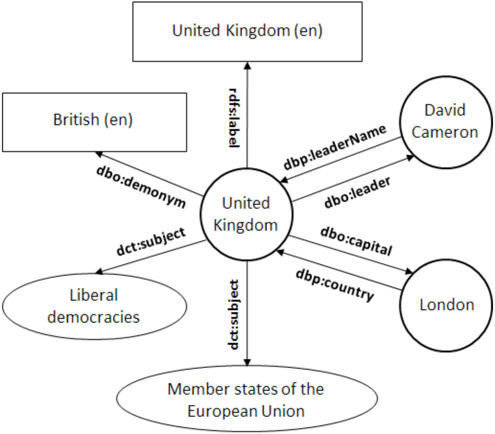
Entity Representation and Retrieval

Laura Dietz
University of New Hampshire

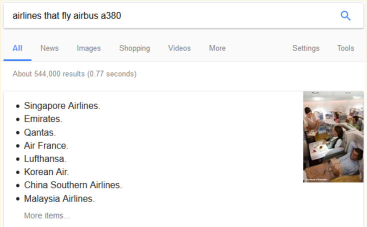
Alexander Kotov
Wayne State University

Edgar Meij
Bloomberg

Knowledge Graph Fragment



Entity Retrieval

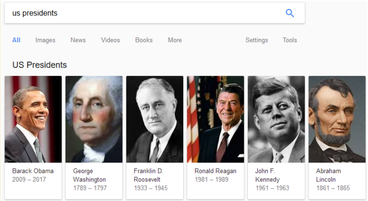


airlines that fly airbus a380

About 544,000 results (0.77 seconds)

- Singapore Airlines.
- Emirates.
- Qantas.
- Air France.
- Lufthansa.
- Korean Air.
- China Southern Airlines.
- Malaysia Airlines.

More items...



us presidents

US Presidents

Barack Obama 2009 – 2017	George Washington 1789 – 1797	Franklin D. Roosevelt 1933 – 1945	Ronald Reagan 1981 – 1989	John F. Kennedy 1961 – 1963	Abraham Lincoln 1861 – 1865

- Besides documents, users often search for concrete or abstract entities/objects (i.e. people, products, organizations, books)
- Users are willing to express these information needs more elaborately than with a few keywords [Balog et al., SIGIR'08]
- Entities (or entity cards) provide immediate answers to such queries → natural units for organizing search results
- Knowledge graphs are built around entities → Entity Retrieval from Knowledge Graph(s) (ERKG)

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)

- Evolution of entity retrieval tasks:
  - **Expert search at TREC 2005–2008 enterprise track:** find experts knowledgeable about a given topic
  - **Entity ranking track at INEX 2007–2009:** find Wikipedia page of entities with a given target type
  - **Related entity search at TREC 2009–2011 entity track:** find Web pages of entities related to a given entity in a certain way
- Can be used for entity linking: fragment of text as query, list of linked entities as result
- 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*. Entities in KG have no textual representation, apart from their names
- **Challenging IR problem:** knowledge graphs are best suited for structured graph pattern-based SPARQL queries, not for traditional IR models

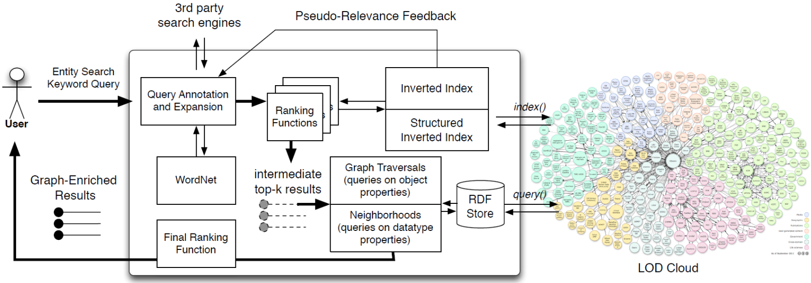
Research Challenges in ERKG

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 semantically match unstructured queries with structured entity representations?
3. How to account for entity types in retrieval?

Architecture of ERKG Methods

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



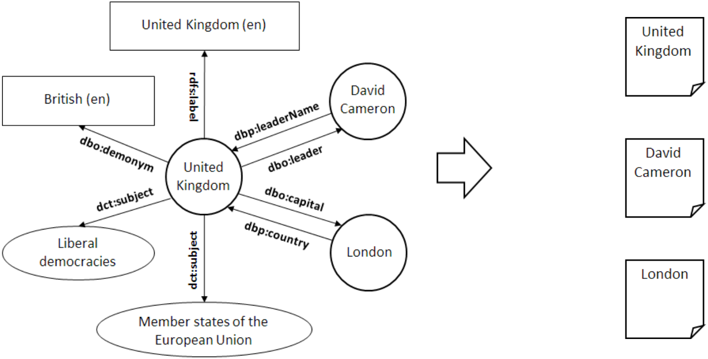
3rd party search engines
Pseudo-Relevance Feedback
Entity Search Keyword Query
User
Query Annotation and Expansion
Ranking Functions
Inverted Index
Structured Inverted Index
index()
WordNet
intermediate top-k results
Graph Traversals (queries on object properties)
Neighborhoods (queries on datatype properties)
RDF Store
query()
Graph-Enriched Results
Final Ranking Function
LOD Cloud

Outline

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

Structured Entity Documents

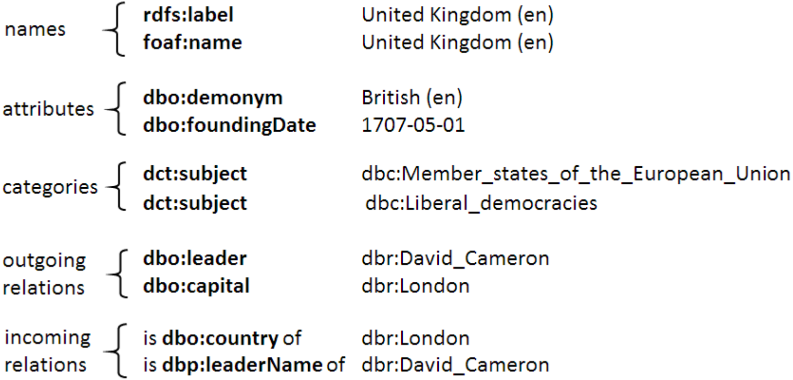
Build a textual representation (i.e. "document") for each entity by considering all triples, where it stands as a subject (or object)



Predicate Folding

- **Simple approach:** each predicate corresponds to one entity 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., SemSearch 2010]
  - by predicate importance (determined based on predicate popularity)[Blanco et al., ISWC 2011]
- The number and type of fields depends on a retrieval task

Predicate Folding Example



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

2-field Entity Document

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

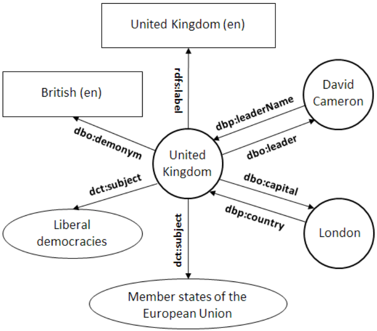
Each entity is represented as a two-field document:

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

This simple scheme is effective for entity retrieval

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:

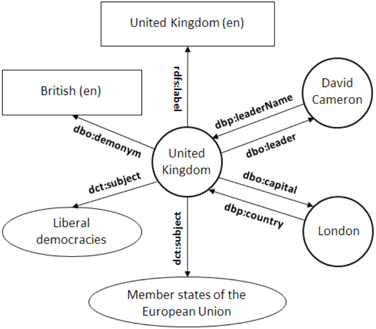
names
: literals of `foaf:name`, `rdfs:label` predicates along with tokens extracted from entity URIs

attributes
: literals of all other predicates

outgoing links
: names of entities in the object position

This scheme is effective for entity retrieval

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
: labels or names of entities

attributes
: all entity properties, other than names

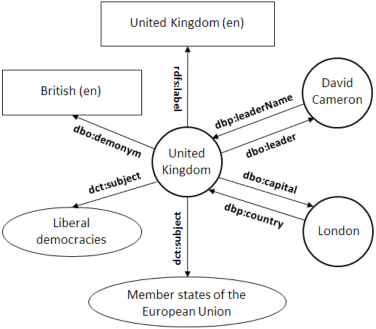
categories
: classes or groups, to which the entity has been assigned

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

This flexible scheme is effective for a variety of tasks: entity search, list search, question answering

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

Challenges related to Entity Representations

- Vocabulary mismatch between relevant entity(ies) description(s) and the query terms that can be used to search for it(them)
- Associations between words and entities depend on the context:
  - `Germany` should be returned for queries related to World War II and 2006 Soccer World Cup
- Real-life events change the descriptions of entities:
  - `Ferguson, Missouri` before and after August 2014

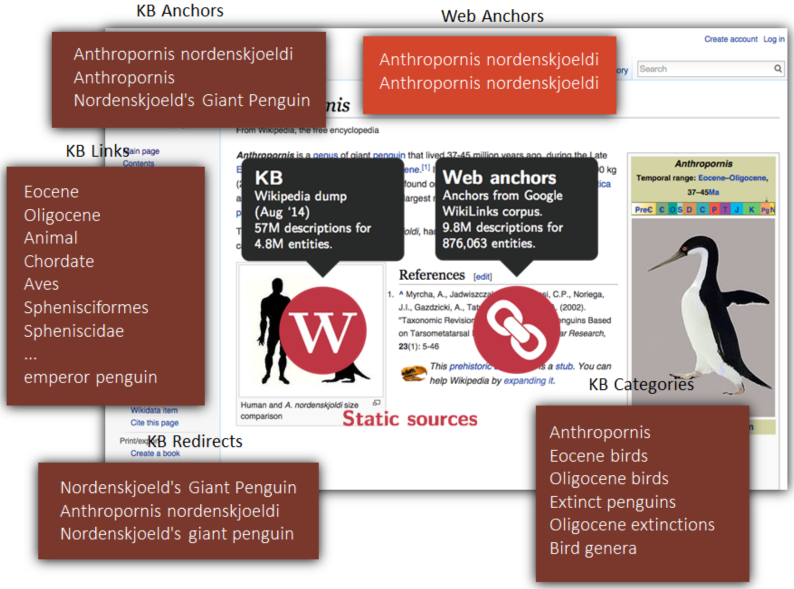
Dynamic Entity Representation

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

Idea: create static entity representations using knowledge bases and leverage different social media sources to dynamically update them

- Represent entities as fielded documents, in which each field corresponds to different source
- Tweak the weights of different fields over time

Static Sources



Create account Log in
Search
From Wikipedia, the free encyclopedia
Main page
Contents
Anthropornis is a genus of giant penguin that lived 37-45 million years ago, during the Late
References [edit]
1. ^ Myrcha, A., Jadwiszczak, P., Tambussi, C.P., Noriega, J.I., Gazdzicki, A., Tatur, A. (2002). "Taxonomic Revision of Eocene Antarctic Penguins Based on Tarsometatarsal Morphology". Polar Research, 23(1): 5-46
This prehistoric ... is a stub. You can help Wikipedia by expanding it.
Human and A. nordenskjoldi size comparison
Anthropornis
Temporal range: Eocene–Oligocene, 37–45Ma
Wikidata item
Cite this page
Create a book

KB Anchors

Anthropornis nordenskjoeldi
Anthropornis
Nordenskjoeld's Giant Penguin

Web Anchors

Anthropornis nordenskjoeldi
Anthropornis nordenskjoeldi

KB Links

Eocene
Oligocene
Animal
Chordate
Aves
Sphenisciformes
Spheniscidae
...
emperor penguin

KB
Wikipedia dump (Aug '14)
57M descriptions for 4.8M entities.

Web anchors
Anchors from Google WikiLinks corpus.
9.8M descriptions for 876,063 entities.

Static sources

KB Categories

Anthropornis
Eocene birds
Oligocene birds
Extinct penguins
Oligocene extinctions
Bird genera

KB Redirects

Nordenskjoeld's Giant Penguin
Anthropornis nordenskjoeldi
Nordenskjoeld's giant penguin

Dynamic Sources

biggest penguin
anthropornis
extinct penguin
prehistoric birds

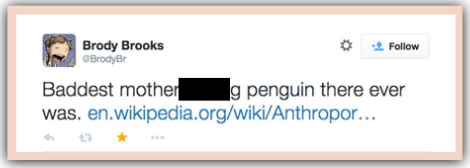
megafauna

Dynamic sources

Queries
Queries from MSN query logs that yield Wikipedia clicks.
47,002 descriptions for 18,724 entities.

Tweets
Tweets w/ links to Wikipedia pages (2011-2014)
52,631 descriptions for 38,269 entities.

Social tags
Delicious tags for Wiki pages, from the SocialBM0311 corpus.
4.4M descriptions for 289,015 entities.



Brody Brooks
@BrodyBr

Baddest mother[redacted]g penguin there ever was. en.wikipedia.org/wiki/Anthropor…

Outline

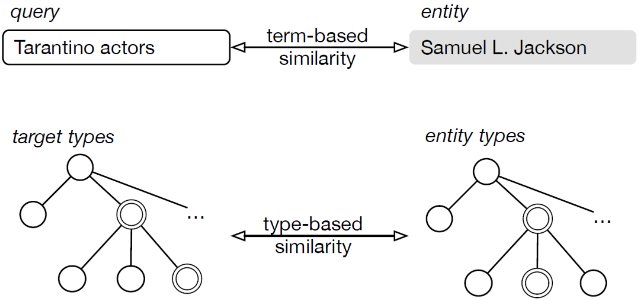
- Entity representation
- Entity retrieval
- Entity set expansion
- Entity ranking

Methods for ERKG

ERKG has been addressed in a probabilistic generative framework:

$$P(e|q) \propto P(q|e)P(e)$$

Besides keywords q_w, query q implicitly or explicitly contains *target entity type(s)* q_t, which can be incorporated into entity retrieval models



Incorporating Entity Types

Two ways to combine term-based similarity $P(q_w|e)$ and type-based similarity $P(q_t|e)$:

- Filtering [Bron et al., CIKM'10]:

$$P(q|e) = P(q_w|e)P(q_t|e)$$

- Interpolation [Balog et al., TOIS'11; Kaptein et al., AI'13; Pehcevski et al., IR'10; Raviv et al., JIWES'12]:

$$P(q|e) = (1 - \lambda_t)P(q_w|e) + \lambda_t P(q_t|e)$$

Term-based Similarity

Possible options for $P(q_w|e)$:

- unigram bag-of-words models for structured document retrieval:
  - Mixture of Language Models (MLM) [Ogilvie and Callan, SIGIR'03]
  - BM25 for multi-field documents (BM25F) [Robertson et al., CIKM'04]
  - Probabilistic Retrieval Model for Semi-structured Data (PRMS) [Kim and Croft, ECIR'09]
- term dependence (bigrams) models:
  - Sequential Dependence Model (SDM) [Metzler and Croft, SIGIR'05]
- term dependence models for structured document retrieval:
  - Fielded Sequential Dependence Model (FSDM) [Zhiltsov et al., SIGIR'15]
  - Parameterized Fielded Sequential Dependence Model (PFSDM) [Nikolaev et al., SIGIR'16]

Fielded Sequential Dependence Model

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

- **Idea:** account both for phrases (bigrams) and document structure
- Document score is a linear combination of matching functions for unigrams and bigrams *in each document field*:

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

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

Fielded Sequential Dependence Model

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

- **Idea:** account both for phrases (bigrams) and document structure
- Document score is a linear combination of matching functions for unigrams and bigrams *in each document field*:

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

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

Fielded Sequential Dependence Model

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

- **Idea:** account both for phrases (bigrams) and document structure
- Document score is a linear combination of matching functions for unigrams and bigrams *in each document field*:

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

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

Fielded Sequential Dependence Model

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

- **Idea:** account both for phrases (bigrams) and document structure
- Document score is a linear combination of matching functions for unigrams and bigrams *in each document field*:

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

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

FSDM ranking function

FSDM matching function for unigrams:

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

Example:

apollo astronauts who walked on the moon

Parameters:

1. Field importance weights for unigrams and bigrams
2. Relative importance weights of matching unigrams and bigrams

FSDM ranking function

FSDM matching function for unigrams:

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

Example:

apollo astronauts who walked on the moon
category

Parameters:

1. Field importance weights for unigrams and bigrams
2. Relative importance weights of matching unigrams and bigrams

FSDM ranking function

FSDM matching function for unigrams:

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

Example:

$$\underbrace{\text{apollo astronauts}}_{\text{category}} \text{ who walked } \underbrace{\text{on the moon}}_{\text{category}}$$

Parameters:

1. Field importance weights for unigrams and bigrams
2. Relative importance weights of matching unigrams and bigrams

Limitation of FSDM

Same field weights for all query unigrams and all query bigrams

Example:

capitals in Europe which were host cities of summer Olympic games

Limitation of FSDM

Same field weights for all query unigrams and all query bigrams

Example:

capitals in Europe which were host cities of summer Olympic games

category

Limitation of FSDM

Same field weights for all query unigrams and all query bigrams

Example:

capitals in Europe which were host cities of summer Olympic games

category attribute

Limitation of FSDM

Same field weights for all query unigrams and all query bigrams

Example:

capitals in Europe which were host cities of summer Olympic games
category attribute category

Parametric extension of FSDM

[Nikolaev, Kotov et al., SIGIR'16]

Idea: calculate field weight for each unigram and bigram based on features:

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

- $\phi_k(q_i, j)$ is the the k-th feature value for unigram q_i in field j
- $\alpha^U_{j,k}$ are feature weights that are learned by coordinate ascent to maximize target retrieval metric

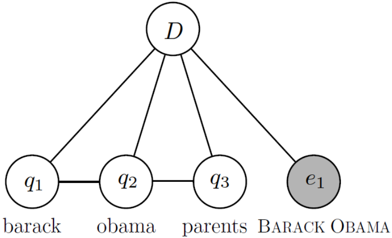
Features

Source	Description	CT
Collection statistics	Posterior probability $P(E_j\|\kappa)$	UG BG
	Top SDM score of the j-th field when κ is used as a query	BG
Stanford POS Tagger	Is concept κ a proper noun?	UG
	Is κ a plural non-proper noun?	UG BG
	Is κ a superlative adjective?	UG
Stanford Parser	Is κ part of a noun phrase?	BG
	Is κ the only singular non-proper noun in a noun phrase?	UG
	Intercept	UG BG

Entity Linking in ERKG

[Hasibi et al., ICTIR'16]

Idea: linked entities as additional feature function in FSDM



$$
\begin{aligned}
P_\Lambda(D|Q) \overset{rank}{=} & \lambda_T \sum_{q \in Q} \tilde{f}_T(q_i, D) + \\
& \lambda_O \sum_{q \in Q} \tilde{f}_O(q_i, q_{i+1}, D) + \\
& \lambda_U \sum_{q \in Q} \tilde{f}_U(q_i, q_{i+1}, D) + \\
& \lambda_E \sum_{e \in E(Q)} \tilde{f}_E(e, D)
\end{aligned}
$$

Type-based Similarity

[Garigliotti and Balog, ICTIR'17]

- If target type(s) q_t are provided with the query, the distribution of types for entity e is estimated as:

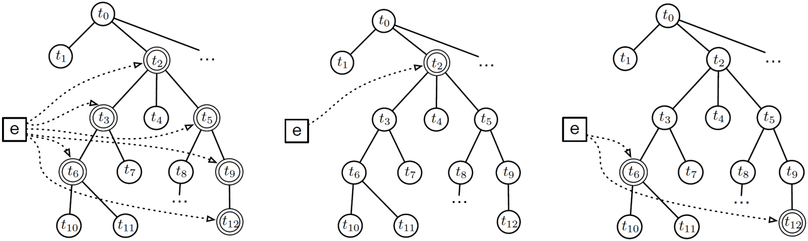
$$P(t|\Theta_e) = \frac{n(t,e) + \mu P(t)}{\sum_{t'} n(t',e) + \mu}$$

- With both Θ_q and Θ_e in place, type-based similarity between q and e is estimated as:

$$P(q_t|e) = z(\max_{e'} KL(\Theta_q||\Theta_{e'}) - KL(\Theta_q||\Theta_e))$$

Entity Type Representation

[Garigliotti and Balog, ICTIR'17]



(a) **all assigned types** (b) **most general types** (c) **most specific types**

Type-based Similarity

If no target type(s) are provided with the query, they can be inferred using:

- **Type-centric approach** [Balog and Neumayer, CIKM'12]: build a document for each type by concatenating the descriptions of all entities that belong to it

$$P(q|t) = \prod_{i=1}^{|q|} P(w_i|\theta_t) = \prod_{i=1}^{|q|} (1-\lambda) \sum_{e:t \in e_t} (P(w|e_d)P(e|t) + \lambda P(w_i))$$

- **Entity-centric approach** [Balog and Neumayer, CIKM'12]: aggregate retrieval scores and type distributions of top retrieved entities

$$P(q|t) = \sum_{e:t \in e_t} P(q|e)P(e|t)$$

Type-based Similarity (cont.)

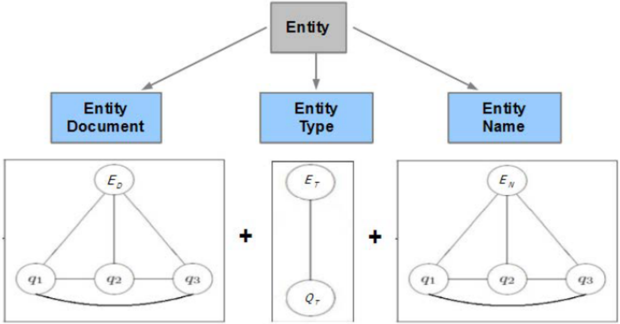
- **Type ranking** [Garigliotti et al., SIGIR'17]: combines scores of entity- and type-centric approaches with taxonomy and type label features
- **Head-modifier approach** [Ma et al., WWW'18]: query and type names are phrases, which consists of a head word (h_q and h_t) and a set of modifiers (M_q and M_t) (e.g. *"Italian Nobel prize <u>winners</u>"*, *"<u>Musicians</u> who appeared in the Blues Brothers movies"*)

$$P(q|t) = P(h_t|h_q)^{\alpha_1} P(M_t|h_q)^{\alpha_2} P(h_t|M_q)^{\alpha_3} P(M_t|M_q)^{\alpha_4}$$

MRF-based Combined Model

[Raviv et al., JIWES'17]

Entity name E_N, description E_D and types E_T can be combined into Markov Random Field-based retrieval model:



$$P(E|Q) = \lambda_{E_N} P(E_N|Q) + \lambda_{E_D} P(E_D|Q) + \lambda_{E_P} P(E_P|Q)$$

Outline

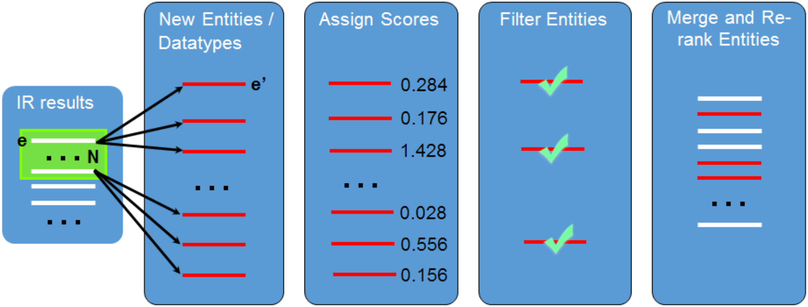
- Entity representation
- Entity retrieval
- Entity set expansion
- Entity ranking

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



score(q, e) (e.g. BM25(F), MLM, (P)FSDM)

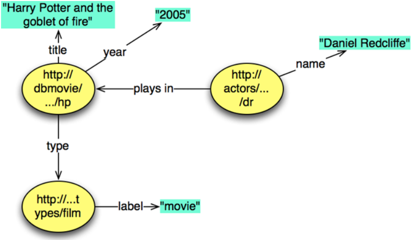
SPARQL queries to follow links/properties from original entities

sim(e', q) (e.g. Jaro-Winkler score)

sim(e', q) > δ ?

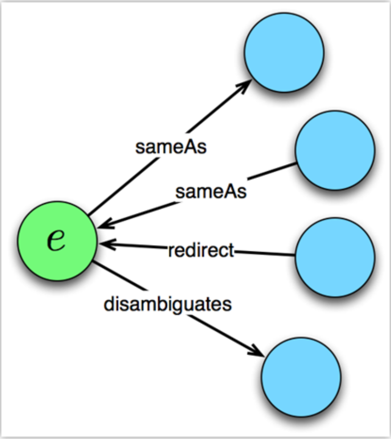
λ·score(q, e) + (1 - λ)·sim(e', q)

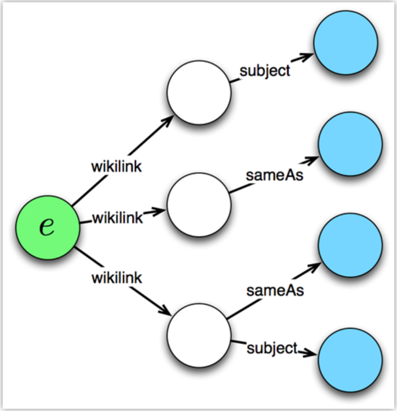
Result Expansion Strategies



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

Predicates to Follow





Outline

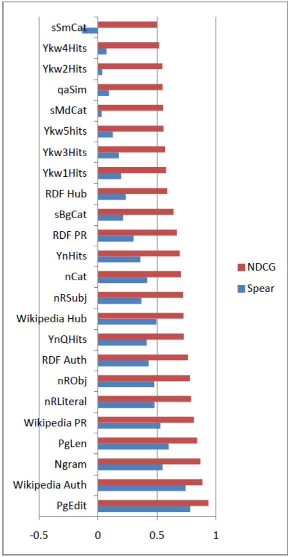
- Entity representation
- Entity retrieval
- Entity set expansion
- Entity ranking

Learning-to-Rank Entities

[Dali and Fortuna, WWW'11]

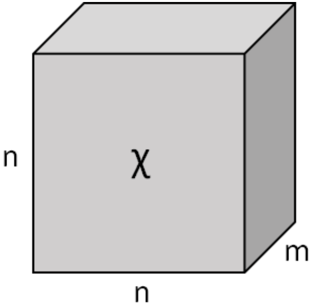
- Potential features:
  - Popularity and importance of Wikipedia page: # of accesses from logs, # of edits, page length
  - RDF features: # of triples E is subject/object/subject and object is a literal, # of categories Wikipedia page for E belongs to, size of the biggest/smallest/median category
  - 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

Feature Importance



- Features approximating the entity 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

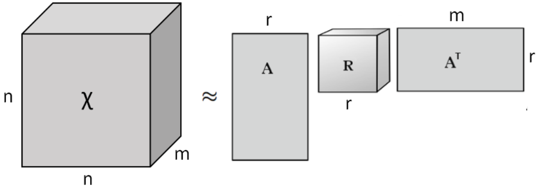
Knowledge Graph as Tensor



- 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

RESCAL Tensor Factorization

[Nickel et al., ICML'11, WWW'12]



- Given r is the number of latent factors, factorize each X_k:

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

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

KG entity embedding methods

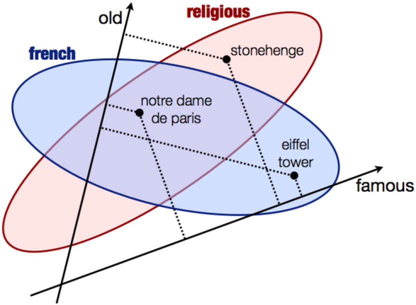
Idea: Represent KG entities and relations as dense real-valued vectors (i.e. embeddings) and predict relation between entities e_s and e_o in a KG based on $f(\mathbf{e_s}, \mathbf{e_o}, \Theta)$

- Interaction-based methods
  - `RESCAL` [Nikel et al., ICML'11]: $w_k^T \mathbf{e_s} \otimes \mathbf{e_o}$
  - `LFM` [Jenatton et al., NIPS'12]: $\mathbf{e_s} W_p \mathbf{e_o}$
  - `HolE` [Nickel et al., AAAI'16]: $\sigma(\mathbf{p}^T(\mathbf{e_s} \star \mathbf{e_o}))$
- Neural network-based methods
  - `ER-MLP` [Dong et al., KDD'14]: $w^T g\left(C^T[\mathbf{e_s}; \mathbf{p}; \mathbf{e_o}]\right)$
  - `NTN` [Socher et al., NIPS'13]: $w_p^T g\left(\mathbf{e_s}^T W_p^{[1:k]} \mathbf{e_o} + C_p^T[\mathbf{e_s}]; \mathbf{e_o}\right)$
  - `ConvE` [Dettmers et al., AAAI'18]: $g(\text{vec}(g([\overline{\mathbf{e_s}}; \overline{\mathbf{p}}] * \omega))W)\mathbf{e_o}$
- Distance-based methods
  - `Unstructured` [Bordes et al., AAAI'11]: $-\|\mathbf{e_s} - \mathbf{e_o}\|_2^2$
  - `SE` [Bordes et al., AAAI'11]: $-\|W_{e_s}\mathbf{e_s} - W_{e_o}\mathbf{e_o}\|_1$
  - `TransE` [Bordes et al., NIPS'13]: $-\|\mathbf{e_s} + \mathbf{p} - \mathbf{e_o}\|_{1/2}$

$\otimes$, $\star$, $*$, $\bar{\cdot}$, $[\cdot;\cdot]$ and vec denote tensor product, cross-correlation, convolution, 2D reshaping, vector concatenation and tensor vectorization operators

Interpretable KG Entity Embeddings

[Jameel et al., SIGIR'17]



- Salient properties of entities are modeled as hyperplanes that separate entities that have a property in their descriptions from the ones that do not
- Normals of separating hyperplanes point to the regions where entities with a salient property occur

Utilizing entity embeddings for entity re-ranking

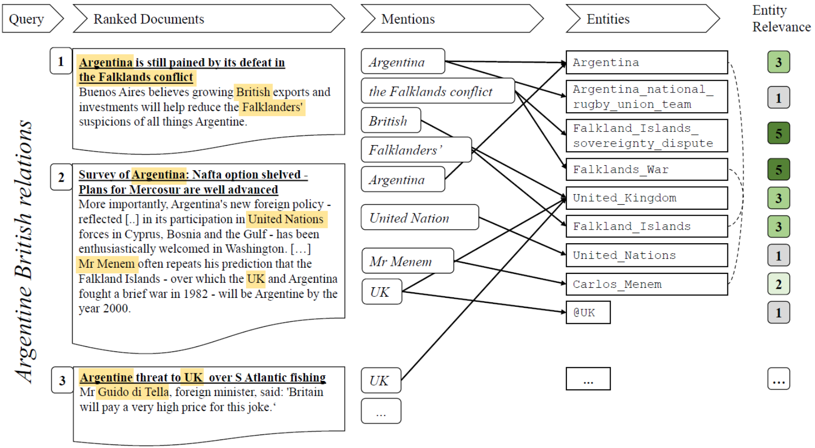
[Zhiltsov and Agichtein, CIKM'13]

1. Retrieve initial set of entities
2. Re-rank retrieved entities using similarity metrics to top-k retrieved entities in low-dimensional space as features:
   - cosine similarity: $cos(\mathbf{e}, \mathbf{e}_{top})$
   - Euclidean distance: $\|\mathbf{e} - \mathbf{e}_{top}\|_2$
   - heat kernel: $e^{-\frac{\|\mathbf{e}-\mathbf{e}_{top}\|_2^2}{\sigma}}$

Ranking KG Entities using Top Documents

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

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



1 **Argentina is still pained by its defeat in the Falklands conflict**
Buenos Aires believes growing British exports and investments will help reduce the Falklanders' suspicions of all things Argentine.

2 **Survey of Argentina: Nafta option shelved - Plans for Mercosur are well advanced**
More importantly, Argentina's new foreign policy - reflected [..] in its participation in United Nations forces in Cyprus, Bosnia and the Gulf - has been enthusiastically welcomed in Washington. […]
Mr Menem often repeats his prediction that the Falkland Islands - over which the UK and Argentina fought a brief war in 1982 - will be Argentine by the year 2000.

3 **Argentine threat to UK over S Atlantic fishing**
Mr Guido di Tella, foreign minister, said: 'Britain will pay a very high price for this joke.'

Takeaway messages

- Use dynamic entity representations built from different sources (not only KG)
- Use retrieval models that account for query unigram and bigrams (FSDM and PFSDM) rather than bag-of-words structured document retrieval models (BM25F and MLM) to obtain candidate entities
- Leverage entity links and types in entity retrieval models
- Expand candidate entities by following KG links
- Re-rank candidate entities by using a variety of features including the ones based on KG entity embeddings

Thank you!