Part II

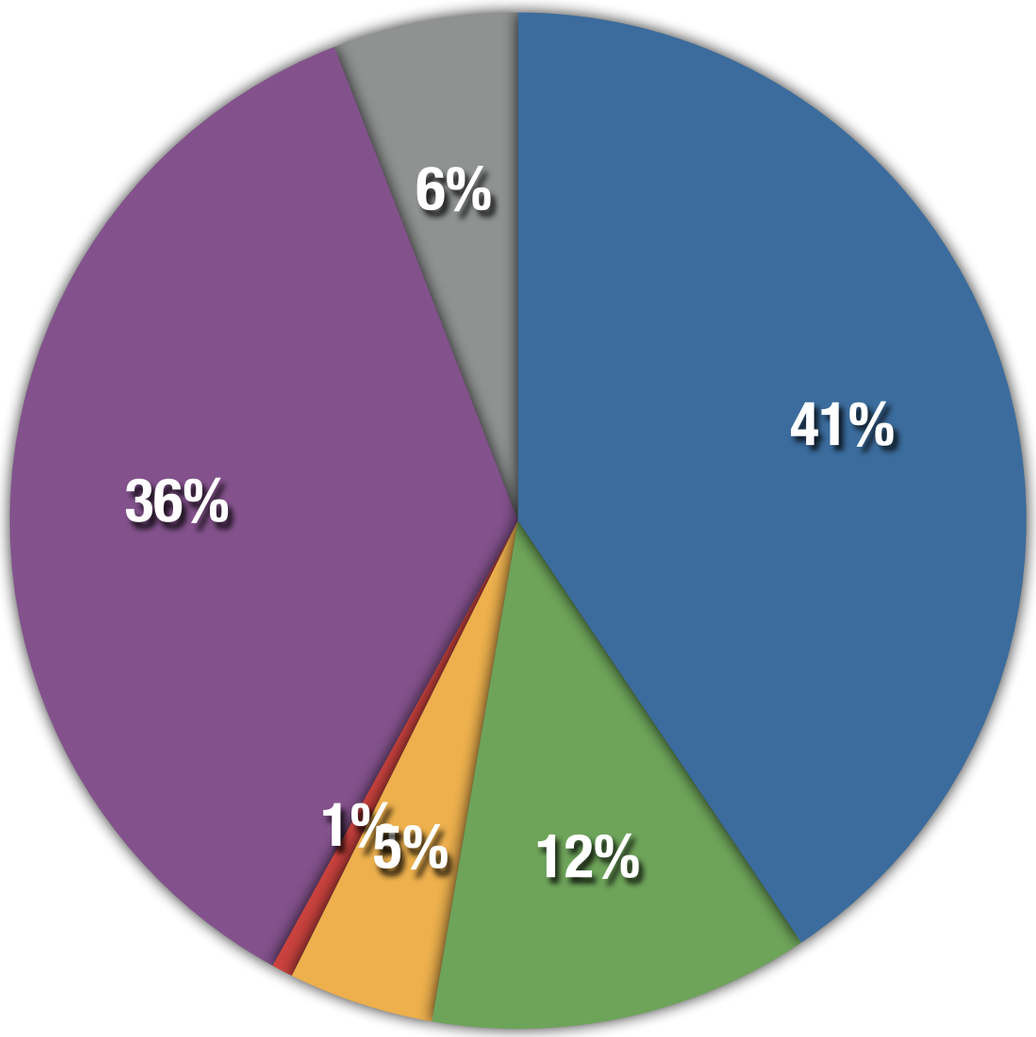
Entity Retrieval

Tutorial organized by Radialpoint | Montreal, Canada, 2014

Entity retrieval

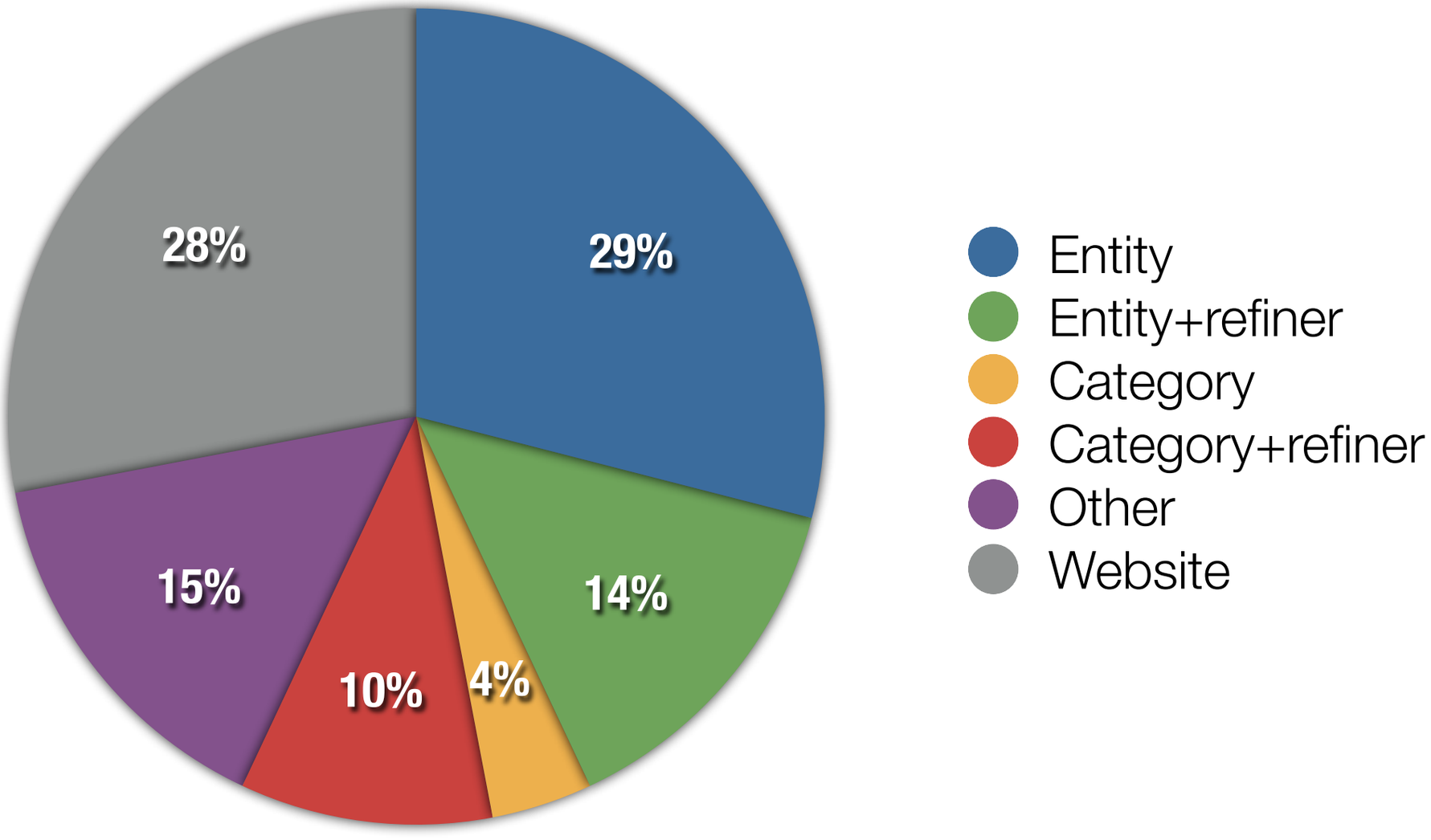
*Addressing information needs that are better answered by **returning specific objects** (entities) instead of just any type of documents.*

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



- Entity (“1978 cj5 jeep”)
- Type (“doctors in barcelona”)
- Attribute (“zip code waterville Maine”)
- Relation (“tom cruise katie holmes”)
- Other (“nightlife in Barcelona”)
- Uninterpretable

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

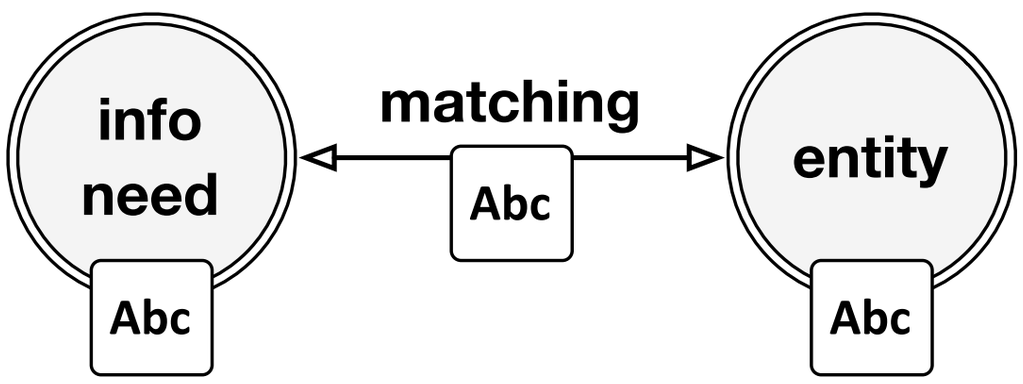


What's so special here?

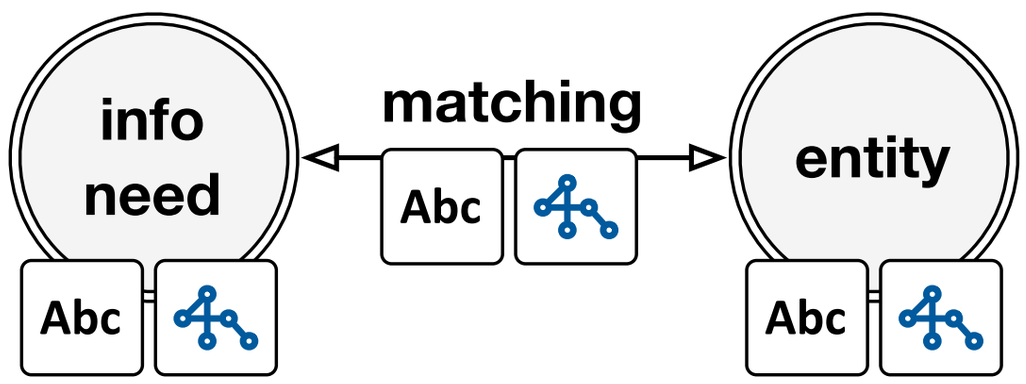
- Entities are not always directly represented
  - Recognize and disambiguate entities in text (that is, entity linking)
  - Collect and aggregate information about a given entity from multiple documents and even multiple data collections
- More structure than in document-based IR
  - Types (from some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

Semantics in our context

- working definition:
  references to meaningful structures
- How to capture, represent, and use structure?
- It concerns all components of the retrieval process!



Text-only representation



Text+structure representation

Overview of core tasks

	Queries	Data set	Results
(adhoc) entity retrieval	keyword	unstructured/ semi-structured	ranked list
	keyword++ (target type(s))	semi-structured	ranked list
list completion	keyword++ (examples)	semi-structured	ranked list
related entity finding	keyword++ (target type, relation)	unstructured & semi-structured	ranked list

In this part

- Input: keyword(++) query
- Output: a ranked list of entities
- Data collection: unstructured and (semi)structured data sources (and their combinations)
- Main RQ: **How to incorporate structure into text-based retrieval models?**

Outline

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

Attributes (/Descriptions)
Type(s)
Relationships

Probabilistic models (mostly)

- Estimating conditional probabilities

$$P(A|B)$$
$$P(A, B|C)$$

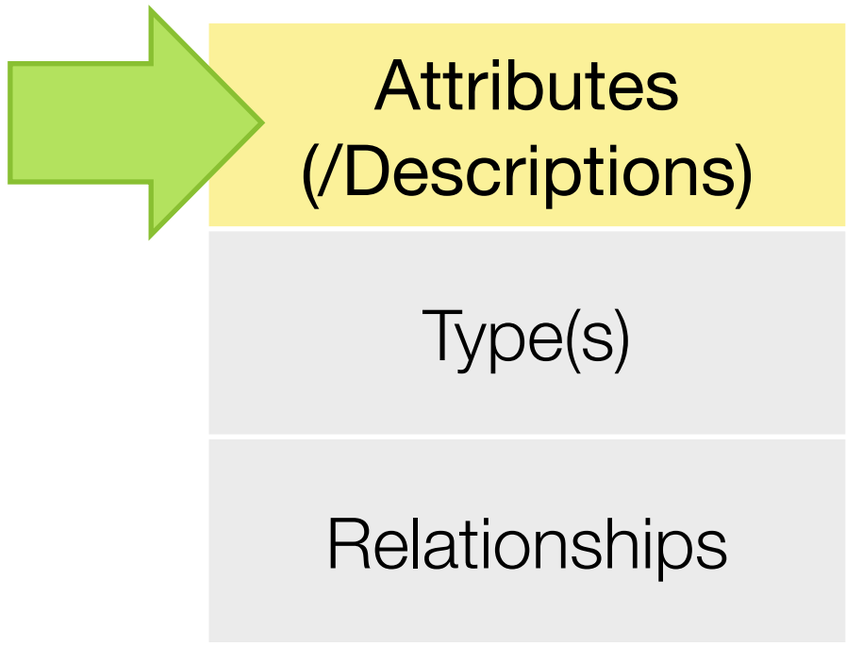
- Conditional independence

$$P(A|B) = P(A)$$
$$P(A, B|C) = P(A|C) \cdot P(B|C)$$

- Conditional dependence

$$P(A|B) = P(B|A)P(A)/P(B)$$
$$P(A, B|C) = P(A|B, C)P(B|C)$$

Ranking 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

american embassy nairobi

ben franklin

Chernobyl

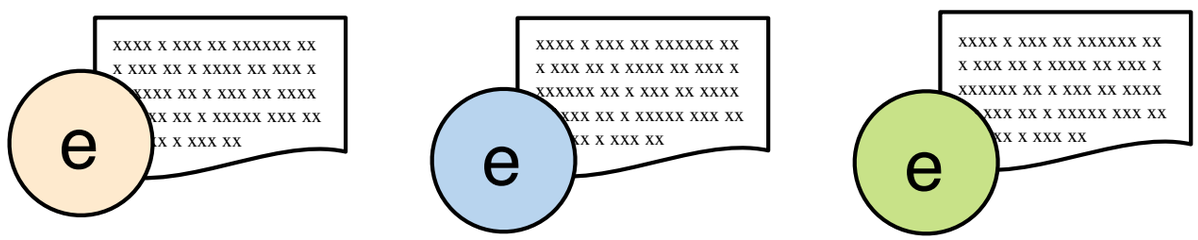
meg ryan war

Worst actor century

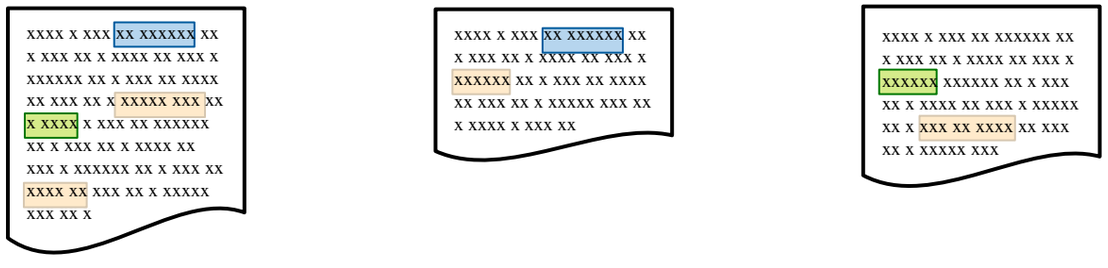
Sweden Iceland currency

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

Crash course into Language modeling

Example

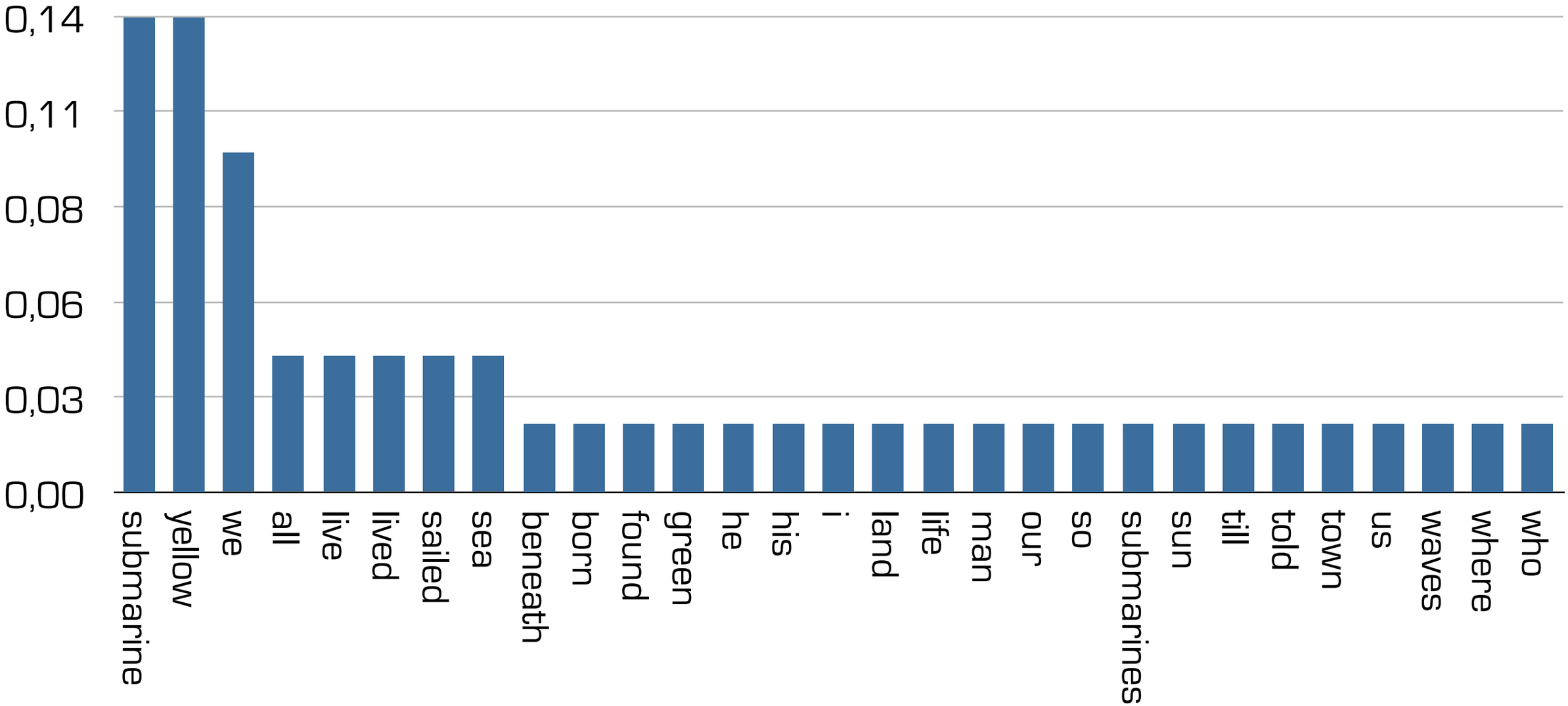
In the town where I was born,
Lived a man who sailed to sea,
And he told us of his life,
In the land of submarines,

So we sailed on to the sun,
Till we found the sea green,
And we lived beneath the waves, In our yellow submarine,

We all live in yellow submarine, yellow submarine, yellow submarine, We all live in yellow submarine, yellow submarine, yellow submarine.

Empirical document LM

$$P(t|d) = \frac{n(t,d)}{|d|}$$



Alternatively...

Scoring a query

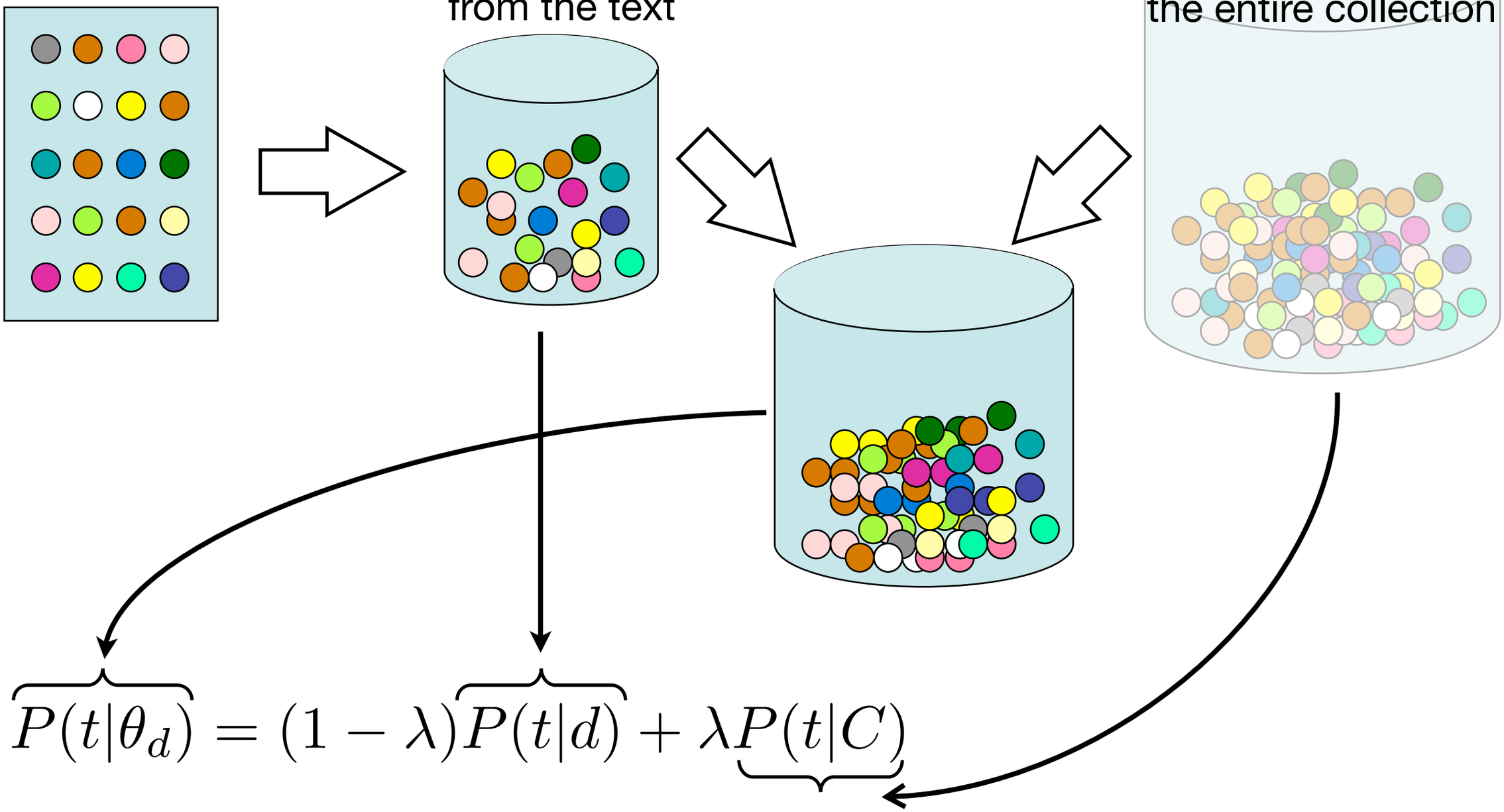
$q = \{\text{sea}, \text{submarine}\}$

$P(q|d) = P(\text{“sea”}|\theta_d) \cdot P(\text{“submarine”}|\theta_d)$

Language Modeling

Estimate a multinomial probability distribution from the text

Smooth the distribution with one estimated from the entire collection



$$\overbrace{P(t|\theta_d)} = (1-\lambda)\overbrace{P(t|d)} + \lambda\underbrace{P(t|C)}$$

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)

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

Number of times *t* appears in *q*

Document language model
Multinomial probability distribution over the vocabulary of terms

Smoothing parameter

$$P(t|\theta_d) = (1-\lambda)P(t|d) + \lambda P(t|C)$$

Empirical document model

Maximum likelihood estimates

Collection model

$$\frac{n(t,d)}{|d|} \qquad \frac{\sum_d n(t,d)}{\sum_d |d|}$$

Scoring a query

$$q = \{\text{sea}, \text{submarine}\}$$

$$P(q|d) = \overbrace{P(\text{“sea”}|\theta_d) \cdot P(\text{“submarine”}|\theta_d)}^{0.03602}$$

$$(1-\lambda)P(\text{“sea”}|d) + \lambda P(\text{“sea”}|C)$$

0.9 0.04 0.1 0.0002

t	P(t|d)
submarine	0,14
sea	0,04
...	

t	P(t|C)
submarine	0,0001
sea	0,0002
...	

Scoring a query

$$q = \{\text{sea}, \text{submarine}\}$$

0.04538 0.03602 0.12601

$$P(q|d) = P(\text{“sea”}|\theta_d) \cdot P(\text{“submarine”}|\theta_d)$$

0.9 0.14 0.1 0.0001

$$(1-\lambda)P(\text{“submarine”}|d) + \lambda P(\text{“submarine”}|C)$$

t	P(t|d)
submarine	0,14
sea	0,04
...	

t	P(t|C)
submarine	0,0001
sea	0,0002
...	

Here, documents==entities, so

$$P(e|q) \propto P(e)P(q|\theta_e) = \underbrace{P(e)}_{\text{Entity prior}} \prod_{t \in q} \underbrace{P(t|\theta_e)}_{\text{Entity language model}}^{n(t,q)}$$

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

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact Wikipedia

Toolbox

Print/export

Languages
العربية
Беларуская
Български
Česky
Dansk
Deutsch

Audi A4

From Wikipedia, the free encyclopedia

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.[2]

The Audi A4 automobile layout consists of a longitudinally oriented engine at the front, with transaxle-type transmissions mounted at the rear of the engine. The cars are front-wheel drive, or on some models, "quattro" all-wheel drive.

The A4 is available as a saloon/sedan and estate/wagon. The second (B6) and third generations (B7) of the A4 also had a convertible version, but the B8 version of the convertible became a variant of the Audi A5 instead as Audi got back into the compact executive coupé segment. The facebook fans of the Audi A4 page are more than 870,000.

Contents [show]

Audi A4

Manufacturer	Audi
Production	1994–present
Assembly	Ingolstadt, Germany Changchun, China[1] Tokyo, Japan (AMA; B5 only) Jakarta, Indonesia (Garuda Mataram Motor; B5 & B8) Solomonovo, Ukraine (Eurocar; B7 only) Aurangabad, India
Predecessor	Audi 80
Class	Compact executive car (globally)
Layout	front-engine, front-wheel-drive front-engine, four-wheel-drive
Platform	Volkswagen Group B

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**
is **dbpedia-owl:predecessor** of	**dbpedia:Audi_A5**
is **dbpprop:similar** of	**dbpedia:Cadillac_BLS**

Mixture of Language Models

[Ogilvie & Callan 2003]

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

$$P(t|\theta_d) = \sum_{j=1}^{m} \underbrace{\mu_j}_{\text{Field weights}} P(t|\underbrace{\theta_{d_j}}_{\text{Field language model}})$$

$$\sum_{j=1}^{m} \mu_j = 1$$

Field language model
Smoothed with a collection model built from all document representations of the same type in the collection

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]

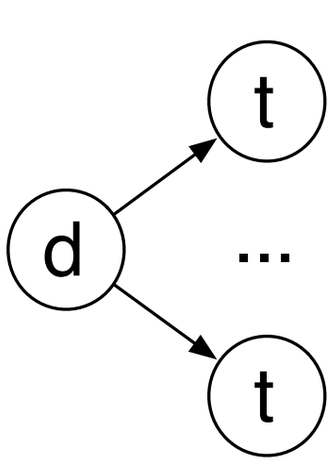
- Organize fields into a 2-level hierarchy
  - Field types (4) on the top level
  - Individual fields of that type on the bottom level
- Estimate field weights
  - Using training data for field types
  - Using heuristics for bottom-level types

Two-level hierarchy

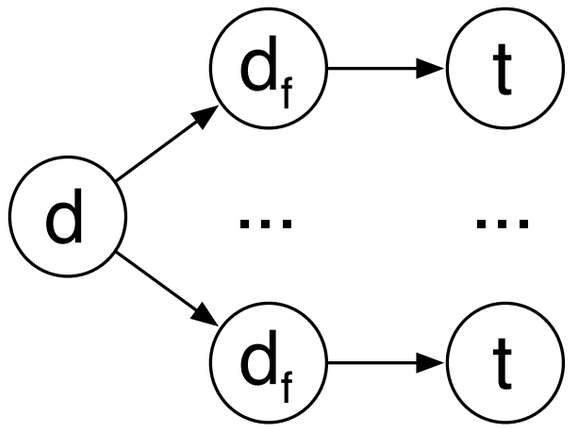
[Neumayer et al. 2012]

Name	**foaf:name**	Audi A4
	rdfs:label	Audi A4
Attributes	**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
Out-relations	**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**
In-relations	is **dbpedia-owl:predecessor** of	**dbpedia:Audi_A5**
	is **dbpprop:similar** of	**dbpedia:Cadillac_BLS**

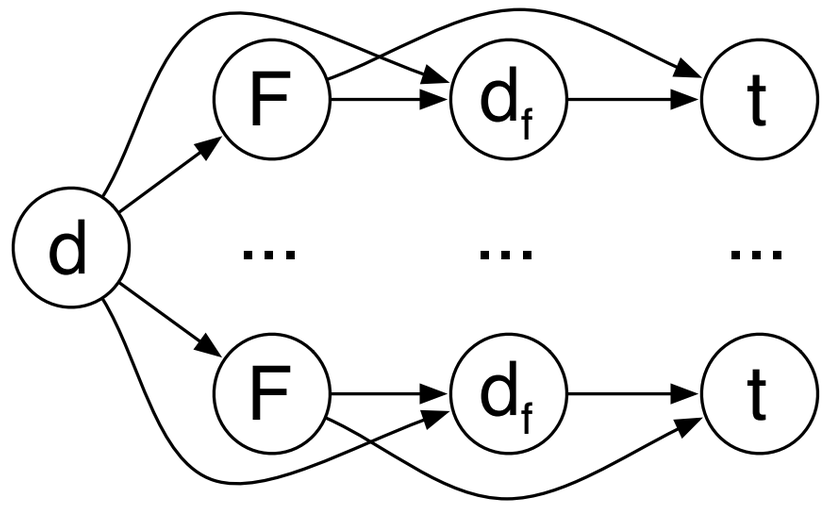
Comparison of models



Unstructured document model



Fielded document model



Hierarchical document model

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

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

Estimating the mapping probability

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

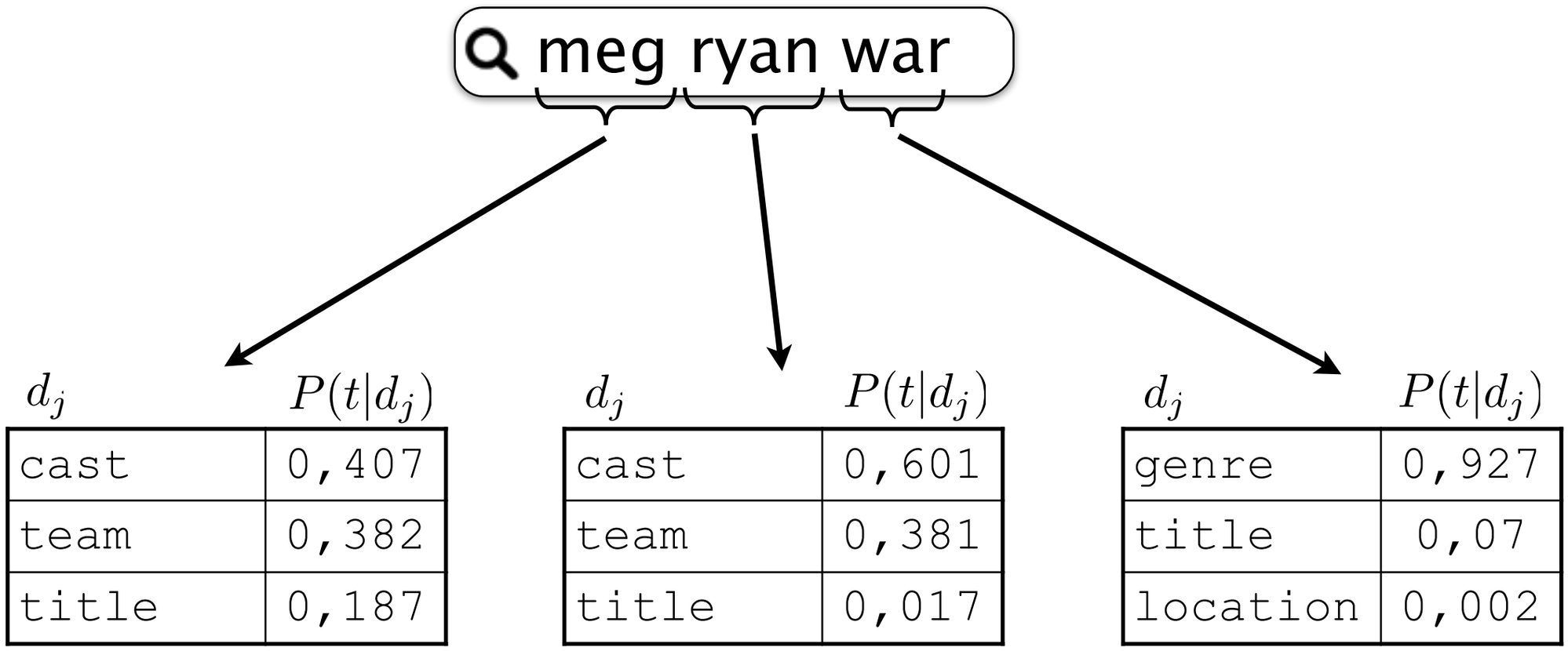
Term likelihood
Probability of a query term occurring in a given field type

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

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

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

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

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

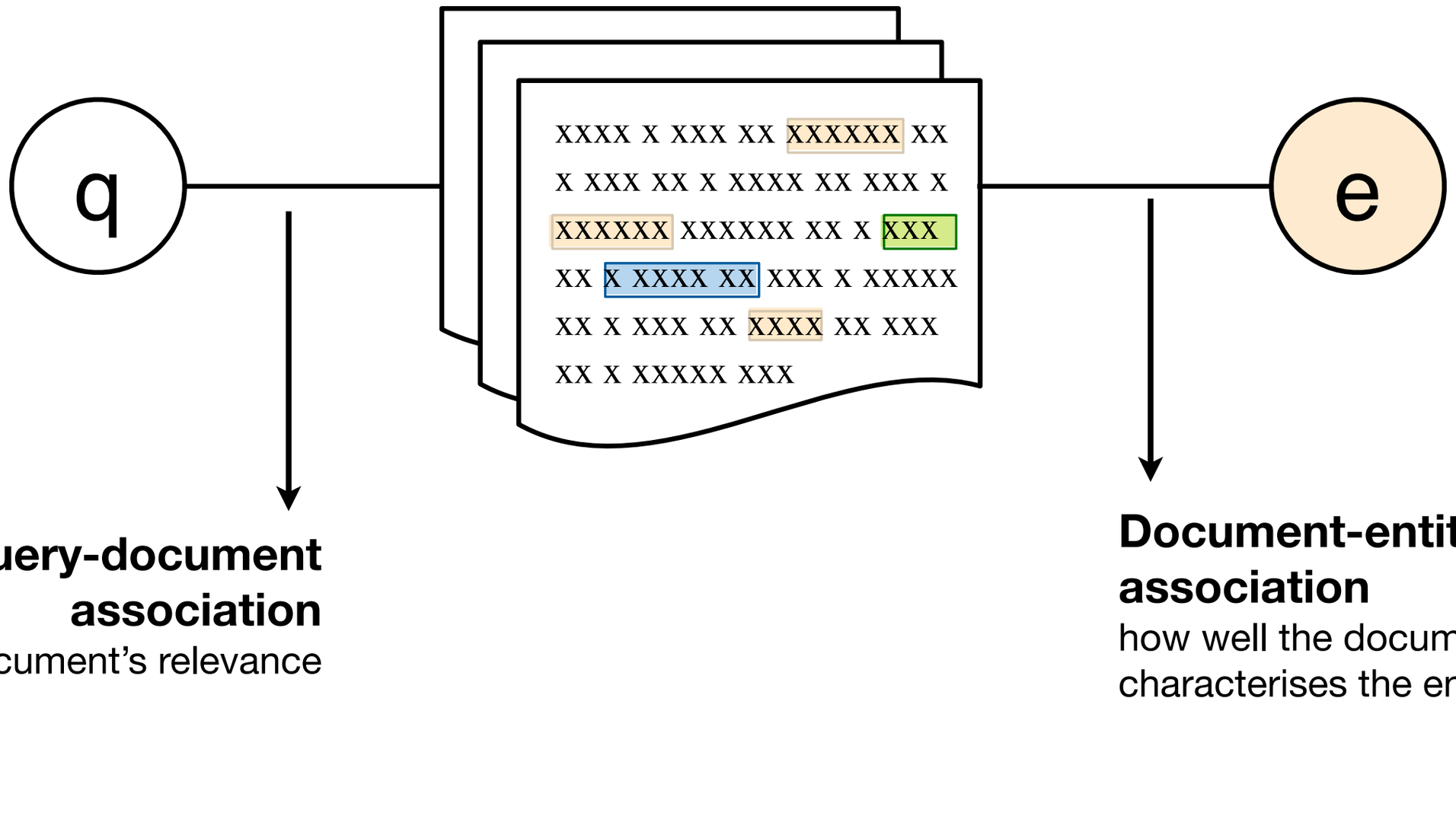
Ranking without explicit entity representations

Scenario

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

The basic idea

Use documents to go from queries to entities



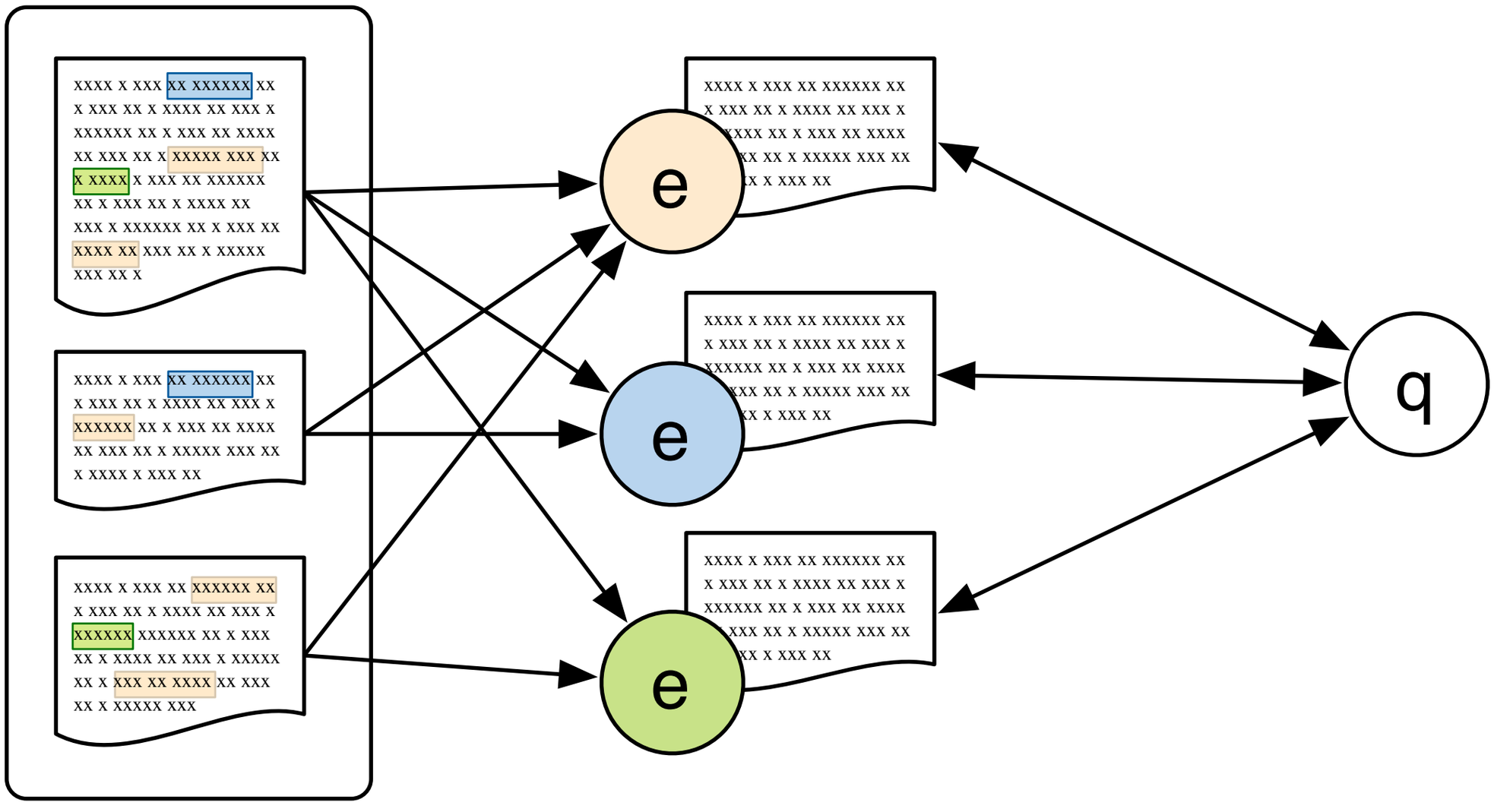
Query-document association
the document's relevance

Document-entity association
how well the document characterises the entity

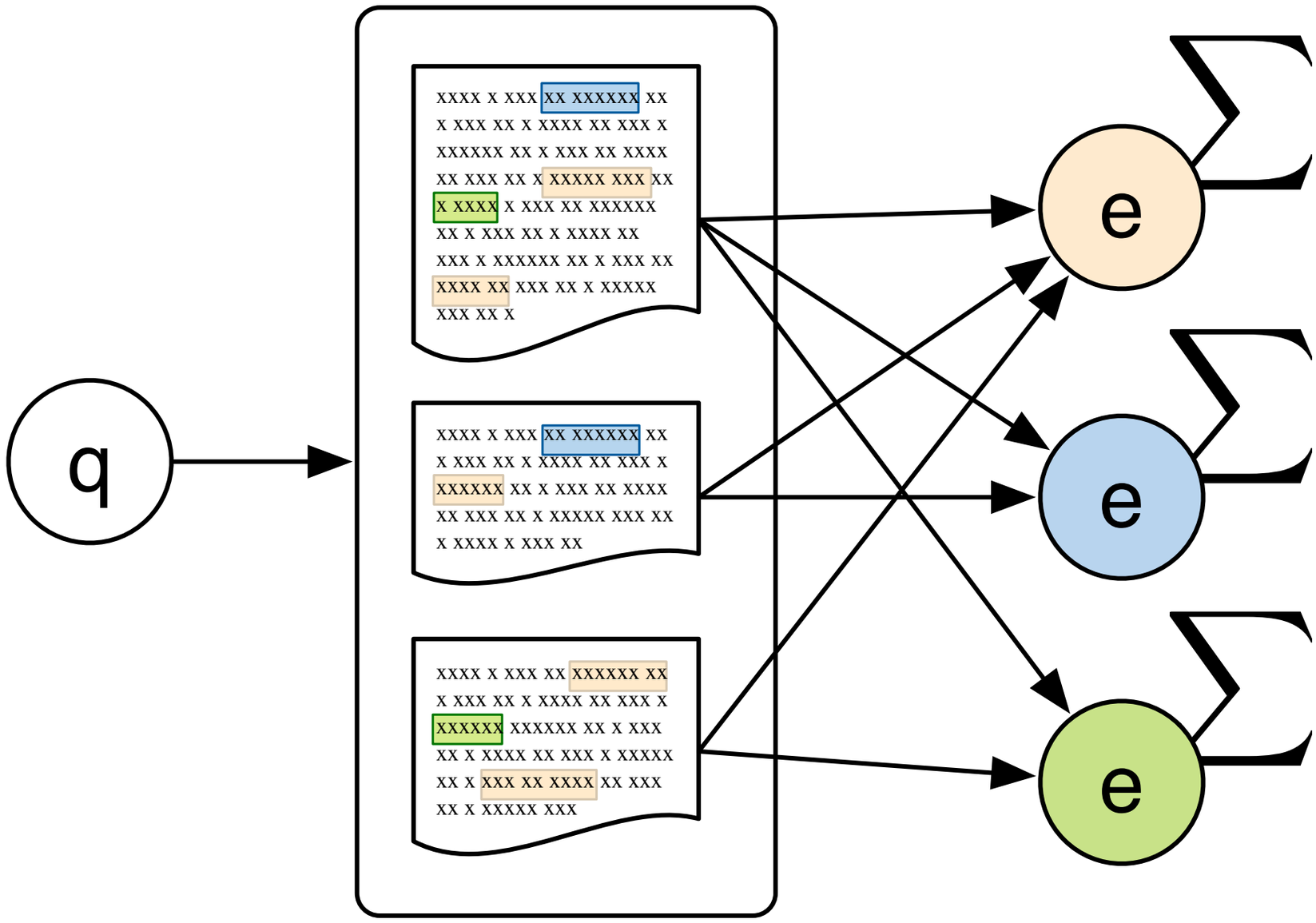
Two principal approaches

- **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]

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

Smoothing
With collection-wide background model

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

$$\sum_d \underbrace{P(t|d,e)}\underbrace{P(d|e)}$$

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]

$$P(q|e) = \sum_d \underbrace{P(q|d,e)}_{\text{Document relevance}} \underbrace{P(d|e)}_{\text{Document-entity association}}$$

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

Document-entity association

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

Simplifying assumption (t and e are conditionally independent given 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]

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

Normalizing contant

Indicator function
1 if the term at position i is t,
0 otherwise

Proximity-based kernel
- constant function
- triangle kernel
- Gaussian kernel
- step function

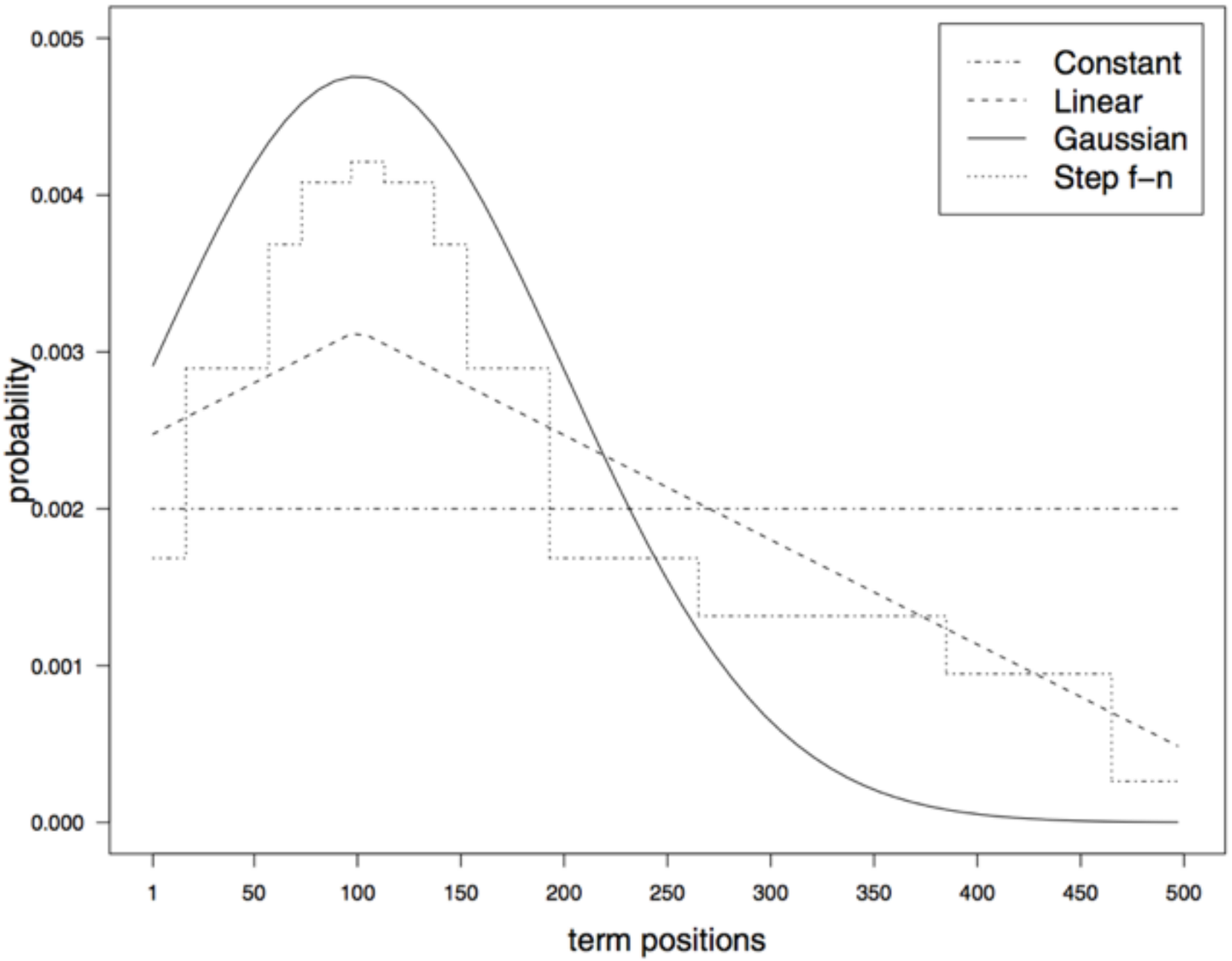


Figure taken from D. Petkova and W.B. Croft. Proximity-based document representation for named entity retrieval. **CIKM'07.**

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]

$$P_\theta(r=1|e,q) = \sum_d \underbrace{P(r_1=1|q,d)}_{\text{Query-document relevance}} \underbrace{P(r_2=1|e,d)}_{\text{Document-entity relevance}} \underbrace{P(d)}_{\text{Document prior}}$$

logistic function
over a linear combination of features

$$\underbrace{\sigma}_{\text{standard logistic function}}\Big(\sum_{i=1}^{N_f} \underbrace{\alpha_i}_{\text{weight parameters (learned)}} \underbrace{f_i(q,d_t)}_{\text{features}}\Big) \qquad \sigma\Big(\sum_{j=1}^{N_g} \beta_j g_j(e,d_t)\Big)$$

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

$$Score(e, q) = |\{M(e) \cap R(q)\}| \sum_{\{M(e) \cap R(q)\}} s(d, q)$$

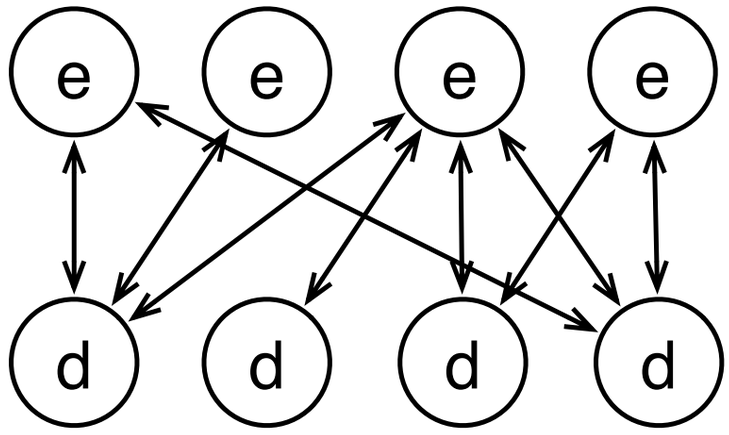
Graph-based models

[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]

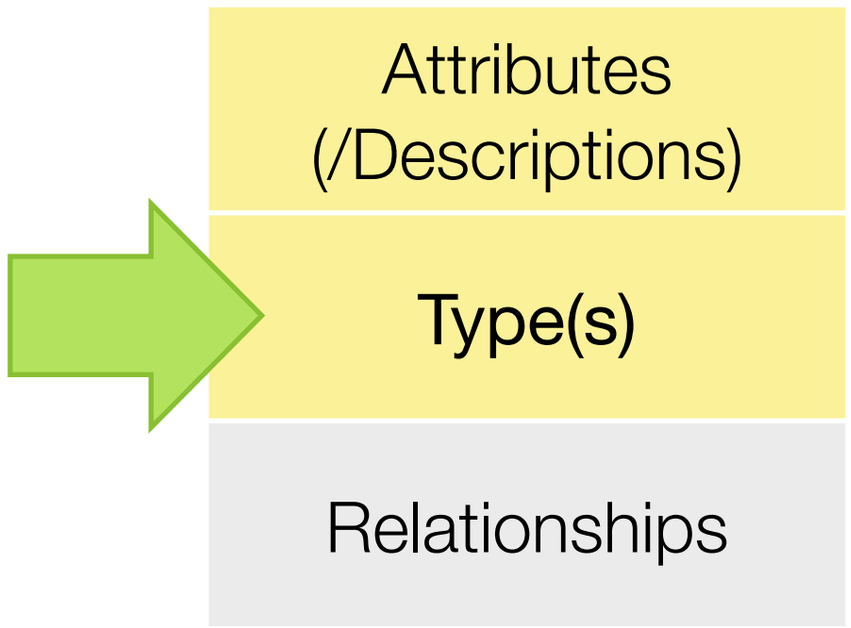


$$
\begin{aligned}
P_i(d) &= \lambda P_J(d) + (1-\lambda) \sum_{e \to d} P(d|e) P_{i-1}(e), \\
P_i(e) &= \sum_{d \to e} P(e|d) P_{i-1}(d), \\
P_J(d) &= P(d|q),
\end{aligned}
$$

Evaluation

- Expert finding task @TREC Enterprise track
  - 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

Incorporating entity types



people

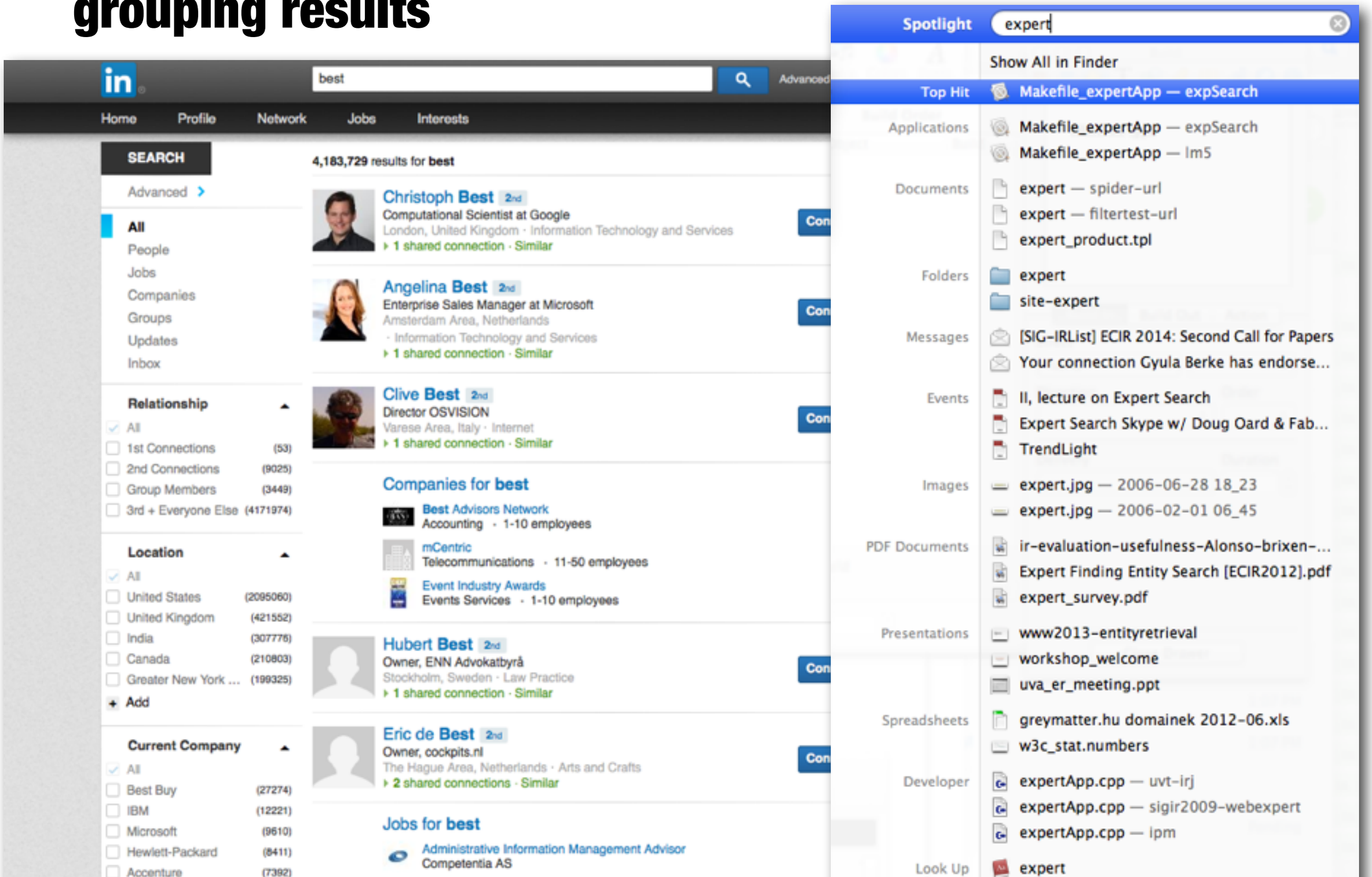
organizations

locations

products

Interacting with types

grouping results



Interacting with types

filtering results

amazon Try Prime

Krisztian's Amazon.com | Today's Deals | Gift Cards | Sell | Help

Get ready for Summer › Shop now

Shop by Department | Search | All | information retrieval book | Go | Hello, Krisztian Your Account | Try Prime | 4 Cart | Wish List

1-16 of 16,451 results for "information retrieval book"

Choose a Department to sort

Show results for

Books ›
- Computer Network Administration
- Databases
- Reference
- Object-Oriented Design
- Computer Programming
- + See more

Kindle Store ›
- Computers & Technology
- Computer Databases
- Computer Programming

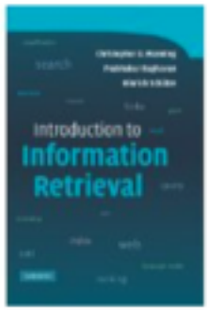
+ See All 11 Departments

Refine by

Eligible for Free Shipping
Free Shipping by Amazon

Book Format
Hardcover
Kindle Edition

Avg. Customer Review
★★★★☆ & Up
★★★☆☆ & Up
★★☆☆☆ & Up
★☆☆☆☆ & Up



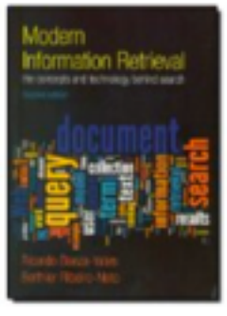
Introduction to Information Retrieval by Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze (Jul 7, 2008)

$18.25 to rent Hardcover Prime
$56.26 to buy
Usually ships in 2 to 4 weeks

$39.36 Kindle Edition
Auto-delivered wirelessly

More Buying Choices - Hardcover
$47.97 new (46 offers)
$30.00 used (47 offers)

★★★★½ (18)
FREE Shipping
Trade-in eligible for an Amazon gift card
Books: See all 16,430 items



Modern Information Retrieval: The Concepts and Technology behind Search (2nd Edition) (ACM Press Books) by Ricardo Baeza-Yates and Berthier Ribeiro-Neto (Feb 10, 2011)

$32.20 to rent Paperback Prime
$61.86 to buy
Only 19 left in stock - order soon.

More Buying Choices - Paperback
$55.00 new (40 offers)
$52.49 used (20 offers)

★★★½☆ (3)
FREE Shipping
Trade-in eligible for an Amazon gift card
Books: See all 16,430 items

Introduction to Modern Information Retrieval, 3rd Edition by G. G. Chowdhurry (Jul 31, 2010)

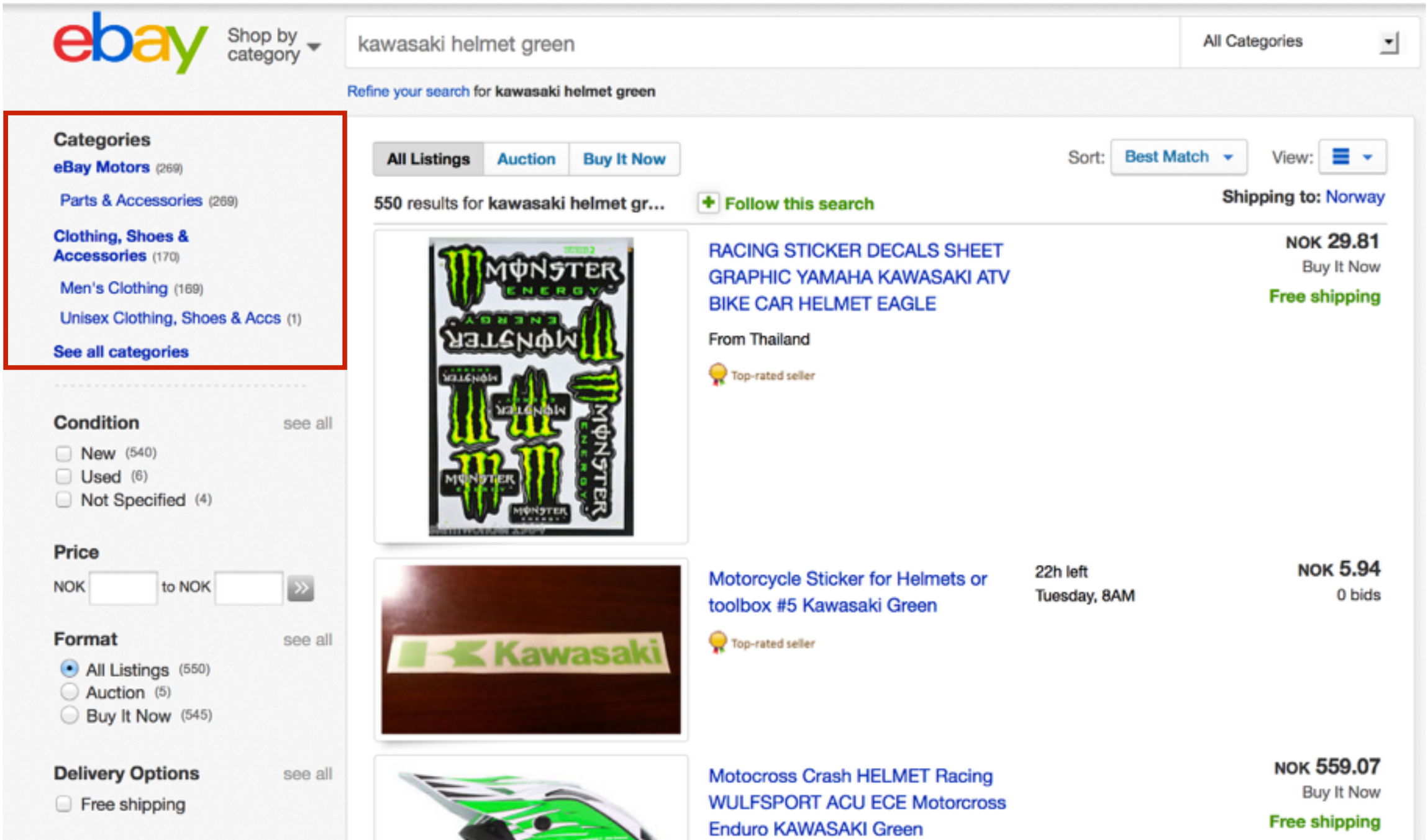
$66.24 to rent Paperback Prime
$90.20 to buy
Only 8 left in stock - order soon.

More Buying Choices - Paperback
$90.20 new (5 offers)
$70.00 used (17 offers)

★★★★★ (1)
FREE Shipping
Trade-in eligible for an Amazon gift card
Books: See all 16,430 items

Interacting with types

filtering results



Type-aware ranking

- Typically, a two-component model:

$$P(q|e) = P(q_T, q_t|e) = \underbrace{P(q_T|e)}_{\text{Type-based similarity}} \underbrace{P(q_t|e)}_{\text{Term-based similarity}}$$

#1 **Where to get the *target type* from?**

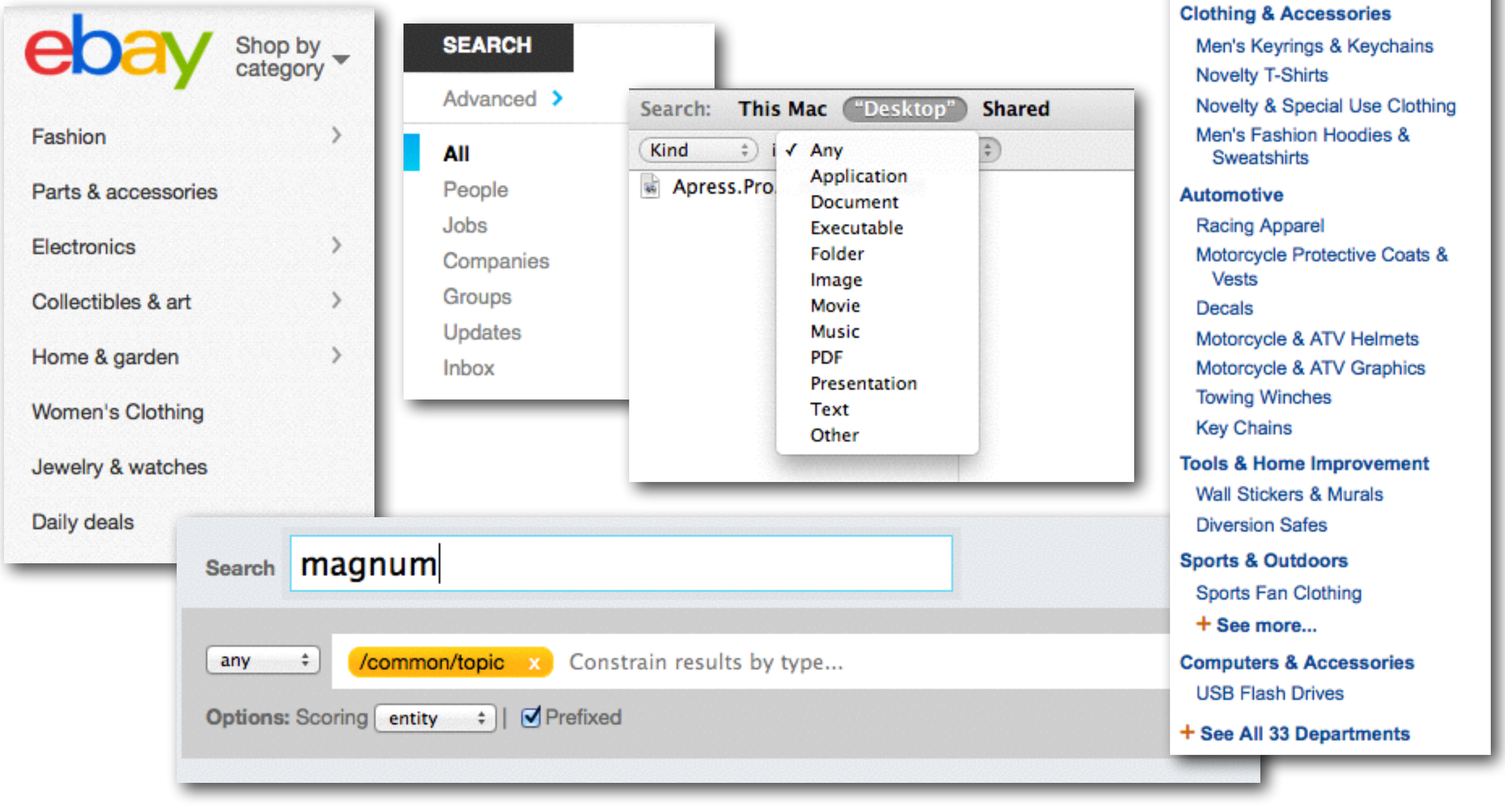
#2 **How to estimate *type-based similarity*?**

Target type

- Provided by the user
  - keyword++ query
- Need to be automatically identified
  - keyword query

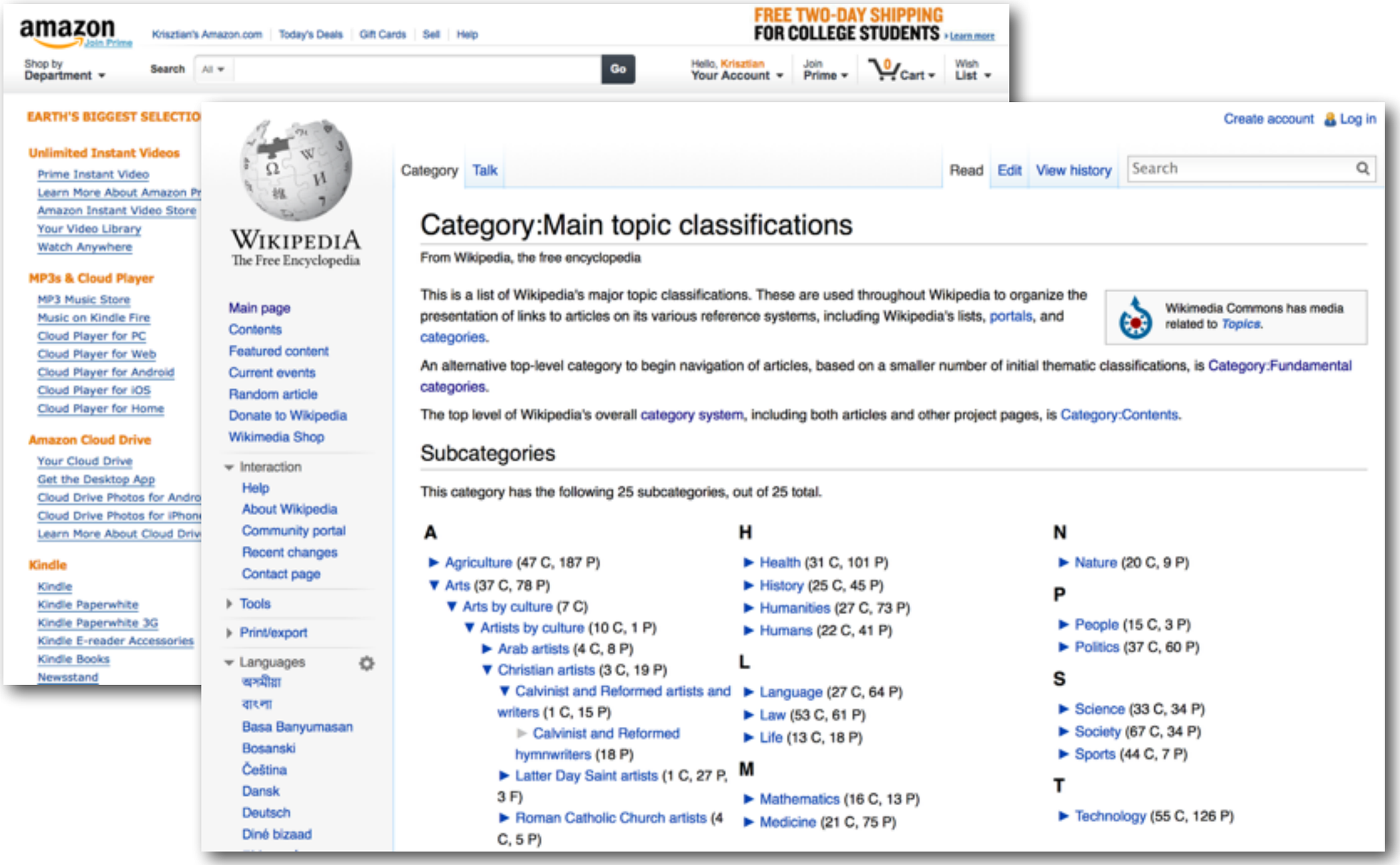
Target type(s) are provided

faceted search, form fill-in, etc.



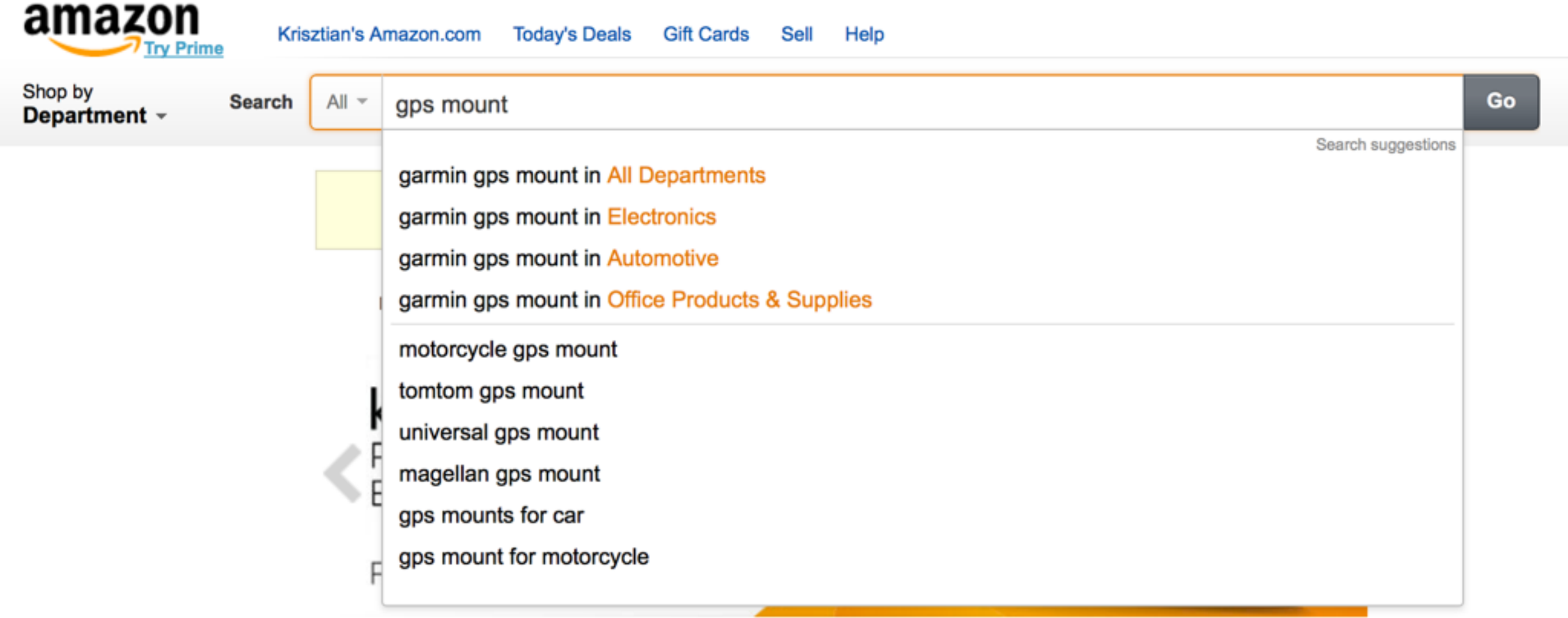
But what about very many types?

which are typically hierarchically organized



Challenges

- Users are not familiar with the type system



In general, categorizing things can be hard

- What is *King Arthur*?
  - Person / Royalty / British royalty
  - Person / Military person
  - Person / Fictional character

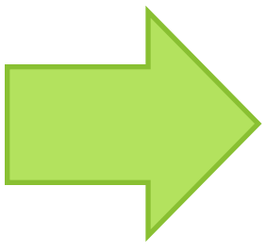
Which King Arthur?!

Upshot for type-aware ranking

- Need to be able to handle the imperfections of the type system
  - Inconsistencies
  - Missing assignments
  - Granularity issues
    - Entities labeled with too general or too specific types
- User input is to be treated as a hint, not as a strict filter

Two settings

- ~~Target type(s) are provided by the user~~
  - ~~keyword++ query~~

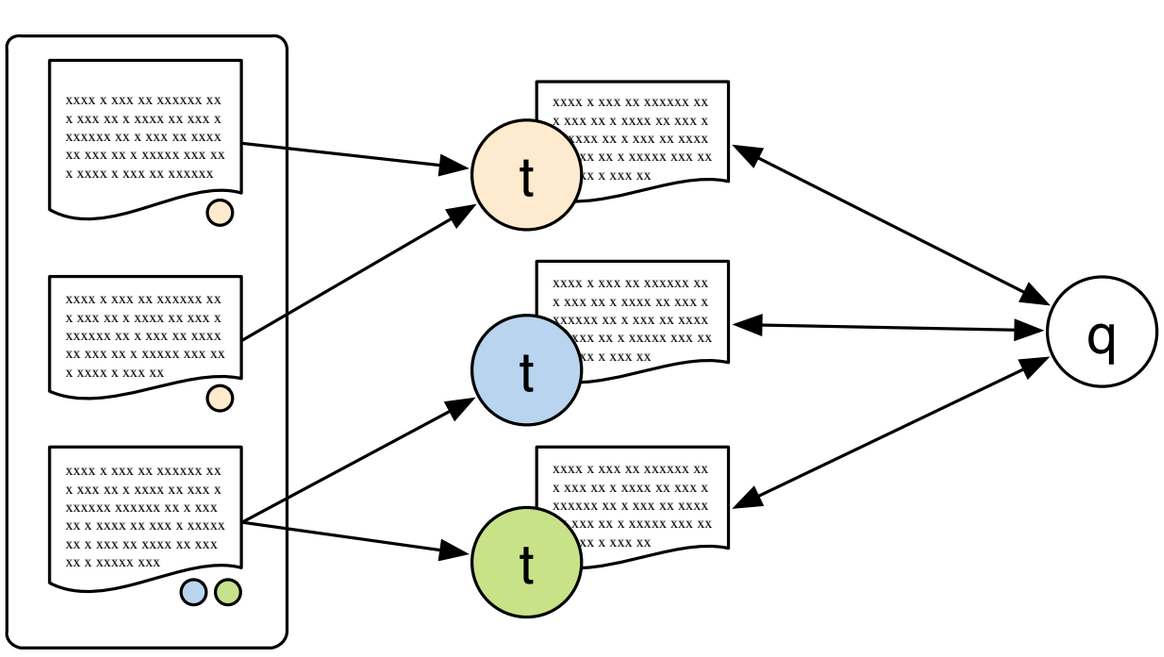


- Target types need to be automatically identified
  - keyword query

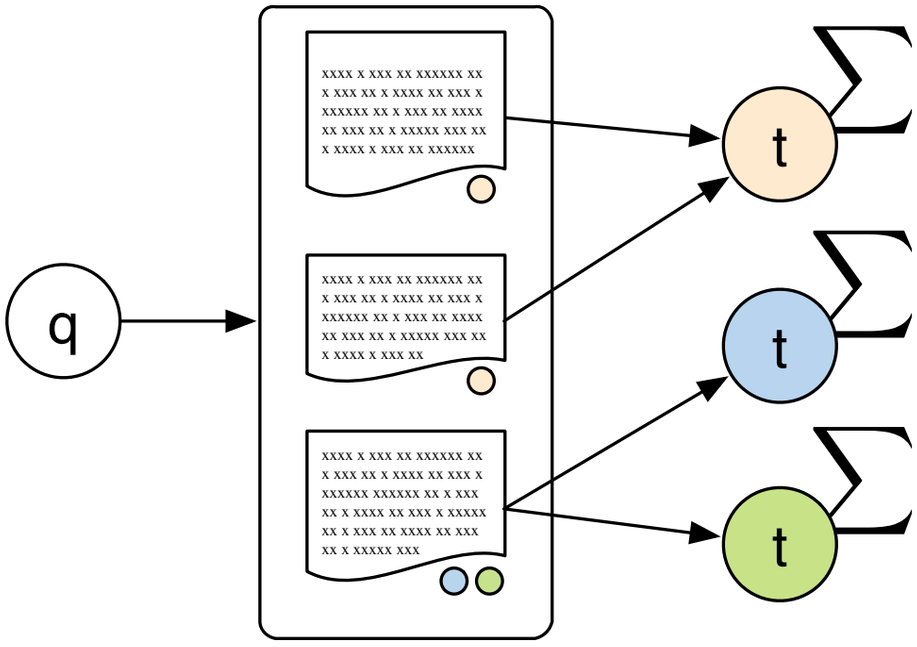
Identifying target types for queries

- Types can be ranked much like entities **[Balog & Neumayer 2012]**
  - Direct term-based representations (“Model 1”)
  - Types of top ranked entities (“Model 2”) **[Vallet & Zaragoza 2008]**

Type-centric vs. entity-centric type ranking



Type-centric



Entity-centric

Hierarchical target type identification

- *Finding the single most specific type [from an ontology] that is general enough to cover all entities that are relevant to the query.*
- Finding the right granularity is difficult…
  - Models are good at finding either general (top-level) or specific (leaf-level) types

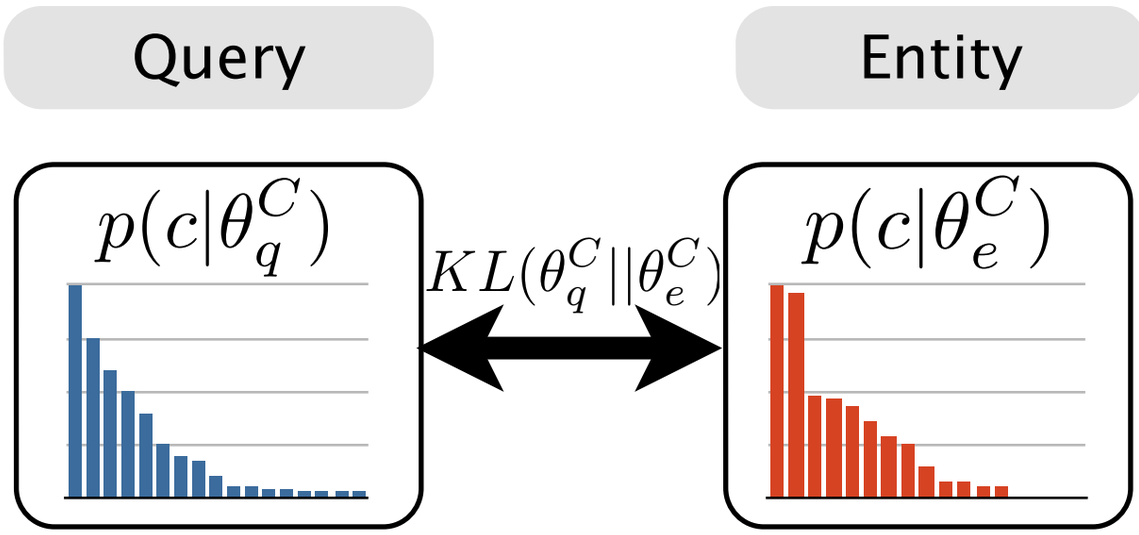
Type-based similarity

$$P(q|e) = P(q_T, q_t|e) = P(q_T|e)P(q_t|e)$$

- Measuring similarity
  - Set-based
  - Content-based (based on type labels)
- Need “soft” matching to deal with the imperfections of the category system
  - Lexical similarity of type labels
  - Distance based on the hierarchy
  - Query expansion

Modeling types as probability distributions [Balog et al. 2011]

- Analogously to term-based representations



- Advantages
  - Sound modeling of uncertainty associated with category information
  - Category-based feedback is possible

Joint type detection and entity ranking [Sawant & Chakrabarti 2013]

- Assumes “telegraphic” queries with target type
  - woodrow wilson president university
  - dolly clone institute
  - lead singer led zeppelin band
- Type detection is integrated into the ranking
  - Multiple query interpretations are considered
- Both generative and discriminative formulations

Approach

- Each query term is either a “type hint” ($h(\vec{q}, \vec{z})$) or a “word matcher” ($s(\vec{q}, \vec{z})$)
  - Number of possible partitions is manageable ($2^{|q|}$)

losing baseball team world series 1998

Type: Major league baseball teams

instanceOf

Entity: San Diego Padres

mentionOf

Evidence snippets:

By comparison, the Padres have been to two World Series, losing in 1984 and 1998.

Generative approach

Generate query from entity

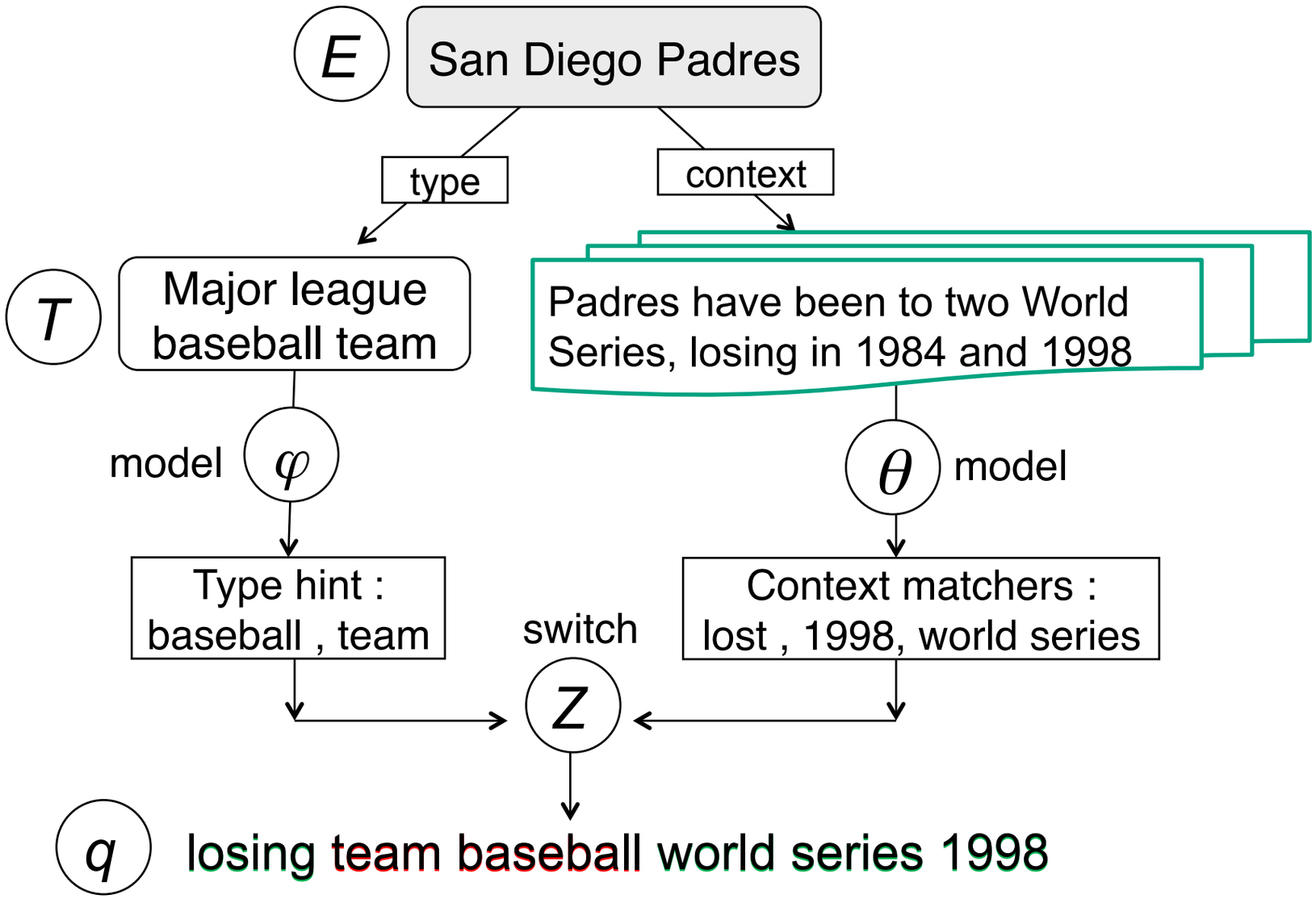


Figure taken from Sawant & Chakrabarti (2013). Learning Joint Query Interpretation and Response Ranking. In WWW '13. (see presentation)

Generative formulation

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

Entity prior

Type prior
Estimated from answer types in the past

Query switch
Probability of the interpretation

Type model
Probability of observing t in the type model

Entity model
Probability of observing t in the entity model

Discriminative approach

Separate correct and incorrect entities

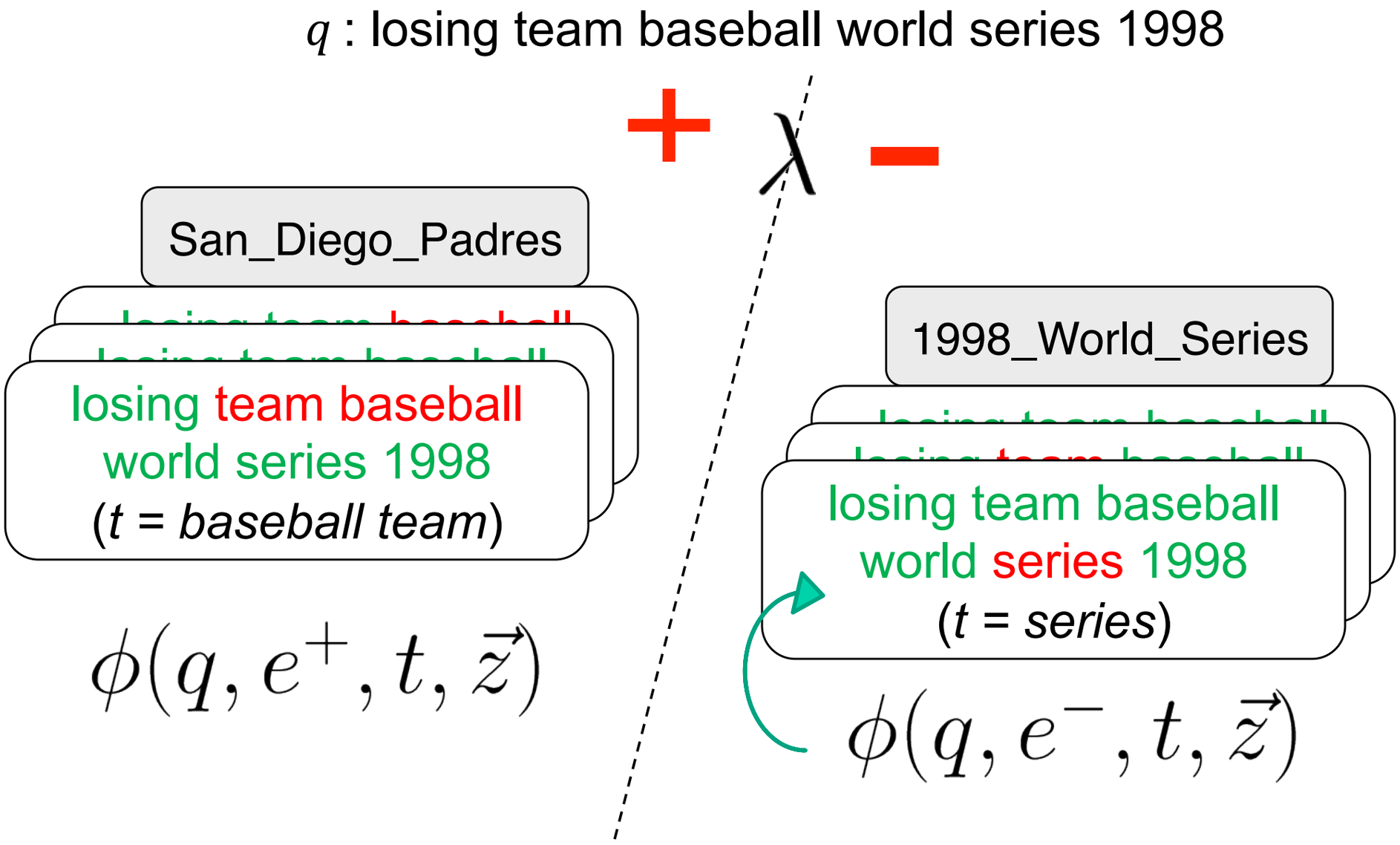


Figure taken from Sawant & Chakrabarti (2013). Learning Joint Query Interpretation and Response Ranking. In WWW '13. (see presentation)

Discriminative formulation

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

Models the type prior P(t|e)

Models the entity prior P(e)

Comparability between hint words and type

Comparability between matchers and snippets that mention e

Evaluation

- INEX Entity Ranking track
  - Entities are represented by Wikipedia articles
  - Topic definition includes target categories

Titanic (1997 film)

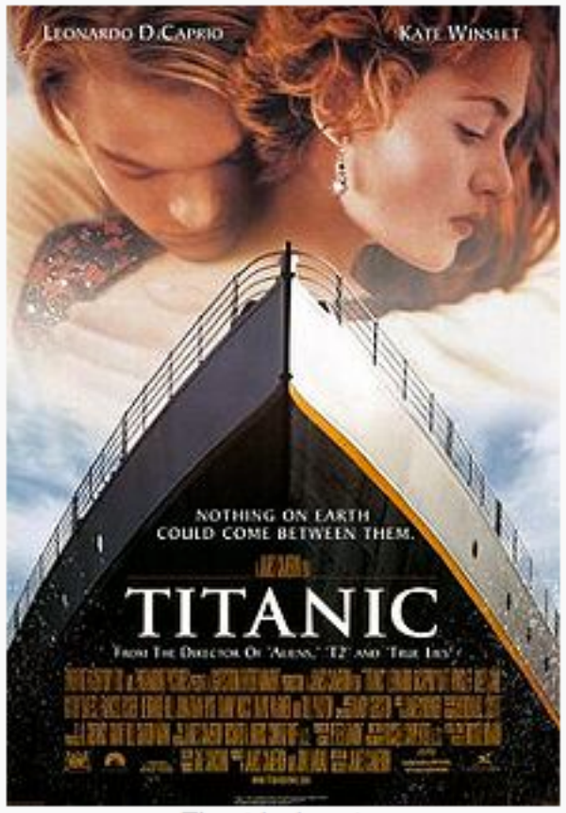
From Wikipedia, the free encyclopedia

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 Mstislav Keldysh*, which Cameron had used as a base when filming the actual wreck. A reconstruction of the *Titanic* was built at Playas 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.[3][4][5][6]

The film was originally scheduled to open on July 2, 1997, however, post-production delays pushed back its release to December 19 instead.[7] *Titanic* was an enormous critical and commercial success. It was nominated for fourteen Academy Awards, eventually winning eleven, including Best Picture and Best Director.[8] 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.[9][10] *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.[11] The film is due for theatrical re-release in 2012 after Cameron completes its conversion into 3-D.[12]

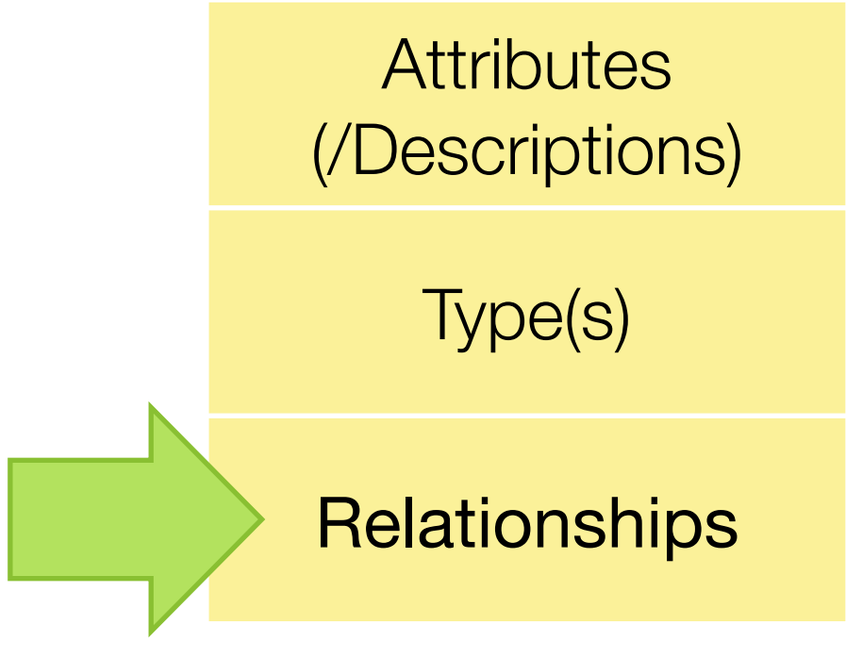
Titanic



Theatrical poster

Categories: 1997 films | American films | English-language films | American disaster films | Best Drama Pict
winners | Best Song Academy Award winners | Films directed by James Cameron | Films set in 1912 | Films
won the Best Visual Effects Academy Award | Films whose art director won the Best Art Direction Academy
Cinematography Academy Award | Films whose director won the Best Director Academy Award | Films who
editor won the Best Film Editing Academy Award | Epic films | RMS Titanic | Romantic epic films | Romantic
shot in Nova Scotia | Films shot in Vancouver | Paramount films | 20th Century Fox films | Lightstorm Enterta

Entity relationships



Related entities

Google kimi raikkonen Krisztian Balog

Web Images Maps Shopping News More Search tools

About 8,700,000 results (0.41 seconds)

Kimi Raikkonen - Lotus

3rd in Formula One World Championship - 116 points - 1 wins - 9 starts

Recent races		Place	Points	Time
Jun 30	British Grand Prix	5	10	01:33:10
Jul 7	German Grand Prix	2	18	01:41:15
Jul 28	Hungarian Grand Prix			today 8:00 AM (EST)

News for **kimi raikkonen**

Kimi Raikkonen leaves future to fate and gut instinct
The Guardian - 1 day ago
Kimi Raikkonen, favourite to replace Mark Webber at Red Bull, has said he will decide his team for next season on what feels right for him.

DECISION TIME ... **Raikkonen** insists he has no idea what will happen
The Sun - 1 day ago

Kimi Räikkönen's manager says driver still in running for Red Bull, Lotus F1 rid...
AutoWeek - 15 hours ago

Kimi Räikkönen - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Kimi_Räikkönen**
Kimi-Matias **Räikkönen** (Finnish pronunciation: [ˈkimi ˈmɑtiɑs ˈræikːønen]; born 17 October 1979) is a Finnish racing driver. After nine seasons racing in ...
Jenni Dahlman - List of largest sports contracts - List of Finns - Flying Finn

KIMI RÄIKKÖNEN Official Web Site | Lotus Formula 1 Driver
www.**kimiraikkonen**.com/
Official site features news, biography, pictures, videos, fan club and chat.

Kimi Räikkönen - Formula 1® - The Official F1® Website
www.formula1.com/teams_and_drivers/drivers/12/
Kimi Raikkonen (FIN) Lotus F1. Formula One World Championship, Rd7, Canadian. 2013. Emerges as an early championship contender after brilliantly winning ...

Kimi Räikkönen Space
kimiraikkonenspace.com/

Kimi Räikkönen

Race car driver

Kimi-Matias Räikkönen is a Finnish racing driver. After nine seasons racing in Formula One, in which he won the 2007 Formula One World Drivers' Championship, he competed in the World Rally Championship in 2010 and 2011. Wikipedia

Born: October 17, 1979 (age 33), Espoo, Finland
Height: 5' 9" (1.75 m)
Full name: Kimi-Matias Räikkönen
Spouse: Jenni Dahlman (m. 2004–2013)
Parents: Matti Räikkönen
Siblings: Rami Räikkönen

People also search for


Fernando Alonso


Sebastian Vettel


Lewis Hamilton


Felipe Massa


Mark Webber

Google tom cruise a

- tom cruise **and katie holmes**
- tom cruise **age**
- tom cruise **and cameron diaz**
- tom cruise **and nicole kidman**

Google tom cruise wives

Krisztian Balog + Share

Web Images Maps Shopping More Search tools

About 2,650,000 results (0.23 seconds)

Tom Cruise Spouse

Katie Holmes
(m. 2006–2012)

Nicole Kidman
(m. 1990–2001)

Mimi Rogers
(m. 1987–1990)

Feedback / More info

Each of **Tom Cruise's wives** has been 11 years younger than the ...
www.omg-facts.com › Celebrity Facts
Mimi Rogers was born in 1956, **Nicole Kidman** was born in 1967, and **Katie Holmes** was born in 1978. **Tom** himself was born in 1962, meaning that he was six ...

Tom Cruise

Actor

Follow

Thomas Cruise Mapother IV, widely known as Tom Cruise, is an American film actor and producer. He has been nominated for three Academy Awards and has won three Golden Globe Awards. He started his career at age 19 in the 1981 film Taps.
Wikipedia

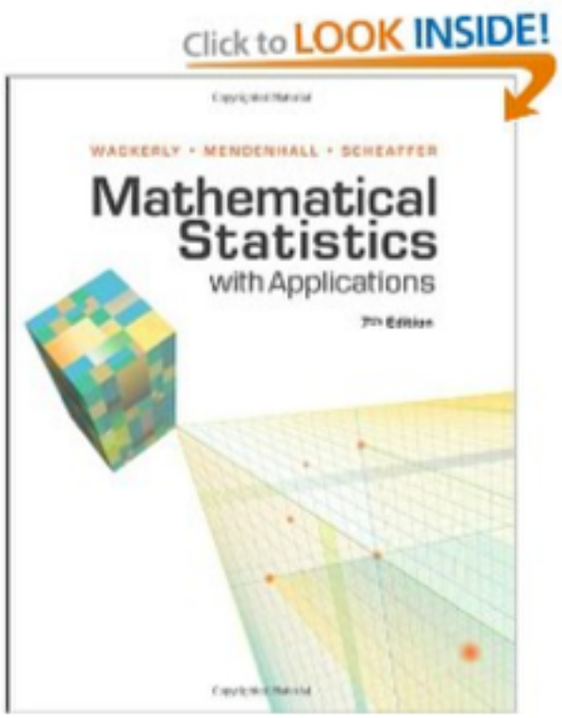
Born: July 3, 1962 (age 51), Syracuse, New York, United States

Height: 5' 7" (1.70 m)

Upcoming movies: All You Need Is Kill, Mission: Impossible 5

Spouse: Katie Holmes (m. 2006–2012), Nicole Kidman (m. 1990–2001), Mimi Rogers (m. 1987–1990)

Children: Suri Cruise, Isabella Jane Cruise, Connor Cruise



Click to open expanded view

Mathematical Statistics with Applications [Hardcover]

Dennis Wackerly (Author), William Mendenhall (Author), Richard L. Scheaffer (Author)

★★★☆☆ (29 customer reviews)

Buy New

$221.98 & **FREE Shipping**. Details

In Stock.

Ships from and sold by **Amazon.com**. Gift-wrap available.

Want it Wednesday, Feb. 26? Order within **7 hrs 3 mins** and choose **AmazonGlobal Priority Shipping** at checkout. Details

Rent

$28.00

In Stock.

Rented by **apex_media** and **Fulfilled by Amazon**.

39 new from $110.98 **93 used** from $60.00

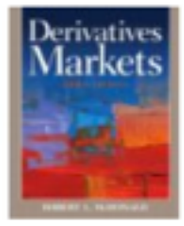
Customers Who Bought This Item Also Bought

Student Solution Manual for Mathematical ...
William J. Owen
★★☆☆☆ (9)
Paperback
$59.83 Prime

Linear Algebra and Its Applications, 4th ...
› David C. Lay
★★★½☆ (73)
Hardcover
$147.85 Prime

Contemporary Abstract Algebra
Joseph Gallian
★★★★☆ (8)
Hardcover
$140.86 Prime

Microeconomic Theory: Basic Principles and ...
› Walter Nicholson
★★★★½ (5)
Hardcover
$180.98 Prime



Derivatives Markets (3rd Edition) (Pearson ...
Robert L. McDonald
★★★★★ (1)
Hardcover
$209.69 Prime

A First Course in A... Algebra, 7th ...
› John B. Fraleigh
★★★☆☆ (24)
Hardcover
$135.10 Prime

Searching for arbitrary relations*

*given an input entity and target type

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?

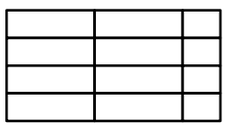
LOC Eurail

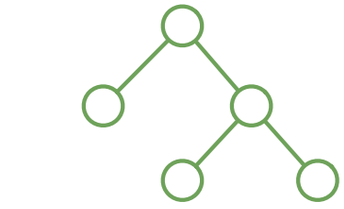
Modeling related entity finding

[Bron et al. 2010]

- Ranking entities of a given type (T) that stand in a required relation (R) with an input entity (E)
- Three-component model

$$p(e|E,T,R) \propto \underbrace{p(e|E)}_{\text{Co-occurrence model}} \cdot \underbrace{p(T|e)}_{\text{Type filtering}} \cdot \underbrace{p(R|E,e)}_{\text{Context model}}$$





Evaluation

- TREC Entity track
- 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

Wrapping up

- Entity retrieval in different flavors using generative approaches based on language modeling techniques
- Increasingly more discriminative approaches over generative ones
  - Increasing amount of components (and parameters)
  - Easier to incrementally add informative but correlated features
  - But, (massive amounts of) training data is required!

Future challenges

- It's "easy" when the "query intent" is known
  - Desired results: single entity, ranked list, set, …
  - Query type: ad-hoc, list search, related entity finding, …
- Methods specifically tailored to specific types of requests
- Understanding query intent still has a long way to go

Google

sports events in montreal june 2014

Web News Images Shopping Videos More Search tools

About 53,700,000 results (0.52 seconds)

Montreal | Event | What to do | Major | Festival | Music ...
www.tourisme-**montreal**.... Montreal Official Tourist Information Web Site
Sport ... Festival Mondial de la bière (**Montreal** Beer Fest). **June 11** to **15, 2014**. A beer-tasting ... March 21 to **June 21, 2014** ... Troupes and performers from near and far take the cultural metropolis by storm for a summer **event** not to be missed.
Attraction - Cirque du Soleil KURIOS - Activity - Nightlife

Events directory : **Sports event** (**Montréal**, Canada ...
www.bonjourquebec.com › ... › Sports event Bonjour Québec
Montréal, **Sports event** 15 result(s) ... Don't miss one of the biggest action **sports** festivals in Canada. Discover six ... **June 06, 2014 - June 08, 2014. Montréal** ...

Montreal Events - best **events in Montreal** - World Travel G...
www.worldtravelguide.net › ... › Canada › Quebec › Montreal
Make your trip to **Montreal** memorable with information on the top **events in Montreal** including everything from cultural ... **June 2014** - August 2014 (June-July.).

Montreal Upcoming **Events**, Festivals / What to do in ...
www.resto**montreal**.ca/**events**/index.php?lang=en RestoMontreal.ca
... **Events** and Festivals. RestoMontreal.ca is your guide to **events**, arts, entertainment and dining in **Montreal**. ... Televised **Sports**; Theme / Unique ..... Tribute to the great immortals with Snooksta on **June 19th, 2014 at 8pm**. DINNER-SHOW ...

May 1 - Oct 1	Montreal Food Tours	Montreal
May 18 - Sep 21	Piknic Electronik at Parc ...	Parc Jean-Drapeau
Jun 3 - Jun 28	For the month of June ...	Café St-Paul