Building and Querying Semantic Layers for Web Archives

Pavlos Fafalios, Vaibhav Kasturia, Helge Holzmann, Wolfgang Nejdl

fafalios@l3s.de

L3S Research Center, University of Hannover, Germany

Web Archives

- Valuable sources for research in many disciplines
- Comprehensive documentation of society
- + News Archives and Social Media Archives (non-versioned)

- Accessing Web Archives
    - Limited query and exploration capabilities
    - Difficult to integrate information and identify interesting parts
    - Laborious to derive interesting (aggregated) information

- Analysts want to see, compare and understand information about **entities**
    - Thus, calling for entity-centric exploration and analysis methods!

The 6 Motivating Questions

1 - Information exploration

- **How to explore web archives in a more advanced and exploratory way?**
- Find documents of a specific time period, discussing about a specific category of entities, or about entities sharing some characteristics

2 - Information integration

- **How to explore web archives by also integrating information from existing knowledge bases like DBpedia?**
- How to integrate information coming form multiple (web) archives?

The 6 Motivating Questions

3 – Information/Knowledge discovery

- **How to infer knowledge by exploiting the contents of a Web Archive?**
- Identify important time periods related to one or more entities
- Find out popular entities of a specific type in specific time periods

4 - Robustness in information change

- **How to explore web archives by automatically taking into account the change of entities over time?**
- Find documents without worrying about their correct reference

The 6 Motivating Questions

5 - Multilinguality

- **How to explore documents about entities independently of the document language** (and thus of the language of the entity mentions)?

6 - Interoperability

- **How to facilitate exploration of web archives by other systems and tools?**
- Expose information about web archives in the Web, in a standard and machine understandable format
- Identify interesting parts for further analysis, easily and fast

Existing Approaches

- **Exploring** Web Archives
  - Search services provided by Internet Archive (Wayback Machine), Memento (Time Travel), Archive-It, Portuguese Web Archive
  - Research works: [Holzmann and Anand, 2016], [Kanhabua et al., 2016], [Vo et al., 2016], [Jackson et al., 2016], [Singh et al., 2016]

- **Profiling** Web Archives
  - Improve effectiveness of query routing strategies in distributed archive search [AlSum et al., 2014], [Alam et al., 2015], [Bornand et al., 2016], [Alam et al., 2016]

- **Analyzing** Web Archives
  - Frameworks for distributed analysis of Web Archives
  - ArchiveSpark [Holzmann et al., 2016], Warcbase [Lin et al., 2014]

Our approach: building and querying **Semantic Layers**

*for **profiling** and **exploring** web archives*

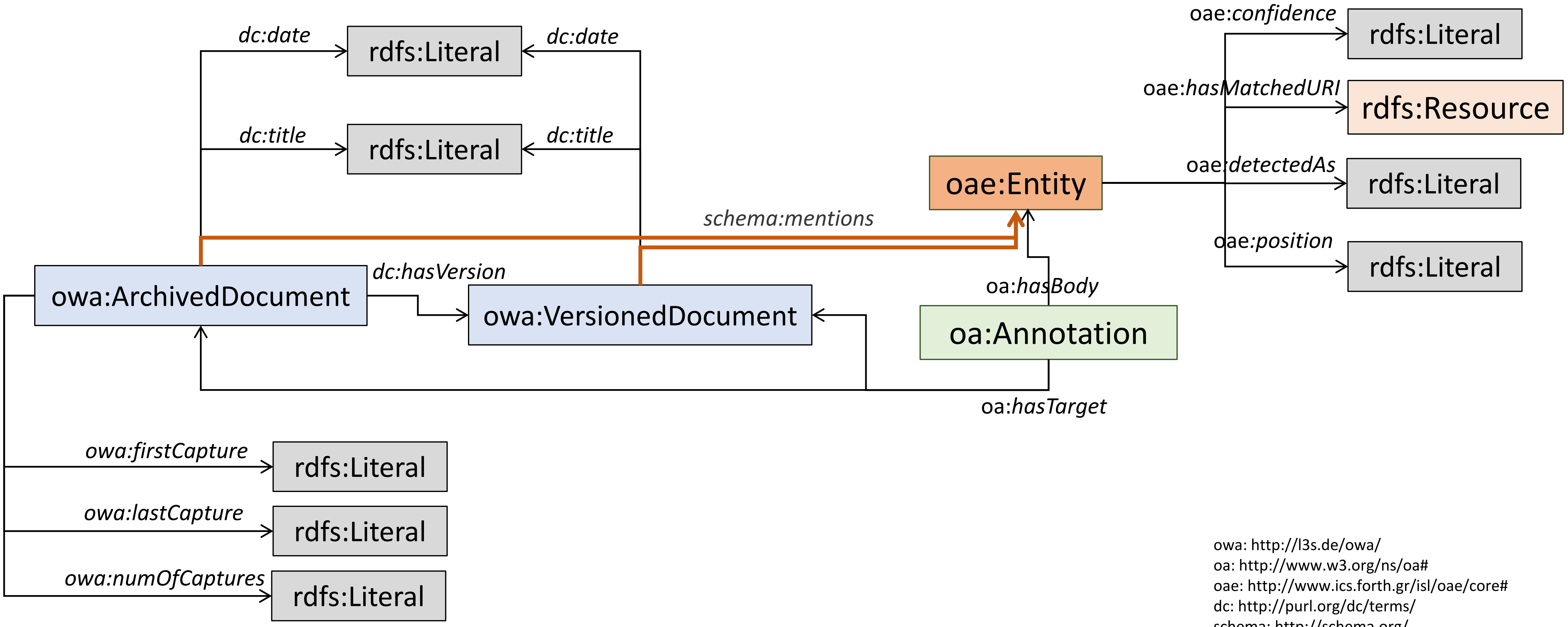
- Semantic Layer:
  - An **RDF** repository (RDF graph) of **structured data** (triples) about an archived collection of documents
- It allows:
  - Describing useful **metadata** information about the archived documents
  - **Annotating** the documents with semantic information, like entities, events and concepts mentioned in the documents
  - **Publishing** all this data on the Web (as Linked Data or though a SPARQL endpoint)
- Why?
  - Advanced, entity-centric query capabilities (using SPARQL)
  - Real-time data integration
  - Directly accessible and exploitable by other systems and tools

Outline of next slides

- Building Semantic Layers
  - RDF/S **data model**: “Open Web Archive”
  - Construction **process**
  - Open source **framework**: “ArchiveSpark2Triples”
- Querying Semantic Layers
  - Case Studies and Query Capabilities
  - Evaluation
  - Problems and Limitations
- Ranking in Semantic Layers
  - How to rank the results returned by a SPARQL query?
  - Baseline Probabilistic Modeling
- Conclusion and Future Work

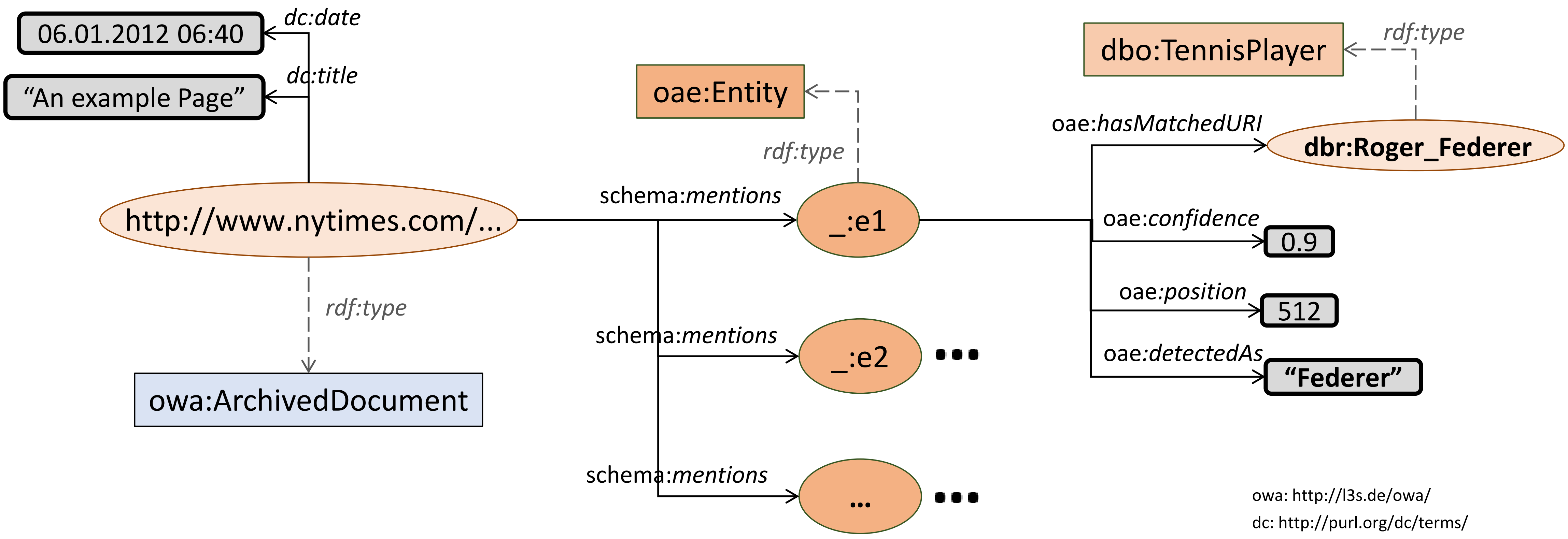
Building Semantic Layers

RDF/S data model: *Open Web Archive* http://l3s.de/owa/



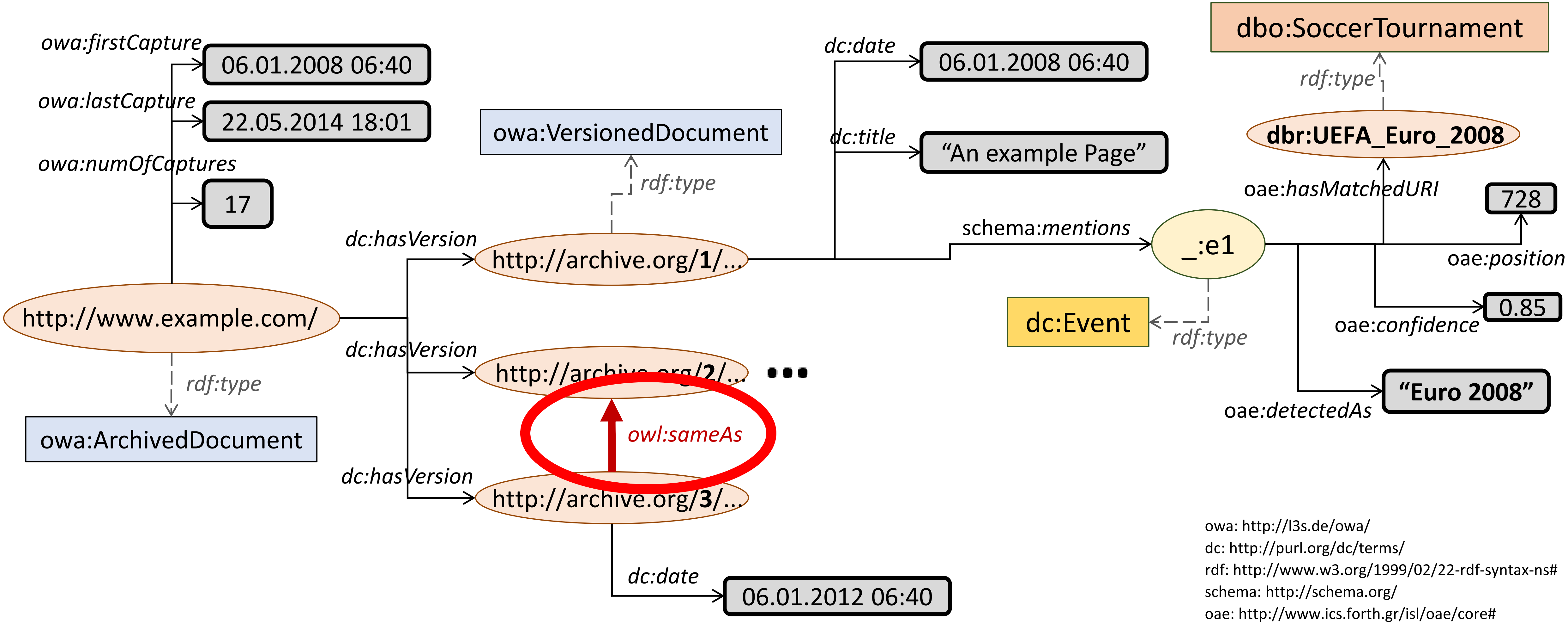
owa: http://l3s.de/owa/
oa: http://www.w3.org/ns/oa#
oae: http://www.ics.forth.gr/isl/oae/core#
dc: http://purl.org/dc/terms/
schema: http://schema.org/
rdfs: http://www.w3.org/2000/01/rdf-schema#

Open Web Archive – Example of Non-versioned Web Page



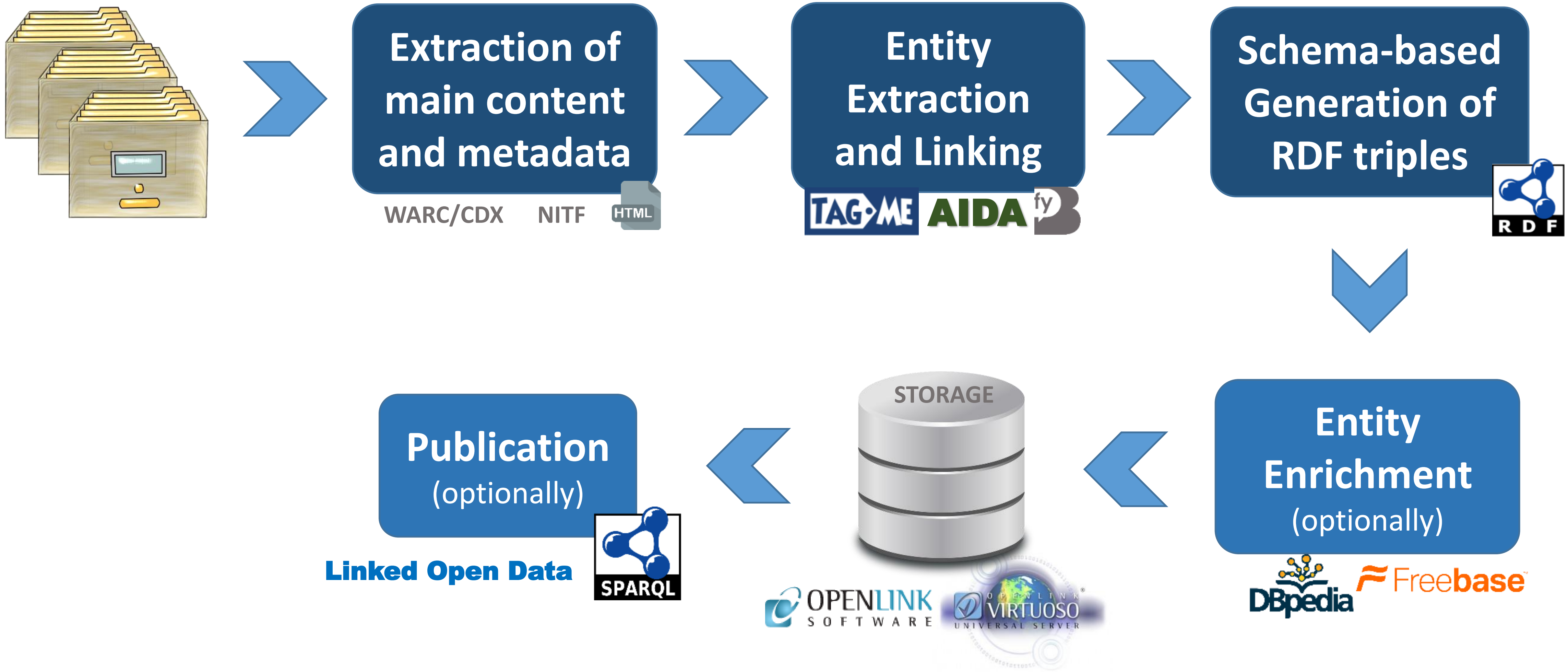
owa: http://l3s.de/owa/
dc: http://purl.org/dc/terms/
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
schema: http://schema.org/
oae: http://www.ics.forth.gr/isl/oae/core#

Open Web Archive – Example of Versioned Web Page



owa: http://l3s.de/owa/
dc: http://purl.org/dc/terms/
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
schema: http://schema.org/
oae: http://www.ics.forth.gr/isl/oae/core#
owl:http://www.w3.org/2002/07/owl#

The construction process



Extraction of main content and metadata

Entity Extraction and Linking

Schema-based Generation of RDF triples

Entity Enrichment (optionally)

Publication (optionally)

Apache Spark framework: ***ArchiveSpark2Triples***

https://github.com/helgeho/ArchiveSpark2Triples

- Based on **ArchiveSpark** framework https://github.com/helgeho/ArchiveSpark
  - Programming framework for efficiently analyzing large web archives
  - Unified data model storing records in an hierarchical way
  - Very fast filtering, grouping and sorting based on metadata
  - Support of external modules, called **enrich functions**
- **ArchiveSpark2Triples**
  - Extension that automates the construction of a semantic layer
  - Output: Notation3 (N3) files
  - Customizable assignment of URLs and vocabularies to use → Extendable!
  - Extraction of entities using **Yahoo FEL** entity linking tool
    - Enrich function available under **FEL4ArchiveSpark** https://github.com/helgeho/FEL4ArchiveSpark

Apache Spark framework: ***ArchiveSpark2Triples***

https://github.com/helgeho/ArchiveSpark2Triples

- **Efficiency**
  - Very efficient for operations that only rely on metadata information (in CDX files)
  - Actual contents are accessed only for applying **enrich functions** to the versioned documents that do **not** constitute duplicates of older versions
  - **Entity extraction** is the most expensive task
  - Actual time for the entire workflow depends on:
    - Dataset size and nature of data
    - Computing infrastructure and available resources
    - Indicatively: 24 hours for creating a semantic layer for a web archive of 9 million web pages (474.6 GB of compressed WARC and CDX files)
      - Hadoop cluster of 25 compute nodes, 268 CPU cores, 2,688 GB RAM, 110 executors in parallel most of the time

Querying Semantic Layers

Case Studies

- **Web Archive** (versioned)
  - **Occupy Movement** 2011/2012 collection (provided by **Archive-It**)
  - >9M captures of >3M URLs
  - **URIs for versions:** links to the collection's Wayback Machine provided by ArchiveIt
  - >10B triples; >1.3M **same-as** properties; 939,960 distinct entities
- **News Archive** (non-versioned)
  - New York Times Annotated Corpus
  - ≈ 1.5M articles published by NYT between 1987 and 2007
  - >195M triples; 856,283 distinct entities
- **Social Media Archive**
  - ≈ 1.4M tweets posted in 2016 by 469 twitter accounts of USA newspapers
  - Metadata: *creation date, username, favorite count, retweet count*
  - >19 million triples; 146,854 distinct entities

Available at: http://l3s.de/owa/semanticlayers/

Case Studies – Query Capabilities

- Information Exploration and Integration
  - Articles of **summer 1989** mentioning **New York lawyers born in Brooklyn** (and for each lawyer show its **birth date** and a **description in French**)

Semantic Layer over NYT articles

```
SELECT DISTINCT ?article ?title ?date ?nylawyer ?birthDate ?abstr WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    ?nylawyer dc:subject dbc:New_York_lawyers ;
              dbo:birthPlace dbr:Brooklyn .
    OPTIONAL {
       ?nylawyer dbo:birthDate ?birthDate ;
                 dbo:abstract ?abstr FILTER(lang(?abstr)='fr') } }

  ?article dc:date ?date FILTER(?date >= "1989-06-01"^^xsd:date &&
                                ?date <= "1989-08-31"^^xsd:date)
  ?article schema:mentions ?entity .
  ?entity oae:hasMatchedURI  ?nylawyer .
  ?article dc:title ?title } ORDER BY ?nylawyer
```

44 lawyers

Result: 184 articles

Case Studies – Query Capabilities

- Information Exploration and Integration
  - **Popular tweets** (with >50 re-tweets) posted during the **summer of 2016**, mentioning **basketball players** of the NBA team **Los Angeles Lakers**

Semantic Layer
over tweets collection

```
SELECT DISTINCT ?tweet ?count ?date ?entityUri WHERE {
  SERVICE <http://dbpedia.org/sparql> {
     ?entityUri dc:subject dbc:Los_Angeles_Lakers_players }
  ?t a tw:Tweet ; dc:date ?date FILTER(?date>="2016-06-01"^^xsd:dateTime &&
                                     ?date<="2016-08-31"^^xsd:dateTime)
  ?t tw:retweetCount ?count FILTER (?count > 50) .
  ?t schema:text ?tweet ; schema:mentions ?entity .
  ?entity oae:hasMatchedURI ?entityUri }
```

7 players

Result: 14 tweets

Case Studies – Query Capabilities

- Information/Knowledge Discovery
  - Most discussed **journalists** in **Occupy Movement** collection

Semantic Layer
over Occupy Movement

```
SELECT ?journ (COUNT(DISTINCT ?page) AS ?num) WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    ?journ a yago:Journalist110224578 }
  ?page a owa:ArchivedDocument ; dc:hasVersion ?version .
  ?version schema:mentions ?entity .
  ?entity oae:hasMatchedURI  ?journ
} GROUP BY ?journ ORDER BY DESC(?num)
```

- Ralph Nader
- Chris Hedges
- Dylan Ratigan

Case Studies – Query Capabilities

- Information/Knowledge Discovery
  - Number of articles per year mentioning **Nelson Mandela**

Semantic Layer over NYT articles

```
SELECT ?year (COUNT(DISTINCT ?article) AS ?num) WHERE {
  ?article dc:date ?date ; schema:mentions ?entity .
  ?entity oae:hasMatchedURI dbr:Nelson_Mandela
} GROUP BY (year(?date) AS ?year) order by ?year
```

1988	54
1989	80
1990	**509**
1991	193
1992	142
1993	157
1994	**282**
1995	172
1996	130

1990: released from prison

1994: become president
(South African multiracial general election)

Case Studies – Query Capabilities

- Information/Knowledge Discovery
  - Most discussed **Drugs** in **1987 (left) and 1997 (right)**

Semantic Layer over NYT articles

```
SELECT DISTINCT ?drug (count(DISTINCT ?article) as ?numOfArticles) WHERE {
  SERVICE <http://dbpedia.org/sparql> { ?drug a dbo:Drug }
  ?article dc:date ?date FILTER(year(?date) = "1987") .
  ?article schema:mentions ?entity . ?entity oae:hasMatchedURI  ?drug .
} GROUP BY ?drug ORDER BY DESC(?numOfArticles)
```

Drug	**Num of articles (1987)**
http://dbpedia.org/resource/Cocaine	778
http://dbpedia.org/resource/Heroin	248
http://dbpedia.org/resource/Aspirin	63
http://dbpedia.org/resource/Zidovudine	53
http://dbpedia.org/resource/Furosemide	53

Drug	**Num of articles (1997)**
http://dbpedia.org/resource/Cocaine	462
http://dbpedia.org/resource/Heroin	275
http://dbpedia.org/resource/**Nicotine**	125
http://dbpedia.org/resource/**Fluoxetine**	61
http://dbpedia.org/resource/**Caffeine**	58

Case Studies – Query Capabilities

- Robustness and Multilinguality
  - Extracted entities are assigned unique URIs
    - Different mentions of an entity are assigned the same unique URI (multilingual name variants)
  - For multilinguality, the entity linking system should support the identification of entities in different languages
  - Time-awareness and correct disambiguation of the entity linking system affect the results!

Case Studies – Query Capabilities

- Other exploitation scenarios
  - **Time-Aware Entity Recommendation** (based on entity co-occurrences)

```
SELECT ?politician (count(distinct ?article) as ?num) WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    ?politician a dbo:Politician }
  ?article dc:date ?date FILTER(?date >= "2007-06-01"^^xsd:date && ?date <= "2007-08-30"^^xsd:date) .
  ?article schema:mentions ?entity .  ?entity oae:hasMatchedURI dbr:Barack_Obama .
  ?article schema:mentions ?entityPolit .
  ?entityPolit oae:hasMatchedURI ?politician FILTER (?politician != dbr:Barack_Obama)
} GROUP BY ?politician ORDER BY DESC(?num) LIMIT 5
```

  - Identification of **Similar** or **Identical documents**

```
SELECT ?article2 (count(?entUri2) as ?numOfCommon) WHERE {
  nyt:9504E4D71530F932A35755C0A9619C8B63 schema:mentions ?entity1 .
  ?entity1 oae:hasMatchedURI ?entUri1 .
  ?article2 schema:mentions ?entity2 FILTER (?article2 != nyt:9504E4D71530F932A35755C0A9619C8B63) .
  ?entity2 oae:hasMatchedURI ?entUri2 FILTER(?entUri2 = ?entUri1)
} GROUP BY ?article2 ORDER BY DESC(?numOfCommon) LIMIT 5
```

Evaluation

- Objectives:
  - to show that for a bit more complex information needs, keyword-based search systems return poor results
    - **Thus, calling for new, more advanced information seeking strategies!**
  - to identify possible problems and limitations of our approach

- Setup
  - Archived collection: **NYT corpus**
  - 20 information needs of *exploratory nature*
    - each one requesting documents of a **specific time period**, related to some **entities of interest**
  - Each information need corresponds to **one SPARQL query** and **one free-text query**
  - Example of information need:
    - *"find articles of* ***June 2010*** *discussing about* ***African-American film producers****"*
    - Corresponding free text query: *"African-American film producers"* (we manually specify the data range to each system)

Evaluation

- Comparison:
  - SPARQL query on **Semantic Layer**
  - Free-text query on **Google News** (appending the string “ `site:nytimes.com`”)
  - Free-text query on **HistDiv** [Singh et al., 2016]
    - Time-aware and diversity-oriented approach
- Manual evaluation of all returned results
  - Considering only articles existing in all systems!

Evaluation – Results

Big number of disambiguation errors

	Information need	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
SPARQL	Num of results	27	34	37	16	11	14	**18**	8	**11**	15	**15**	12	13	16	15	12	15	13	16	15
	Num of **relevant** results	27	27	33	16	9	14	**2**	8	**1**	14	**1**	8	13	15	9	10	13	11	15	15
GOOGLE NEWS	Num of results	8	1	0	0	0	1	1	1	0	0	0	0	0	2	0	6	1	1	1	1
	Num of **relevant** results returned by SPARQL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0
	Num of **relevant** results **<u>not returned by SPARQL</u>**	0	**1**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**
HISTDIV	Num of results	0	3	1	0	0	0	0	4	0	0	0	0	0	0	0	25	2	0	0	0
	Num of **relevant** results returned by SPARQL	0	**2**	0	0	0	0	0	0	0	0	0	0	0	0	0	**3**	**1**	0	0	0
	Num of **relevant** results **<u>not returned by SPARQL</u>**	0	**1**	0	0	0	0	0	**3**	0	0	0	0	0	0	0	**3**	0	0	0	0

Information needs and full results available at: http://l3s.de/owa/semanticlayers/SemLayerEval.zip

Problems and Limitations

- False positive:
    - A returned document is <u>not relevant</u> due to disambiguation error
- False negative
    - A relevant document is <u>not returned</u> because:
        - An entity of interest was not recognized by the entity linking tool
        - Disambiguation error
        - **Confidence score** of extracted entity of interest below used threshold
- Temporal inconsistency
    - Change of entity properties
    - Completeness and freshness of used knowledge bases

Efficiency of Query Answering

- Query execution time depends on:
  - Efficiency of triplestore and server
  - Query itself
    - Use of costly operators (like FILTER, OPTIONAL and SERVICE)
- Indicatively:
  - ≈400 ms is the average execution time of the 20 queries used in our evaluation
    - Min: 56 ms, max: 2.4 sec
    - They all make use of SERVICE operator for querying DBpedia
    - Experiments on Openlink Virtuoso server installed in a modest personal computer (Intel Core i5, 8GB RAM)

Ranking in Semantic Layers

Ranking in Semantic Layers

- The results returned by a SPARQL query:
  - can be numerous
  - all equally match the query
- How can we identify and promote the most important results?
- Focus on **ranking documents**
  - Two main cases:
    - Conjunctive (AND) semantics: requesting documents containing **ALL** query entities
    - Disjunctive (OR) semantics: requesting documents containing **AT LEAST ONE** of the query entities

"AND" semantics

```
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***

"OR" semantics

```
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***

Ranking in Semantic Layers

- Related Works
    - 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 in knowledge graphs
        - Availability of metadata and entity annotations
        - No access to full contents!

Ranking in Semantic Layers

- What makes an archived document important given a **time period** and one or more **query entities**?

- **Relativeness**
  - the document should talk about the query entities as its main topic
- **Timeliness**
  - the document should have been published in a time period which is important for the query entities
- **Relatedness**
  - the document should discuss the relation of the query entities with other entities (that are important for the query entities in important time periods)

Baseline Probabilistic Modeling

- **Relativeness**: the probability to pick a document based (only) on the *query entities* it mentions

$$P(d|E_Q) = \frac{score^f(d, E_Q)}{\sum_{d' \in D_Q} score^f(d', E_Q)} \quad \text{where:} \quad score^f(d, E_Q) = \frac{\sum_{e \in E_Q} count(e, d)}{\sum_{e' \in ents(d)} count(e', d)}$$

- **Timeliness**: the probability to pick a document based (only) on its *publication date*

$$P(d|t_d) = \frac{score^t(t_d)}{\sum_{d' \in D_Q} score^t(t_{d'})} \quad \text{where:} \quad score^t(t) = \frac{|docs(t) \cap D_Q|}{|D_Q|}$$

- **Relatedness**: the probability to pick a document based (only) on *other entities* it mentions (no query entities)

$$P(d|E_{D_Q}) = \frac{\sum_{e \in ents(d) \setminus E_Q} score^r(e)}{\sum_{d' \in D_Q} \sum_{e' \in ents(d') \setminus E_Q} score^r(e')} \quad \text{where:} \quad score^r(e) = idf_{\wedge}(e) \cdot \sum_{t \in T_Q} \frac{|docs(t) \cap D_Q \cap docs(e)|}{|D_Q|}$$

Baseline Probabilistic Modeling - Evaluation

- No existing ground truth dataset
- Manual evaluation of the results returned by 28 queries
  - 14 of “AND” semantics (7 of single entity + 7 of 2-3 entities)
  - 14 of “OR” semantics (7 of 2-3 entities + 7 of entity category)

- Graded relevance scale:
  - **Score 3**: The topic of the document is about the query entities
  - **Score 2**: The topic of the document is **not** about the query entities, however the query entities are important for the document context
  - **Score 1**: The topic of the document is **not** about the query entities, however the query entities are related to the document context
  - **Score 0**: The document has almost nothing to do with the query entities

Baseline Probabilistic Modeling - Evaluation

- All queries – NDCG scores

K	RANDOM RANKING	RELATIVENESS [A]	TIMELINESS [B]	RELATEDNESS [C]	[A]*[B]	[A]*[C]	[B]*[C]	[A]*[B]*[C]
5	0.27	0.42	0.26	0.40	0.47	0.47	0.44	**0.50**
10	0.34	0.46	0.33	0.48	0.49	0.51	0.49	**0.52**
20	0.45	0.58	0.47	0.62	0.61	0.62	0.61	**0.62**
ALL	0.68	0.76	0.69	0.76	0.77	0.78	0.76	**0.79**

- ➢ Improvement of joined model compared to RELATIVENESS is **statistically significant** (paired t-test, $p \leq 0.05$)
- ➢ Relativeness can be considered a baseline model since it relies on term (entity) frequency which is a classic numerical statistic reflecting the importance of the term (entity) to a document

Baseline Probabilistic Modeling - Evaluation

- Queries of “AND” semantics only – NDCG scores

K	RANDOM RANKING	RELATIVENESS [A]	TIMELINESS [B]	RELATEDNESS [C]	[A]*[B]	[A]*[C]	[B]*[C]	[A]*[B]*[C]
5	0.26	0.44	0.27	0.35	0.49	0.47	0.40	**0.50**
10	0.33	0.49	0.33	0.43	0.52	0.52	0.47	**0.53**
20	0.43	0.60	0.45	0.57	0.61	0.62	0.56	**0.62**
ALL	0.68	0.79	0.69	0.74	0.80	0.79	0.74	**0.80**

- Queries of “OR” semantics only – NDCG scores

K	RANDOM RANKING	RELATIVENESS [A]	TIMELINESS [B]	RELATEDNESS [C]	[A]*[B]	[A]*[C]	[B]*[C]	[A]*[B]*[C]
5	0.27	0.40	0.24	0.46	0.46	0.47	0.49	**0.49**
10	0.34	0.43	0.32	**0.53**	0.47	0.50	0.52	0.52
20	0.47	0.57	0.49	**0.67**	0.60	0.62	0.67	0.62
ALL	0.68	0.73	0.68	**0.77**	0.75	0.76	0.77	**0.77**

Conclusion and Future Work

Conclusions

- **Data model** and **framework** for constructing Semantic Layers for (web) archives
- Semantic Layers allow:
  - ❖ **Exploring** web archives in more advanced and exploratory ways (entity-centric)
  - ❖ **Integrating** information (at query-execution time) coming from other semantic layers and knowledge bases
  - ❖ **Inferring** new knowledge that is very laborious and time-consuming to derive otherwise (just with one query)
  - ❖ Coping with common problems like **temporal reference variants** and **multilinguality**
  - ❖ Making the contents of web archives **machine understandable**
- Quality of results depends on quality of entity annotations
- Baseline Probabilistic Modeling for ranking results
  - Considering TIMELINESS and RELATEDNESS improves the results significantly

Future Work

- Development of **user-friendly interfaces** on top of Semantic Layers
  - Faceted Search and Exploration
  - Translation of free-text queries to SPARQL
- Ranking of SPARQL results
  - Experimentation with other ranking schemes (Random Walk with Restart)
- Cope with temporal inconsistencies
  - Use entity URIs that lead to old DBpedia descriptions?

Thank you

Questions/Suggestions/Comments?