?entity .
  ?entity oae:hasMatchedURI ?entURI .
  ?entURI dc:subject dbc:State_Presidents_of_South_Africa }
```

*Retrieve articles of **1990** discussing about **state presidents of South Africa***

3. The Problem

- The results returned by a SPARQL query:
  - *can be numerous*
  - *all equally match the query*
- How to rank them for identifying and promoting the most important ones?
  - What makes an archived document important for a given query?

4. Related Work

- **Ranking of archived documents** (for free-text queries)
  - Time-aware Retrieval and Ranking [Kanhabua and Anand, 2016]
  - Tempas [Holzmann and Anand, 2016], HistDiv [Singh et al., 2016]
  - Works by Kanhabua et al. (2016), Vo et al. (2016)
- **Ranking in knowledge graphs**
  - Learning to rank for RDF entity search [Dali et al., 2012]
  - Swoogle [Ding et al., 2005], SemRank [Anyanwu et al., 2005]
  - NAGA [Kasneci et al., 2008], DING [Delbru et al., 2010],
  - ReconRank [Hogan et al., 2006], Noc-order [Graves et al., 2008]
- **Our approach:** Ranking archived documents for structured queries in knowledge graphs
  - Availability of metadata and entity annotations
  - No access to full contents!

5. Problem Definition

- **Ranking Documents for Structured Queries over Semantic Layers**
  - Consider a **semantic layer** over a collection of **archived documents D** published within a set of **time periods T** of fixed granularity (e.g., day), and a set of **entities E** mentioned in documents of D.
  - Given a **SPARQL query Q** requesting documents from D published within a **time period** $T_Q \subseteq T$ and related to one or more **Entities of Interest** (EoI) $E_Q \subseteq E$ with logical AND (mentioning all EoI) or OR (mentioning at least one EoI) semantics, the **problem** is how to rank the returned documents $D_Q \subseteq D$ that match Q.

6. Baseline Probabilistic Modeling

- What makes an archived document **important** for one or more entities of interest (EoI)?
  - **Relativeness:** the document should talk about the EoI (as its main topic)
  - **Timeliness:** the document should have been published in an important (for the EoI) time period
  - **Relatedness:** the document should discuss the relation of the EoI with other important (for the EoI) entities

- **Relativeness**
  - *Consider the frequency of the EoI in the document d*

$$score^f_\wedge(d) = \frac{\sum_{e \in E_Q} count(e,d)}{\sum_{e' \in ents(d)} count(e',d)}$$ *"AND" (conjunctive) semantics*

$$score^f_\vee(d) = \frac{\sum_{e \in E_Q} count(e,d)}{\sum_{e' \in ents(d)} count(e',d)} \cdot \frac{|ents(d) \cap E_Q|}{|E_Q|}$$ *"OR" (disjunctive) semantics*

  - *Probability to select a document given only the query entities*

$$P(d|E_Q) = \frac{score^f(d)}{\sum_{d' \in D_Q} score^f(d')}$$

- **Timeliness**
  - *Consider the number of documents mentioning the EoI during time period t*

$$score^t_\wedge(t) = \frac{|docs(t) \cap D_Q|}{|D_Q|}$$ *"AND" (conjunctive) semantics*

$$score^t_\vee(t) = \frac{|docs(t) \cap D_Q|}{|D_Q|} \cdot N(E_Q,t)$$ *where*

$$N(E_Q,t) = \frac{\sum_{d \in docs(t) \cap D_Q} \frac{|ents(d) \cap E_Q|}{|E_Q|}}{|docs(t) \cap D_Q|}$$ *Avg. percentage of EoI in articles of t*

  - *Probability to select a document given only its publication date*

$$P(d|T_Q) = \frac{score^t(t_d)}{\sum_{d' \in D_Q} score^t(t_{d'})}$$

- **Relatedness**
  - *Consider the number of co-occurrences of e with the EoI in important time periods*
  - *Avoid over-emphasizing common and general entities*

$$score^r_\wedge(e) = idf_\wedge(e) \cdot \sum_{t \in T_Q} \frac{|docs(t) \cap D_Q \cap docs(e)|}{|D_Q|}$$ *"AND" (conjunctive) semantics* *where*

$$idf_\wedge(e) = 1 - \frac{|docs(e) \cap (\cap_{e' \in E_Q} docs(e'))|}{|\cap_{e' \in E_Q} docs(e')|}$$

$$score^r_\vee(e) = idf_\vee(e) \cdot N(E_Q,e) \cdot \sum_{t \in T_Q} (N(E_Q,t) \cdot \frac{|docs(t) \cap D_Q \cap docs(e)|}{|D_Q|})$$ *"OR" (disjunctive) semantics* *where*

$$idf_\vee(e) = 1 - \frac{|docs(e) \cap (\cup_{e' \in E_Q} docs(e'))|}{|\cup_{e' \in E_Q} docs(e')|}$$

$$N(E_Q,e) = \frac{\sum_{d \in docs(e) \cap D_Q} \frac{|ents(d) \cap E_Q|}{|E_Q|}}{|docs(e) \cap D_Q|}$$ *Avg. percentage of EoI in articles of e*

  - *Probability to select a document given only other entities mentioned in the retrieved documents*

$$P(d|E_{D_Q}) = \frac{\sum_{e \in ents(d) \setminus E_Q} score^r(e)}{\sum_{d' \in D_Q} \sum_{e' \in ents(d') \setminus E_Q} score^r(e')}$$

- **Joining the models:** $P(d|E_Q, t_d, E_{D_Q}) = \frac{P(d|E_Q)P(d|T_Q)P(d|E_{D_Q})}{\sum_{d' \in D_Q} P(d'|E_Q)P(d'|T_Q)P(d'|E_{D_Q})}$

7. Evaluation

- We create ground truth consisting of 28 queries (14 AND, 14 OR Semantics)
- Articles in ground truth are judged on a graded relevance score from 0 to 3 depending upon importance of article to query entities.
- NDCG for different rankings and their combinations was calculated

Ranking Model	Normalized Discounted Cumulative Gain (NDCG)							
	AND Semantics				OR Semantics			
	@5	@10	@20	end	@5	@10	@20	end
Random List	0.264	0.352	0.435	0.681	0.271	0.345	0.473	0.676
Relativeness [A]	0.437	0.490	0.595	0.786	0.399	0.434	0.572	0.732
Timeliness [B]	0.274	0.335	0.445	0.685	0.242	0.352	0.488	0.682
Relatedness [C]	0.352	0.434	0.574	0.743	0.457	**0.527**	**0.671**	**0.775**
[A]*[B]	0.490	0.518	0.611	0.796	0.456	0.470	0.601	0.753
[A]*[C]	0.466	0.518	0.618	0.794	0.469	0.497	0.620	0.760
[B]*[C]	0.403	0.471	0.559	0.743	0.486	0.517	0.665	0.772
[A]*[B]*[C]	**0.501**	**0.527**	**0.622**	**0.800**	**0.493**	0.520	0.624	0.771

NDCG for different ranking models and their combinations

8. Next Steps

- Evaluate a Random Walk with Restart (RWR) model for each query graph
  - Nodes consist of documents returned for SPARQL query, entities mentioned in the returned documents and entities of interest
  - Edge traversal possible between documents to their mentioned entities and vice-versa

Leibniz Universität Hannover

Forschungszentrum · L3S Research Center

ALEXANDRIA

http://alexandria-project.eu