

Entity Representation and Retrieval

Laura Dietz

University of New Hampshire

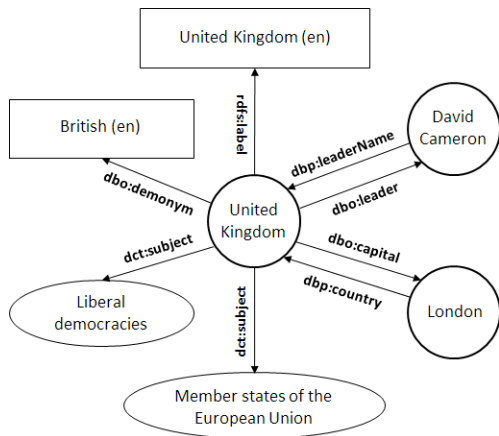
Alexander Kotov

Wayne State University

Edgar Meij

Bloomberg L.P.

Knowledge Graph Fragment



Entity Retrieval

A screenshot of a search engine results page for the query "airlines that fly airbus a380". The search bar at the top contains the query and a magnifying glass icon. Below the search bar, there are navigation tabs for "All", "News", "Images", "Shopping", "Videos", and "More", along with "Settings" and "Tools". The results section shows "About 544,000 results (0.77 seconds)". A list of airlines is displayed, including Singapore Airlines, Emirates, Qantas, Air France, Lufthansa, Korean Air, China Southern Airlines, and Malaysia Airlines. A small thumbnail image of an airplane cabin is visible on the right side of the results.

A screenshot of a search engine results page for the query "us presidents". The search bar at the top contains the query and a magnifying glass icon. Below the search bar, there are navigation tabs for "All", "Images", "News", "Videos", "Books", and "More", along with "Settings" and "Tools". The results section shows "US Presidents" and a grid of six portrait images of US Presidents: Barack Obama (2009 - 2017), George Washington (1789 - 1797), Franklin D. Roosevelt (1933 - 1945), Ronald Reagan (1981 - 1989), John F. Kennedy (1961 - 1963), and Abraham Lincoln (1861 - 1865).

- ▶ Users often search for concrete or abstract objects (i.e. people, products or locations), rather than documents
- ▶ Search results are names of entities or entity representations (i.e. entity cards)
- ▶ Users are willing to express their information need more elaborately than with a few keywords [Balog et al. 2008]
- ▶ Knowledge graphs are perfectly suited for entity retrieval

Typical Entity Retrieval Tasks

- ▶ **Entity Search:** simple queries aimed at finding a particular entity or an entity which is an attribute of another entity
 - ▶ *"Ben Franklin"*
 - ▶ *"Einstein Relativity theory"*
 - ▶ *"England football player highest paid"*
- ▶ **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?"*

Entity Retrieval from Knowledge Graph(s) (ERKG)

- ▶ Assumes *keyword* queries (structured queries are studied in database community)
- ▶ Different from ad-hoc entity retrieval, which is focused on retrieving entities embedded in documents, e.g:
 - ▶ Entity track at TREC 2009–2011
 - ▶ Entity Ranking track at INEX 2007–2009
 - ▶ Expert Finding in Enterprise Search
- ▶ Different from entity linking, which aims at identifying entities mentioned in queries (part 1 of this tutorial)
- ▶ Can be combined with methods using KGs for ad-hoc or Web search (part 3 of this tutorial)

Why ERKG?

- ▶ **Unique IR problem:** there are *no documents*
- ▶ **Challenging IR problem:** knowledge graphs are designed for graph pattern-based SPARQL queries

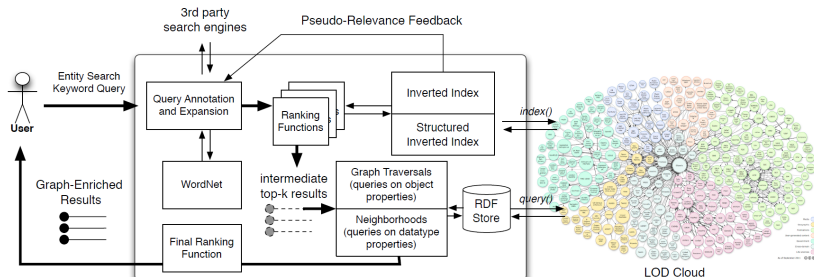
Research challenges in ERKG

ERKG requires accurate interpretation of unstructured textual queries and matching them with entity semantics:

1. How to design entity representations that capture the semantics of entity properties and relations to other entities?
2. How to develop accurate and efficient entity retrieval models?

Architecture of ERKG Methods

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

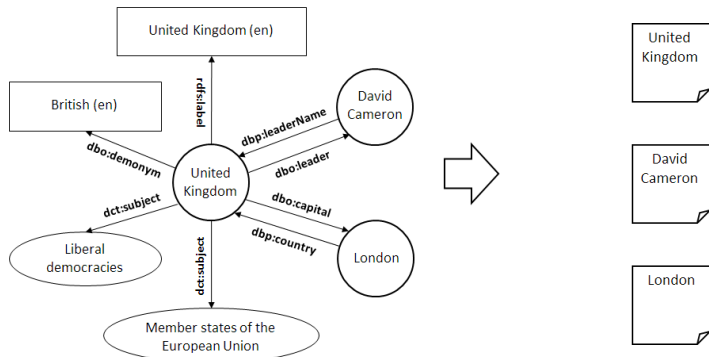


Outline

- ▶ Entity representation
- ▶ Entity retrieval
- ▶ Entity set expansion
- ▶ Entity ranking

Structured Entity Documents

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



Predicate Folding

- ▶ **Simple approach:** each predicate corresponds to one document field
- ▶ **Problem:** there are infinitely many predicates → optimization of field importance weights is computationally intractable
- ▶ **Predicate folding:** group predicates into a small set of predefined categories → entity documents with smaller number of fields
 - ▶ By predicate type (attributes, incoming/outgoing links)[[Pérez-Agüera et al. 2010](#)]
 - ▶ By predicate importance (determined based on predicate popularity)[[Blanco et al. 2010](#)]

Predicate Folding Example

names	{ rdfs:label foaf:name	United Kingdom (en) United Kingdom (en)
attributes	{ dbo:demonym dbo:foundingDate	British (en) 1707-05-01
categories	{ dct:subject dct:subject	dbc:Member_states_of_the_European_Union dbc:Liberal_democracies
outgoing relations	{ dbo:leader dbo:capital	dbr:David_Cameron dbr:London
incoming relations	{ is dbo:country of is dbp:leaderName of	dbr:London dbr:David_Cameron

2-field Entity Document

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

Each entity is represented as a two-field document:

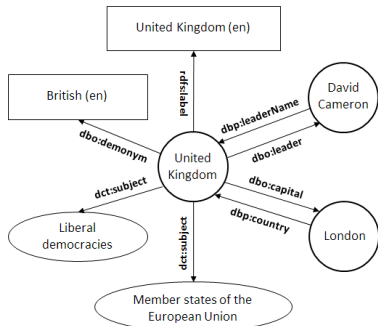
title

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

2-field Entity Document Example



title	united kingdom
content	british founding date 1707-05-01 united kingdom great britain northern ireland capital london leader david cameron

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

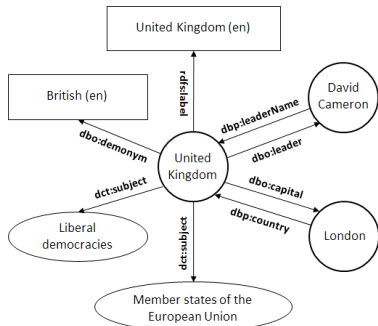
attributes

literals of all other predicates

outgoing links

names of entities in the object position

3-field Entity Document Example



names	united kingdom
attributes	british founding date 1707-05-01
outgoing links	united kingdom great britain northern ireland capital london leader david cameron

5-field Entity Document

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

Each entity is represented as a five-field document:

names

conventional names of 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

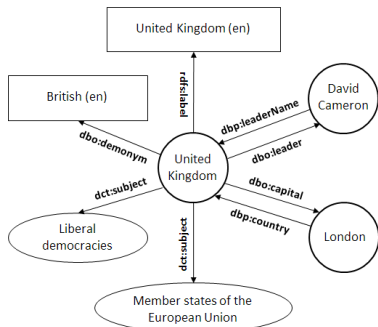
similar entity names

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

related entity names

names of entities in the object position

5-field Entity Document Example



names	united kingdom
attributes	british founding date 1707-05-01
categories	member state european union liberal democracy
similar entity names	united kingdom great britain northern ireland
related entity names	capital london leader david cameron

Dynamic Entity Representation

[Graus, Tsagkias et al., WSDM'16]

- ▶ **Problem:** vocabulary mismatch between entity's description in a knowledge base and the way people refer to the entity when searching for it
- ▶ Entity representations should account for:
 - ▶ **Context:** entities can appear in different contexts (e.g. Germany should be returned for queries related to World War II and 2014 Soccer World Cup)
 - ▶ **Time:** entities are not static in how they are perceived (e.g. Ferguson, Missouri before and after August 2014)

Approach (1)

Leverage collective intelligence provided by different entity description sources (KBs, web anchors, tweets, social tags, query log) to fill in the “vocabulary gap”:

- ▶ Create and update entity representations based on different sources
- ▶ Combine different entity descriptions for retrieval at specific time intervals by dynamically assigning weights to different sources

Approach (2)



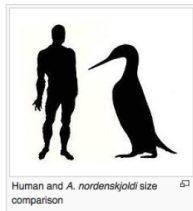
Baddest motherf**ing penguin there ever was.**

Anthropornis

From Wikipedia, the free encyclopedia

Anthropornis is a [genus](#) of giant [penguin](#) that lived 37-45 million years ago, during the Late [Eocene](#) and the earliest part of the [Oligocene](#).^[1] It reached 1.7 m (5 ft 7 in) in height and 90 kg (200 lb) in weight. Fossils of it have been found on [Seymour Island](#) off the coast of [Antarctica](#) and in [New Zealand](#). By comparison, the largest modern penguin species, the [emperor penguin](#), is just 1.2 m (3 ft 11 in) tall.

The type species, *Anthropornis nordenskjoldi*, had a bent joint in the wing, probably a carryover from flying ancestors.



References [\[edit\]](#)

- [↑] Myrcha, A., Jadwiszczak, P., Tambussi, C.P., Noriega, J.I., Gazdzicki, A., Tatur, A., and Valle, R.A. (2002). "Taxonomic Revision of Eocene Antarctic Penguins Based on Tarsometatarsal Morphology". *Polish Polar Research*, **23**(1): 5-46



This prehistoric bird article is a stub. You can help Wikipedia by expanding it.

Dynamic Entity Representation

Represent entities as fielded documents, in which each field corresponds to the content that comes from one description source:

- ▶ **Knowledge base:** anchor text of inter-knowledge base hyperlinks, redirects, category titles, names of entities that are linked from and to each entity in Wikipedia
- ▶ **Web anchors:** anchor text of links to Wikipedia pages from Google Wikilinks corpus
- ▶ **Twitter:** all English tweets that contain links to Wikipedia pages representing entities in the used snapshot
- ▶ **Delicious:** tags associated with Wikipedia pages in SocialBM0311 dataset
- ▶ **Queries:** queries that result in clicks on Wikipedia pages in the used snapshot

Entity Updates

The fields of entity document:

$$e = \{\bar{f}_{title}^e, \bar{f}_{text}^e, \bar{f}_{anchors}^e, \dots, \bar{f}_{query}^e\}$$

are updated at each discretized time point $T = \{t_1, t_2, t_3, \dots, t_n\}$

$$\bar{f}_{query}^e(t_i) = \bar{f}_{query}^e(t_{i-1}) + \begin{cases} \bar{q}, & \text{if } e_{clicked} \\ 0, & \text{otherwise} \end{cases}$$

$$\bar{f}_{tweets}^e(t_i) = \bar{f}_{tweets}^e(t_{i-1}) + \overline{tweet}_e$$

$$\bar{f}_{tags}^e(t_i) = \bar{f}_{tags}^e(t_{i-1}) + \overline{tag}_e$$

Each field's contribution towards the final entity score is determined based on features

Features

- ▶ **Field similarity**: TF-IDF cosine similarity of query and field f at time t_i
- ▶ **Field importance** (favor fields with more novel content): field's length in terms; field's length in characters; field's novelty at time t_i (favor fields with unseen, newly associated terms); number of updates to the field from t_0 through t_1
- ▶ **Entity importance** (favor recently updated entities): time since the last entity update

Classification-based ranker supervised by clicks learns the optimal feature weights

Results

Run	MAP (10k)	MAP (end)	Rate	P@1 (10k)	P@1 (end)	Rate
KBER _{sim}	0.5274	0.5579	+5.8%	0.4648	0.4967	+6.9%
KB+Web _{sim}	0.5485	0.5787 [▲]	+5.5%	0.4965	0.5282 [▲]	+6.4%
KB+Tags _{sim}	0.5455	0.5804 [▲]	+6.4%	0.4930	0.5317 [▲]	+7.8%
KB+Tweets _{sim}	0.5290	0.5612 [▲]	+6.1%	0.4673	0.5021 [▲]	+7.5%
KB+Queries _{sim}	0.5379	0.5750 [▲]	+6.9%	0.4813	0.5242 [▲]	+8.9%
DCER _{sim}	0.5620	0.5971[▲]	+6.2%	0.5178	0.5573[▲]	+7.6%

(a) adaptive runs

Run	MAP (10k)	MAP (end)	Rate	P@1 (10k)	P@1 (end)	Rate
KBER _{na}	0.5040	0.5198	+3.1%	0.4286	0.4392	+2.5%
KB+Web _{na}	0.5318	0.5493 [▲]	+3.3%	0.4698	0.4829 [▲]	+2.8%
KB+Tags _{na}	0.5298	0.5546 [▲]	+4.7%	0.4671	0.4904 [▲]	+5.0%
KB+Tweets _{na}	0.5074	0.5269 [▲]	+3.8%	0.4334	0.4490 [▲]	+3.6%
KB+Queries _{na}	0.5275	0.5650 [▲]	+7.1%	0.4659	0.5090 [▲]	+9.2%
DCER _{na}	0.5548	0.5872[▲]	+5.8%	0.5063	0.5408 [▲]	+6.8%

(b) non-adaptive runs

- ▶ Social tags are the best performing single entity description source
- ▶ KB+queries yields substantial relative improvement → added queries provide a strong signal for ranking the clicked entities
- ▶ Rankers that incorporate dynamic description sources (i.e KB+tags, KB+tweets and KB+queries) show the highest learning rate → entity content from these sources accounts for changes in entity representations over time

Outline

- ▶ Entity representation
- ▶ Entity retrieval
- ▶ Entity set expansion
- ▶ Entity ranking

Setting Field Weights

- ▶ Structured entity documents can be retrieved using structured document retrieval models (B25F, MLM)
- ▶ **Problem:** how to set the weights of document fields?
 - ▶ Heuristically: proportionate to the length of content in the field
 - ▶ Empirically: by optimizing the target retrieval metric using training queries

Fielded Sequential Dependence Model

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

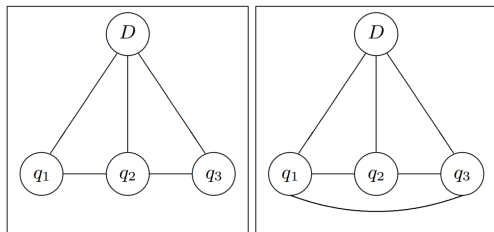
Previous research in ad-hoc IR has focused on two major directions:

- ▶ unigram bag-of-words retrieval models for multi-fielded documents
 - Ogilvie and Callan. Combining Document Representations for Known-item Search, SIGIR'03 (MLM)
 - Robertson et al. Simple BM25 Extension to Multiple Weighted Fields, CIKM'04 (BM25F)
- ▶ retrieval models incorporating term dependencies
 - Metzler and Croft. A Markov Random Field Model for Term Dependencies, SIGIR'05 (SDM)

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:

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

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:

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

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:

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

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:

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

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

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

Experiments

- ▶ DBpedia 3.7 as a knowledge graph
- ▶ 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

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* _†	0.286* _†	0.204* _†	0.750* _†
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* _†	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* _†	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* _†	0.231* _†	0.179* _†	0.517* _†

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

$$w_{q_i, j}^T = \sum_k \alpha_{j, k}^U \phi_k(q_i, j)$$

Parametric extension of FSDM

$$w_{q_i,j}^T = \sum_k \alpha_{j,k}^U \phi_k(q_i, j)$$

- ▶ $\phi_k(q_i, j)$ is the the k -th feature value for unigram q_i in field j .

Parametric extension of FSDM

$$w_{q_i,j}^T = \sum_k \alpha_{j,k}^U \phi_k(q_i,j)$$

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

Parametric extension of FSDM

$$w_{q_i,j}^T = \sum_k \alpha_{j,k}^U \phi_k(q_i, j)$$

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

$$\sum_j w_{q_i,j}^T = 1, w_{q_i,j}^T \geq 0, \alpha_{j,k}^U \geq 0, 0 \leq \phi_k(q_i, j) \leq 1$$

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

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

Outline

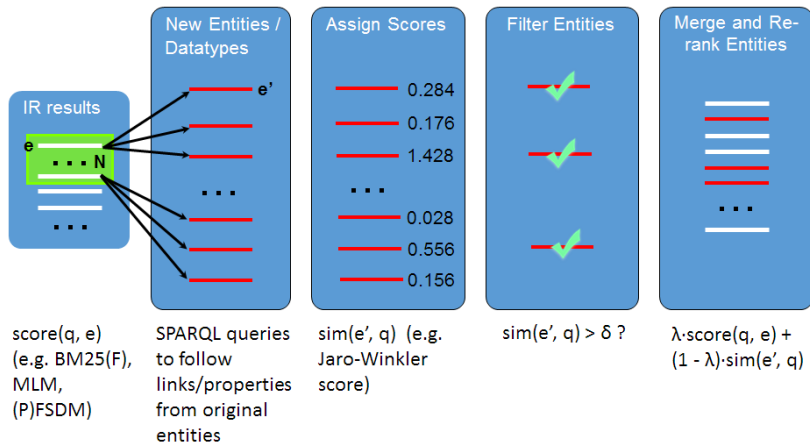
- ▶ Entity representation
- ▶ Entity retrieval
- ▶ Entity set expansion
- ▶ Entity ranking

Combining IR and Structured Search

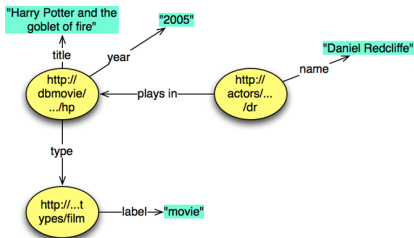
[Tonon, Demartini et al., SIGIR'12]

- ▶ Maintain inverted index for entity representations and triple store for entity relations
- ▶ **Hybrid approach:** IR models for initial entity retrieval and SPARQL queries for expansion

Pipeline

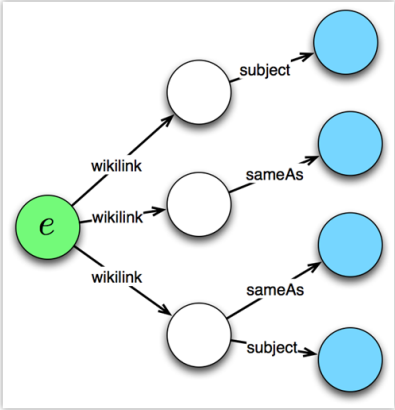
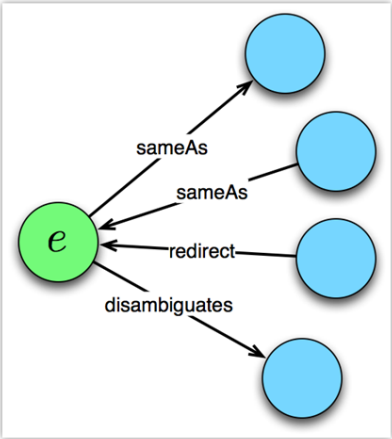


Result Expansion Strategies



- ▶ Follow predicates leading to other entities
- ▶ Follow predicates leading to entity attributes
- ▶ Explore entity neighbors and the neighbors of neighbors

Predicates to Follow



Results

	2010 Collection		2011 Collection	
	MAP	P@10	MAP	P@10
BM25	0.2070	0.3348	0.1484	0.2020
SAS	0.2293* (+11%)	0.363* (+8%)	0.1612 (+9%)	0.2200 (+9%)
SAS+DIS+RED	0.2586* (+25%)	0.3848* (+15%)	0.1657 (+12%)	0.2140 (+6%)

- ▶ Best performing method exploits entity neighbors by following `<owl:sameAs>` (SAS) as well as `<dbpedia:redirect>` (RED) and `<dbpedia:disambiguates>` predicates (DIS)
- ▶ Looking further into KG for related entities and following general predicates (`<dbpedia:wikilink>`, `<skos:subject>`, `<foaf:homepage>`, etc.) does not improve results

Outline

- ▶ Entity representation
- ▶ Entity retrieval
- ▶ Entity set expansion
- ▶ Entity ranking

Learning-to-Rank Entities

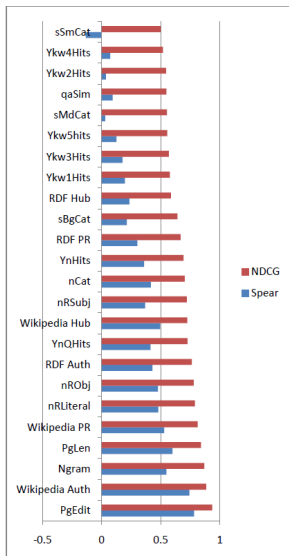
[Dali and Fortuna, WWW'11]

- ▶ 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

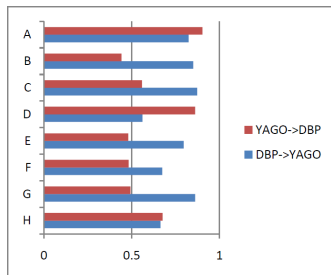
- ▶ Initial set of entities obtained using SPARQL queries
- ▶ 14 example queries for DBpedia and 27 example queries for Yago
- ▶ Example queries: “Which athlete was born in Philadelphia?”, “List of Schalke 04 players”, “Which countries have French as an official language?”, “Which objects are heavier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

Feature Importance



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

Transfer Learning



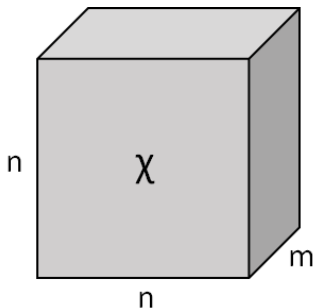
- ▶ Ranking model was trained on DBpedia questions and applied to Yago questions
- ▶ Only feature set A (all features) results in robust ranking model transfer
- ▶ In general, the ranking models for different knowledge graphs are non-transferable, unless they have been learned on large number of features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

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

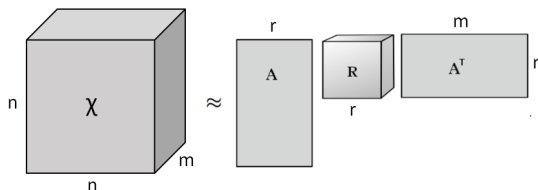
Knowledge Graph as Tensor



- ▶ 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 -th 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

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, $e^{-\frac{\ \mathbf{e} - \mathbf{e}_{top}\ ^2}{\sigma}}$

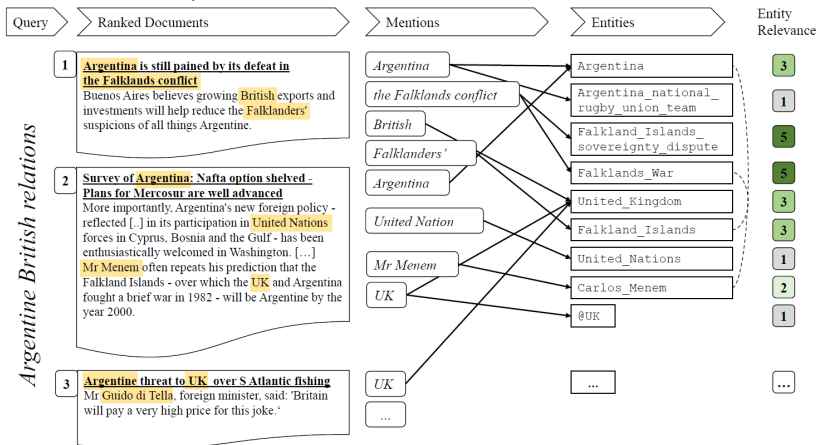
Results

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%)*

Ranking KG Entities using Top Documents

[Schuhmacher, Dietz et al., CIKM'15]

Aim: complex entity-focused informational queries (e.g. “Argentine British relations”)



Features and rankers

Mention Features	
MenFrq	# of entity occurrences in top documents
MenFrqIdf	entity IDF
Query-Mention Features	
SED	normalized Levenshtein distance
Glo	similarity based on GloVe embeddings
Jo	similarity based on JoBimText embeddings
Query-Entity Features	
QEnt	is document entity linked in query
QEntEntSim	is there a path in KG between document and query entities
WikiBoolean	is entity retrieved by query using Boolean model over Wikipedia articles
WikiSDM	SDM retrieval score of entity by query over Wikipedia articles
Query-Entity Features	
Wikipedia	is there a path between two entities in DBpedia KG

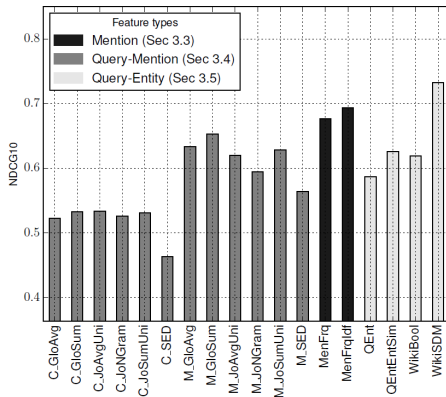
Rankers:

- ▶ rankSVM with linear kernel and linear+semantic smoothing kernels (pairwise)
- ▶ coordinate ascent

Results

- ▶ Authoritativeness marginally correlates with relevance (entities ranked high by PageRank are very general)
- ▶ Best results are obtained when ranking using SDM (supported by INEX results) and normalized mention frequencies
- ▶ RankLib performs better than SVM-rank with or without semantic kernel

Feature importance



- ▶ Context query mention features (prefix C_) perform worse than their no-context counterparts (prefix M_)
- ▶ Context features based on edit distance and distributional similarity are not effective
- ▶ DBpedia-based features have positive but insignificant influence on the overall performance, while Wikipedia-based features show strong and significant influence

Takeaway messages

- ▶ Use dynamic entity representations built from different sources (not only KB)
- ▶ Use retrieval models that account for different query concept types (FSDM and PFSDM) rather than standard fielded document retrieval models (BM25F and MLM) to obtain candidate entities
- ▶ Expand candidate entities by following KG links and using top-retrieved documents
- ▶ Re-rank candidate entities by using a variety of features including latent dimensional entity representations

Thank you!

References (1)

Entity representation methods:

- Neumayer, Balog et al. When Simple is (more than) Good Enough: Effective Semantic Search with (almost) no Semantics, ECIR'12
- Zhiltsov, Kotov et al. Fielded Sequential Dependence Model for Ad-hoc Entity Retrieval in the Web of Data, SIGIR'15
- Graus, Tsagkias et al. Dynamic Collective Entity Representations for Entity Ranking, WSDM'16

References (2)

Entity retrieval and ranking:

- 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
- Tonon, Demartini et al. Combining Inverted Indices and Structured Search for Ad-hoc Object Retrieval, SIGIR'12
- Dali and Fortuna. Learning to Rank for Semantic Search, WWW'11
- Zhiltsov and Agichtein. Improving Entity Search over Linked Data by Modeling Latent Semantics, CIKM'13
- Schuhmacher, Dietz et al. Ranking Entities for Web Queries through Text and Knowledge, CIKM'15