

Part II

Entity Retrieval

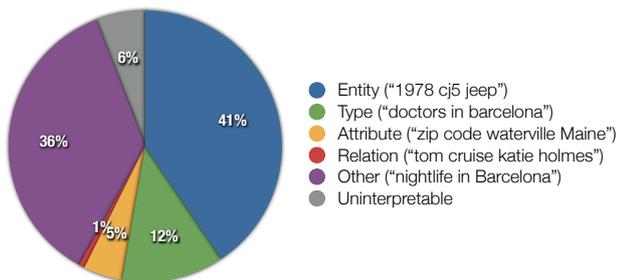
Krisztian Balog
University of Stavanger

Full-day tutorial at the SIGIR'13 conference | Dublin, Ireland, 2013

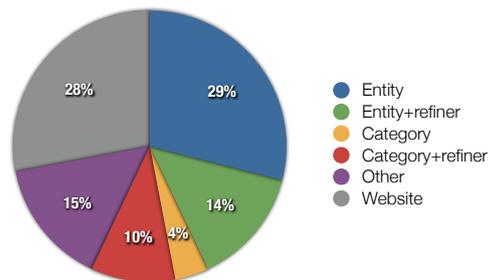
Entity retrieval

Addressing information needs that are better answered by returning specific objects instead of just any type of documents.

Distribution of web search queries [Pound et al. 2010]



Distribution of web search queries [Lin et al. 2011]



What's so special here?

- Entities are not always directly represented
 - Recognize and disambiguate entities in text (that is, entity linking)
 - Collect and aggregate information about a given entity from multiple documents and even multiple data collections
- More structure than in document-based IR
 - Types (from some taxonomy)
 - Attributes (from some ontology)
 - Relationships to other entities ("typed links")

In this part

- Look at a number of entity ranking tasks
 - Motivating real-world use-cases
 - Abstractions at evaluation benchmarking campaigns (TREC, INEX)
 - Methods and approaches
- In all cases
 - Input: (semi-structured) query
 - Output: ranked list of entities
 - Evaluation: standard IR metrics

Outline

1. Ranking based on entity descriptions
2. Incorporating entity types
3. Entity relationships

Attributes (/Descriptions)
Type(s)
Relationships

Setting boundaries

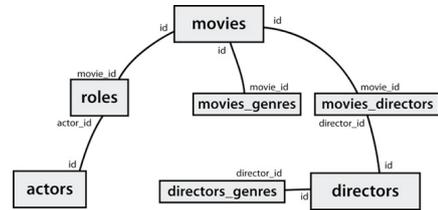


We are IR people ...

- ... but that doesn't mean that we are the only ones who thought about this
- Entity retrieval is an active research area in neighbouring communities
 - Databases
 - Semantic web
 - Natural language processing

Databases

- Keyword search in DBs
 - Return tuples with matching keywords, minimally joined through primary-foreign key relationships

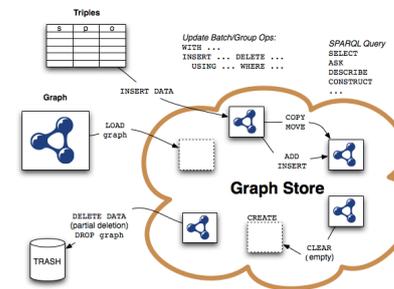


Databases (2)

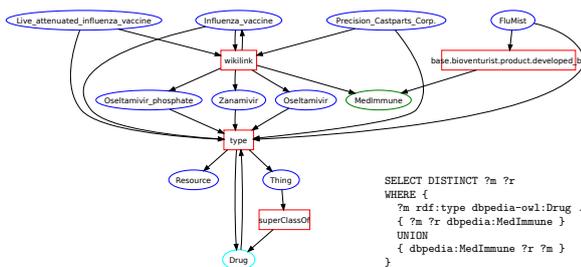


Semantic web

- Indexing
- Retrieval
- Querying
- Inference



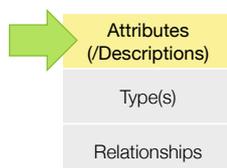
Semantic web (2)



Natural language processing

- Question answering
 - "Who invented the paper clip?"
 - "What museums have displayed Chanel clothing?"
- Relationship extraction

Ranking based on entity descriptions



Task: ad-hoc entity retrieval

- **Input:** unconstrained natural language query
 - "telegraphic" queries (neither well-formed nor grammatically correct sentences or questions)
- **Output:** ranked list of entities
- **Collection:** unstructured and/or semi-structured documents

Example information needs

- Q american embassy nairobi
- Q ben franklin
- Q Chernobyl
- Q meg ryan war
- Q Worst actor century
- Q Sweden Iceland currency

Two settings

1. With ready-made entity descriptions



2. Without explicit entity representations



Ranking with ready-made entity descriptions

This is not unrealistic...



Document-based entity representations

- Most entities have a “home page”
- I.e., each entity is described by a document
- In this scenario, ranking entities is much like ranking documents
 - unstructured
 - semi-structured

Evaluation initiatives

- INEX Entity Ranking track (2007-09)
 - Collection is the (English) Wikipedia
 - Entities are represented by Wikipedia articles
- Semantic Search Challenge (2010-11)
 - Collection is a Semantic Web crawl (BTC2009)
 - ~1 billion RDF triples
 - Entities are represented by URIs
- INEX Linked Data track (2012-13)
 - Wikipedia enriched with RDF properties from DBpedia and YAGO

Standard Language Modeling approach

- Rank documents d according to their likelihood of being relevant given a query q : $P(d|q)$

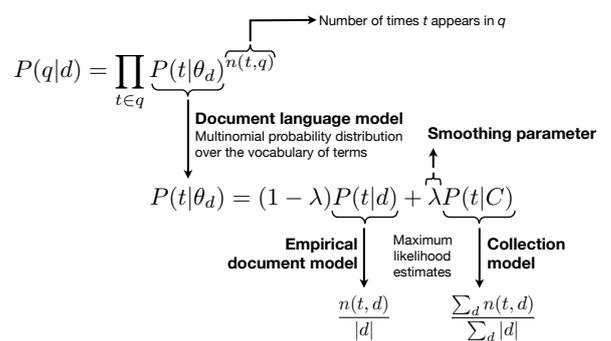
$$P(d|q) = \frac{P(q|d)P(d)}{P(q)} \propto P(q|d)P(d)$$

Query likelihood
Probability that query q was “produced” by document d

Document prior
Probability of the document being relevant to any query

$$P(q|d) = \prod_{t \in q} P(t|\theta_d)^{n(t,q)}$$

Standard Language Modeling approach (2)



Here, documents==entities, so

$$P(e|q) \propto P(e)P(q|\theta_e) = P(e) \prod_{t \in q} P(t|\theta_e)^{n(t,q)}$$

Entity prior
Probability of the entity
being relevant to any query
Entity language model
Multinomial probability distribution
over the vocabulary of terms

Semi-structured entity representation

- Entity description documents are rarely unstructured
- Representing entities as
 - Fielded documents – the IR approach
 - Graphs – the DB/SW approach

The Audi A4 is a line of compact executive cars produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group.

The A4 has been built in four generations and is based on Volkswagen's B platform. The first generation A4 succeeded the Audi 80. The automaker's internal numbering treats the A4 as a continuation of the Audi 80 lineage, with the initial A4 designated as the B5-series, followed by the B6, B7, and the current B8. The B8 A4 is built on the Volkswagen Group MLB platform shared with many other Audi models and potentially one Porsche model within Volkswagen Group.^[c]

dbpedia:Audi_A4	
foaf:name	Audi A4
rdfs:label	Audi A4
rdfs:comment	The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group. The A4 has been built [...]
dbpprop:production	1994 2001 2005 2008
rdf:type	dbpedia-owl:MeanOfTransportation dbpedia-owl:Automobile
dbpedia-owl:manufacturer	dbpedia:Audi
dbpedia-owl:class	dbpedia:Compact_executive_car
owl:sameAs	freebase:Audi_A4 dbpedia:Audi_A5
is dbpedia-owl:predecessor of	dbpedia:Cadillac_BLS
is dbpprop:similar of	

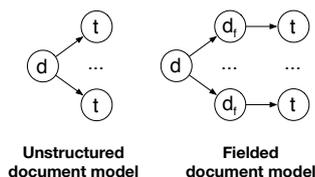
Mixture of Language Models [Ogilvie & Callan 2003]

- Build a separate language model for each field
- Take a linear combination of them

$$P(t|\theta_d) = \sum_{j=1}^m \mu_j P(t|\theta_{d_j})$$

Field weights
 $\sum_{j=1}^m \mu_j = 1$
Field language model
Smoothed with a collection model built from all document representations of the same type in the collection

Comparison of models



Setting field weights

- Heuristically
 - Proportional to the length of text content in that field, to the field's individual performance, etc.
- Empirically (using training queries)
- Problems
 - Number of possible fields is huge
 - It is not possible to optimise their weights directly
- Entities are sparse w.r.t. different fields
 - Most entities have only a handful of predicates

Predicate folding

- **Idea**: reduce the number of fields by grouping them together
- Grouping based on (BM25F and)
 - type [Pérez-Agüera et al. 2010]
 - manually determined importance [Blanco et al. 2011]

Hierarchical Entity Model [Neumayer et al. 2012]

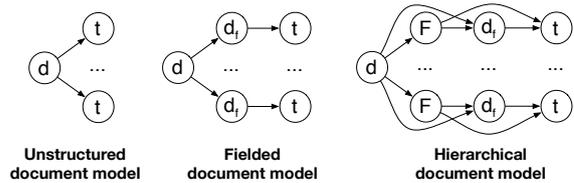
- Organize fields into a 2-level hierarchy
 - Field types (4) on the top level
 - Individual fields of that type on the bottom level
- Estimate field weights
 - Using training data for field types
 - Using heuristics for bottom-level types

Two-level hierarchy

[Neumayer et al. 2012]

Name	foaf:name rdfs:label rdfs:comment	Audi A4 Audi A4 The Audi A4 is a compact executive car produced since late 1994 by the German car manufacturer Audi, a subsidiary of the Volkswagen Group. The A4 has been built [...]
Attributes	dbpprop:production	1994 2001 2005 2008
Out-relations	rdf:type dbpedia-owl:manufacturer dbpedia-owl:class owl:sameAs	dbpedia-owl:MeanOfTransportation dbpedia-owl:Automobile dbpedia:Audi dbpedia:Compact_executive_car freebase:Audi_A4
In-relations	is dbpprop:similar of	dbpedia:Audi_A5 dbpedia:Cadillac_BLS

Comparison of models



Probabilistic Retrieval Model for Semistructured data

[Kim et al. 2009]

- Extension to the Mixture of Language Models
- Find which document field each query term may be associated with

$$P(t|\theta_d) = \sum_{j=1}^m \mu_j P(t|\theta_{d_j})$$

Mapping probability
Estimated for each query term

$$P(t|\theta_d) = \sum_{j=1}^m P(d_j|t) P(t|\theta_{d_j})$$

Estimating the mapping probability

$$P(t|C_j) = \frac{\sum_d n(t, d_j)}{\sum_d |d_j|}$$

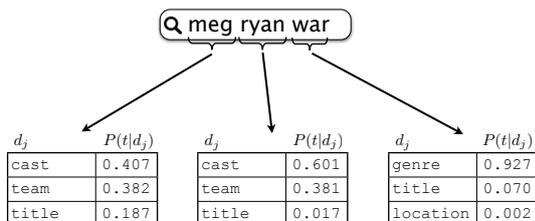
Term likelihood
Probability of a query term occurring in a given field type

Prior field probability
Probability of mapping the query term to this field before observing collection statistics

$$P(d_j|t) = \frac{P(t|d_j)P(d_j)}{P(t)}$$

$$\sum_{d_k} P(t|d_k)P(d_k)$$

Example



Ranking without explicit entity representations

Scenario

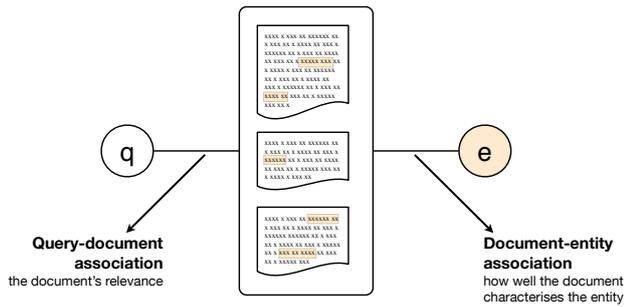
- Entity descriptions are not readily available
- Entity occurrences are annotated
 - manually
 - automatically (~entity linking)

TREC Enterprise track

- Expert finding task (2005-08)
 - Enterprise setting (intranet of a large organization)
 - Given a query, return people who are experts on the query topic
 - List of potential experts is provided
- We assume that the collection has been annotated with <person>...</person> tokens

The basic idea

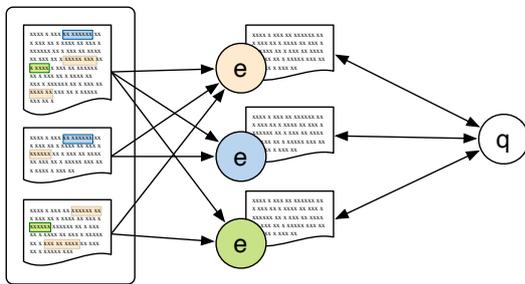
Use documents to go from queries to entities



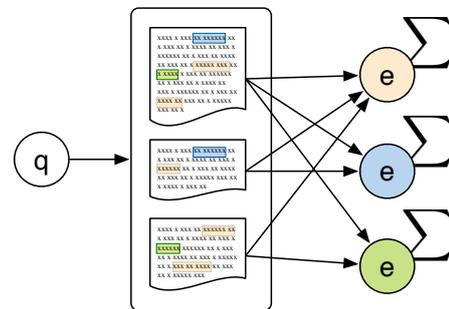
Two principal approaches

- **Profile-based methods**
 - Create a textual profile for entities, then rank them (by adapting document retrieval techniques)
- **Document-based methods**
 - Indirect representation based on mentions identified in documents
 - First ranking documents (or snippets) and then aggregating evidence for associated entities

Profile-based methods



Document-based methods



Many possibilities in terms of modeling

- Generative (probabilistic) models
- Discriminative (probabilistic) models
- Voting models
- Graph-based models

Generative probabilistic models

- Candidate generation models ($P(e|q)$)
 - Two-stage language model
- **Topic generation models ($P(q|e)$)**
 - Candidate model, a.k.a. Model 1
 - Document model, a.k.a. Model 2
 - Proximity-based variations
- Both families of models can be derived from the Probability Ranking Principle [Fang & Zhai 2007]

Candidate models ("Model 1")

[Balog et al. 2006]

$$P(q|\theta_e) = \prod_{t \in q} P(t|\theta_e)^{n(t,q)}$$

Smoothing
With collection-wide background model

$$(1 - \lambda)P(t|e) + \lambda P(t)$$

$$\sum_d P(t|d, e)P(d|e)$$

Term-candidate co-occurrence
In a particular document.
In the simplest case: $P(t|d)$

Document-entity association

Document models ("Model 2")

[Balog et al. 2006]

$$P(q|e) = \sum_d P(q|d, e)P(d|e)$$

Document relevance
How well document d supports the claim that e is relevant to q

Document-entity association

$$\prod_{t \in q} P(t|d, e)^{n(t,q)}$$

Simplifying assumption
(t and e are conditionally independent given d)

$$P(t|\theta_d)$$

Document-entity associations

- Boolean (or set-based) approach
- Weighted by the confidence in entity linking
- Consider other entities mentioned in the document

Proximity-based variations

- So far, conditional independence assumption between candidates and terms when computing the probability $P(t|d,e)$
- Relationship between terms and entities that in the same document is ignored
 - Entity is equally strongly associated with everything discussed in that document
- Let's capture the dependence between entities and terms
 - Use their distance in the document

Using proximity kernels

[Petkova & Croft 2007]

$$P(t|d,e) = \frac{1}{Z} \sum_{i=1}^N \delta_d(i,t) k(t,e)$$

$\frac{1}{Z}$ → Normalizing constant
 $\delta_d(i,t)$ → Indicator function
 1 if the term at position i is t ,
 0 otherwise
 $k(t,e)$ → Proximity-based kernel
 - constant function
 - triangle kernel
 - Gaussian kernel
 - step function

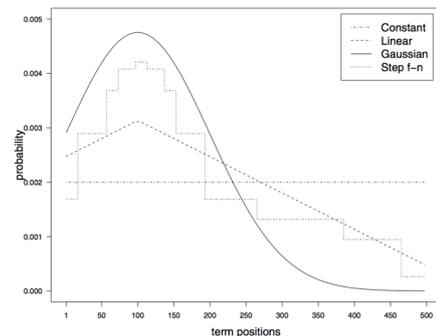


Figure taken from D. Petkova and W.B. Croft. Proximity-based document representation for named entity retrieval. *CIKM'07*.

Many possibilities in terms of modeling

- Generative probabilistic models
- Discriminative probabilistic models
- Voting models
- Graph-based models

Discriminative models

- Vs. generative models:
 - Fewer assumptions (e.g., term independence)
 - "Let the data speak"
 - Sufficient amounts of training data required
 - Incorporating more document features, multiple signals for document-entity associations
 - Estimating $P(r=1|e,q)$ directly (instead of $P(e,q|r=1)$)
 - Optimization can get trapped in a local maximum/minimum

Arithmetic Mean Discriminative (AMD) model

[Yang et al. 2010]

$$P_{\theta}(r=1|e,q) = \sum_d P(r_1=1|q,d) P(r_2=1|e,d) P(d)$$

$P(r_1=1|q,d)$ → Query-document relevance
 $P(r_2=1|e,d)$ → Document-entity relevance
 $P(d)$ → Document prior

$\sigma\left(\sum_{i=1}^{N_f} \alpha_i f_i(q, d_i)\right)$ → standard logistic function over a linear combination of features
 $\sigma\left(\sum_{j=1}^{N_g} \beta_j g_j(e, d_i)\right)$ → standard logistic function over a linear combination of features

weight parameters (learned) → α_i
 features → f_i
 weight parameters (learned) → β_j
 features → g_j

Learning to rank & entity retrieval

- Pointwise
 - AMD, GMD [Yang et al. 2010]
 - Multilayer perceptrons, logistic regression [Sorg & Cimiano 2011]
 - Additive Groves [Moreira et al. 2011]
- Pairwise
 - Ranking SVM [Yang et al. 2009]
 - RankBoost, RankNet [Moreira et al. 2011]
- Listwise
 - AdaRank, Coordinate Ascent [Moreira et al. 2011]

Voting models

[Macdonald & Ounis 2006]

- Inspired by techniques from data fusion
 - Combining evidence from different sources
- Documents ranked w.r.t. the query are seen as “votes” for the entity

Voting models

Many different variants, including...

- Votes
 - Number of documents mentioning the entity
$$Score(e, q) = |M(e) \cap R(q)|$$
- Reciprocal Rank
 - Sum of inverse ranks of documents
$$Score(e, q) = \sum_{\{M(e) \cap R(q)\}} \frac{1}{rank(d, q)}$$
- CombSUM
 - Sum of scores of documents
$$Score(e, q) = |\{M(e) \cap R(q)\}| \sum_{\{M(e) \cap R(q)\}} s(d, q)$$

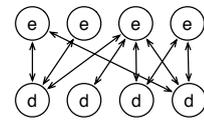
Graph-based models

[Serdyukov et al. 2008]

- One particular way of constructing graphs
 - Vertices are documents and entities
 - Only document-entity edges
- Search can be approached as a random walk on this graph
 - Pick a random document or entity
 - Follow links to entities or other documents
 - Repeat it a number of times

Infinite random walk

[Serdyukov et al. 2008]

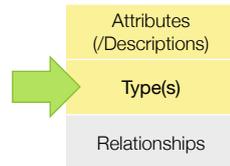


$$P_i(d) = \lambda P_J(d) + (1 - \lambda) \sum_{e \rightarrow d} P(d|e)P_{i-1}(e),$$

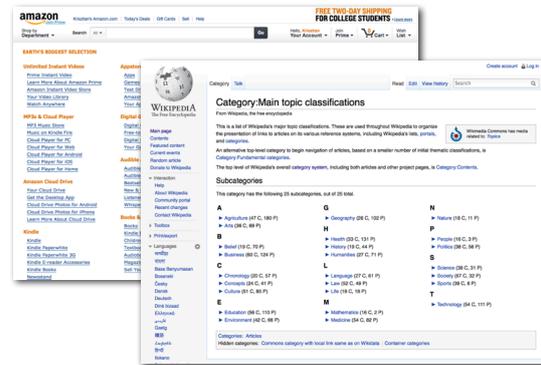
$$P_i(e) = \sum_{d \rightarrow e} P(e|d)P_{i-1}(d),$$

$$P_J(d) = P(d|q),$$

Incorporating entity types



Entities are typed...



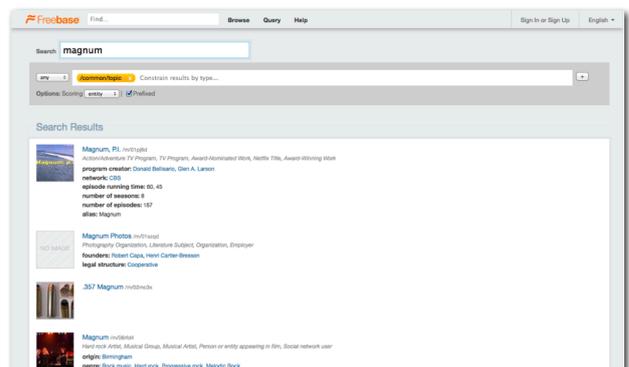
If target type is not provided

Rely on techniques from...

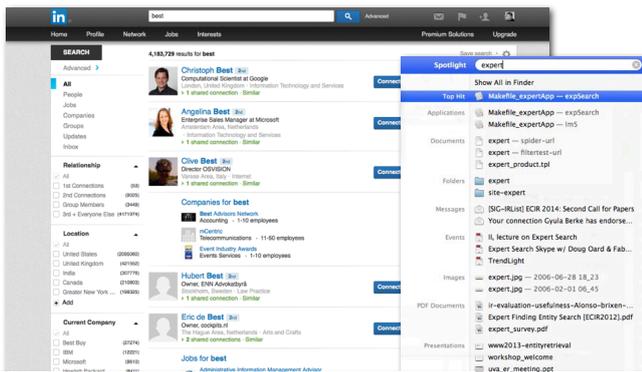
- Federated search
 - Obtain a separate ranking for each type of entity, then merge [Kim & Croft 2010]
- Aggregated search
 - Return top ranked entities from each type [Lalmas 2011]

Example (1)

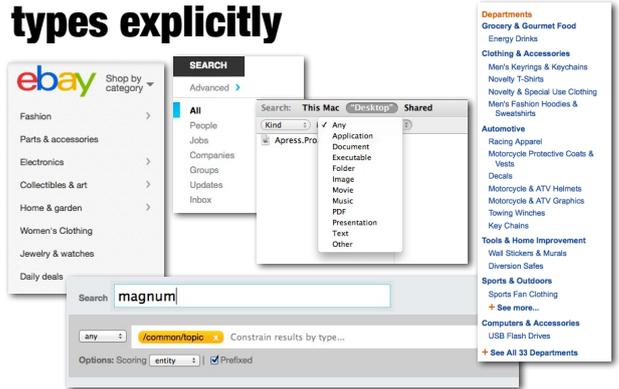
Mixing all types together (“federated search”)



Example (2) Grouping results by entity type ("aggregated search")



Often, users provide target types explicitly



Type-aware entity ranking

- Assume that the user provides target type(s)
- Challenges
 - Target type information is imperfect
 - Users are not familiar with the classification system
 - Categorisation of entities is imperfect
 - Entity might belong to multiple categories
 - E.g. is King Arthur "British royalty", "fictional character", or "military person"?
 - Types can be hierarchically organised
 - Although it may not be a strict "is-a" hierarchy

INEX Entity Ranking track

- Entities are represented by Wikipedia articles
- Topic definition includes target categories



Titanic (1997 film)

From Wikipedia, the free encyclopedia

Titanic is a 1997 American epic romance and disaster film directed, written, co-produced, and co-edited by James Cameron. A fictionalized account of the sinking of the RMS *Titanic*, it stars Leonardo DiCaprio as Jack Dawson and Kate Winslet as Rose DeWitt Bukater, members of different social classes who fall in love aboard the ship during its ill-fated maiden voyage. Although the central roles and love story are fictitious, some characters are based on genuine historical figures. Gloria Stuart portrays the elderly Rose, who narrates the film in a modern-day framing device, and Billy Zane plays Cal Hockley, the overbearing fiancé of the younger Rose. Cameron saw the love story as a way to engage the audience with the real-life tragedy.

Production on the film began in 1995, when Cameron shot footage of the actual *Titanic* wreck. The modern scenes were shot on board the *Akademik Mistral* *Keldysh*, which Cameron had used as a base when filming the actual wreck. A reconstruction of the *Titanic* was built at Playa de Rosarito, Baja California, and scale models and computer-generated imagery were also used to recreate the sinking. The film was partially funded by Paramount Pictures and 20th Century Fox – respectively, its American and international distributor – and at the time, it was the most expensive film ever made, with an estimated budget of US\$200 million.^{[a][b][c]}

The film was originally scheduled to open on July 2, 1997, however, post-production delays pushed back its release to December 19 instead.^[d] *Titanic* was an enormous critical and commercial success. It was nominated for fourteen Academy Awards, eventually winning eleven, including Best Picture and Best Director.^[e] It became the highest-grossing film of all time, with a worldwide gross of over \$1.8 billion, and remained so for twelve years until Cameron's next directorial effort, *Avatar*, surpassed it in 2010.^{[f][g]} *Titanic* also has been ranked as the sixth best epic film of all time in AFI's 10 Top 10 by the American Film Institute.^[h] The film is due for theatrical re-release in 2012 after Cameron completes its conversion into 3-D.^{[i][j]}



Categories: 1997 films | American films | English-language films | American disaster films | Best Drama Picture Golden Globe winners | Best Picture Academy Award winners | Best Song Academy Award winners | Films directed by James Cameron | Films set in 1912 | Films that won the Best Sound Mixing Academy Award | Films that won the Best Visual Effects Academy Award | Films whose art director won the Best Art Direction Academy Award | Films whose cinematographer won the Best Cinematography Academy Award | Films whose director won the Best Director Academy Award | Films whose director won the Best Director Golden Globe | Films whose editor won the Best Film Editing Academy Award | Epic films | RMS Titanic | Romantic epic films | Romantic period films | Seafaring films based on actual events | Films shot in Nova Scotia | Films shot in Vancouver | Paramount films | 20th Century Fox films | Lightstorm Entertainment films | 2-D films converted to 3-D

Using target type information

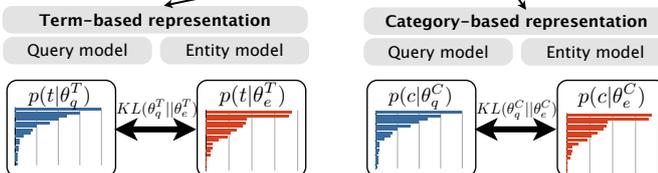
- Constraining results
 - Soft/hard filtering
 - Different ways to measure type similarity (between target types and the types associated with the entity)
 - Set-based
 - Content-based
 - Lexical similarity of type labels
- Query expansion
 - Adding terms from type names to the query
- Entity expansion
 - Categories as a separate metadata field

Modeling terms and categories

[Balog et al. 2011]

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

$$P(q|e) = (1 - \lambda)P(\theta_q^T | \theta_e^T) + \lambda P(\theta_q^C | \theta_e^C)$$



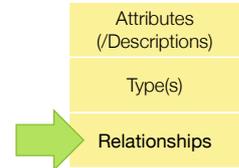
Identifying target types for queries

- Types of top ranked entities [Vallet & Zaragoza 2008]
- Direct term-based vs. indirect entity-based representations [Balog & Neumayer 2012]
- Hierarchical case is difficult... [Sawant & Chakrabarti 2013]

Expanding target types

- Pseudo relevance feedback
- Based on hierarchical structure
- Using lexical similarity of type labels

Entity relationships



Related entities

TREC Entity track

- Related Entity Finding task
- Given
 - Input entity (defined by name and homepage)
 - Type of the target entity (PER/ORG/LOC)
 - Narrative (describing the nature of the relation in free text)
- Return (homepages of) related entities

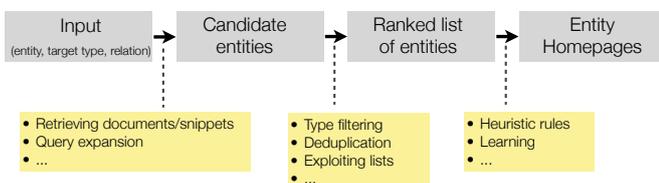
Example information needs

airlines that currently use Boeing 747 planes
ORG Boeing 747

Members of The Beaux Arts Trio
PER The Beaux Arts Trio

What countries does Eurail operate in?
LOC Eurail

A typical pipeline



Modeling related entity finding [Bron et al. 2010]

- Three-component model

$$p(e|E, T, R) \propto p(e|E) \cdot p(T|e) \cdot p(R|E, e)$$

Co-occurrence model
Context model

Type filtering

Anything else?

The usual suspects from document retrieval...

- Priors
 - HITS, PageRank
 - Document link indegree [Kamps & Koolen 2008]
- Pseudo relevance feedback
 - Document-centric vs. entity-centric [Macdonald & Ounis 2007; Serdyukov et al. 2007]
 - sampling expansion terms from top ranked documents and/or (profiles of) top ranked candidates
 - Field-based [Kim & Croft 2011]

Query understanding

- Structuring and segmentation [Bendersky et al. 2010, Bendersky et al. 2011]
- Aiding the user with context-sensitive suggestions [Bast et al. 2012]
- Query interpretation with the help of knowledge bases [Pound et al. 2012]

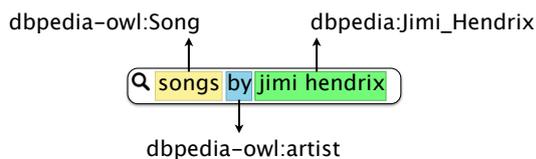
Specialized interfaces

[Bast et al. 2012]

The screenshot shows a search interface with a search bar at the top. Below it, there's a 'Your Query' section with a tree view of filters: 'Plant' (selected), 'occurs-with' (with a sub-filter 'edible leaves'), and 'native-to' (with a sub-filter 'Europe'). The 'Hits' section shows two results: 'Broccoli' and 'Cabbage'. For each, there's an ontology snippet (e.g., 'Broccoli: is a plant; native to Europe.') and a document snippet (e.g., 'The edible portions of Broccoli are the stem tissue, the flower buds, as well as the leaves.'). Images of broccoli and cabbage are also shown.

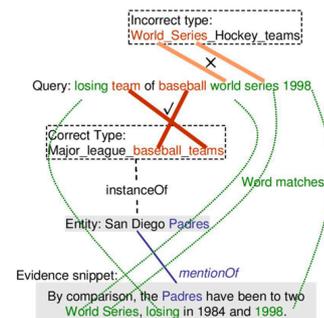
Figure taken from Bast et al. Broccoli: Semantic Full-Text Search at your Fingertips. <http://arxiv.org/abs/1207.2615/>

Semantic query understanding



Joint query interpretation and response ranking

[Sawant & Chakrabarti 2013]



Tools & services



Public Toolkits and Web Services for Entity Retrieval

- EARS
- Sindice & SIREn
- Sig.ma

EARS



- Entity and Association Retrieval System
 - open source, built on top of Lemur in C++
 - not actively maintained anymore (but still works)
- Entity-topic association finding models
 - suited for other tasks, e.g. blog distillation
 - focuses on two entity-related tasks:
 - finding entities:
 - "Which entities are associated with topic X?"
 - profiling entities:
 - "What topics is an entity associated with?"
- See <https://code.google.com/p/ears/>

Sindice/SIREn

- Handling of semi-structured data
 - efficient, large scale
 - typically based on DBMS backends
 - uses Lucene for semi-structured search
- Open source
- Online demo, local install
- See <http://siren.sindice.com/>

Sig.ma

- Search, aggregate, and visualize LOD data
- Powered by Sindice
- See <http://sig.ma/>



Test collections

Test collections

Campaign	Task	Collection	Entity repr.	#Topics
TREC Enterprise (2005-08)	Expert finding	Enterprise intranets (W3C, CSIRO)	Indirect	99 (W3C) 127 (CSIRO)
TREC Entity (2009-11)	Rel. entity finding	Web crawl (ClueWeb09)	Indirect	120
	List completion			70
INEX Entity Ranking (2007-09)	Entity search	Wikipedia	Direct	55
	List completion			
SemSearch Chall. (2010-11)	Entity search	Semantic Web crawl (BTC2009)	Direct	142
	List search			50
INEX Linked Data (2012-13)	Ad-hoc search	Wikipedia + RDF (Wikipedia-LOD)	Direct	100 (*12)
				144 (*13)

Test collections (2)

- Entity search as Question Answering
 - TREC QA track
 - QALD-2 challenge
 - INEX-LD Jeopardy task
- DBpedia entity search [Balog & Neumayer 2013]
 - synthesized queries and assessments, distilled from previous campaigns
 - from short keyword queries to natural language questions
 - 485 queries in total; mapped to DBpedia

Open challenges

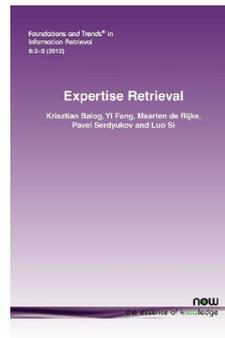
Open challenges

- Combining text and structure
 - Knowledge bases and unstructured Web documents
- Query understanding and modeling
- UI/UX/Result presentation
 - How to interact with entities
- Hyperlocal
 - Siri/Google Now/...
 - Recommendations

Open challenges (2)

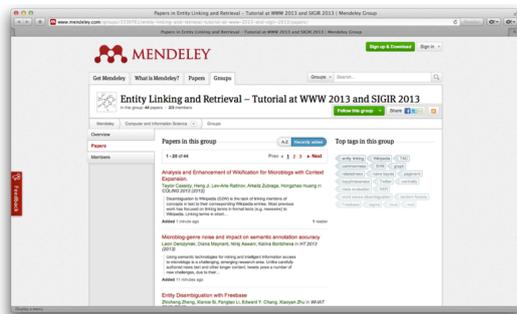
- There is more to types than currently exploited
 - Multiple category systems, hierarchy, ...
- Entity retrieval is typically part of some more complex task
 - Buying a product, planning a vacation, etc.

Follow-up reading



K. Balog, Y. Fang, M. de Rijke, P. Serdyukov, and L. Si. **Expertise Retrieval**. *FnTIR'12*.

References – Entity retrieval



<http://www.mendeley.com/groups/3339761/entity-linking-and-retrieval-tutorial-at-www-2013-and-sigir-2013/papers/added/0/tag/entity+retrieval/>