Entity Representation and Retrieval from Knowledge Graphs

Alexander Kotov

Textual Data Analytics Lab, Department of Computer Science, Wayne State University

RuSSIR 2016

Overview

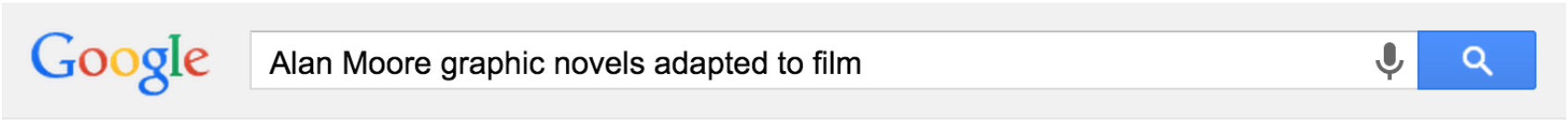
Entities and Entity Retrieval

- **Entities**: material objects or concepts that exist in the real world or fiction (e.g. people, books, conferences, colors etc.)
- Entities (named entities) are typically designated by proper nouns or proper noun phrases (e.g. Barack Obama)
- **Entity retrieval**: answering arbitrary information needs related to particular aspects of objects (entities), expressed in unconstrained natural language and resolved using a collection of entities [Pound, Mika et al., WWW'10]

Ad-hoc Entity Retrieval

- **Query**: keyword query corresponding to an entity name, description of property (properties) of the target entity or a set of entities
- "Telegraphic" queries – neither well-formed, nor grammatically correct sentences or questions
- **Results**: rank list of entities (entity representations) instead of or in addition to documents.

Ad-hoc Entity Retrieval



About 186,000 results (0.37 seconds)

Alan Moore - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Alan_Moore** ▾ Wikipedia ▾
For other people named **Alan Moore**, see **Alan Moore** (disambiguation). ... Frequently described as the best **graphic novel** writer in history, he has been called "one of the most important British writers of the ... 4 **Film adaptations**; 5 Personal life.

From Hell - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/From_Hell ▾ Wikipedia ▾
From Hell is a **graphic novel** by writer **Alan Moore** and artist Eddie Campbell, ... The comic was loosely **adapted** into a **film** of the same title, released in 2001.

Watchmen - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Watchmen ▾ Wikipedia ▾
For the 2009 **film adaptation**, see Watchmen (**film**). ... The series was created by writer **Alan Moore**, artist Dave Gibbons, and colorist John Higgins. .... Moore reasoned that MLJ **Comics**' Mighty Crusaders might be available for such a project, ...

Alan Moore - IMDb
www.imdb.com/name/nm0600872/ ▾ Internet Movie Database ▾
Alan Moore was born on November 18, 1953 in Northampton, England. ... Includes clips from GLORY DAZE, COMMUNITY and horror **film** URBAN ... 2009 Tales of the Black Freighter (Video short) (**graphic novel** "Watchmen" - uncredited).
Biography - Awards - Photo Gallery - Publicity

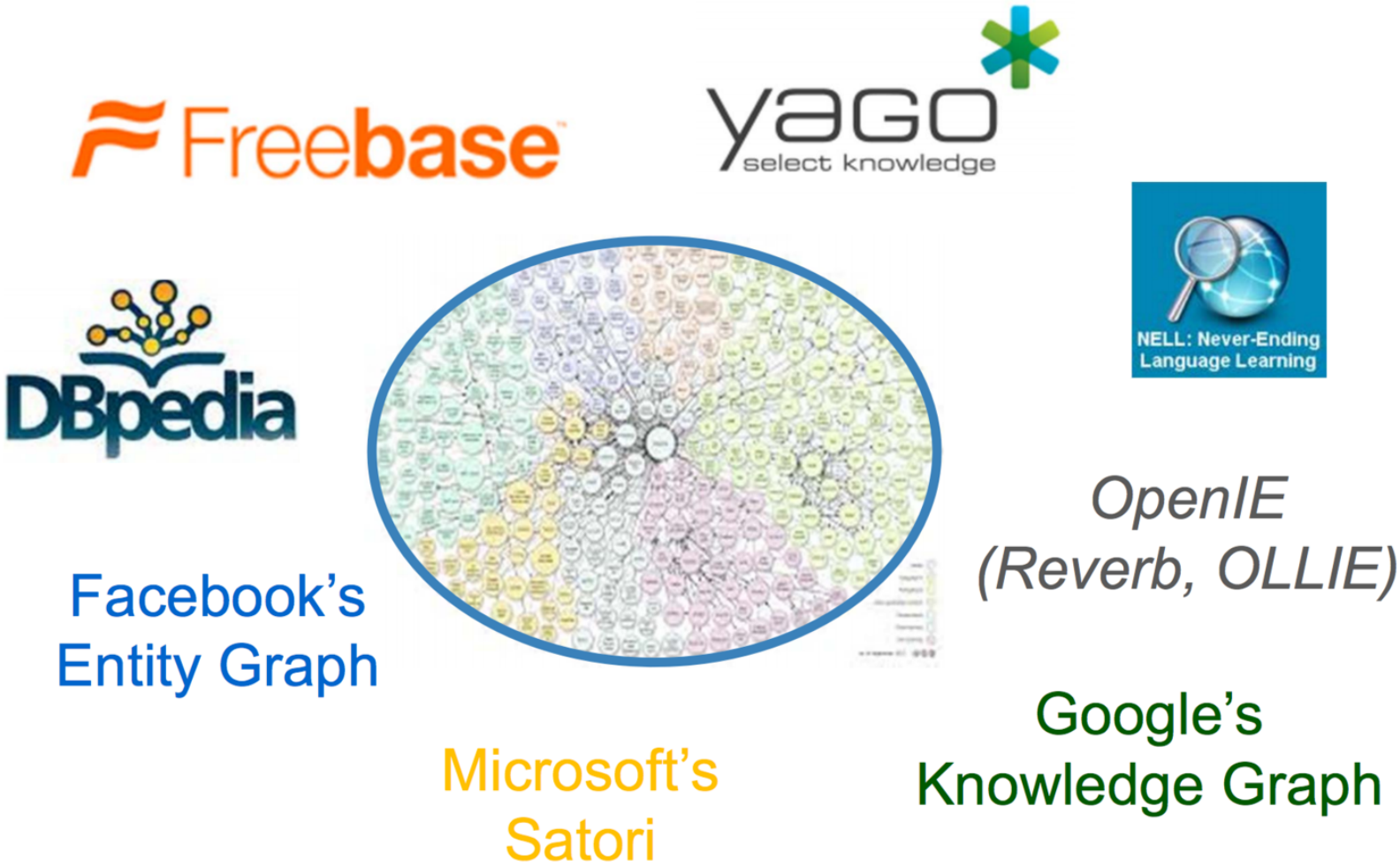
IMDb: Graphic Novel Adaptations - a list by deano11
www.imdb.com/list/ls005377599/ ▾ Internet Movie Database ▾

Overview

Subject-Predicate-Object (RDF) triples

- One way to represent knowledge in machine readable way
- *Subjects* correspond to entities designated by an identifier (URI `http://dbpedia.org/page/Barack_Obama` in case of DBpedia)
- Entities are connected with other entities, literals or scalars by relations or *predicates* (e.g. *hasGenre*, *knownFor*, *marriedTo*, *isPCmemberOf* etc.)
- Each triple represents a simple fact (e.g. <`http://dbpedia.org/page/Barack_Obama`, marriedTo, `http://dbpedia.org/page/Michelle_Obama`>)
- Many SPO triples → *knowledge graph*

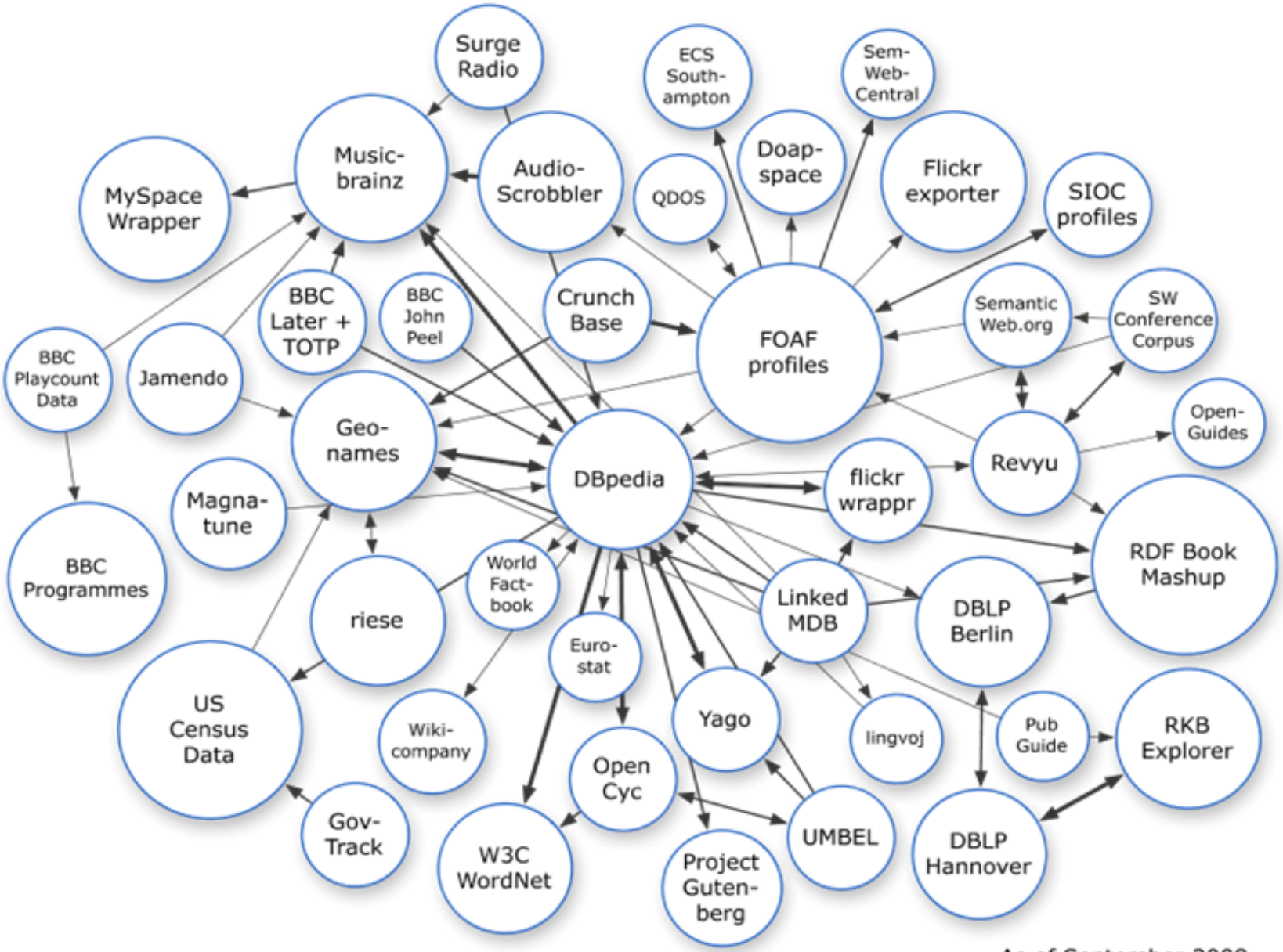
Existing Knowledge Graphs



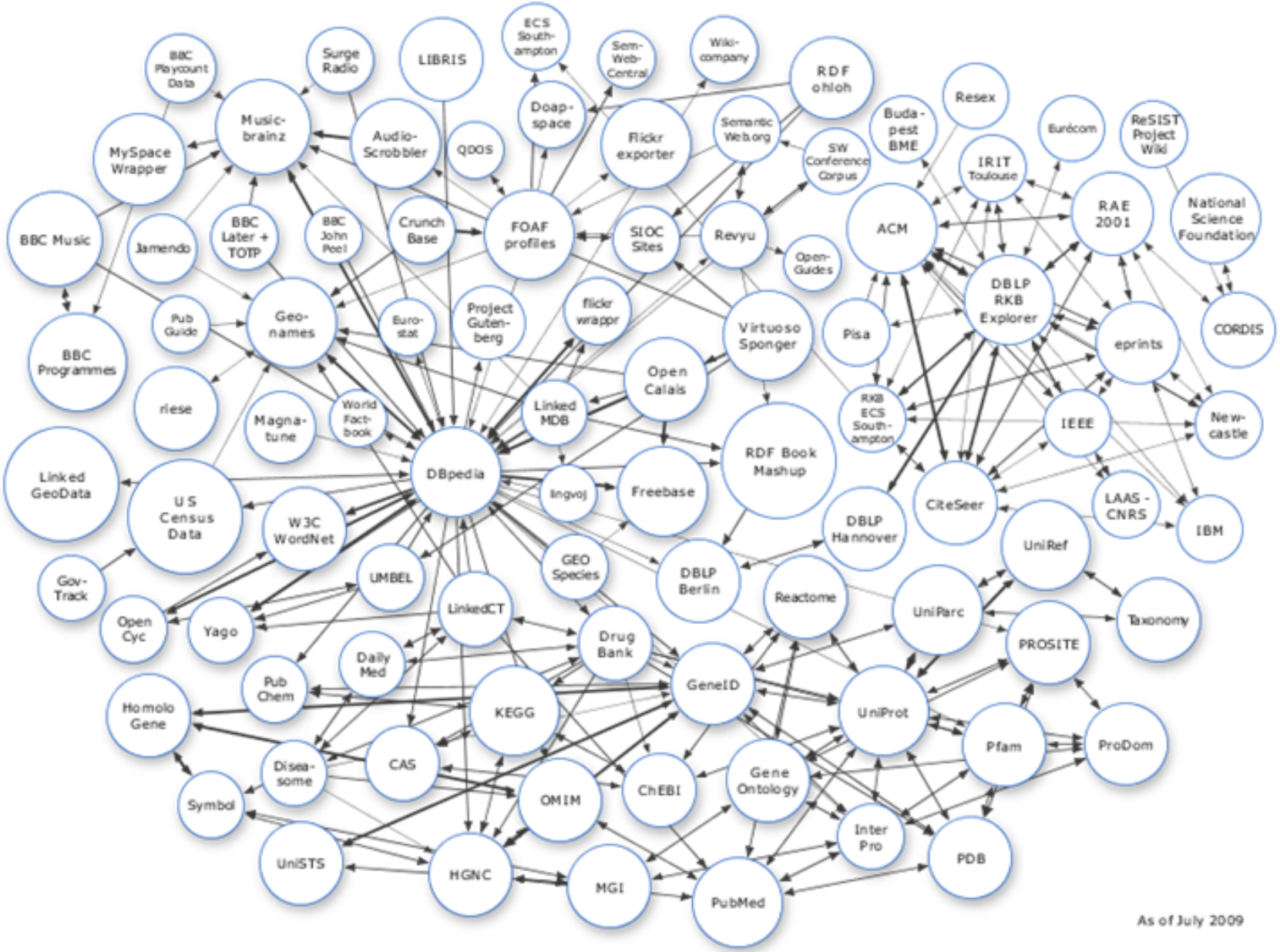
Linked Open Data

- Individual knowledge repositories can be published in machine readable form (RDF)
- The repositories can be connected to each other $\rightarrow$ Liked Open Data (LOD) cloud

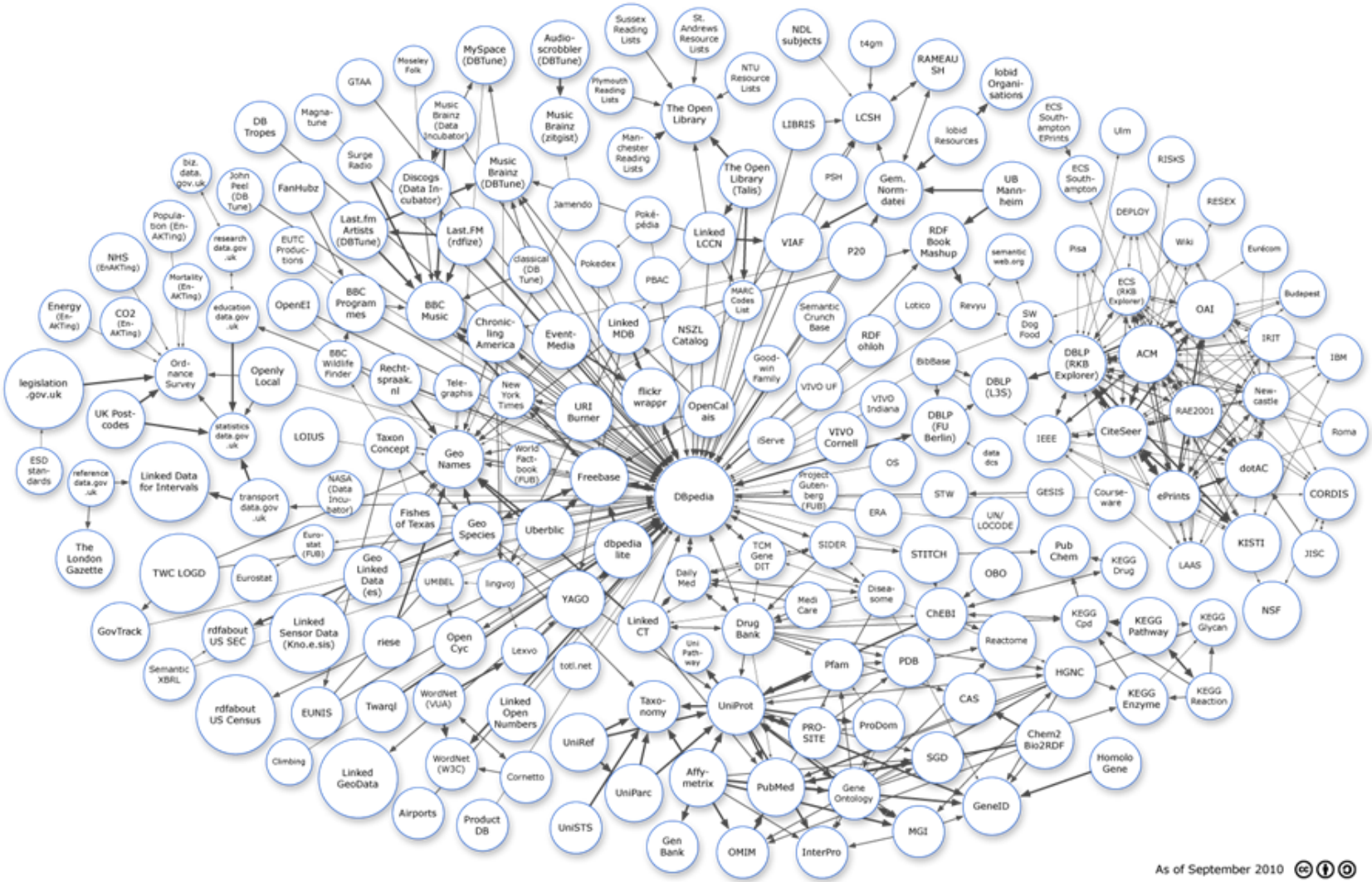
LOD Cloud (circa 2008)



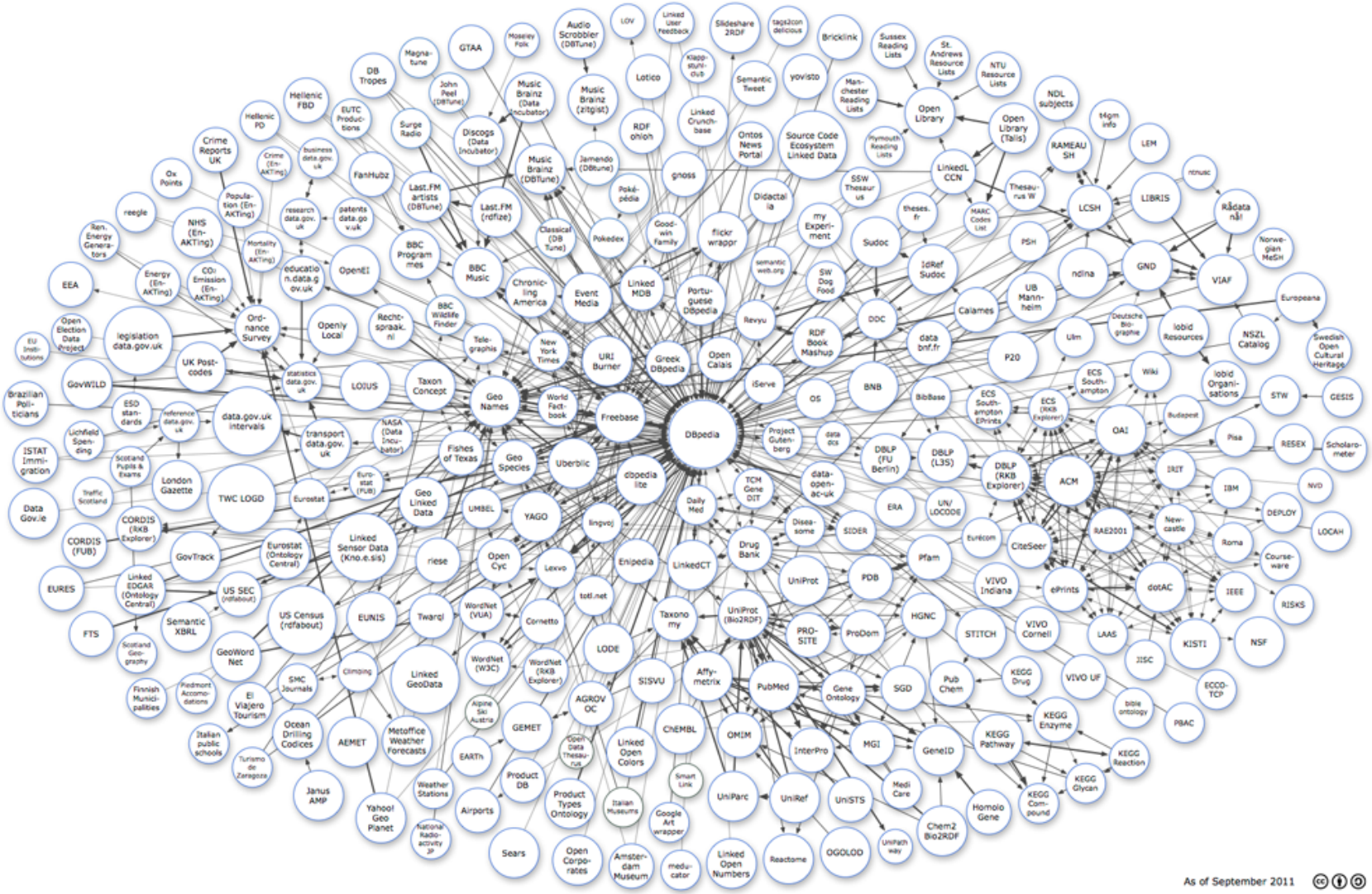
LOD Cloud (circa 2009)



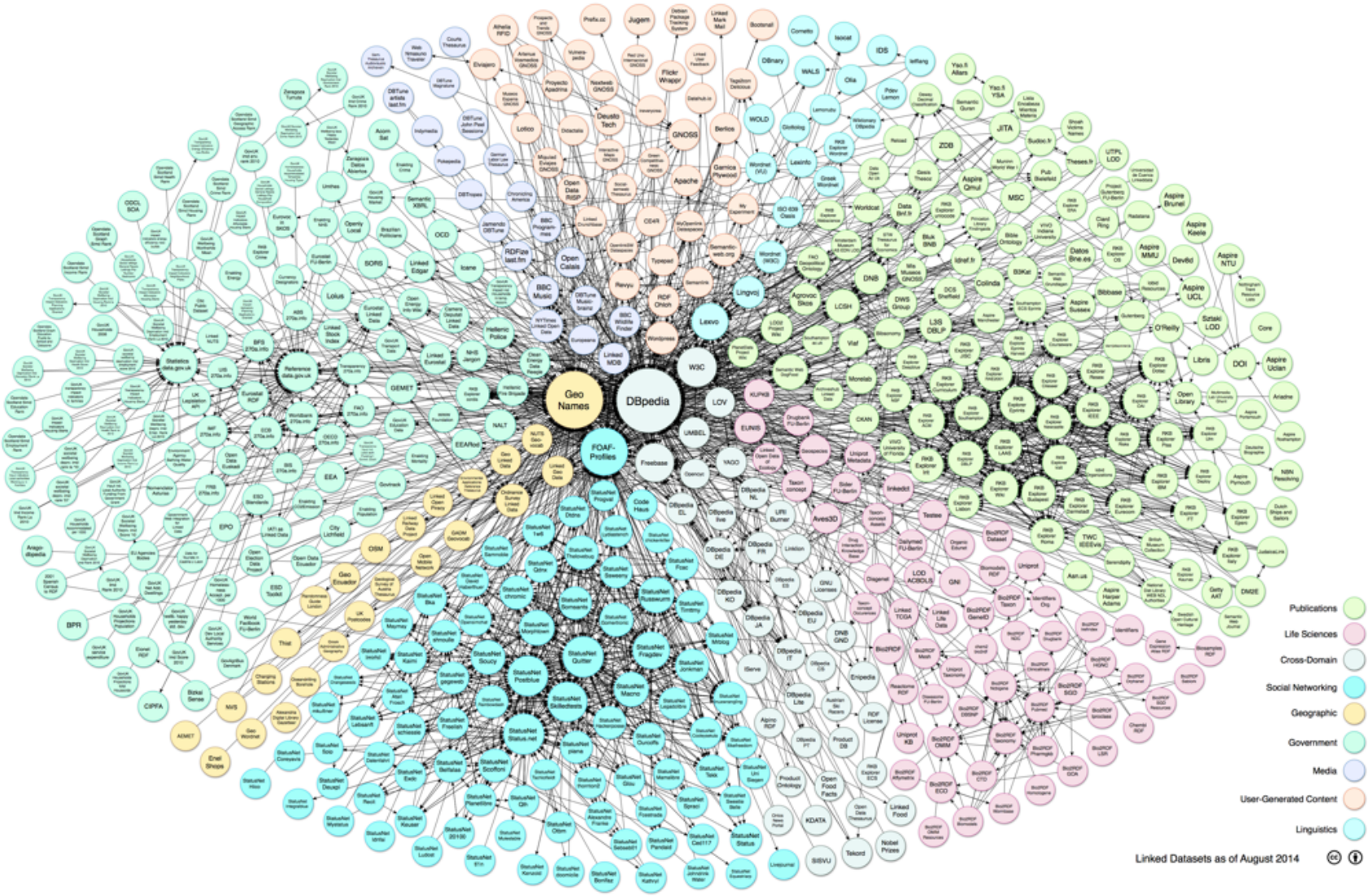
LOD Cloud (circa 2010)



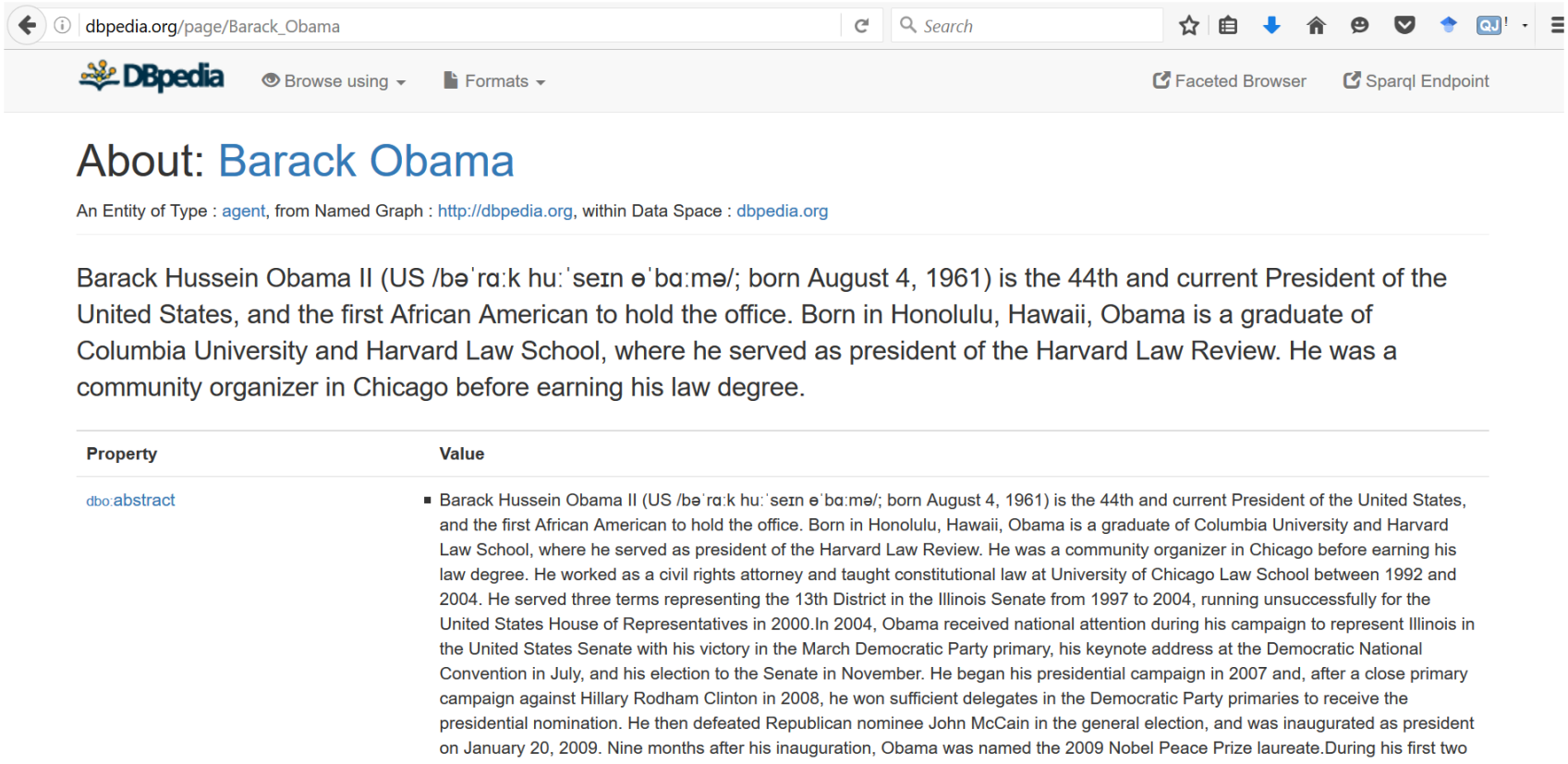
LOD Cloud (circa 2011)



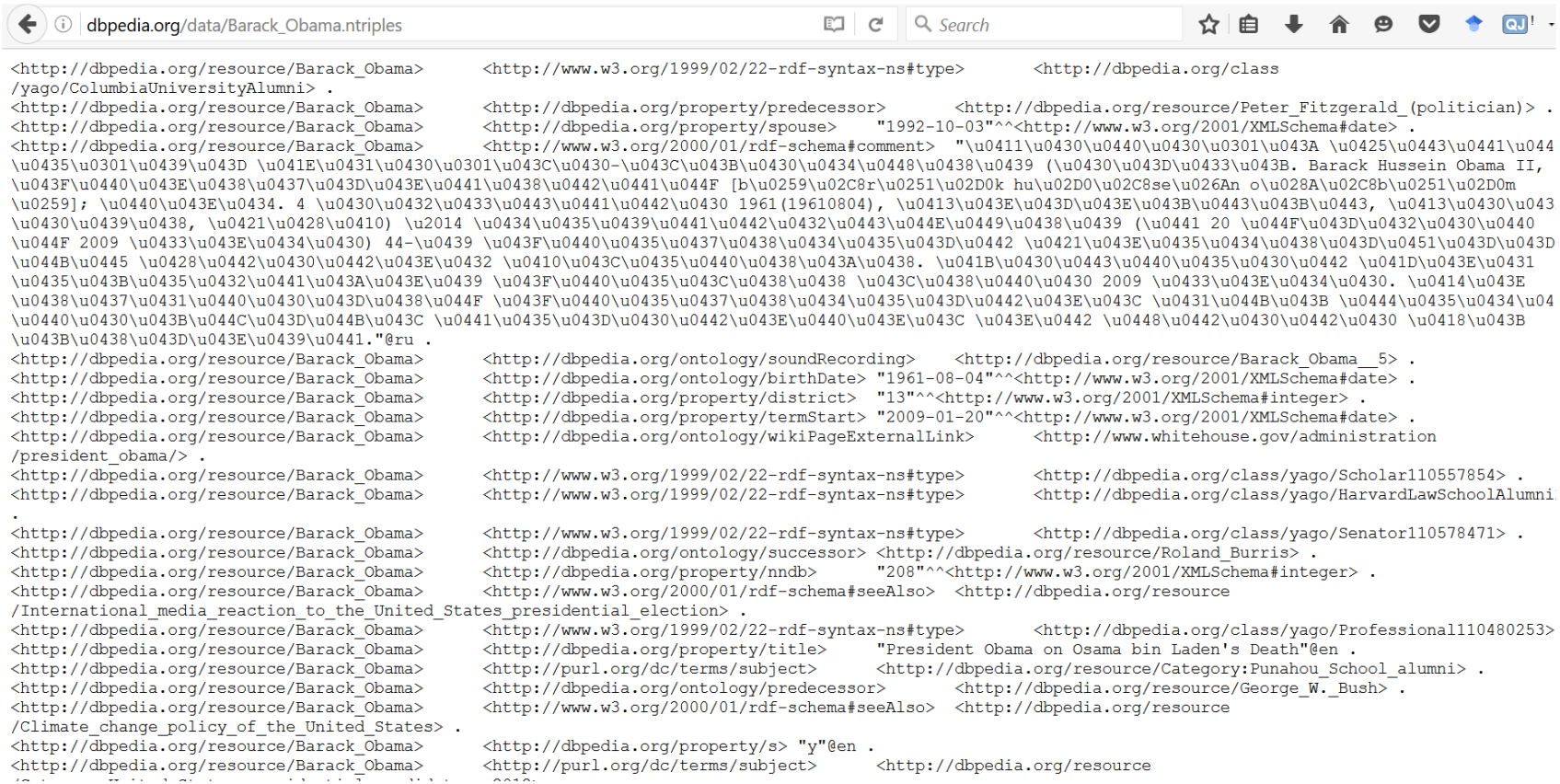
LOD Cloud (Current State)



DBpedia Entity Page



DBpedia Entity Page




```
<http://dbpedia.org/resource/Barack_Obama>      <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>       <http://dbpedia.org/class
/yago/ColumbiaUniversityAlumni> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/property/predecessor>       <http://dbpedia.org/resource/Peter_Fitzgerald_(politician)> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/property/spouse>    "1992-10-03"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://dbpedia.org/resource/Barack_Obama>      <http://www.w3.org/2000/01/rdf-schema#comment>  "\u0411\u0430\u0440\u0430\u0301\u043A \u0425\u0443\u0441\u0441
\u0435\u0301\u0439\u043D \u041E\u0431\u0430\u0301\u043C\u0430-\u043C\u043B\u0430\u0434\u0448\u0438\u0439 (\u0430\u043D\u0433\u043B. Barack Hussein Obama II,
\u043F\u0440\u043E\u0438\u0437\u043D\u043E\u0441\u0438\u0442\u0441\u044F [b\u0259\u02C8r\u0251\u02D0k hu\u02D0\u02C8se\u026An o\u028A\u02C8b\u0251\u02D0m
\u0259]; \u0440\u043E\u0434. 4 \u0430\u0432\u0433\u0443\u0441\u0442\u0430 1961(19610804), \u0413\u043E\u043D\u043E\u043B\u0443\u043B\u0443, \u0413\u0430\u043
\u0430\u0439\u0438, \u0421\u0428\u0410) \u2014 \u0434\u0435\u0439\u0441\u0442\u0432\u0443\u044E\u0449\u0438\u0439 (\u0441 20 \u044F\u043D\u0432\u0430\u0440
\u044F 2009 \u0433\u043E\u0434\u0430) 44-\u0439 \u043F\u0440\u0435\u0437\u0438\u0434\u0435\u043D\u0442 \u0421\u043E\u0435\u0434\u0438\u043D\u0451\u043D\u043D
\u044B\u0445 \u0428\u0442\u0430\u0442\u043E\u0432 \u0410\u043C\u0435\u0440\u0438\u043A\u0438. \u041B\u0430\u0443\u0440\u0435\u0430\u0442 \u041D\u043E\u0431
\u0435\u043B\u0435\u0432\u0441\u043A\u0438\u043E\u0439 \u043F\u0440\u0435\u043C\u0438\u0438 \u043C\u0438\u0440\u0430 2009 \u0433\u043E\u0434\u0430. \u0414\u043E
\u0438\u0437\u0431\u0440\u0430\u043D\u043D\u044F \u043F\u0440\u0435\u0437\u0438\u0434\u0435\u043D\u0442\u043E\u043C \u0431\u044B\u043B \u0444\u0435\u0434\u04
\u0440\u0430\u043B\u044C\u043D\u044B\u043C \u0441\u0435\u043D\u0430\u0442\u043E\u0440\u043E\u043C \u043E\u0442 \u0448\u0442\u0430\u0442\u0430 \u0418\u043B
\u043B\u0438\u043D\u043E\u0439\u0441."@ru .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/ontology/soundRecording>    <http://dbpedia.org/resource/Barack_Obama__5> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/ontology/birthDate> "1961-08-04"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/property/district>  "13"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/property/termStart> "2009-01-20"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/ontology/wikiPageExternalLink>      <http://www.whitehouse.gov/administration
/president_obama/> .
<http://dbpedia.org/resource/Barack_Obama>      <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>       <http://dbpedia.org/class/yago/Scholar110557854> .
<http://dbpedia.org/resource/Barack_Obama>      <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>       <http://dbpedia.org/class/yago/HarvardLawSchoolAlumni
.
<http://dbpedia.org/resource/Barack_Obama>      <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>       <http://dbpedia.org/class/yago/Senator110578471> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/ontology/successor> <http://dbpedia.org/resource/Roland_Burris> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/property/nndb>      "208"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://dbpedia.org/resource/Barack_Obama>      <http://www.w3.org/2000/01/rdf-schema#seeAlso>  <http://dbpedia.org/resource
/International_media_reaction_to_the_United_States_presidential_election> .
<http://dbpedia.org/resource/Barack_Obama>      <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>       <http://dbpedia.org/class/yago/Professional110480253>
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/property/title>     "President Obama on Osama bin Laden's Death"@en .
<http://dbpedia.org/resource/Barack_Obama>      <http://purl.org/dc/terms/subject>      <http://dbpedia.org/resource/Category:Punahou_School_alumni> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/ontology/predecessor>       <http://dbpedia.org/resource/George_W._Bush> .
<http://dbpedia.org/resource/Barack_Obama>      <http://www.w3.org/2000/01/rdf-schema#seeAlso>  <http://dbpedia.org/resource
/Climate_change_policy_of_the_United_States> .
<http://dbpedia.org/resource/Barack_Obama>      <http://dbpedia.org/property/s> "y"@en .
<http://dbpedia.org/resource/Barack_Obama>      <http://purl.org/dc/terms/subject>      <http://dbpedia.org/resource
```

Entity Retrieval from Knowledge Graph(s) (ERKG)

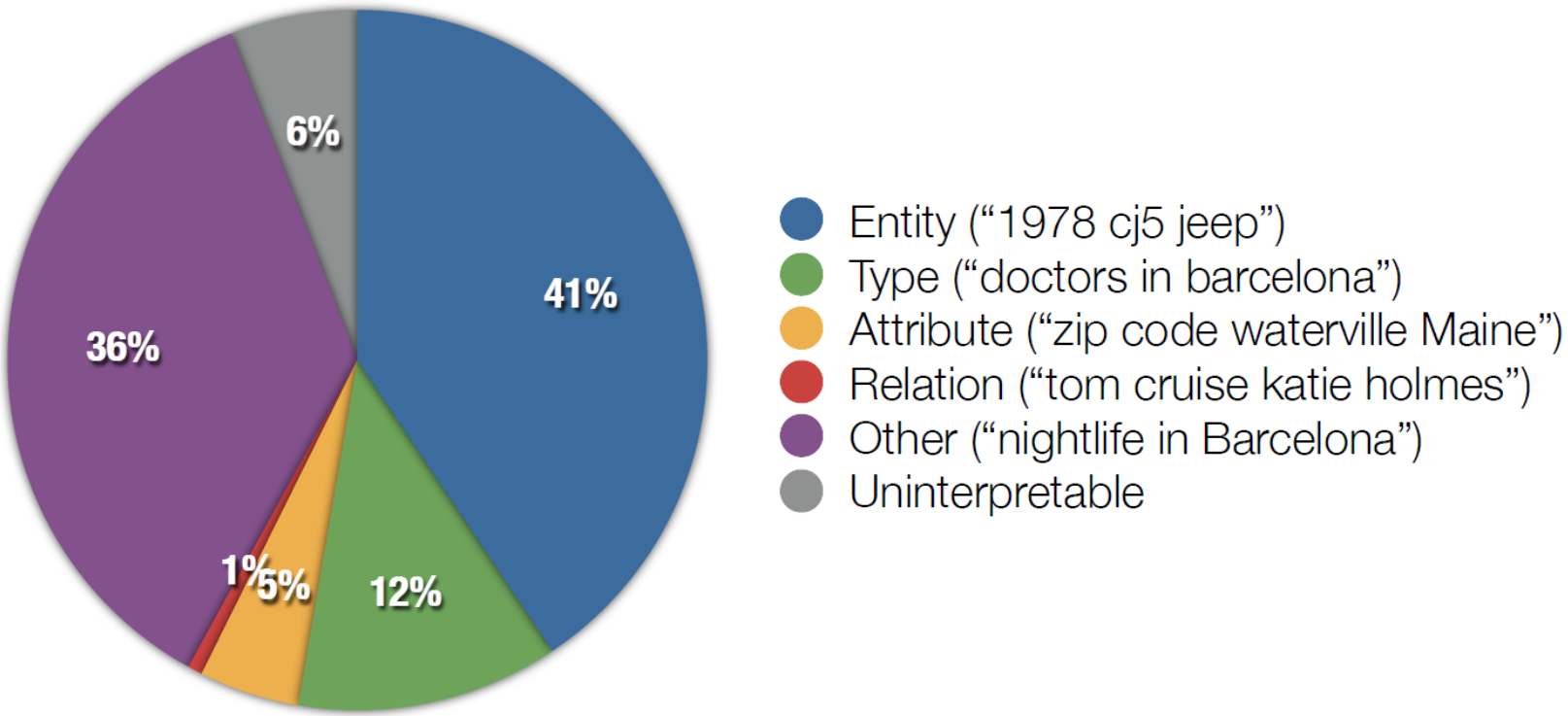
- Knowledge graphs are perfectly suited for addressing the information needs that aim at finding specific objects (entities) rather than documents
- Entity retrieval is a unique and interesting IR problem, since there is *no notion of a document*
- *Ad-hoc* Entity Retrieval assumes keyword queries (structured queries are studied more in the DB community)

Typical ERKG tasks

- **Entity Search**: simple queries aimed at finding a particular entity or an entity which is an attribute of another entity
  - *"Ben Franklin"*
  - *"England football player highest paid"*
  - *"Einstein Relativity theory"*
- **List Search**: descriptive queries with several relevant entities
  - *"US presidents since 1960"*
  - *"animals lay eggs mammals"*
  - *"Formula 1 drivers that won the Monaco Grand Prix"*
- **Question Answering**: queries are questions in natural language
  - *"Who founded Intel?"*
  - *"For which label did Elvis record his first album?"*

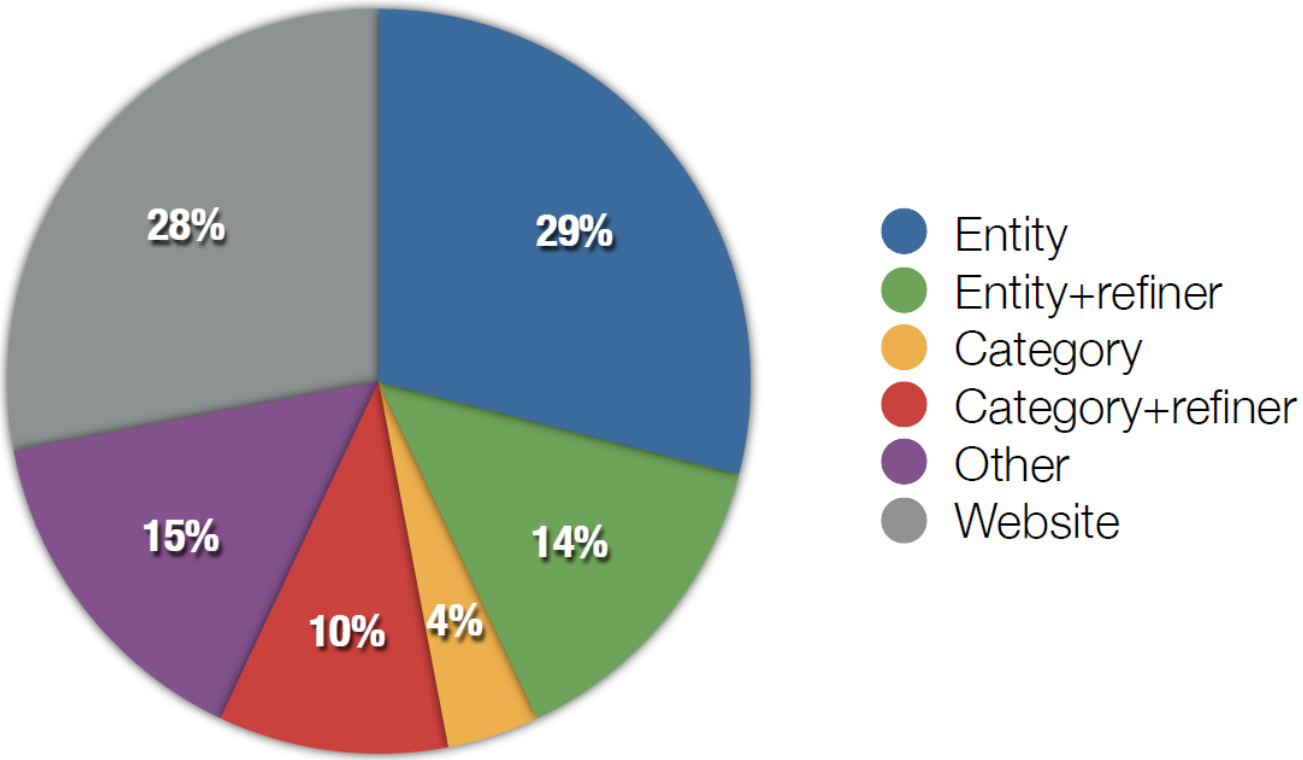
Distribution of Entity Web Search Queries

[Pound et al. WWW'10]



Distribution of Entity Web Search Queries

[Lin et al. WWW'11]



Research challenges in ERKG

1. How to design entity representations that capture the semantics of entity properties/relations and are effective for entity retrieval?
2. How to develop accurate and efficient entity retrieval models?

Entity Representation Methods (day 1)

- Neumayer, Balog et al. On the Modeling of Entities for Ad-hoc Entity Search in the Web of Data, ECIR'12
- Neumayer, Balog et al. When Simple is (more than) Good Enough: Effective Semantic Search with (almost) no Semantics, ECIR'12
- Zhiltsov and Agichtein. Improving Entity Search over Linked Data by Modeling Latent Semantics, CIKM'13
- Zhiltsov, Kotov et al. Fielded Sequential Dependence Model for Ad-hoc Entity Retrieval in the Web of Data, SIGIR'15

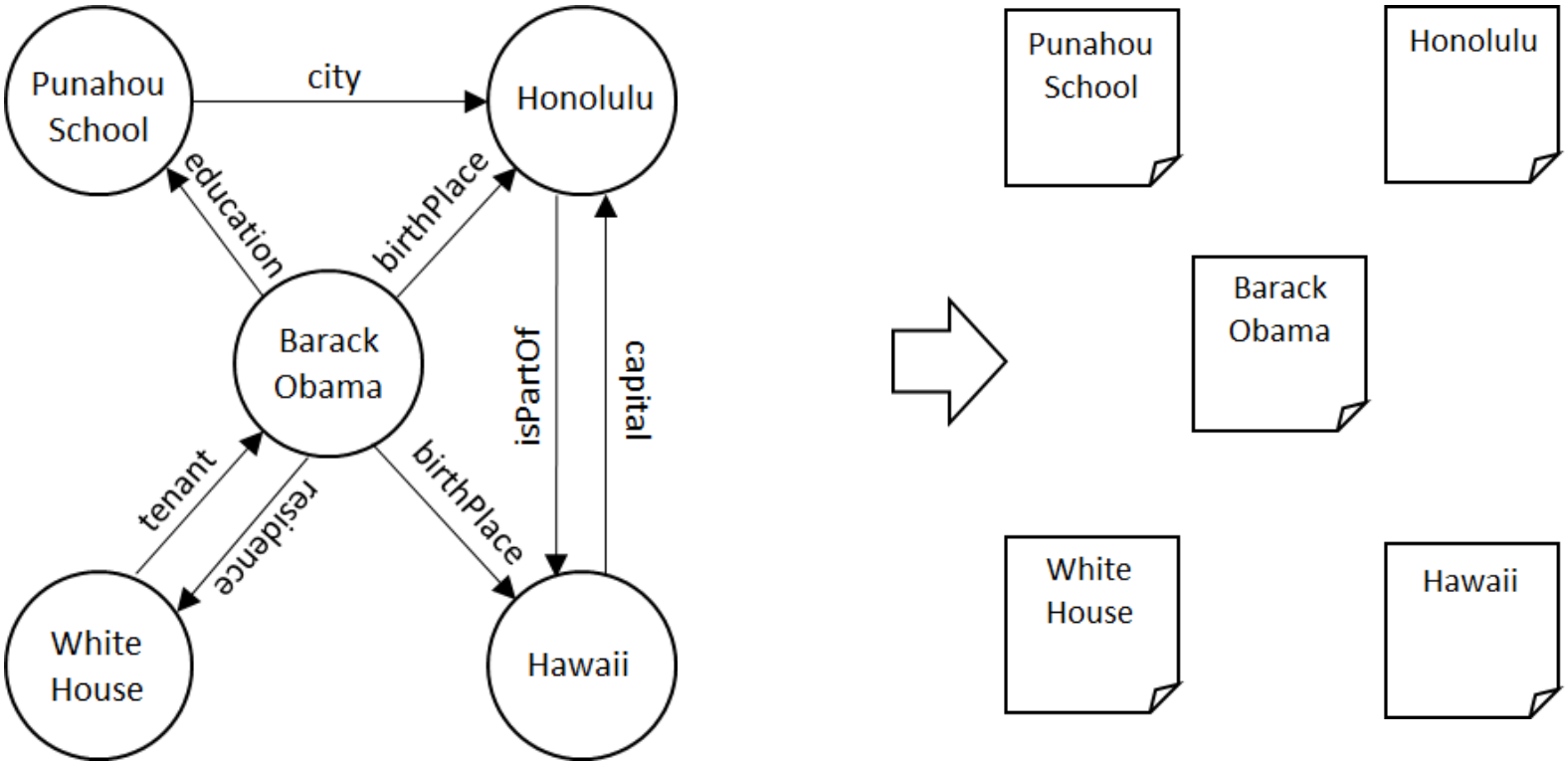
Entity Retrieval Models (day 2)

- Classic unigram bag-of-words models for structured document retrieval, such as BM25F, Mixture of Language Models (MLM), Probabilistic Retrieval Model for Semi-structured Data (PRMS)
- Dali and Fortuna. Learning to Rank for Semantic Search, WWW'11
- Tonon, Demartini et al. Combining Inverted Indices and Structured Search for Ad-hoc Object Retrieval, SIGIR'12
- Sawant and Chakrabarti. Learning Joint Query Interpretation and Response Ranking, WWW'13
- Zhiltsov, Kotov et al. Fielded Sequential Dependence Model for Ad-hoc Entity Retrieval in the Web of Data, SIGIR'15
- Nikolaev, Kotov et al. Parameterized Fielded Term Dependence Models for Ad-hoc Entity Retrieval from Knowledge Graph, SIGIR'16

Overview

From Entity Graph to Entity Documents

Build a textual representation (i.e. "document") for each entity by considering all triples, where it stands as subject (or object)



Language Modeling Approach

- Retrieval score of D is the likelihood of it being relevant to a given query Q
- Query Likelihood retrieval model: retrieval score of D is the likelihood of generating Q from Θ_D, the language model of D

$$P(D|Q) \overset{rank}{=} \frac{P(Q|D)P(D)}{P(Q)} \propto P(Q|D)P(D), where$$

$$P(Q|D) = \prod_{q_i \in Q} P(q_i|\theta_D)^{n(q_i,Q)}$$

Entity Language Model

If each entity is represented as an unstructured document E:

$$P(E|Q) \propto P(E)P(Q|\theta_E) = P(E) \prod_{q_i \in Q} P(q_i|\theta_E)^{n(q_i,Q)}$$

Structured Entity Documents (1)

- Entity descriptions are naturally structured, entities can be represented as fielded documents
- Entity documents can be ranked using conventional IR models
- In the simplest case, each predicate corresponds to one document field
- However, there are infinitely many predicates $\rightarrow$ optimization of field importance weights is computationally intractable

Structured Entity Documents (2)

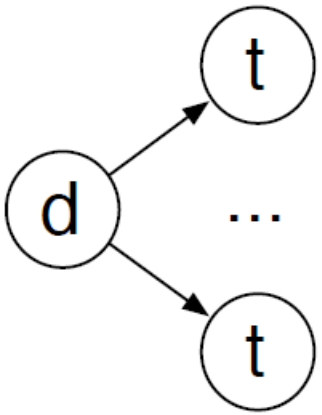
Predicate folding: group predicates together into a small set of predefined categories → entity documents with smaller number of fields

names	**foaf:name**	Barack Obama (en)
	dbp:birthName	Barack Hussein Obama II (en)
attributes	**dbo:birthDate**	1961-08-04 (xsd:date)
	dbp:birthPlace	Honolulu, Hawaii, U.S. (en)
	dbo:office	44th President of the United States
outgoing relations	**dbo:party**	dbr:Democratic_Party_(United_States)
	dbo:region	dbr:Illinois
	dbo:predecessor	dbr:Peter_Fitzgerald_(politician)
		dbr:George_W._Bush
		dbr:Alice_Palmer_(politician)
incoming relations	is **dbo:tenant** of	dbr:White_House
	is **dbo:president** of	dbr:Joe_Biden

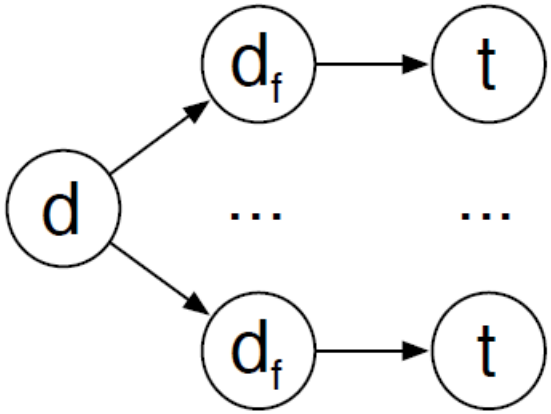
Predicate Folding

- Grouping according to type (attributes, incoming/outgoing links)[Pérez-Agüera et al. 2010]
- Grouping according to importance (determined based on predicate popularity)[Blanco et al. 2010]

Model Comparison



Unstructured document model



Fielded document model

2-field Entity Document

[Neumayer, Balog et al., ECIR'12]

Each entity is represented as a two-field documents:

title
: object values belonging to predicates ending with "name", "label" or "title"

content
: object values for 1000 most frequent predicates concatenated together into a flat text representation

3-field Entity Document

[Zhiltsov and Agichtein, CIKM'13]

Each entity is represented as a three-field document:

names
: literals of `foaf:name`, `rdfs:label` predicates along with tokens extracted from entity URIs

attributes
: literals of all other predicates

outgoing links
: names attributes of entities in the object position

5-field Entity Document

[Zhiltsov, Kotov et al., SIGIR'15]

Each entity is represented as a five-field document:

names
: conventional names of the entities, such as the name of a person or the name of an organization

attributes
: all entity properties, other than names

categories
: classes or groups, to which the entity has been assigned

similar entity names
: names of the entities that are very similar or identical to a given entity

related entity names
: names of the entities that are part of the same RDF triple

5-field Entity Document Example

Entity document for the entity *Barack Obama*.

Field	**Content**
names	barack obama barack hussein obama ii
attributes	44th current president united states birth place honolulu hawaii
categories	democratic party united states senator nobel peace prize laureate christian
similar entity names	barack obama jr barak hussein obama barack h obama ii
related entity names	spouse michelle obama illinois state predecessor george walker bush

Overview

BM25F

[Robertson and Zaragoza, CIKM'04]

- Option 1: aggregation of BM25 scores across fields

$$P(E|Q) \stackrel{rank}{=} \sum_{q_i \in Q} \sum_{j=1}^{F} log \frac{N}{df^j(q_i)} \frac{(k_1+1)\tilde{tf}^j(q_i)}{k_1((1-b)+b\frac{|E^j|}{|E^j|_{avg}})}, 0 \leq b \leq 1$$

- Option 2 (more effective): field-specific length normalization

$$\tilde{tf}^j(q_i) = \sum_{j=1}^{F} w_j \frac{tf^j(q_i)}{B^j}$$

$$B^j = ((1-b^j)+b^j \frac{|E^j|}{|E^j|_{avg}}, 0 \leq b^j \leq 1$$

$$P(E|Q) \stackrel{rank}{=} \sum_{q_i \in Q} log \frac{N}{df^j(q_i)} \cdot \frac{(k_1+1)\tilde{tf}(q_i)}{k_1+\tilde{tf}(q_i)}$$

BM25F

[Robertson and Zaragoza, CIKM'04]

- Option 1: aggregation of BM25 scores across fields

$$P(E|Q) \stackrel{rank}{=} \sum_{q_i \in Q} \sum_{j=1}^{F} log \frac{N}{df^j(q_i)} \frac{(k_1+1)\tilde{tf}^j(q_i)}{k_1((1-b)+b\frac{|E^j|}{|E^j|_{avg}})}, 0 \leq b \leq 1$$

- Option 2 (more effective): field-specific length normalization

$$\tilde{tf}^j(q_i) = \sum_{j=1}^{F} w_j \frac{tf^j(q_i)}{B^j}$$

$$B^j = ((1-b^j)+b^j \frac{|E^j|}{|E^j|_{avg}}, 0 \leq b^j \leq 1$$

$$P(E|Q) \stackrel{rank}{=} \sum_{q_i \in Q} log \frac{N}{df^j(q_i)} \cdot \frac{(k_1+1)\tilde{tf}(q_i)}{k_1+\tilde{tf}(q_i)}$$

Mixture of Language Models

[Ogilvie and Callan, SIGIR'03]

- Separate LM θ_E^j is created for each field j of entity document E
- Document LM is a linear combination of field LMs

$$P(Q|E) \overset{rank}{=} \prod_{q_i \in Q} P(q_i|\theta_E)^{tf(q_i)},$$

where

$$P(q_i|\theta_E) = \sum_{j=1}^{F} w_j P(q_i|\theta_E^j), \sum_j w_j = 1$$

$$P(q_i|\theta_E^j) = \frac{tf_{q_i,E^j} + \mu_j \frac{cf_{q_i}^j}{|C_j|}}{|E^j| + \mu_j}$$

Setting Field Weights

- Heuristically: proportionate to the length of content in the field
- Empirically: by optimizing the target retrieval metric using training queries
- **Problems**:
  - Entities are sparse with respect to different fields (most entities have only a handful of predicates)
  - More fields in entity representations $\rightarrow$ more training data to optimize their weights

Probabilistic Retrieval Model for Semi-Structured Data

[Kim, Xue and Croft, ECIR'09]

Extends Mixture of Language Models by dynamically determining the mapping of query terms onto entity document fields

$$P(q_i|\theta_E) = \sum_{j=1}^{F} w_j P(q_i|\theta_E^j), \sum_j w_j = 1$$

$$P(q_i|\theta_E) = \sum_{j=1}^{F} P(E^j|q_i) P(q_i|\theta_E^j)$$

where

$$P(E^j|q_i) = \frac{P(q_i|E^j)P(E^j)}{\sum_{j=1}^{F} P(q_i|E^j)P(E^j)}$$

Probabilistic Retrieval Model for Semi-Structured Data

[Kim, Xue and Croft, ECIR'09]

Extends Mixture of Language Models by dynamically determining the mapping of query terms onto entity document fields

$$P(q_i|\theta_E) = \sum_{j=1}^{F} w_j P(q_i|\theta_E^j), \sum_j w_j = 1$$

$$P(q_i|\theta_E) = \sum_{j=1}^{F} P(E^j|q_i) P(q_i|\theta_E^j)$$

where

$$P(E^j|q_i) = \frac{P(q_i|E^j)P(E^j)}{\sum_{j=1}^{F} P(q_i|E^j)P(E^j)}$$

Probabilistic Retrieval Model for Semi-Structured Data

[Kim, Xue and Croft, ECIR'09]

Extends Mixture of Language Models by dynamically determining the mapping of query terms onto entity document fields

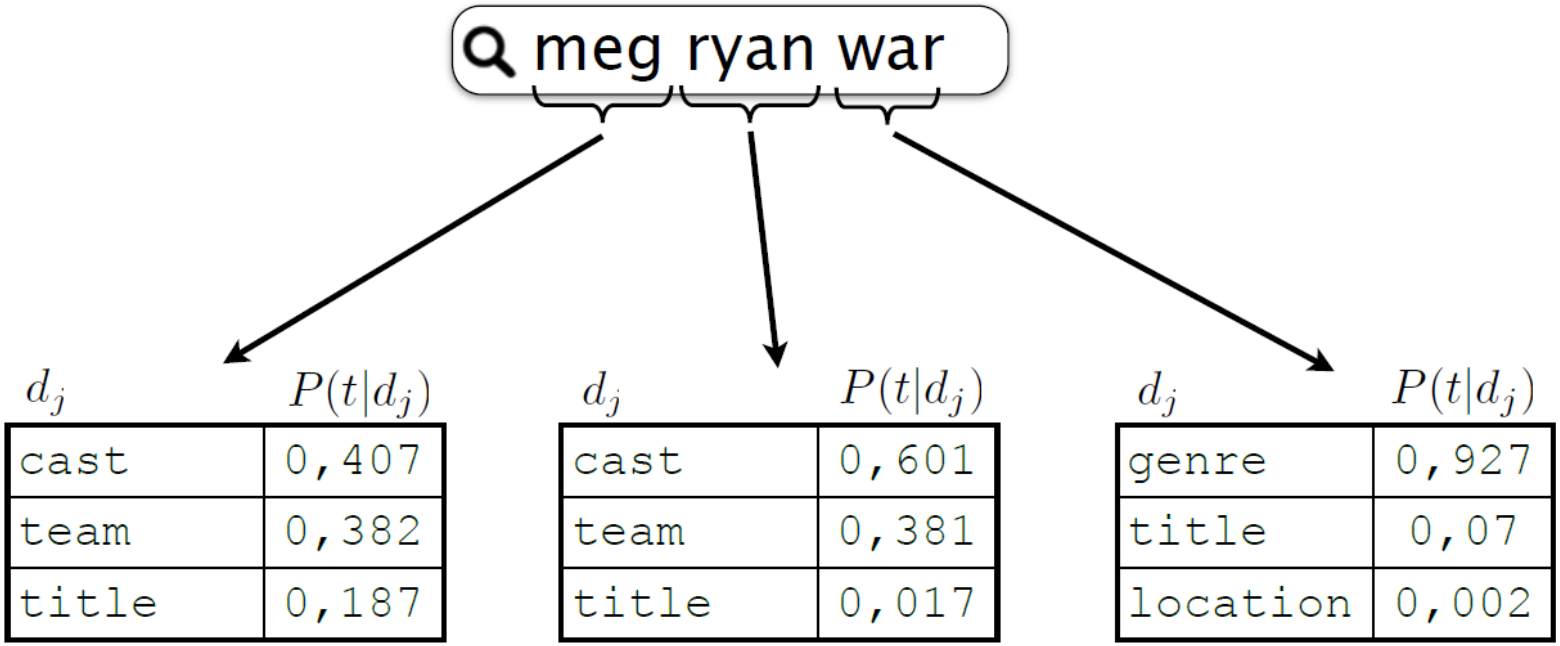
$$P(q_i|\theta_E) = \sum_{j=1}^{F} w_j P(q_i|\theta_E^j), \sum_j w_j = 1$$

$$P(q_i|\theta_E) = \sum_{j=1}^{F} P(E^j|q_i) P(q_i|\theta_E^j)$$

where

$$P(E^j|q_i) = \frac{P(q_i|E^j)P(E^j)}{\sum_{j=1}^{F} P(q_i|E^j)P(E^j)}$$

PRMS (Example)



d_j	$P(t\|d_j)$
cast	0,407
team	0,382
title	0,187

d_j	$P(t\|d_j)$
cast	0,601
team	0,381
title	0,017

d_j	$P(t\|d_j)$
genre	0,927
title	0,07
location	0,002

Hierarchical Entity Model (1)

[Neumayer, Balog et al., ECIR'12]

Entity document fields are organized into a 2-level hierarchy:

- Predicate types are on the top level:

  name
  : subject is E, object is literal and predicate comes from a predefined list (e.g. `foaf:name` or `rdfs:label`) or ends with "name", "label" or "title"

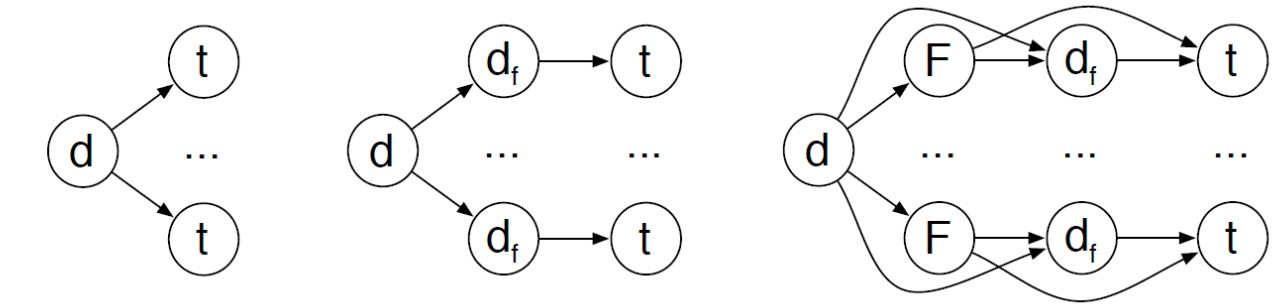
  attributes
  : the subject is E, object is literal and the predicate is not of type name

  outgoing links
  : the subject is E and the object is a URI. URI is resolved by replacing it with entity name

  incoming links
  : E is an object, subject entity URI is resolved

- Individual predicates are at the bottom level

Hierarchical Entity Model (2)



Unstructured document model **Fielded document model** **Hierarchical document model**

$$P(q_i|\theta_E) = \sum_{p_t} P(q_i|p_t, E)P(p_t|E) =$$

$$= \sum_{p_t}(\sum_{p \in p_t} P(q_i|p, p_t)P(p|p_t, E))P(p_t|E)$$

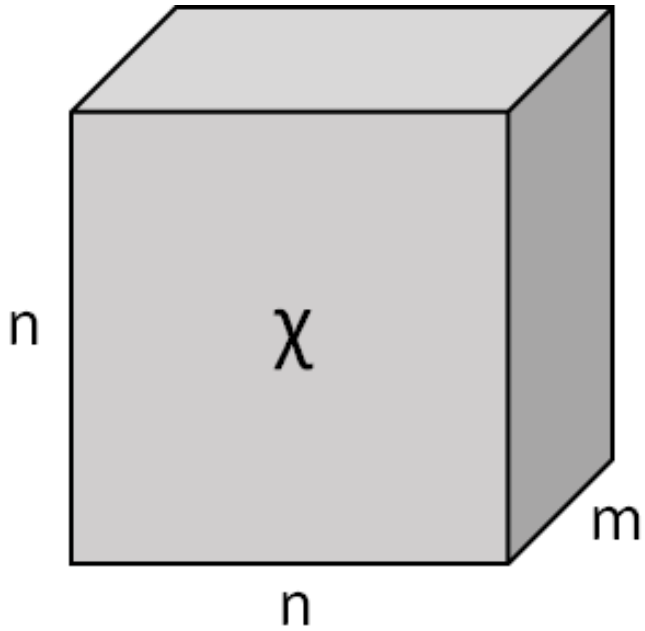
$P(q_i|p, p_t) = (1-\lambda)P(q_i|p) + \lambda P(q_i|\theta_E^{p_t})$, where $P(q_i|p)$ ML estimate and $P(q_i|\theta_E^{p_t})$ is Dirichlet-smoothed LM for predicate type p_t

Latent Dimensional Representation

[Zhiltsov and Agichtein, CIKM'13]

- Compact representation of entities in low dimensional space by using a modified algorithm for tensor factorization
- Entities and entity-query pairs are represented with term-based and structural features

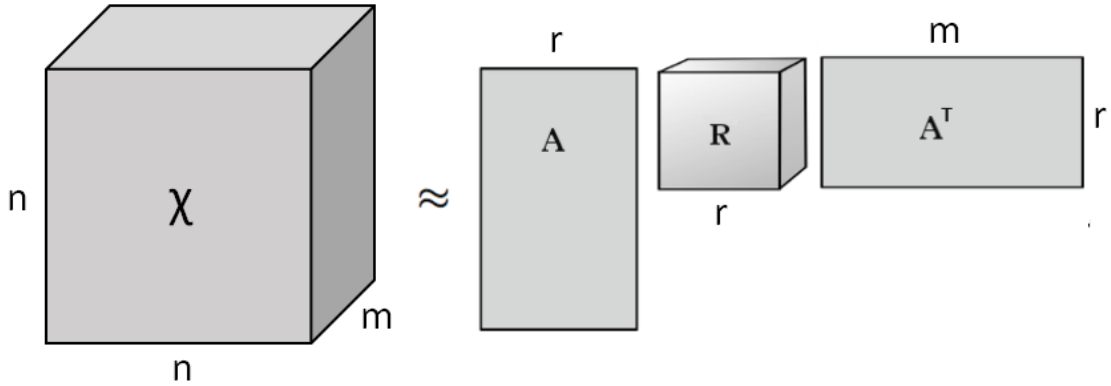
Knowledge Graph as Tensor



- For a knowledge graph with n distinct entities and m distinct predicates, we construct a tensor $\mathcal{X}$ of size $n \times n \times m$, where $\mathcal{X}_{ijk} = 1$, if there is k-th predicate between i-th entity and j-th entity, and $\mathcal{X}_{ijk} = 0$, otherwise
- Each k-th frontal tensor slice $\mathcal{X}_k$ is an adjacency matrix for the k-the predicate, which is sparse

RESCAL Tensor Factorization

[Nikel, Tresp, et al., WWW'12]



- Given r is the number of latent factors, we factorize each X_k into the matrix product:

$$X_k = AR_kA^T, k = \overline{1, m},$$

  where A is a dense $n \times r$ matrix, a matrix of latent embeddings for entities, and R_k is an $r \times r$ matrix of latent factors
- A and R_k are solutions of the following optimization problem:

$$\min_{A,R} \frac{1}{2}\left(\sum_k \|X_k - AR_kA^T\|_F^2\right) + \lambda\left(\|A\|_F^2 + \sum_k \|R_k\|_F^2\right)$$

Retrieval Method

1. Retrieve initial set of entities using MLM
2. Re-rank the entities using Gradient Boosted Regression Tree (GBRT)

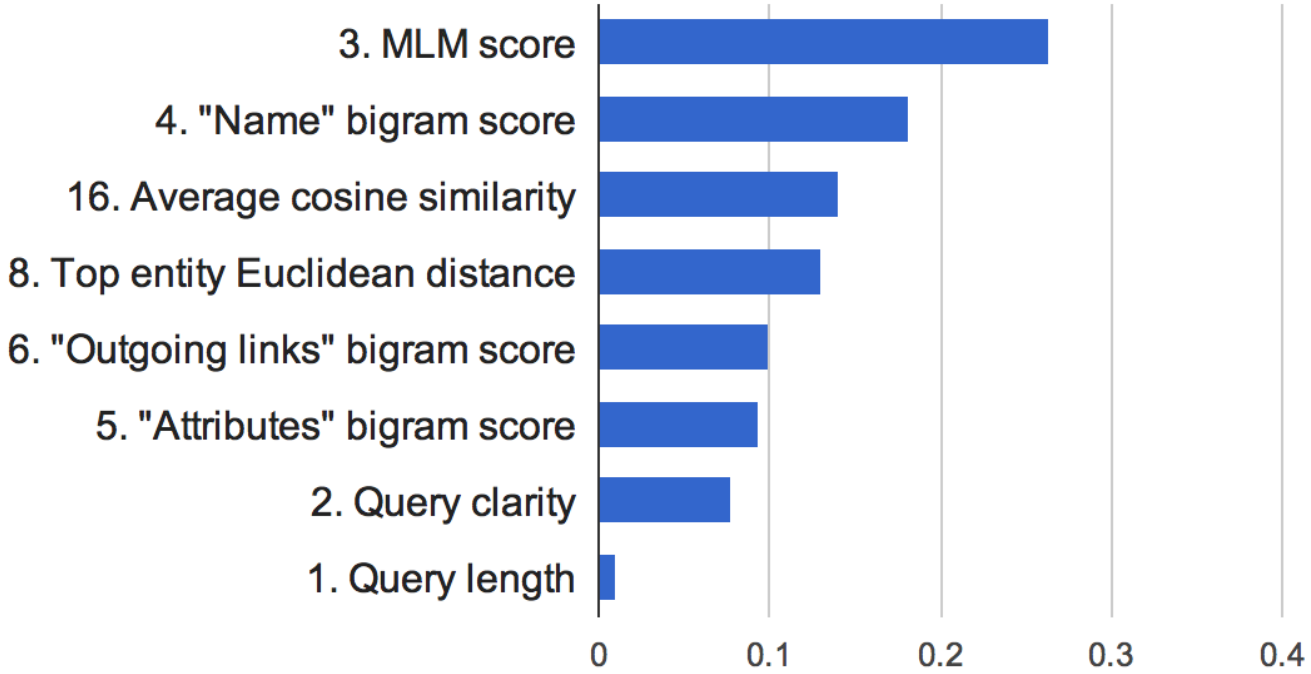
Features

#	Feature
Term-based features	
1	Query length
2	Query clarity
3	Uniformly weighted MLM score
4	Bigram relevance score for the ”name” field
5	Bigram relevance score for the ”attributes” field
6	Bigram relevance score for the ”outgoing links” field
Structural features	
7	Top-3 entity cosine similarity, $cos(\mathbf{e}, \mathbf{e}_{top})$
8	Top-3 entity Euclidean distance, $\|\mathbf{e} - \mathbf{e}_{top}\|$
9	Top-3 entity heat kernel, $e^{-\frac{\|\mathbf{e}-\mathbf{e}_{top}\|^2}{\sigma}}$

Results

Features	Performance		
	NDCG	**MAP**	**P@10**
Term-based baseline	0.382	0.265	0.539
All features	**0.401** (+ 5.0%)*	**0.276** (+ 4.2%)	**0.561** (+ 4.1%)*

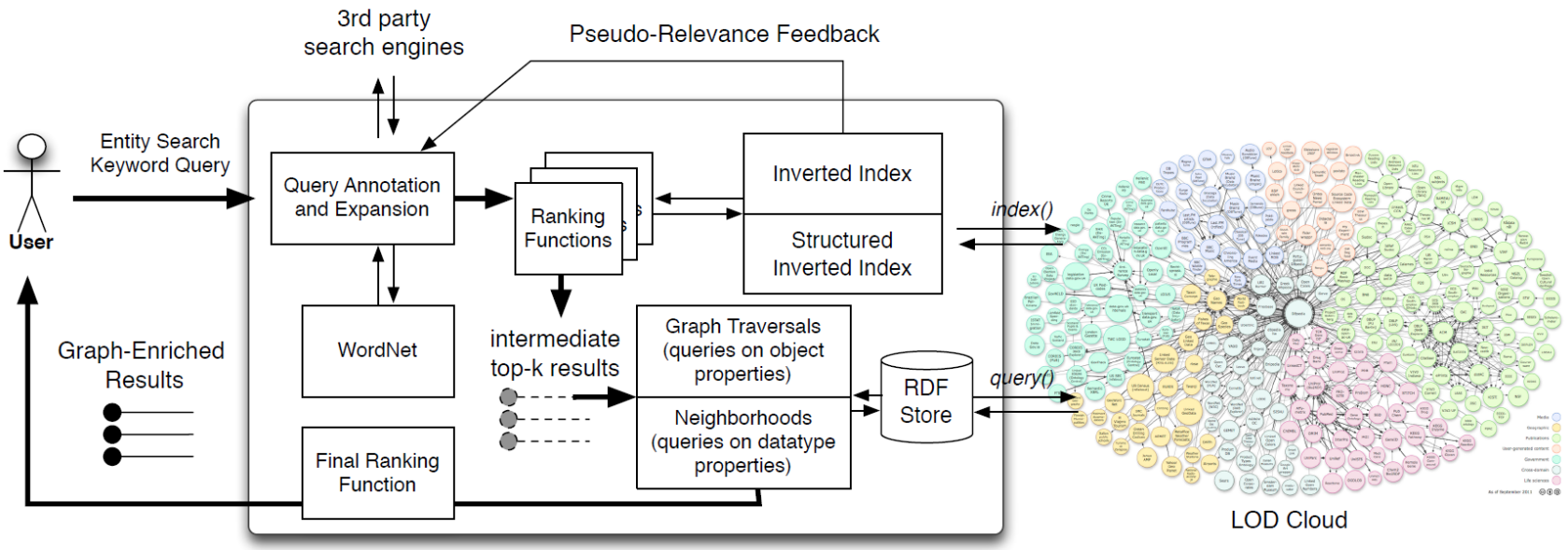
Feature Importance



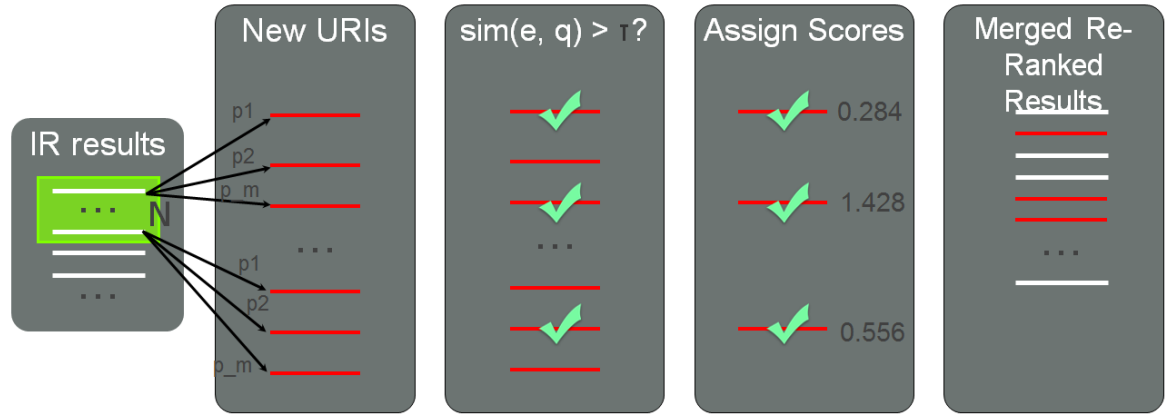
- Exploiting latent semantics of entities helps improve retrieval results (structural features improve NDCG and P@10)
- Most effective distance measures are cosine similarity and Euclidean distance
- However, the overall performance of the method is sensitive to top 3 retrieved results

Hybrid IR and DB ERKG Methods

[Tonon, Demartini et al., SIGIR'12]



Hybrid ERKG Methods



Take top-N docs.

Follow links/properties and get new URIs.

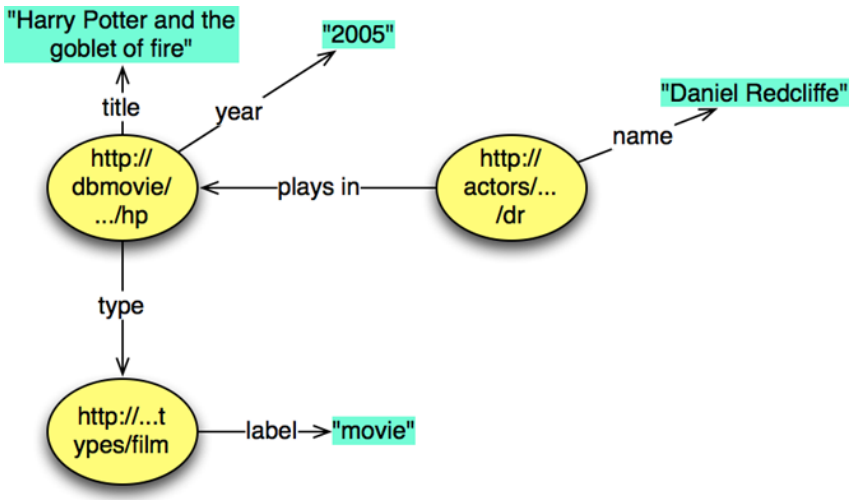
Filter new results by text similarity wrt the user query.

Scoring functions:
count sim > τ,
avg sim > τ,
Sum sim,
Avg sim,
Sum BM25 - ε

$$\lambda \cdot \text{BM25} + (1 - \lambda) \cdot \text{score}$$

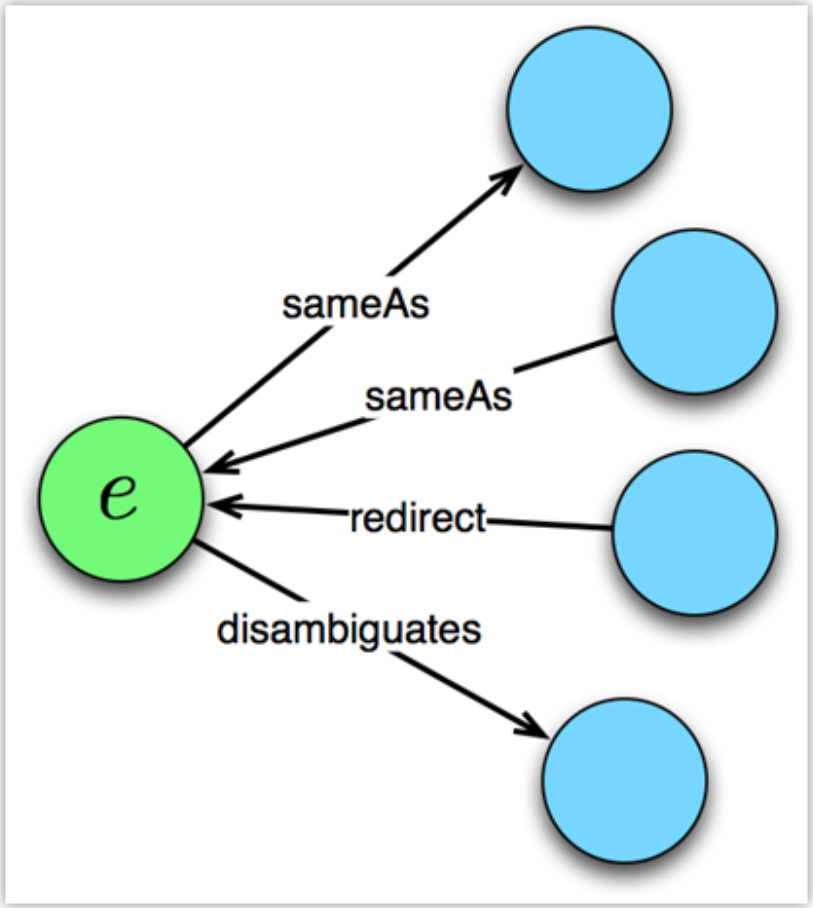
1. Retrieve an initial list of entities matching the query using standard retrieval function (BM25)
2. Expand the retrieved results by exploiting the structure of the knowledge graph (retrieved entities can be used as starting points for simple graph traversals, i.e. finding neighbors)
3. Filter out expanded results removing those with low similarity to the original query
4. Re-rank the results

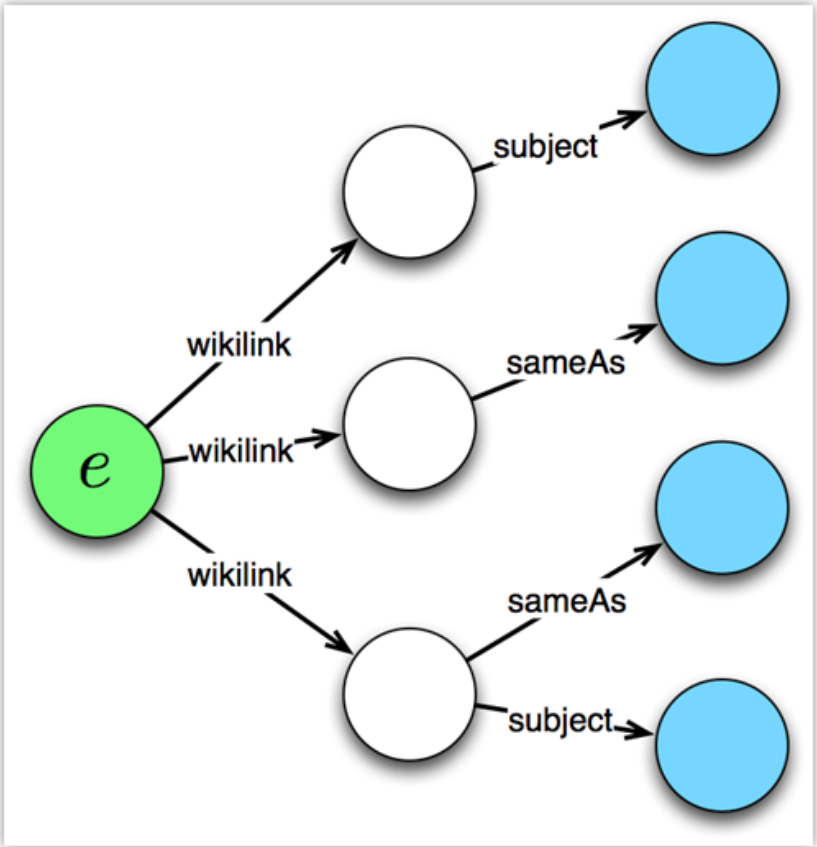
Result Expansion Strategies



- Follow predicates leading to other entities
- Follow datatype properties leading to additional entity attributes
- Explore just the neighborhood of a node and the neighbors of neighbors

Predicates to Follow





Results

	2010 Collection			2011 Collection		
	MAP	P10	NDCG	MAP	P10	NDCG
BM25	0.2070	0.3348	0.5920	0.1484	0.2020	0.4267
SAMEAS	0.2293* (+11%)	0.363* (+8%)	0.5932 (+0%)	0.1612 (+9%)	**0.2200** (+9%)	0.4433 (+4%)
S1_1	**0.2586*** (+25%)	**0.3848*** (+15%)	0.5965 (+1%)	0.1657 (+12%)	0.2140 (+6%)	0.4426 (+4%)
S1_2	0.2305* (+11%)	0.3217 (-4%)	0.5724* (-3%)	**0.1731** (+17%)	0.2180 (+8%)	**0.4532** (+6%)
S1_3	0.2306* (+11%)	0.3217 (-4%)	0.5721* (-4%)	0.1716 (+16%)	0.2140 (+6%)	0.4501 (+5%)
S2_1	0.2118 (+2%)	0.3370 (+1%)	0.5971 (+1%)	0.1550 (+4%)	0.2060 (+2%)	0.4376 (+3%)
S2_2	0.2118 (+2%)	0.3370 (+1%)	0.5965 (+1%)	0.1555 (+5%)	0.2080 (+3%)	0.4379 (+3%)
S2_3	0.2113 (+2%)	0.3402 (+2%)	**0.5978** (+1%)	0.1589 (+7%)	0.2120 (+5%)	0.4385 (+3%)

- The simple `S1_1` approach which exploits <`owl:sameAs`> links plus Wikipedia redirect and disambiguation information performs best obtaining 25% improvement of MAP over the BM25 baseline on the 2010 datatset

Learning-to-Rank Method for Entity Retrieval

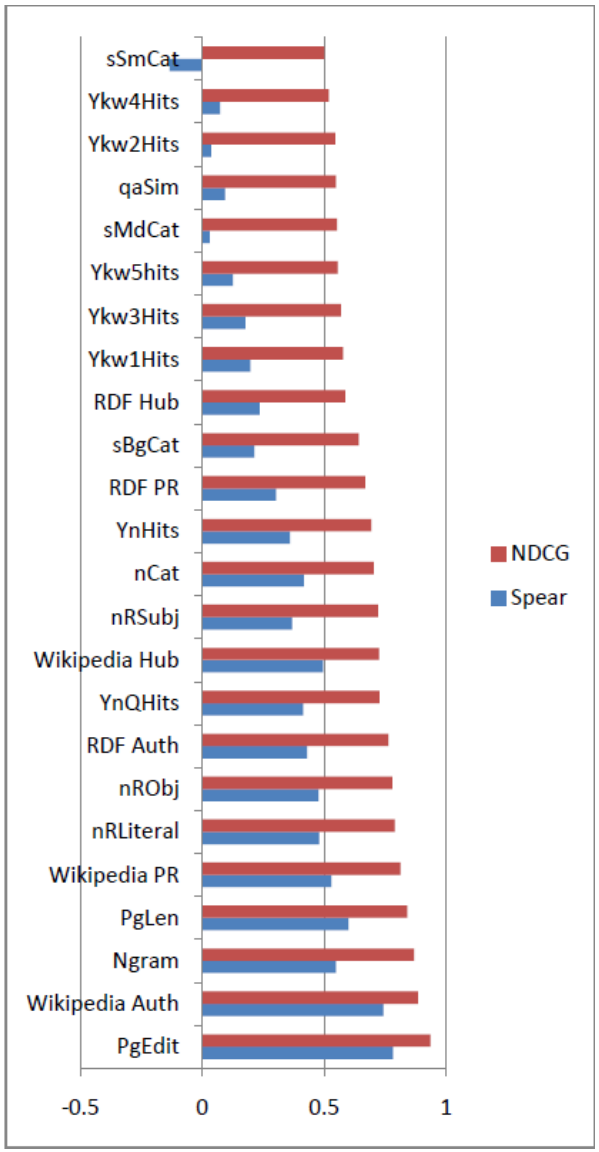
[Dali and Fortuna, WWW'11]

- Variety of features:
  - Popularity and importance of Wikipedia page: # of accesses from logs, # of edits, page length
  - RDF features: # of triples E is subject/object/subject and object is a literal, # of categories Wikipedia page for E belongs to, size of the biggest/smallest/median category
  - HITS scores and Pagerank of Wikipedia page and E in the RDF graph
  - # of hits from search engine API for the top 5 keywords from the abstract of Wikipedia page for E
  - Count of entity name in Google N-grams
- RankSVM learning-to-rank method

Evaluation

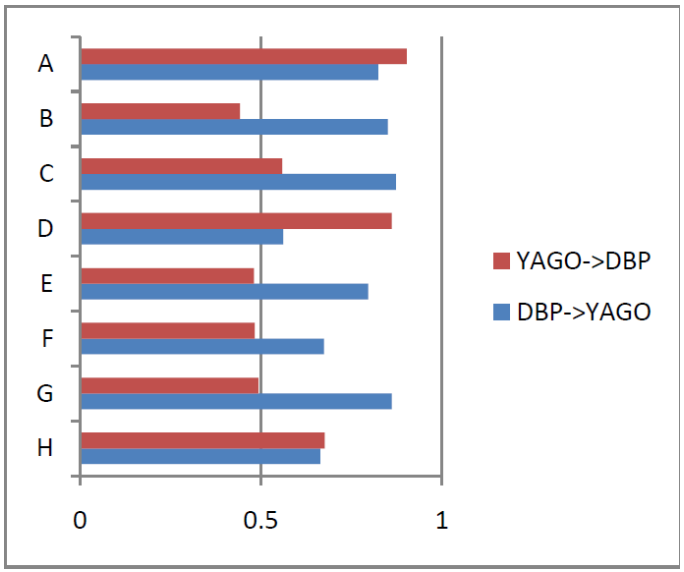
- Initial set of entities obtained using SPARQL queries
- 14 example queries for DBpedia and 27 example queries for Yago
- Example queries: “Which athlete was born in Philadelphia?”, “List of Schalke 04 players”, “Which countries have French as an official language?”, “Which objects are heavier that the Iosif Stalin tank?”

Feature Importance



- Features approximating the importance, hub and authority scores, PageRank of Wikipedia page are effective
- Google N-grams is effective proxy for entity popularity, cheaper than search engine API
- PageRank and HITS scores on RDF graph are not effective (outperformed by simpler RDF features)
- Feature combinations improve both robustness and accuracy of ranking

Transfer Learning



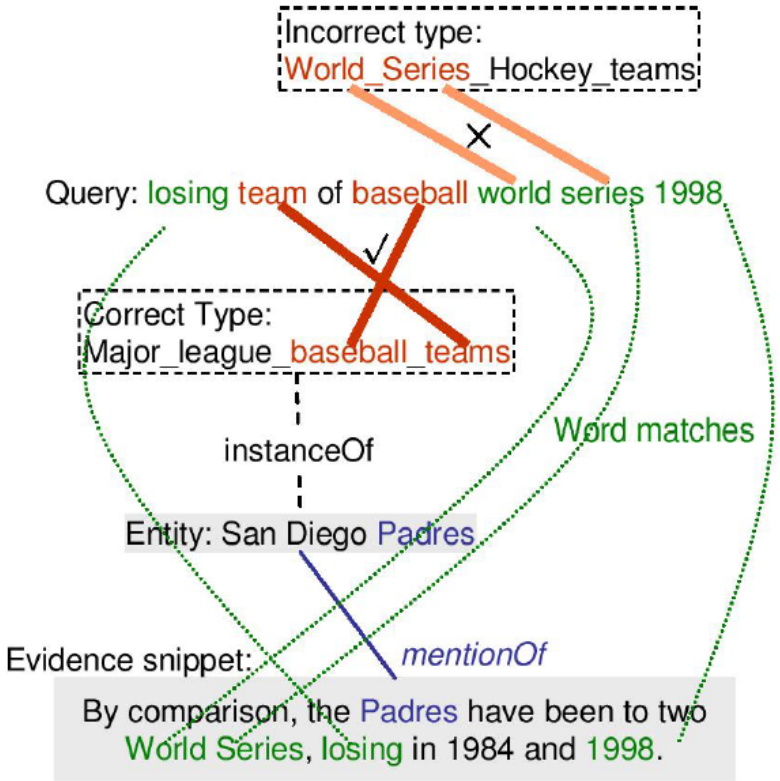
- Ranking model was trained on DBpedia questions and applied to Yago questions
- Only feature set A (all features) results in robust ranking model transfer
- In general, the ranking models for different knowledge graphs are non-transferable, unless they have been learned on large number of features
- The biggest inconsistencies occur on the models trained on graph based features $\rightarrow$ knowledge graphs preserve particularities reflecting their designer decisions

Joint Type Detection and Entity Ranking

[Sawant and Chakrabarti, WWW'13]

- Method for answering "telegraphic" queries with target type
  - woodrow wilson president university
  - dolly clone institute
  - lead singer led zeppelin band
- Integrates type detection into ranking and considers multiple query interpretations
- Has generative and discriminative formulations

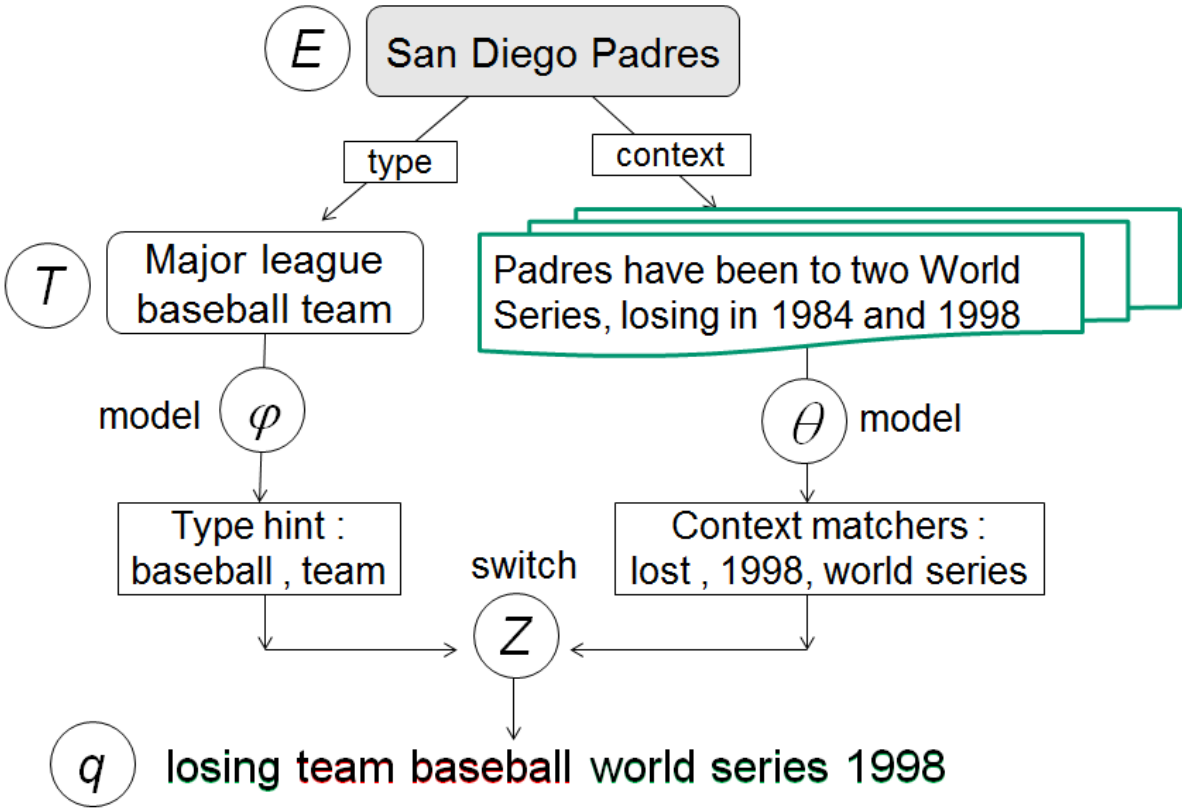
Method



- All possible $2^{|q|}$ query segmentations are considered
- Each query term is either a "type hint" or a "word matcher"

Generative approach

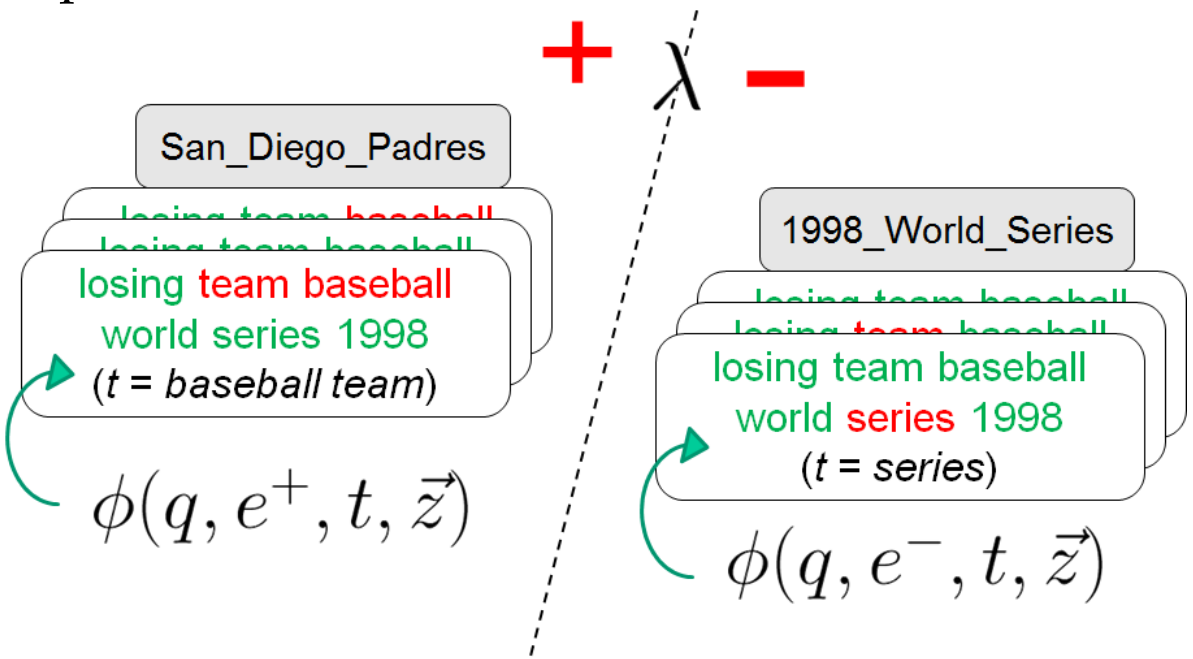
Generate query from entity



$$P(E|Q) \propto P(E) \sum_{t,\vec{z}} P(t|E)P(\vec{z})P(h(\vec{q},\vec{z})|t)P(s(\vec{q},\vec{z})|E)$$

Discriminative approach

Separate correct and incorrect entities



$$\phi(q, e, t, \vec{z}) = \langle \phi_1(q, e), \phi_2(t, e), \phi_3(q, \vec{z}, t), \phi_4(q, \vec{z}, e) \rangle$$

Fielded Sequential Dependence Model

[Zhiltsov, Kotov et al., SIGIR'15]

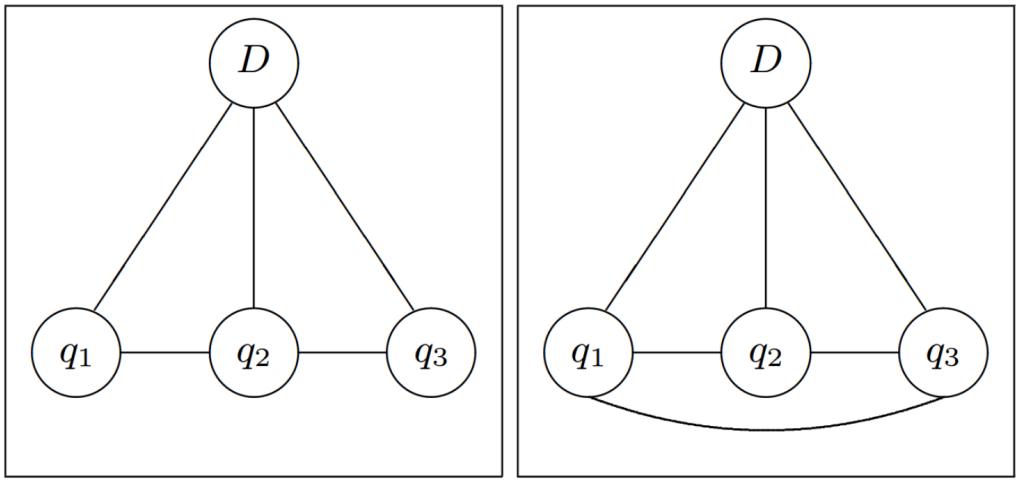
Previous research in ad-hoc IR has focused on two major directions:

- unigram bag-of-words retrieval models for multi-fielded documents
  - Ogilvie and Callan. Combining Document Representations for Known-item Search, SIGIR'03
  - Robertson et al. Simple BM25 Extension to Multiple Weighted Fields, CIKM'04
- retrieval models incorporating term dependencies
  - Metzler and Croft. A Markov Random Field Model for Term Dependencies, SIGIR'05
  - Huston and Croft. A Comparison of Retrieval Models using Term Dependencies, CIKM'14

Goal: to develop a retrieval model that captures both document structure and term dependencies

Sequential and Full Dependence Models

[Metzler and Croft, SIGIR'05]



Ranks w.r.t. $P_\Lambda(D|Q) = \sum_{i \in \{T,U,O\}} \lambda_i f_i(Q,D)$
Potential function for unigrams is QL:

$$f_T(q_i, D) = \log P(q_i|\theta_D) = \log \frac{tf_{q_i,D} + \mu \frac{cf_{q_i}}{|C|}}{|D| + \mu}$$

SDM only considers two-word sequences in queries, FDM considers all two-word combinations.

FSDM ranking function

FSDM incorporates document structure and term dependencies with the following ranking function:

$$P_\Lambda(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, D) + \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, D)$$

Separate MLMs for bigrams and unigrams give FSDM the flexibility to adjust the document scoring depending on the query type

MLM is a special case of FSDM, when $\lambda_T = 1, \lambda_O = 0, \lambda_U = 0$

FSDM ranking function

FSDM incorporates document structure and term dependencies with the following ranking function:

$$P_\Lambda(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, D) + \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, D)$$

Separate MLMs for bigrams and unigrams give FSDM the flexibility to adjust the document scoring depending on the query type

MLM is a special case of FSDM, when $\lambda_T = 1, \lambda_O = 0, \lambda_U = 0$

FSDM ranking function

FSDM incorporates document structure and term dependencies with the following ranking function:

$$P_\Lambda(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, D) + \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, D)$$

Separate MLMs for bigrams and unigrams give FSDM the flexibility to adjust the document scoring depending on the query type

MLM is a special case of FSDM, when $\lambda_T = 1, \lambda_O = 0, \lambda_U = 0$

FSDM ranking function

FSDM incorporates document structure and term dependencies with the following ranking function:

$$P_\Lambda(D|Q) \overset{rank}{=} \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, D) + \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, D) + \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, D)$$

Separate MLMs for bigrams and unigrams give FSDM the flexibility to adjust the document scoring depending on the query type

MLM is a special case of FSDM, when $\lambda_T = 1, \lambda_O = 0, \lambda_U = 0$

5-field Entity Document

[Zhiltsov, Kotov et al., SIGIR'15]

Each entity is represented as a five-field document:

names
: conventional names of the entities, such as the name of a person or the name of an organization

attributes
: all entity properties, other than names

categories
: classes or groups, to which the entity has been assigned

similar entity names
: names of the entities that are very similar or identical to a given entity

related entity names
: names of the entities that are part of the same RDF triple

FSDM ranking function

Potential function for unigrams in case of FSDM:

$$\tilde{f}_T(q_i, D) = \log \sum_j w_j^T P(q_i|\theta_D^j) = \log \sum_j w_j^T \frac{tf_{q_i,D^j} + \mu_j \frac{cf_{q_i}^j}{|C_j|}}{|D^j| + \mu_j}$$

Example

apollo astronauts who walked on the moon

FSDM ranking function

Potential function for unigrams in case of FSDM:

$$\tilde{f}_T(q_i, D) = \log \sum_j w_j^T P(q_i|\theta_D^j) = \log \sum_j w_j^T \frac{tf_{q_i,D^j} + \mu_j \frac{cf_{q_i}^j}{|C_j|}}{|D^j| + \mu_j}$$

Example

apollo astronauts who walked on the moon
category

FSDM ranking function

Potential function for unigrams in case of FSDM:

$$\tilde{f}_T(q_i, D) = \log \sum_j w_j^T P(q_i|\theta_D^j) = \log \sum_j w_j^T \frac{tf_{q_i,D^j} + \mu_j \frac{cf_{q_i}^j}{|C_j|}}{|D^j| + \mu_j}$$

Example

apollo astronauts who walked on the moon
category attribute

Parameters of FSDM

Overall, FSDM has $3 * F + 3$ free parameters: $\langle \boldsymbol{w}^T, \boldsymbol{w}^O, \boldsymbol{w}^U, \boldsymbol{\lambda} \rangle$.

Properties of ranking function

1. Linearity with respect to $\boldsymbol{\lambda}$.

   We can apply any linear learning-to-rank algorithm to optimize the ranking function with respect to $\boldsymbol{\lambda}$.

2. Linearity with respect to $\boldsymbol{w}$ of the arguments of monotonic $\tilde{f}(\cdot)$ functions.

   Optimization of the arguments as linear functions with respect to $\boldsymbol{w}$, leads to optimization of each function $\tilde{f}(\cdot)$.

Optimization algorithm

1: $Q \leftarrow$ Training queries
2: **for** $s \in \{T, O, U\}$ **do** // *Optimize field weights of LMs independently*
3: $\quad \boldsymbol{\lambda} = \boldsymbol{e}_s$
4: $\quad \hat{\boldsymbol{w}}^s \leftarrow CoordAsc(Q, \boldsymbol{\lambda})$
5: **end for**
6: $\hat{\boldsymbol{\lambda}} \leftarrow CoordAsc(Q, \hat{\boldsymbol{w}}_T, \hat{\boldsymbol{w}}_O, \hat{\boldsymbol{w}}_U)$ // *Optimize* $\boldsymbol{\lambda}$

The unit vectors $\boldsymbol{e}_T = (1, 0, 0)$, $\boldsymbol{e}_O = (0, 1, 0)$, $\boldsymbol{e}_U = (0, 0, 1)$ are the corresponding settings of the parameters $\boldsymbol{\lambda}$ in the formula of FSDM ranking function.

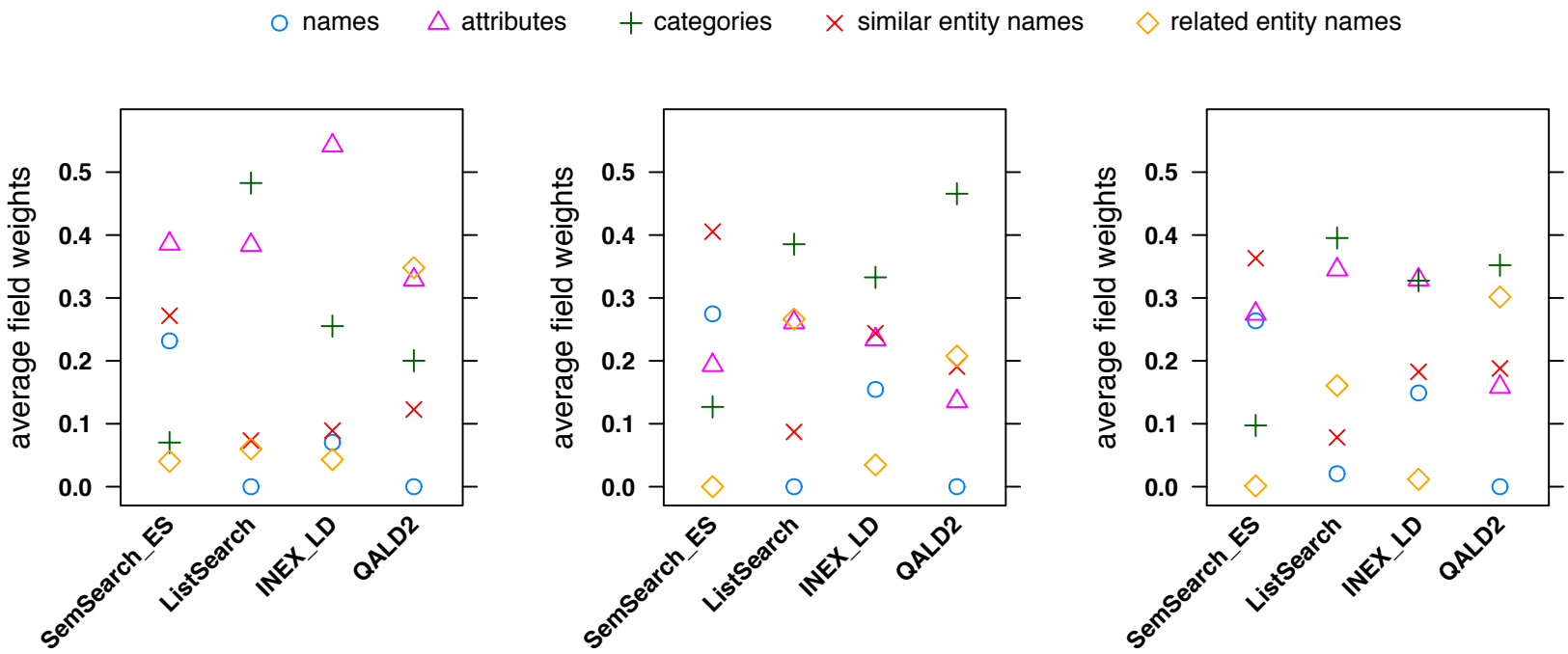
$\Rightarrow$ direct optimization w.r.t. target metric, e.g. MAP

Collection and Query Sets

- DBPedia 3.7 was used as a knowledge
- Queries from Balog and Neumayer. A Test Collection for Entity Search in DBpedia, SIGIR'13.

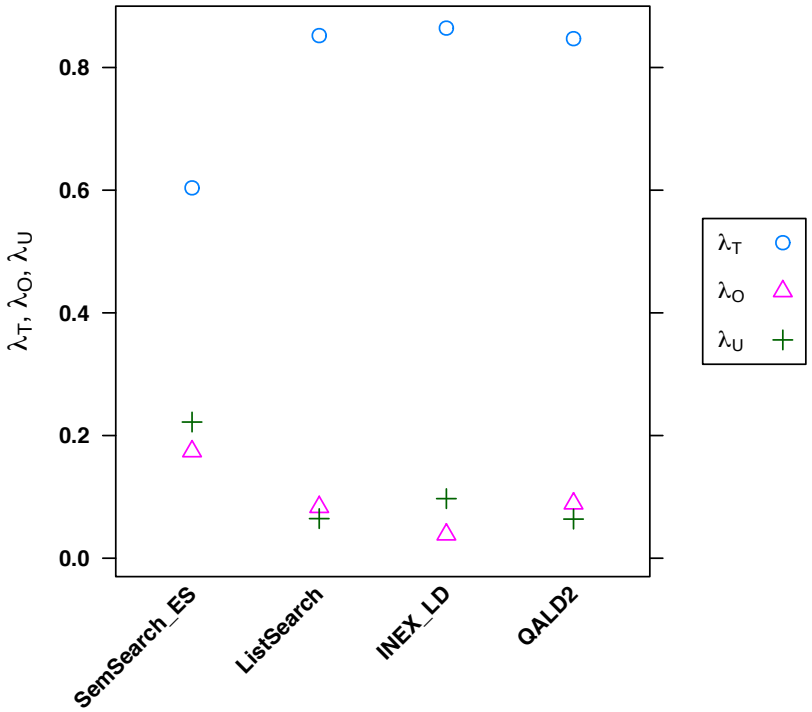
Query set	**Amount**	**Query types [Pound et al., 2010]**
SemSearch ES	130	Entity
ListSearch	115	Type
INEX-LD	100	Entity, Type, Attribute, Relation
QALD-2	140	Entity, Type, Attribute, Relation

Tuning field weights

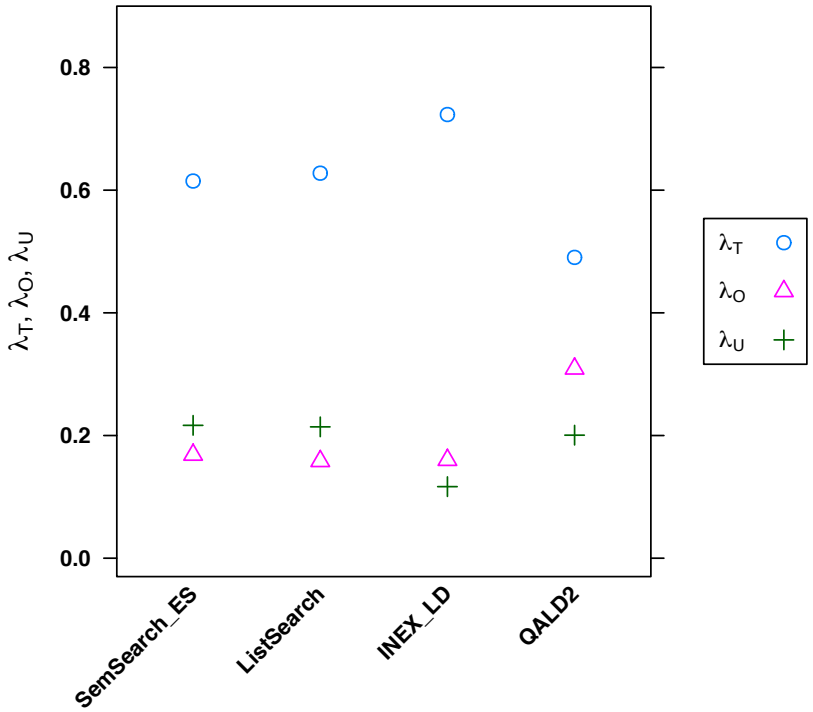


- *Attributes* field is consistently considered to be a very valuable for both unigrams and bigrams.
- The *names* field as well as the *similar entity names* field are highly important for queries aiming at finding named entities.
- Distinguishing *categories* from *related entity names* is particularly important for type queries.

Tuning λ



(a) SDM



(b) FSDM

- Bigram matches are important for named entity queries.
- Transformation of SDM into FSDM increases the importance of bigram matches, which ultimately improves the retrieval performance

Experimental results

Query set	Method	MAP	P@10	P@20	b-pref
SemSearch ES	MLM-CA	0.320	0.250	0.179	0.674
	SDM-CA	0.254*	0.202*	0.149*	0.671
	FSDM	**0.386**$^{*}_{\dagger}$	**0.286**$^{*}_{\dagger}$	**0.204**$^{*}_{\dagger}$	**0.750**$^{*}_{\dagger}$
ListSearch	MLM-CA	0.190	0.252	0.192	0.428
	SDM-CA	0.197	0.252	0.202	**0.471***
	FSDM	**0.203**	**0.256**	**0.203**	0.466*
INEX-LD	MLM-CA	0.102	0.238	0.190	0.318
	SDM-CA	**0.117***	0.258	0.199	0.335
	FSDM	0.111*	**0.263***	**0.215**$^{*}_{\dagger}$	**0.341***
QALD-2	MLM-CA	0.152	0.103	0.084	0.373
	SDM-CA	0.184	0.106	0.090	0.465*
	FSDM	**0.195***	**0.136**$^{*}_{\dagger}$	**0.111***	**0.466***
All queries	MLM-CA	0.196	0.206	0.157	0.455
	SDM-CA	0.192	0.198	0.155	0.495*
	FSDM	**0.231**$^{*}_{\dagger}$	**0.231**$^{*}_{\dagger}$	**0.179**$^{*}_{\dagger}$	**0.517**$^{*}_{\dagger}$

FSDM limitation

In FSDM field weights are the same for all query concepts of the same type.

Example

capitals in Europe which were host cities of summer Olympic games

Parametric extension of FSDM

$$w^T_{q_i,j} = \sum_k \alpha^U_{j,k} \phi_k(q_i, j)$$

- $\phi_k(q_i, j)$ is the the k-th feature value for unigram q_i in field j.
- $\alpha^U_{j,k}$ are feature weights that we learn.

$$\sum_j w^T_{q_i,j} = 1, w^T_{q_i,j} \geq 0, \alpha^U_{j,k} \geq 0, 0 \leq \phi_k(q_i, j) \leq 1$$

PFFDM is the same, but uses full dependence model.

Parametric extension of FSDM

$$w^T_{q_i,j} = \sum_k \alpha^U_{j,k} \phi_k(q_i, j)$$

- $\phi_k(q_i, j)$ is the the k-th feature value for unigram q_i in field j.
- $\alpha^U_{j,k}$ are feature weights that we learn.

$$\sum_j w^T_{q_i,j} = 1, w^T_{q_i,j} \geq 0, \alpha^U_{j,k} \geq 0, 0 \leq \phi_k(q_i, j) \leq 1$$

PFFDM is the same, but uses full dependence model.

Parametric extension of FSDM

$$w^T_{q_i,j} = \sum_k \alpha^U_{j,k} \phi_k(q_i, j)$$

- $\phi_k(q_i, j)$ is the the k-th feature value for unigram q_i in field j.
- $\alpha^U_{j,k}$ are feature weights that we learn.

$$\sum_j w^T_{q_i,j} = 1, w^T_{q_i,j} \geq 0, \alpha^U_{j,k} \geq 0, 0 \leq \phi_k(q_i, j) \leq 1$$

PFFDM is the same, but uses full dependence model.

Parametric extension of FSDM

$$w^T_{q_i,j} = \sum_k \alpha^U_{j,k} \phi_k(q_i, j)$$

- $\phi_k(q_i, j)$ is the the k-th feature value for unigram q_i in field j.
- $\alpha^U_{j,k}$ are feature weights that we learn.

$$\sum_j w^T_{q_i,j} = 1, w^T_{q_i,j} \geq 0, \alpha^U_{j,k} \geq 0, 0 \leq \phi_k(q_i, j) \leq 1$$

PFFDM is the same, but uses full dependence model.

Parametric extension of FSDM

$$w^T_{q_i,j} = \sum_k \alpha^U_{j,k} \phi_k(q_i, j)$$

- $\phi_k(q_i, j)$ is the the k-th feature value for unigram q_i in field j.
- $\alpha^U_{j,k}$ are feature weights that we learn.

$$\sum_j w^T_{q_i,j} = 1, w^T_{q_i,j} \geq 0, \alpha^U_{j,k} \geq 0, 0 \leq \phi_k(q_i, j) \leq 1$$

PFFDM is the same, but uses full dependence model.

Features

Source	Feature	Description	CT
Collection statistics	$FP(\kappa, j)$	Posterior probability $P(E_j\|w)$.	UG BG
	$TS(\kappa, j)$	Top SDM score on j-th field when κ is used as a query.	BG
Stanford POS Tagger	$NNP(\kappa)$	Is concept κ a proper noun?	UG
	$NNS(\kappa)$	Is κ a plural non-proper noun?	UG BG
	$JJS(\kappa)$	Is κ a superlative adjective?	UG
Stanford Parser	$NPP(\kappa)$	Is κ part of a noun phrase?	BG
	$NNO(\kappa)$	Is κ the only singular non-proper noun in a noun phrase?	UG
	INT	Intercept feature ($= 1$).	UG BG

Features

Source	Feature	Description	CT
Collection statistics	$FP(\kappa, j)$	Posterior probability $P(E_j\|w)$.	UG BG
	$TS(\kappa, j)$	Top SDM score on j-th field when κ is used as a query.	BG
Stanford POS Tagger	$NNP(\kappa)$	Is concept κ a proper noun?	UG
	$NNS(\kappa)$	Is κ a plural non-proper noun?	UG BG
	$JJS(\kappa)$	Is κ a superlative adjective?	UG
Stanford Parser	$NPP(\kappa)$	Is κ part of a noun phrase?	BG
	$NNO(\kappa)$	Is κ the only singular non-proper noun in a noun phrase?	UG
	INT	Intercept feature ($= 1$).	UG BG

Parameters of PFSDM

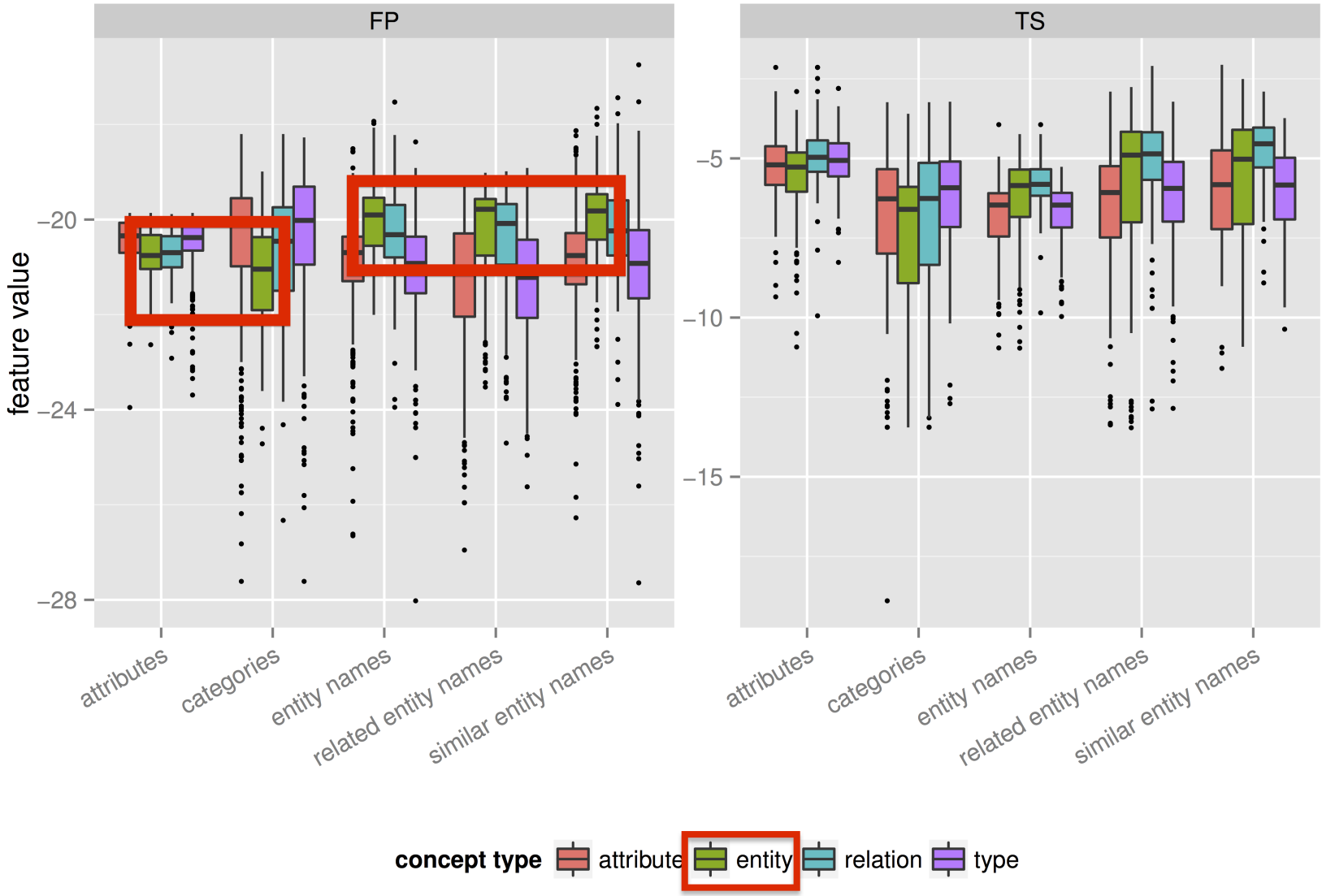
Both PFSDM and PFFDM have $F * U + F * B + 3$ free parameters: $\langle \hat{\boldsymbol{\alpha}}^U, \hat{\boldsymbol{\alpha}}^B, \hat{\boldsymbol{\lambda}} \rangle$.

We perform direct optimization w.r.t. target metric (e.g. MAP) using coordinate ascent.

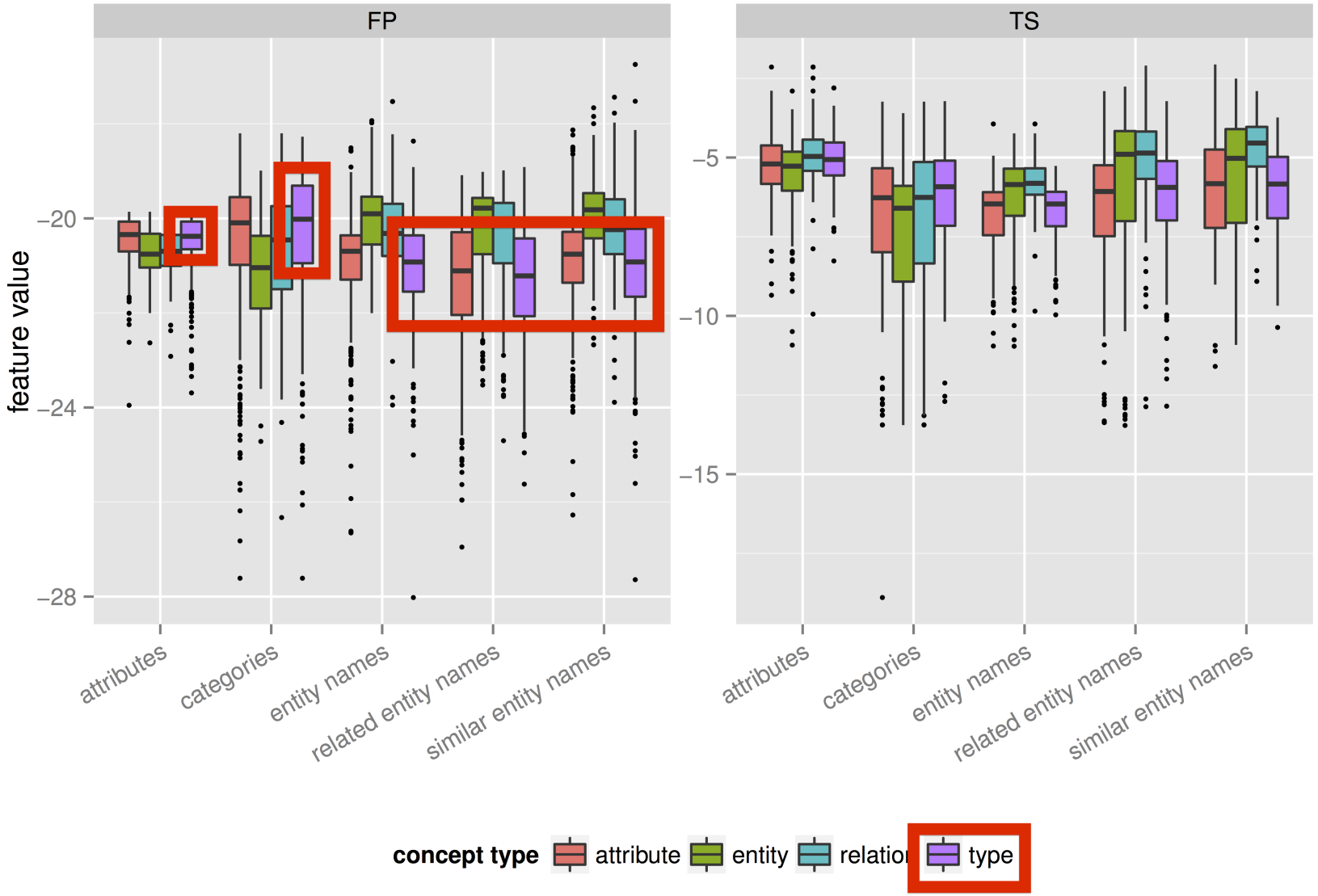
Collections

1. DBPedia 3.7
   - Structured version of on-line encyclopedia Wikipedia
   - Provides the descriptions of over 3.5 million entities belonging to 320 classes
2. BTC-2009
   - Contains entities from multiple knowledge bases.
   - Consists of 1.14 billion RDF triples.

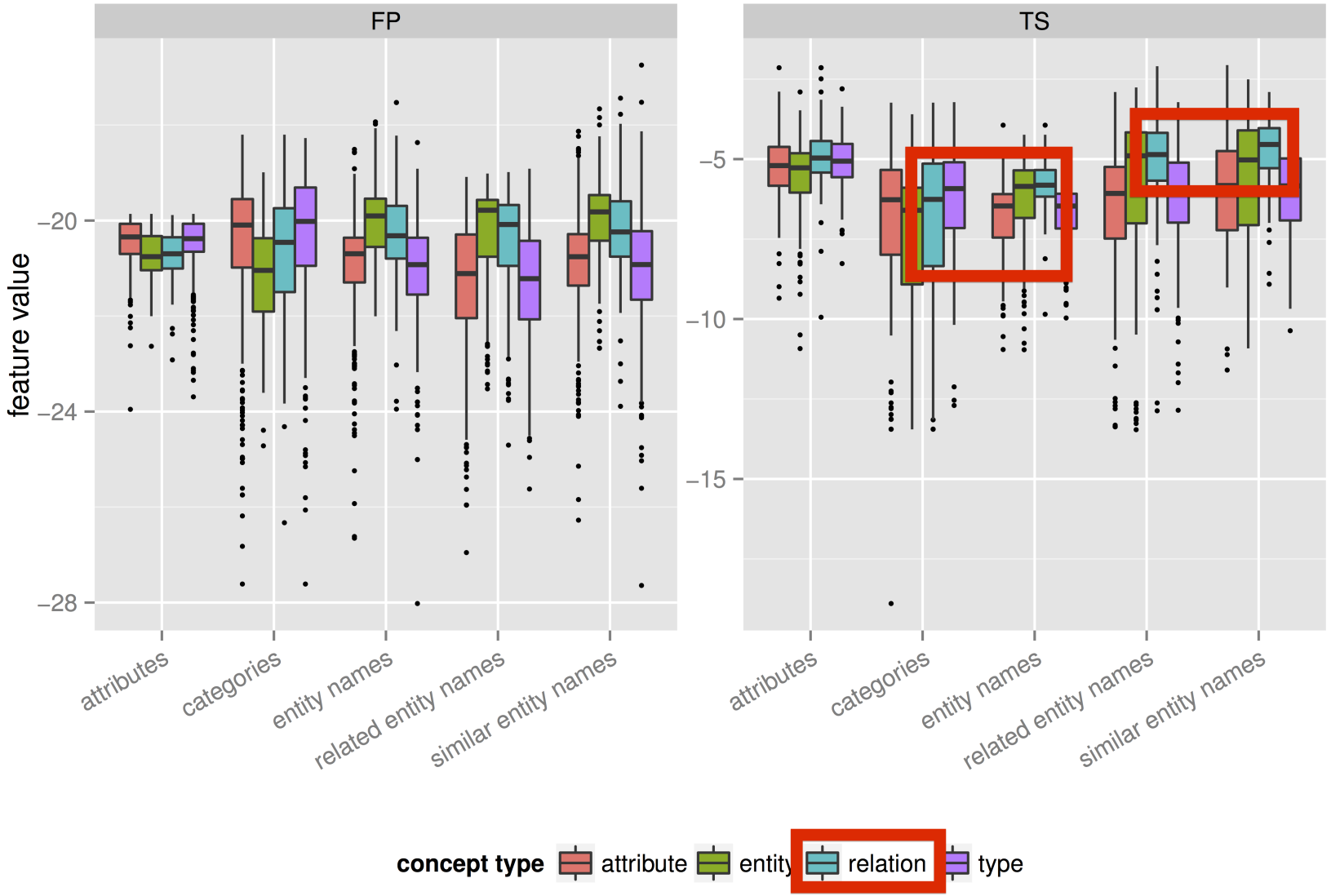
Real-valued features analysis



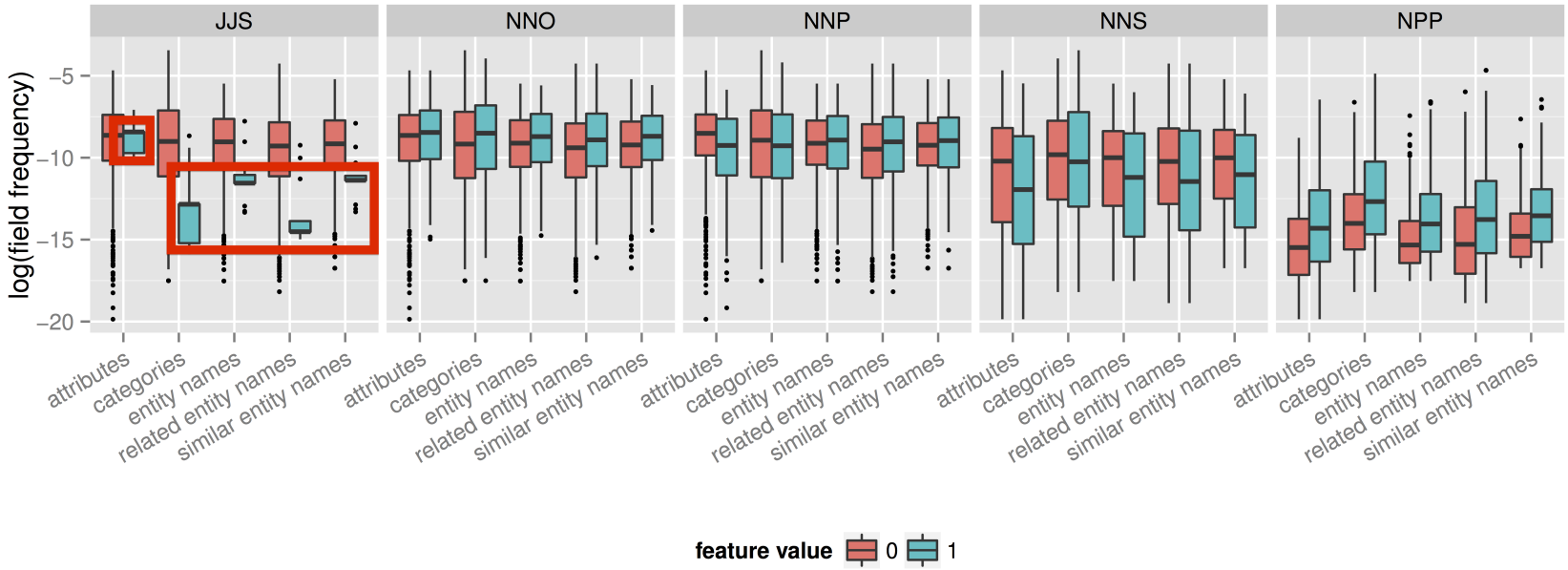
Real-valued features analysis



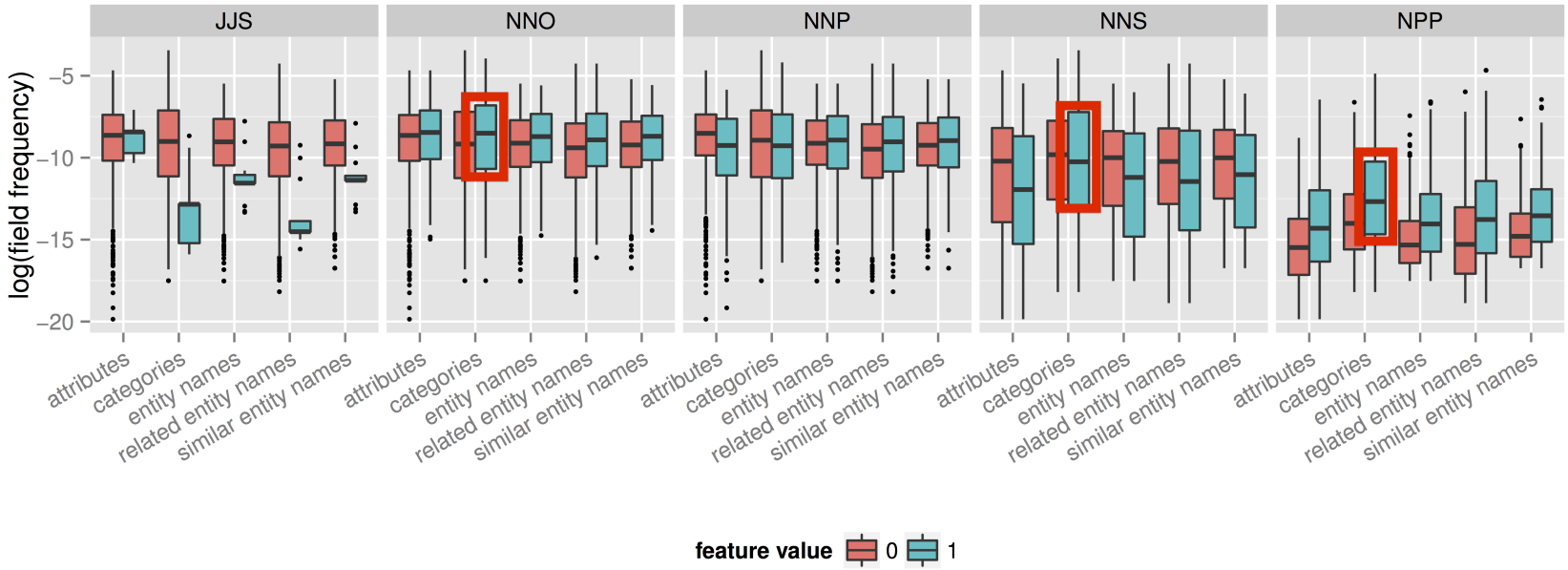
Real-valued features analysis



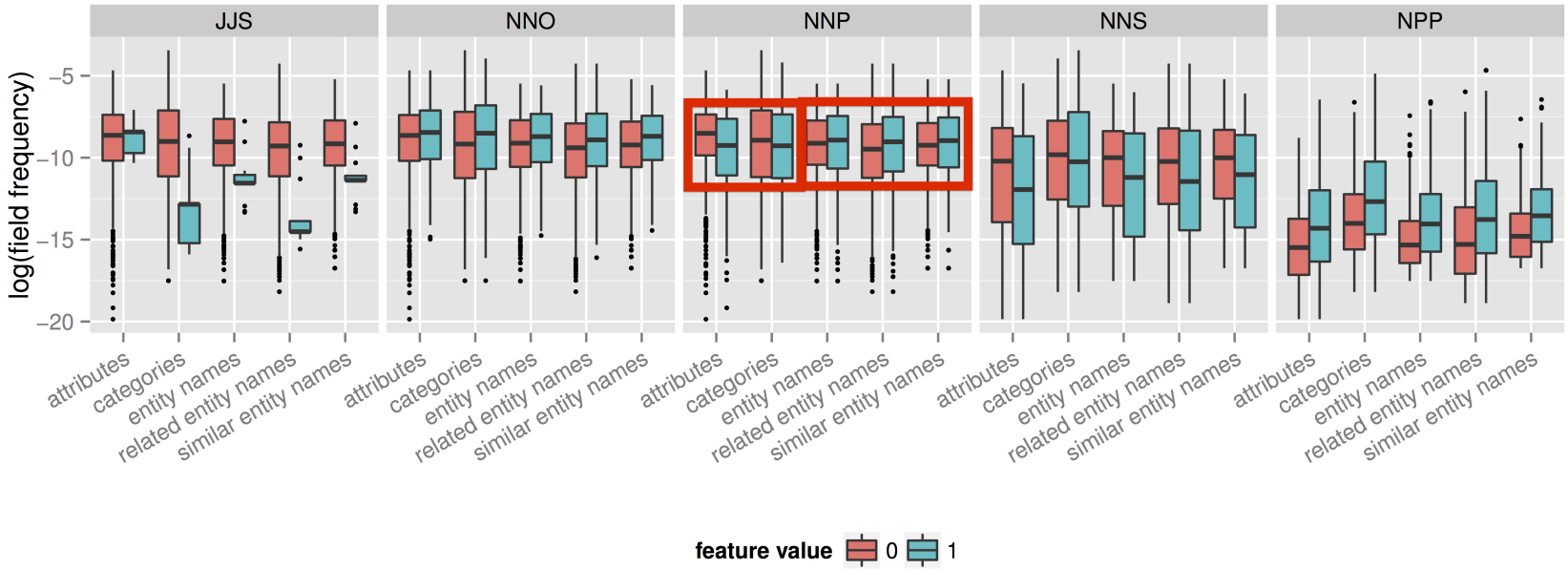
NLP-based features analysis



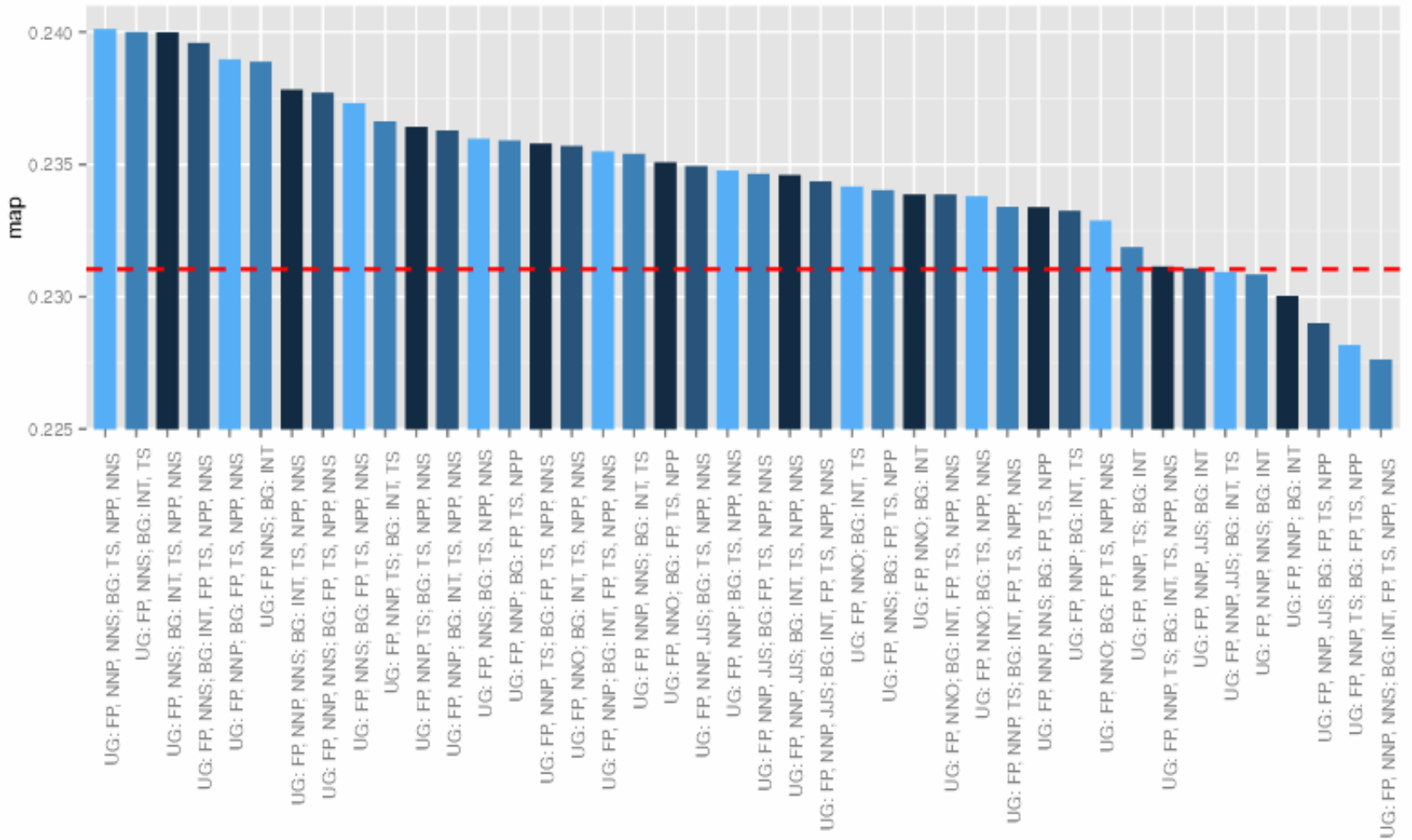
NLP-based features analysis



NLP-based features analysis



Feature Effectiveness



DBpedia results (using best features combination)

Query set	Method	MAP	P@10	P@20	b-pref
SemSearch ES	PRMS	0.230	0.177	0.549	0.317
	FSDM	0.386	0.286	0.737	0.476
	PFSDM	0.394*	0.286*	0.757*	0.494*†
	FFDM	0.389*	0.286*	0.734*	0.479*
	PFFDM	0.380*	0.286*	0.739*	0.477*
ListSearch	PRMS	0.111	0.154	0.355	0.176
	FSDM	0.203	0.256	0.447	0.274
	PFSDM	0.201*	0.253*	0.443*	0.278*
	FFDM	0.226*†	0.282*†	0.499*†	0.313*†
	PFFDM	0.228*†	0.286*†	0.487*	0.302*†
INEX-LD	PRMS	0.064	0.145	0.409	0.216
	FSDM	0.111	0.263	0.546	0.322
	PFSDM	0.116*	0.259*	0.579*	0.341*
	FFDM	0.122*†	0.273*	0.560*	0.345*†
	PFFDM	0.121*†	0.274*	0.556*	0.343*
QALD-2	PRMS	0.120	0.079	0.188	0.147
	FSDM	0.195	0.136	0.283	0.229
	PFSDM	0.218*†	0.140*	0.308*	0.253*†
	FFDM	0.200*	0.139*	0.292*	0.237*
	PFFDM	0.219*†	0.147*	0.310*	0.267*†
All queries	PRMS	0.136	0.136	0.370	0.214
	FSDM	0.231	0.231	0.498	0.325
	PFSDM	0.240*†	0.231*	0.516*†	0.342*†
	FFDM	0.241*†	0.240*†	0.515*†	0.342*†
	PFFDM	0.244*†	0.244*†	0.518*†	0.347*†

DBpedia results (using best features combination)

Query set	Method	MAP	P@10	P@20	b-pref
SemSearch ES	PRMS	0.230	0.177	0.549	0.317
	FSDM	0.386	0.286	0.737	0.476
	PFSDM	0.394*	0.286*	0.757*	0.494*†
	FFDM	0.389*	0.286*	0.734*	0.479*†
	PFFDM	0.380*	0.286*	0.739*	0.477*
ListSearch	PRMS	0.111	0.154	0.355	0.176
	FSDM	0.203	0.256	0.447	0.274
	PFSDM	0.201*	0.253*	0.443*	0.278*
	FFDM	0.226*†	0.282*†	0.499*†	0.313*†
	PFFDM	0.228*†	0.286*†	0.487*	0.302*†
INEX-LD	PRMS	0.064	0.145	0.409	0.216
	FSDM	0.111	0.263	0.546	0.322
	PFSDM	0.116*	0.259*	0.579*	0.341*
	FFDM	0.122*†	0.273*	0.560*	0.345*†
	PFFDM	0.121*†	0.274*	0.556*	0.343*
QALD-2	PRMS	0.120	0.079	0.188	0.147
	FSDM	0.195	0.136	0.283	0.229
	PFSDM	0.218*†	0.140*	0.308*	0.253*†
	FFDM	0.200*	0.139*	0.292*	0.237*
	PFFDM	0.219*†	0.147*	0.310*	0.267*†
All queries	PRMS	0.136	0.136	0.370	0.214
	FSDM	0.231	0.231	0.498	0.325
	PFSDM	0.240*†	0.231*	0.516*†	0.342*†
	FFDM	0.241*†	0.240*†	0.515*†	0.342*†
	PFFDM	0.244*†	0.244*†	0.518*†	0.347*†

DBpedia results (using best features combination)

Query set	Method	MAP	P@10	P@20	b-pref
SemSearch ES	PRMS	0.230	0.177	0.549	0.317
	FSDM	0.386	0.286	0.737	0.476
	PFSDM	0.394*	0.286*	0.757*	0.494*†
	FFDM	0.389*	0.286*	0.734*	0.479*†
	PFFDM	0.380*	0.286*	0.739*	0.477*
ListSearch	PRMS	0.111	0.154	0.355	0.176
	FSDM	0.203	0.256	0.447	0.274
	PFSDM	0.201*	0.253*	0.443*	0.278*
	FFDM	0.226*†	0.282*†	0.499*†	0.313*†
	PFFDM	0.228*†	0.286*†	0.487*	0.302*†
INEX-LD	PRMS	0.064	0.145	0.409	0.216
	FSDM	0.111	0.263	0.546	0.322
	PFSDM	0.116*	0.259*	0.579*	0.341*
	FFDM	0.122*†	0.273*	0.560*	0.345*†
	PFFDM	0.121*†	0.274*	0.556*	0.343*
QALD-2	PRMS	0.120	0.079	0.188	0.147
	FSDM	0.195	0.136	0.283	0.229
	PFSDM	0.218*†	0.140*	0.308*	0.253*†
	FFDM	0.200*	0.139*	0.292*	0.237*
	PFFDM	0.219*†	0.147*	0.310*	0.267*†
All queries	PRMS	0.136	0.136	0.370	0.214
	FSDM	0.231	0.231	0.498	0.325
	PFSDM	0.240*†	0.231*	0.516*†	0.342*†
	FFDM	0.241*†	0.240*†	0.515*†	0.342*†
	PFFDM	0.244*†	0.244*†	0.518*†	0.347*†

DBpedia results (using best features combination)

Query set	Method	MAP	P@10	P@20	b-pref
SemSearch ES	PRMS	0.230	0.177	0.549	0.317
	FSDM	0.386	0.286	0.737	0.476
	PFSDM	0.394*	0.286*	0.757*	0.494*†
	FFDM	0.389*	0.286*	0.734*	0.479*†
	PFFDM	0.380*	0.286*	0.739*	0.477*
ListSearch	PRMS	0.111	0.154	0.355	0.176
	FSDM	0.203	0.256	0.447	0.274
	PFSDM	0.201*	0.253*	0.443*	0.278*
	FFDM	0.226*†	0.282*†	0.499*†	0.313*†
	PFFDM	0.228*†	0.286*†	0.487*	0.302*†
INEX-LD	PRMS	0.064	0.145	0.409	0.216
	FSDM	0.111	0.263	0.546	0.322
	PFSDM	0.116*	0.259*	0.579*	0.341*
	FFDM	0.122*†	0.273*	0.560*	0.345*†
	PFFDM	0.121*†	0.274*	0.556*	0.343*
QALD-2	PRMS	0.120	0.079	0.188	0.147
	FSDM	0.195	0.136	0.283	0.229
	PFSDM	0.218*†	0.140*	0.308*	0.253*†
	FFDM	0.200*	0.139*	0.292*	0.237*
	PFFDM	0.219*†	0.147*	0.310*	0.267*†
All queries	PRMS	0.136	0.136	0.370	0.214
	FSDM	0.231	0.231	0.498	0.325
	PFSDM	0.240*†	0.231*	0.516*†	0.342*†
	FFDM	0.241*†	0.240*†	0.515*†	0.342*†
	PFFDM	0.244*†	0.244*†	0.518*†	0.347*†

DBpedia results (using best features combination)

Query set	Method	MAP	P@10	P@20	b-pref
SemSearch ES	PRMS	0.230	0.177	0.549	0.317
	FSDM	0.386	0.286	0.737	0.476
	PFSDM	0.394*	0.286*	0.757*	0.494*†
	FFDM	0.389*	0.286*	0.734*	0.479*†
	PFFDM	0.380*	0.286*	0.739*	0.477*
ListSearch	PRMS	0.111	0.154	0.355	0.176
	FSDM	0.203	0.256	0.447	0.274
	PFSDM	0.201*	0.253*	0.443*	0.278*
	FFDM	0.226*†	0.282*†	0.499*†	0.313*†
	PFFDM	0.228*†	0.286*†	0.487*	0.302*†
INEX-LD	PRMS	0.064	0.145	0.409	0.216
	FSDM	0.111	0.263	0.546	0.322
	PFSDM	0.116*	0.259*	0.579*	0.341*
	FFDM	0.122*†	0.273*	0.560*	0.345*†
	PFFDM	0.121*†	0.274*	0.556*	0.343*
QALD-2	PRMS	0.120	0.079	0.188	0.147
	FSDM	0.195	0.136	0.283	0.229
	PFSDM	0.218*†	0.140*	0.308*	0.253*†
	FFDM	0.200*	0.139*	0.292*	0.237*
	PFFDM	0.219*†	0.147*	0.310*	0.267*†
All queries	PRMS	0.136	0.136	0.370	0.214
	FSDM	0.231	0.231	0.498	0.325
	PFSDM	0.240*†	0.231*	0.516*†	0.342*†
	FFDM	0.241*†	0.240*†	0.515*†	0.342*†
	PFFDM	0.244*†	0.244*†	0.518*†	0.347*†

BTC2009 results

Method	MAP	P@10	P@20	b-pref
PRMS	0.098	0.198	0.545	0.269
FSDM	0.171	0.323	0.631	0.358
PFSDM	$0.182^{*}_{\dagger}$	0.335^{*}	$0.657^{*}_{\dagger}$	0.371^{*}
FFDM	$0.180^{*}_{\dagger}$	$0.330^{*}_{\dagger}$	0.647^{*}	$0.373^{*}_{\dagger}$
PFFDM	0.187^{*}	$0.342^{*}_{\dagger}$	0.650^{*}	0.377^{*}

Future work

We hypothesize that FSDM and PFDSM can be effective in other structured information retrieval scenarios, such as product and social graph search, and leave verification of this hypothesis to industry or research community.

Overview

- Code and runs are available at:

  `github.com/teanalab`

- Send me an email at `kotov@wayne.edu`, if you have any questions about this tutorial or would like to collaborate on future projects.

Thank you!