

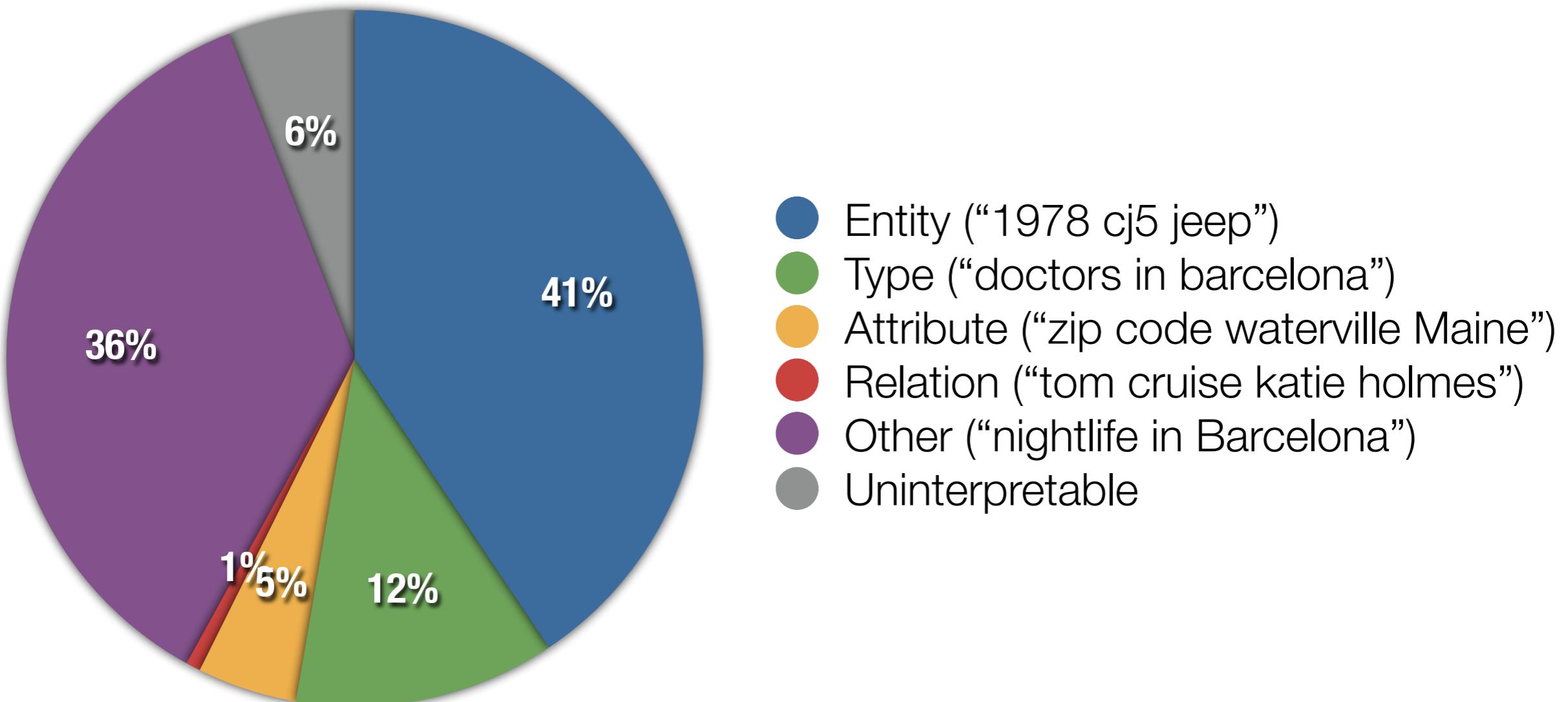
Part II

Entity Retrieval

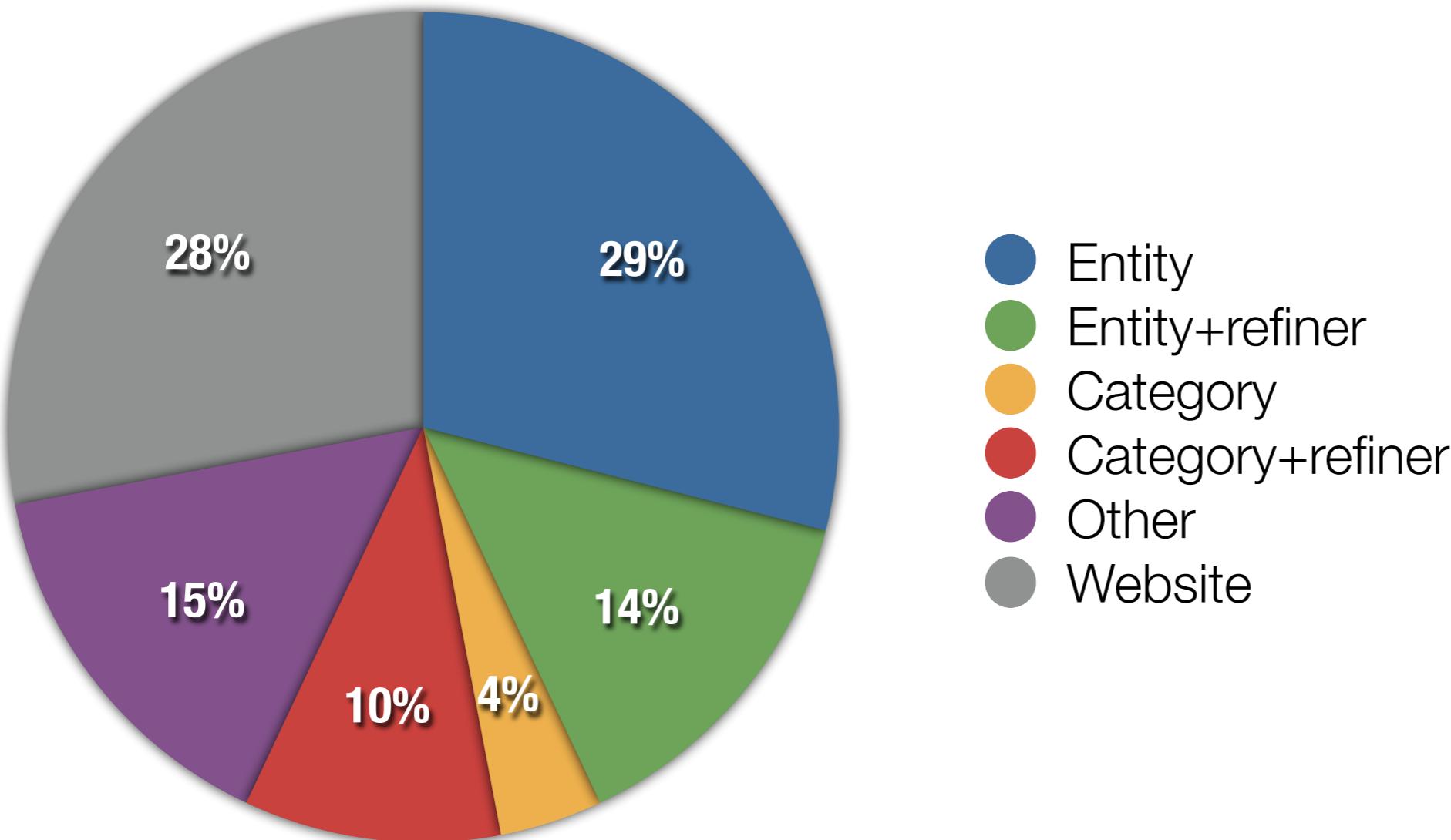
Entity retrieval

*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]



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

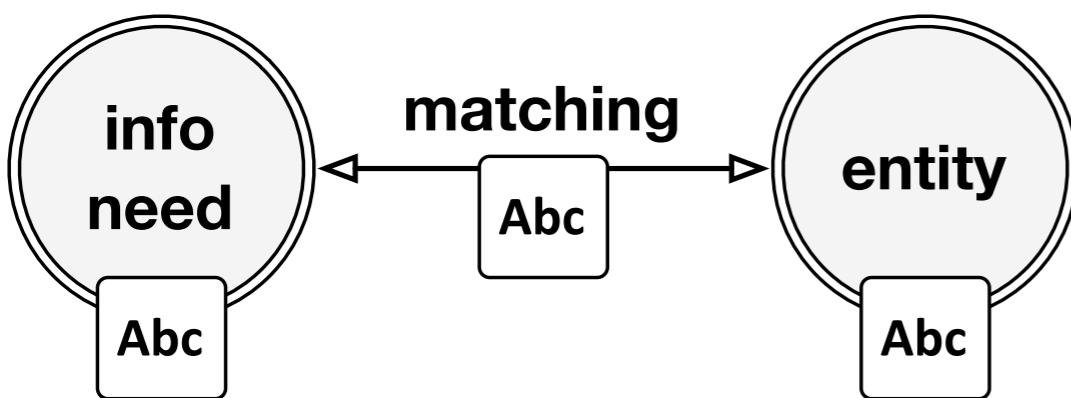


What's so special here?

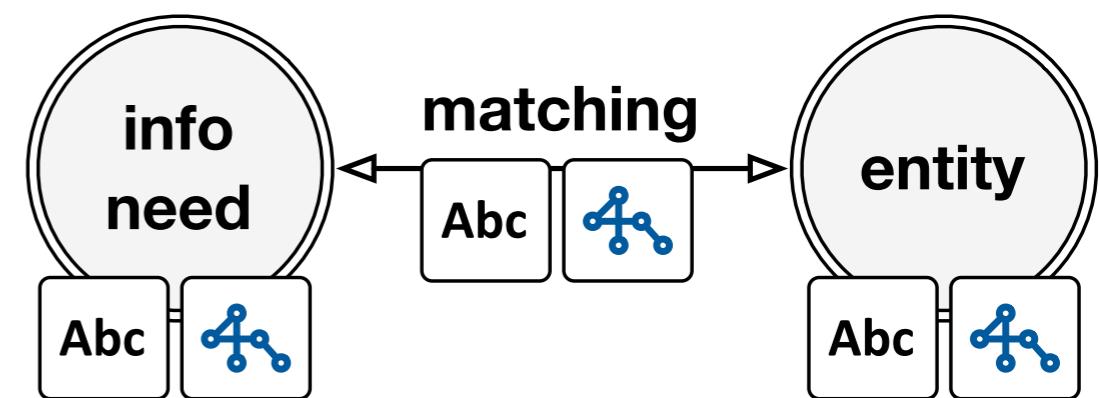
- Entities are not always directly represented
 - Recognize and disambiguate entities in text (that is, entity linking)
 - Collect and aggregate information about a given entity from multiple documents and even multiple data collections
- More structure than in document-based IR
 - Types (from some taxonomy)
 - 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/ semistructured	ranked list
adhoc object retrieval	keyword	structured	ranked list
list completion	keyword+++ (examples)	(semi)structured	ranked list
related entity finding	keyword++ (target type, relation)	unstructured & 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

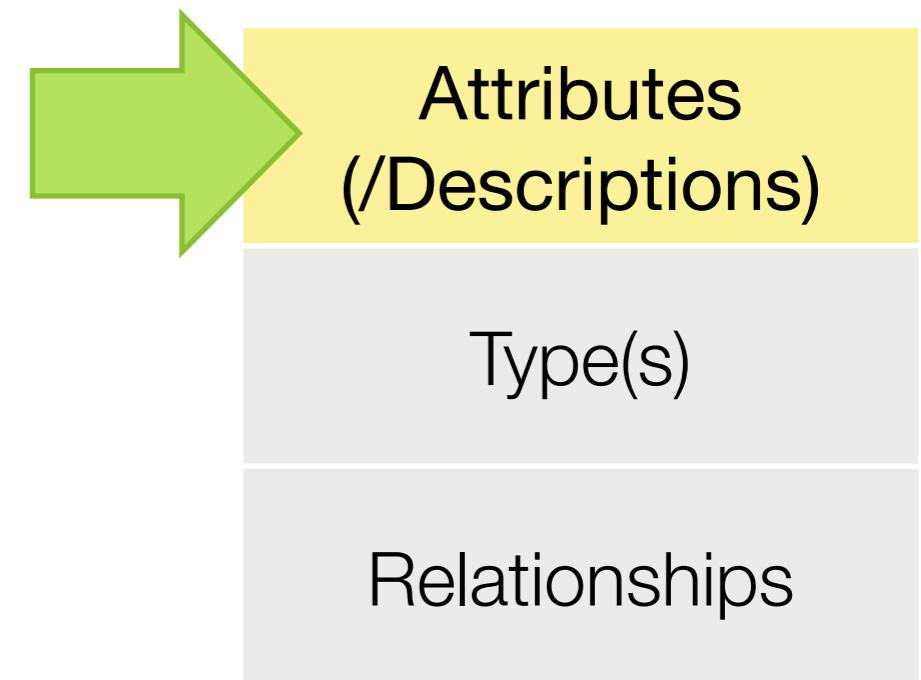
- 1.Ranking based on entity descriptions
- 2.Incorporating entity types
- 3.Entity relationships

Attributes
(/Descriptions)

Type(s)

Relationships

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

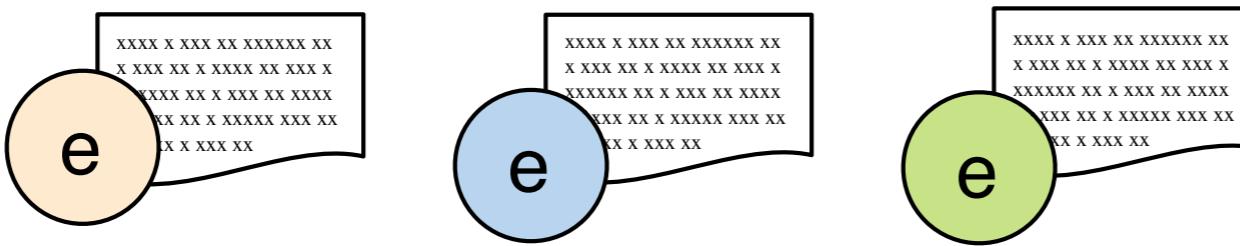
🔍 meg ryan war

🔍 Worst actor century

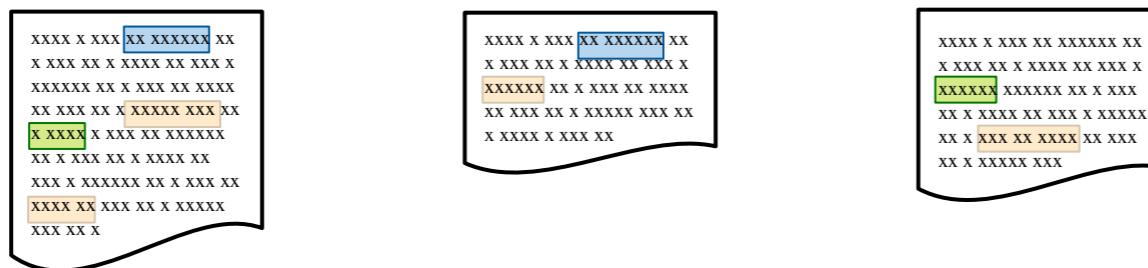
🔍 Sweden Iceland currency

Two settings

1. With ready-made entity descriptions



2. Without explicit entity representations



Ranking with ready-made entity descriptions

This is not unrealistic...

The image displays a composite screenshot of four overlapping web pages:

- Wikipedia** (leftmost): Shows the main page with the title "WIKIPEDIA The Free Encyclopedia".
- IMDb**: Shows a search bar with "Find Movies, TV shows, Celebrities and more..." and a search icon.
- LinkedIn**: Shows the user profile of "Krisztian Balog" with sections for Home, Profile, Contacts, Groups, Jobs, Inbox, Companies, News, and More. It also shows a search bar and a sidebar for "Activity".
- Amazon** (bottom right): Shows the product page for "Modern Information Retrieval: The Concepts and Technology behind Search (2nd Edition) (ACM Press Books) [Paperback]". The price is listed as \$62.49, and there is a "Buy New" button. The page also features a "FREE TWO-DAY SHIPPING FOR COLLEGE STUDENTS" offer and a "SELL BOOKS" section.

The overlapping nature of the windows illustrates how multiple web applications can be used simultaneously by a single user.

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 : $P(d|q)$

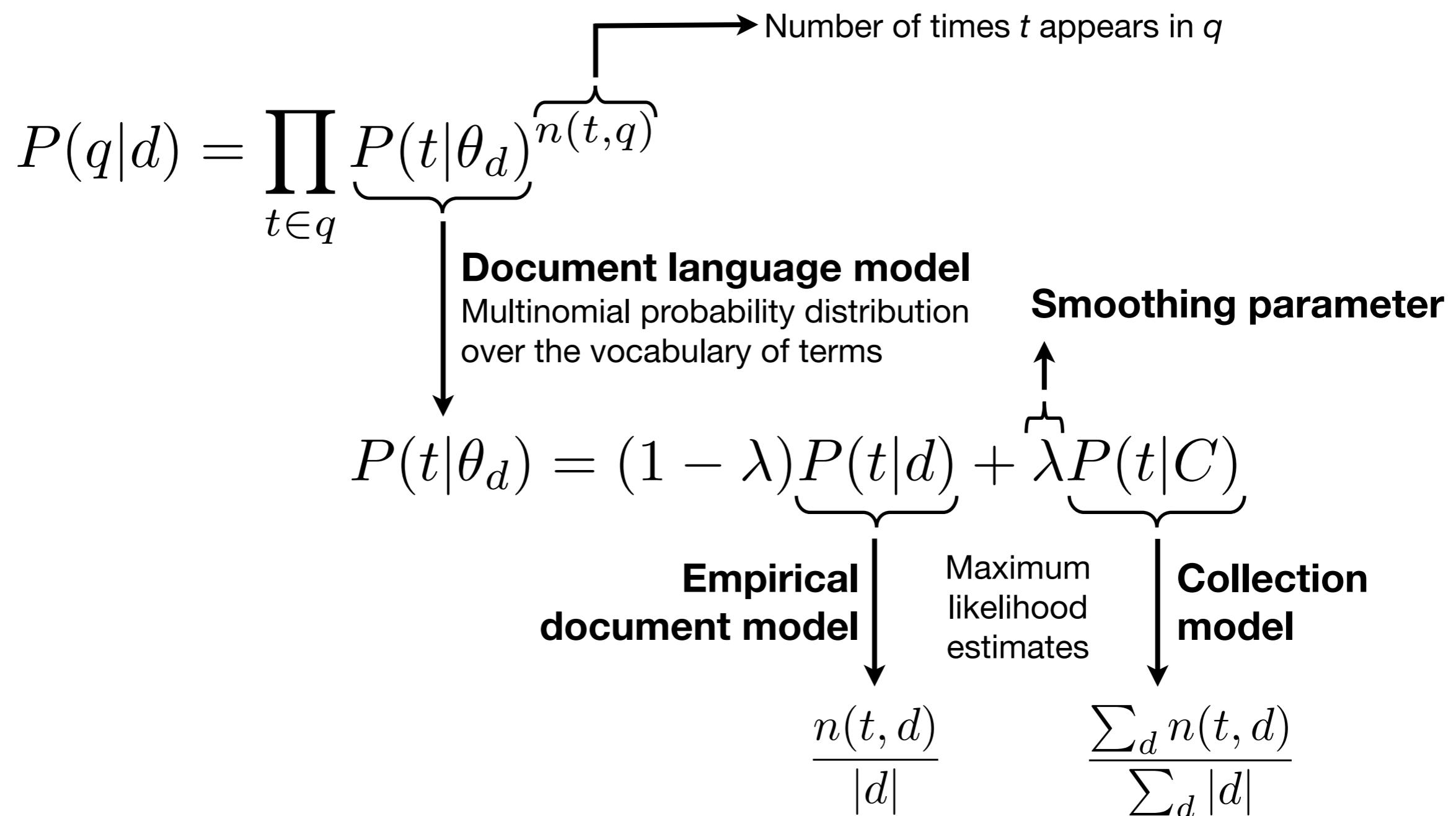
$$P(d|q) = \frac{P(q|d)P(d)}{P(q)} \propto P(q|d)P(d)$$

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

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

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



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]

Audi A4



Manufacturer Audi

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:Audi](#)

[dbpedia:Compact_executive_car](#)

[freebase:Audi_A4](#)

[dbpedia:Audi_A5](#)

[dbpedia:Cadillac_BLS](#)

dbpedia-owl:manufacturer

dbpedia-owl:class

owl:sameAs

is [dbpedia-owl:predecessor](#) of

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

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

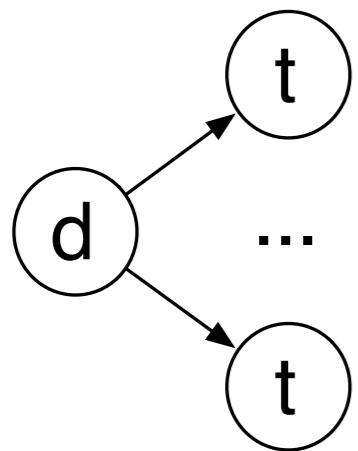
Field weights

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

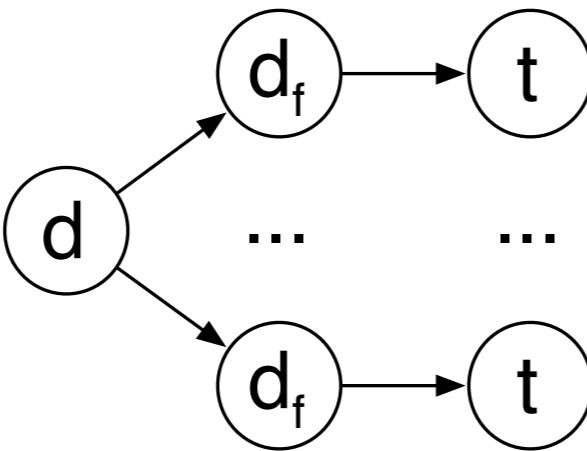
Field language model

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

Comparison of models



**Unstructured
document model**



**Fielded
document model**

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]

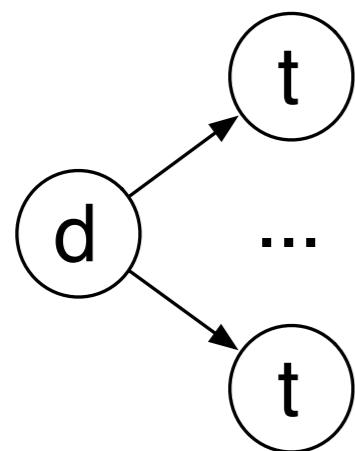
- Organize fields into a 2-level hierarchy
 - Field types (4) on the top level
 - Individual fields of that type on the bottom level
- Estimate field weights
 - Using training data for field types
 - Using heuristics for bottom-level types

Two-level hierarchy

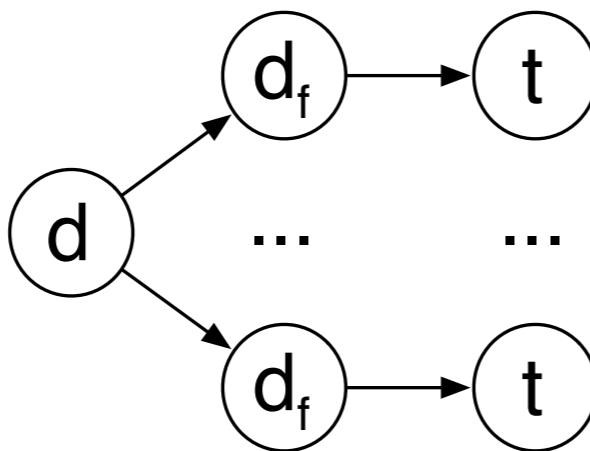
[Neumayer et al. 2012]

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

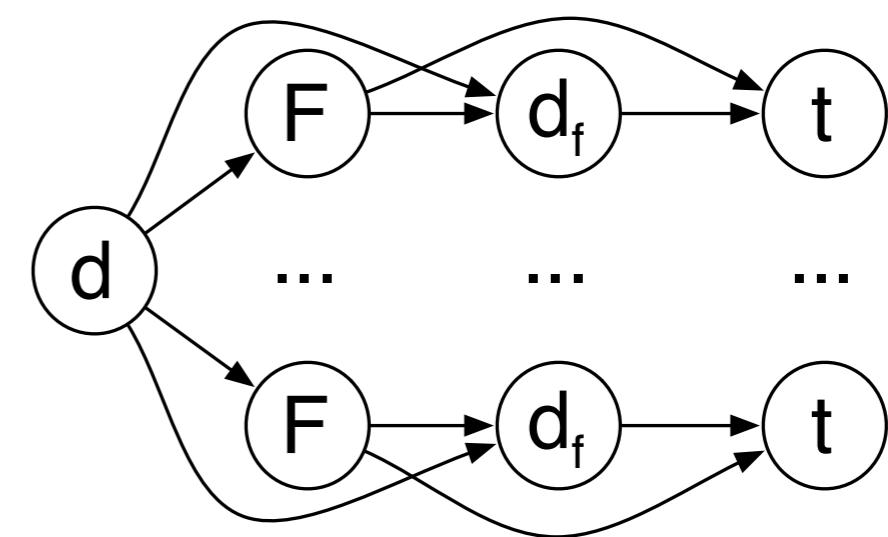
Comparison of models



**Unstructured
document model**



**Fielded
document model**



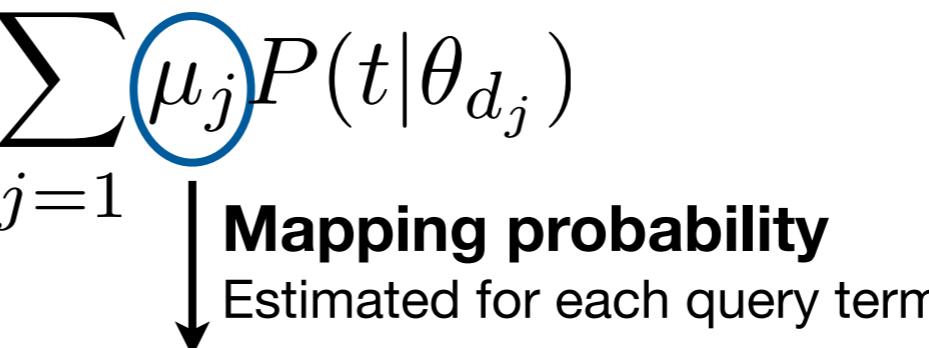
**Hierarchical
document model**

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

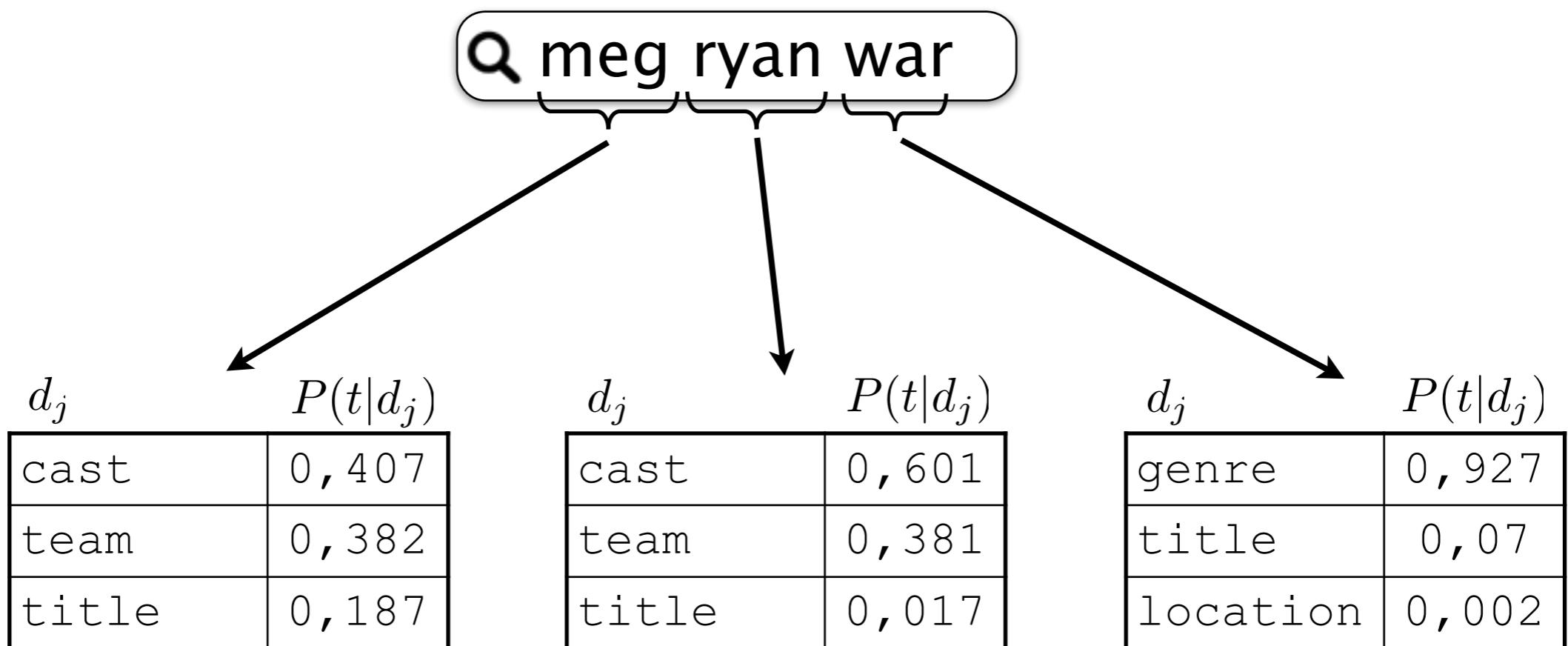
$$P(t|C_j) = \frac{\sum_d n(t, d_j)}{\sum_d |d_j|}$$

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

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

Example



Ranking without explicit entity representations

Scenario

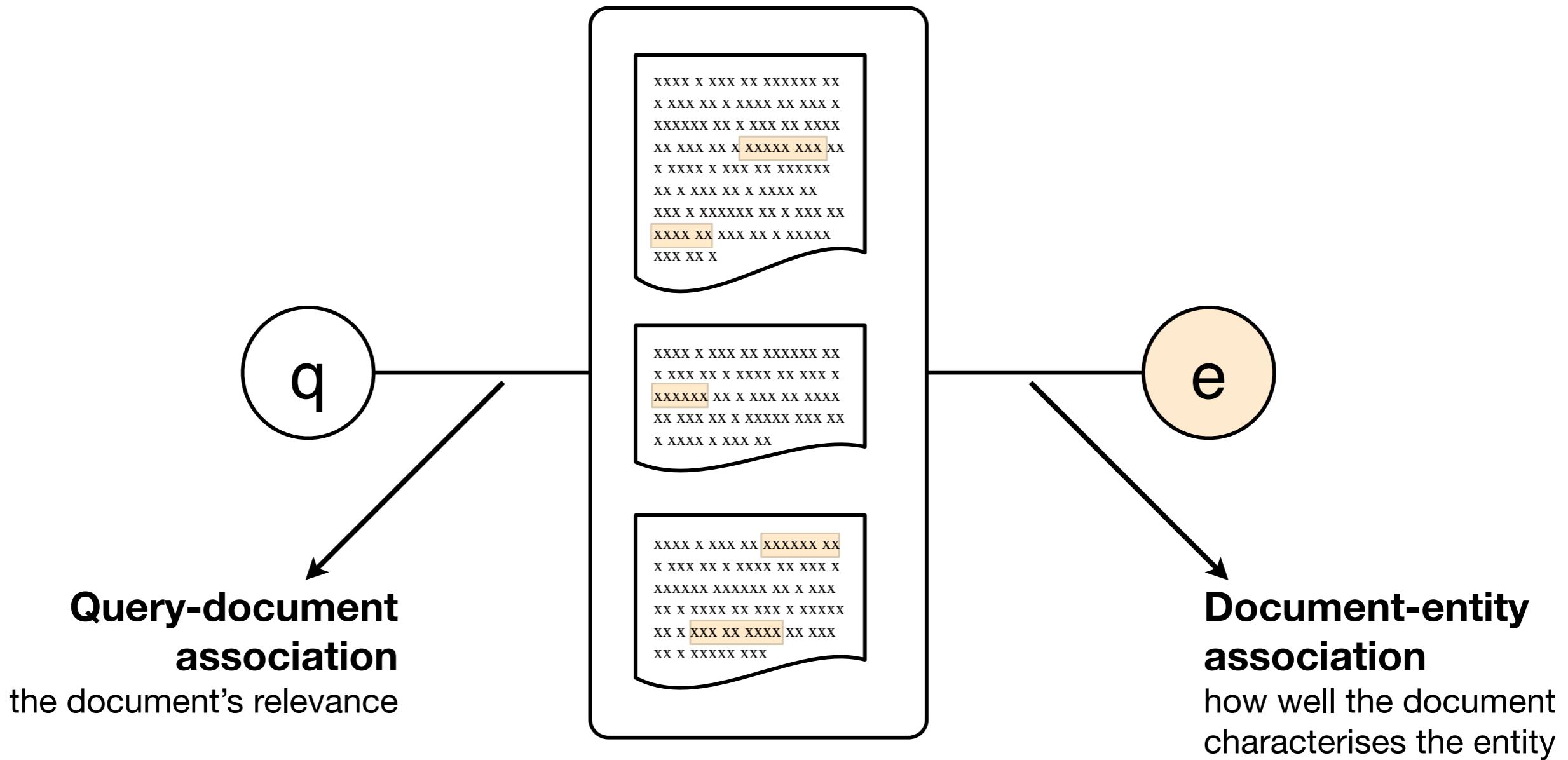
- Entity descriptions are not readily available
- Entity occurrences are annotated
 - manually
 - automatically (~entity linking)

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

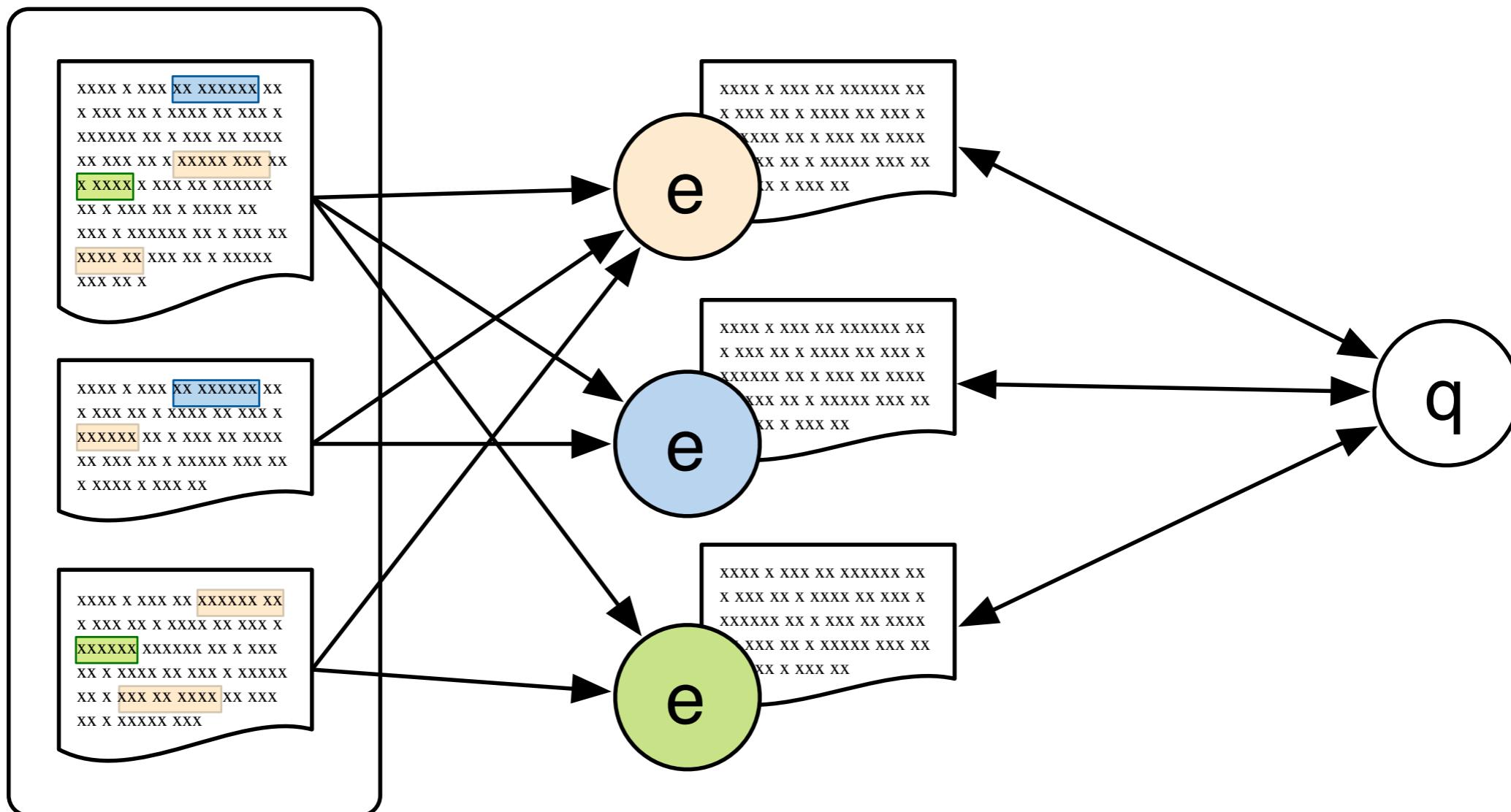
Use documents to go from queries to entities



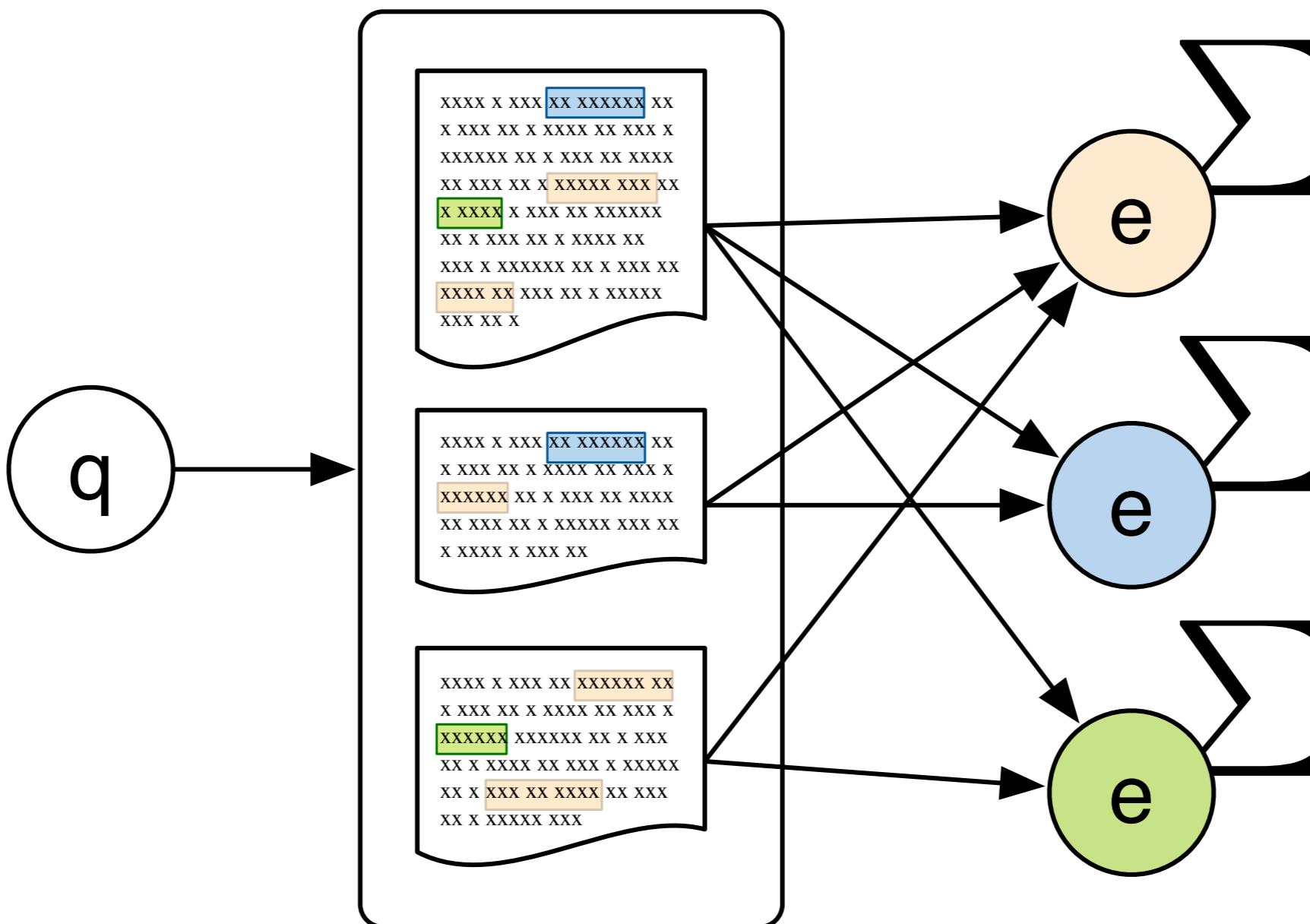
Two principal approaches

- **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

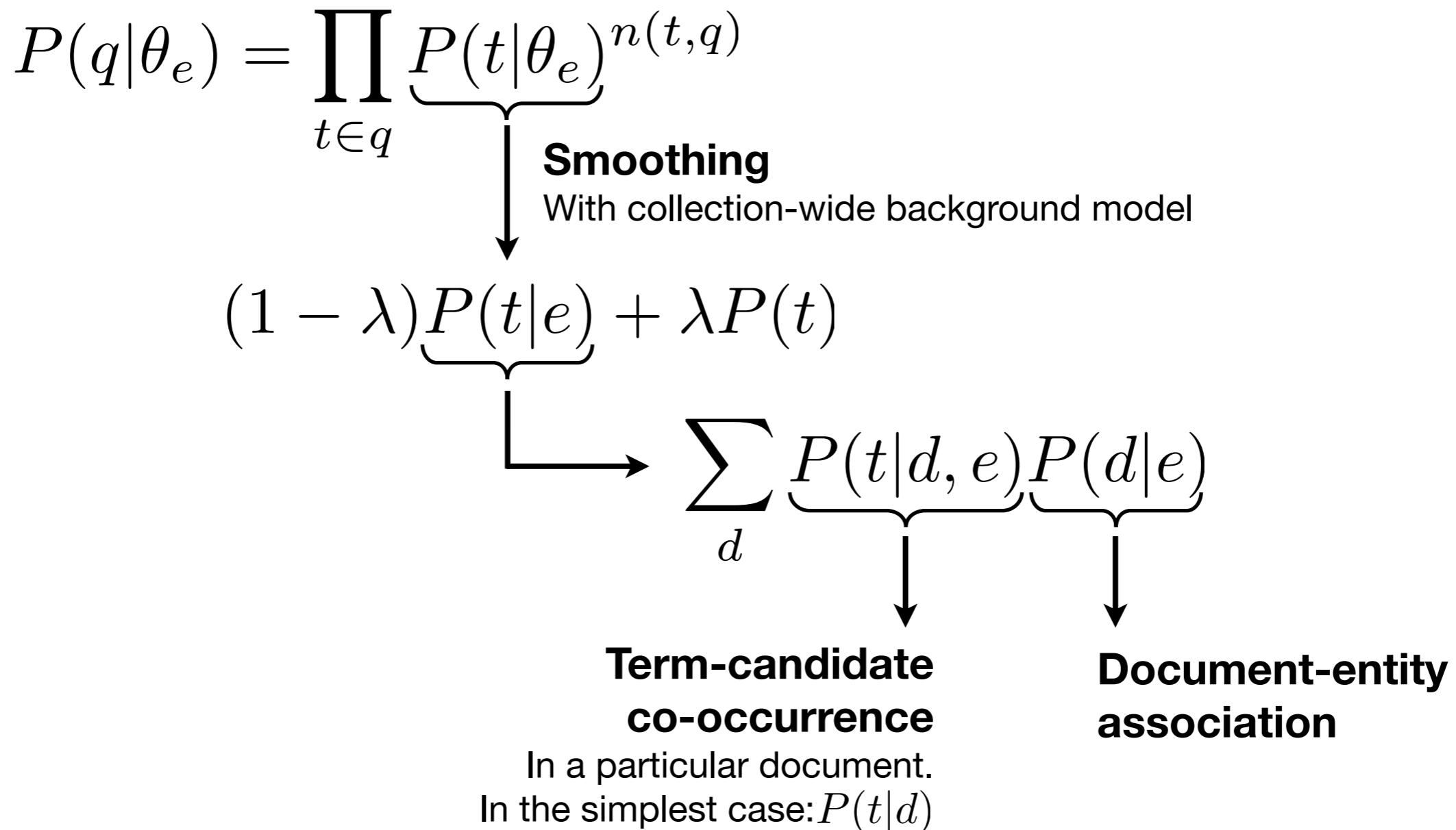
Generative probabilistic models

- 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]



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

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

$P(t|\theta_d)$

The diagram illustrates the decomposition of the probability $P(q|e)$ into two components. The first component, $\sum_d P(q|d, e) P(d|e)$, is labeled "Document relevance" and is described as "How well document d supports the claim that e is relevant to q ". The second component, $\prod_{t \in q} P(t|d, e)^{n(t,q)}$, is labeled "Document-entity association". A bracket under this second component is labeled "Simplifying assumption" with the text "(t and e are conditionally independent given d)". Arrows point from the labels to their respective parts in the equation.

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]

$$P(t|d, e) = \frac{1}{Z} \sum_{i=1}^N \underbrace{\delta_d(i, t)}_{\text{Normalizing constant}} \underbrace{k(t, e)}_{\text{Indicator function}}$$

Proximity-based kernel

- constant function
- triangle kernel
- Gaussian kernel
- step function

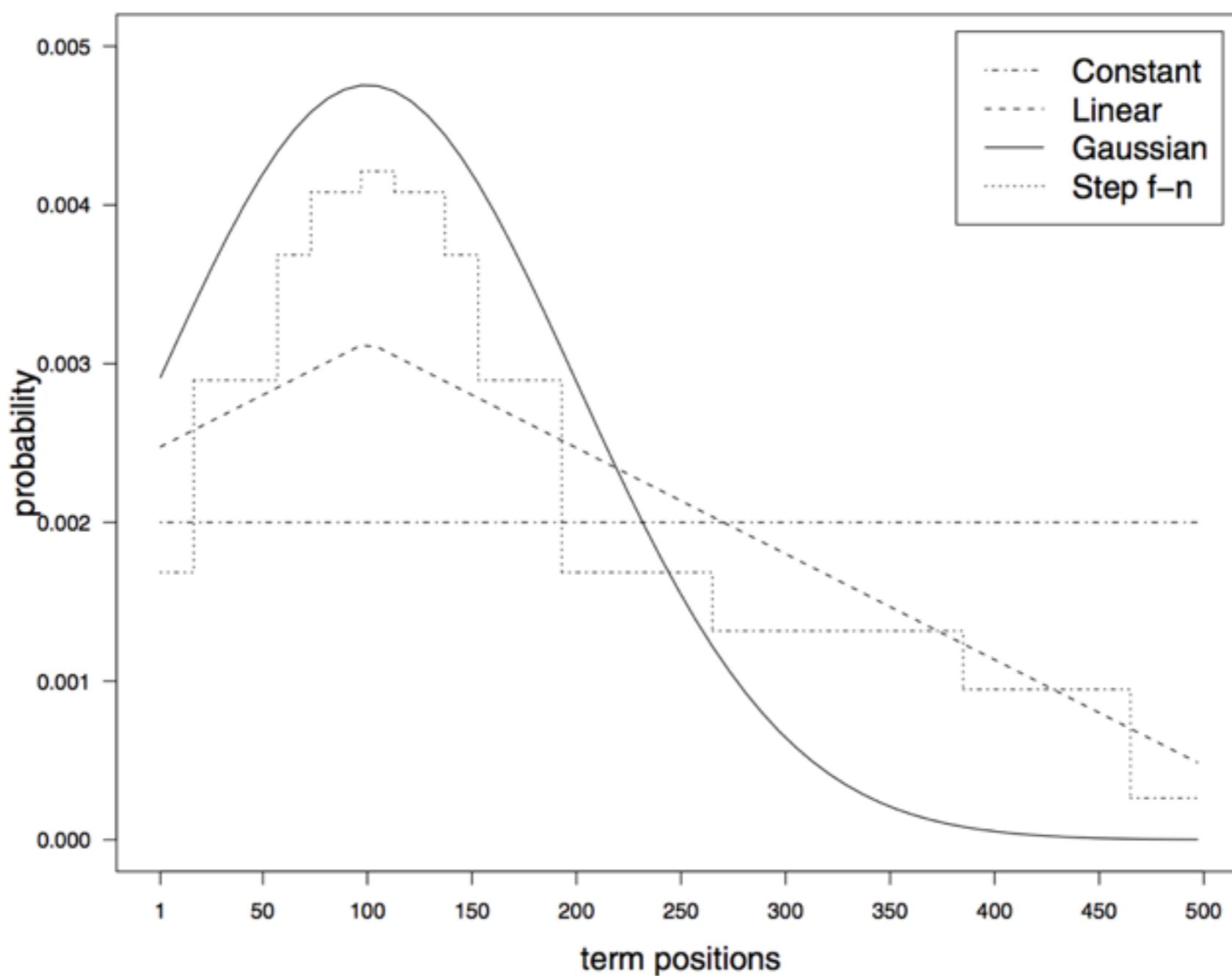


Figure taken from D. Petkova and W.B. Croft. Proximity-based document representation for named entity retrieval. CIKM'07.

Many possibilities in terms of modeling

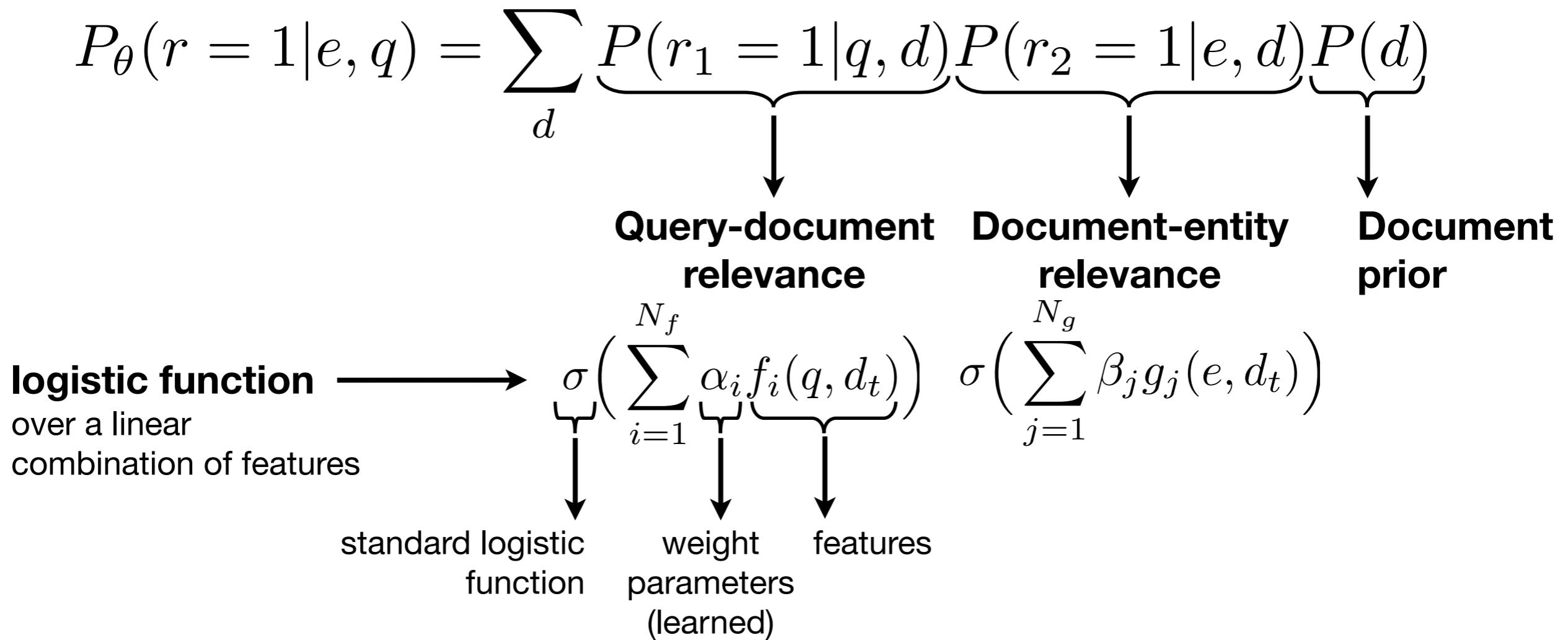
- Generative probabilistic models
- Discriminative probabilistic models
- Voting models
- Graph-based models

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]



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

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

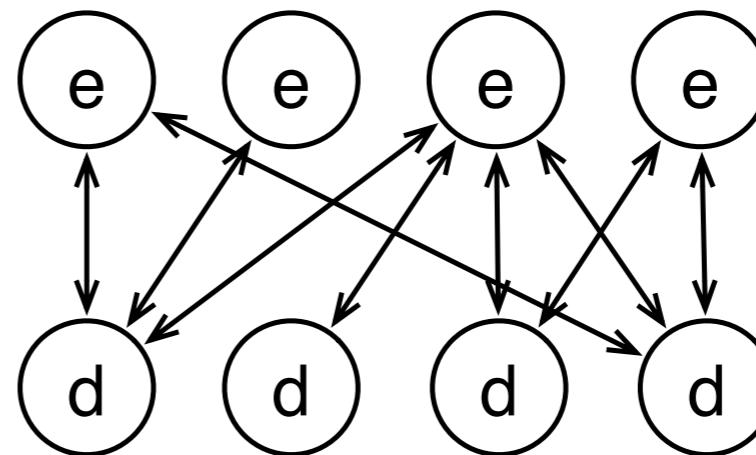
Graph-based models

[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]

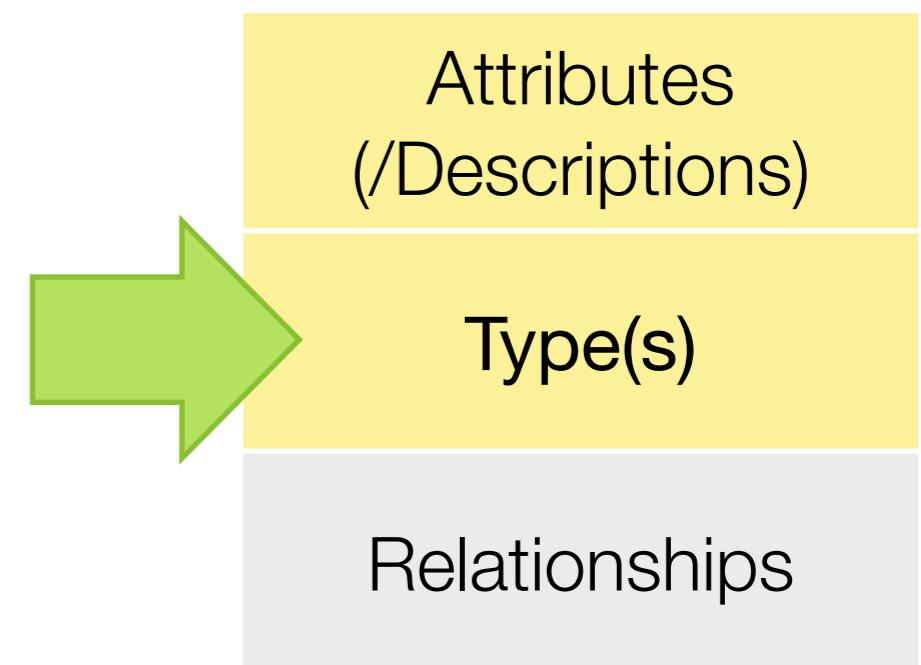


$$P_i(d) = \lambda P_J(d) + (1 - \lambda) \sum_{e \rightarrow d} P(d|e) P_{i-1}(e),$$

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

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

Incorporating entity types



For a handful of types grouping results by entity type is a viable solution

LinkedIn search results for "best".

4,183,729 results for best

SEARCH

Advanced >

All

- People
- Jobs
- Companies
- Groups
- Updates
- Inbox

Relationship

- All
- 1st Connections (53)
- 2nd Connections (9025)
- Group Members (3449)
- 3rd + Everyone Else (4171974)

Location

- All
- United States (2095060)
- United Kingdom (421552)
- India (307776)
- Canada (210803)
- Greater New York ... (199325)
- + Add

Current Company

- All
- Best Buy (27274)
- IBM (12221)
- Microsoft (9610)
- Hewlett-Packard (8411)

Christoph Best 2nd
Computational Scientist at Google
London, United Kingdom · Information Technology and Services
» 1 shared connection · Similar

Connect

Angelina Best 2nd
Enterprise Sales Manager at Microsoft
Amsterdam Area, Netherlands · Information Technology and Services
» 1 shared connection · Similar

Connect

Clive Best 2nd
Director OSVISION
Varese Area, Italy · Internet
» 1 shared connection · Similar

Connect

Companies for best

- Best Advisors Network**
Accounting · 1-10 employees
- mCentric**
Telecommunications · 11-50 employees
- Event Industry Awards**
Events Services · 1-10 employees

Hubert Best 2nd
Owner, ENN Advokatbyrå
Stockholm, Sweden · Law Practice
» 1 shared connection · Similar

Connect

Eric de Best 2nd
Owner, cockpits.nl
The Hague Area, Netherlands · Arts and Crafts
» 2 shared connections · Similar

Connect

Premium Search
Find the right people in half the time.

Premium Search Tools

- Premium filters
- Automatic search alerts
- Full profile access

Upgrade

Ads

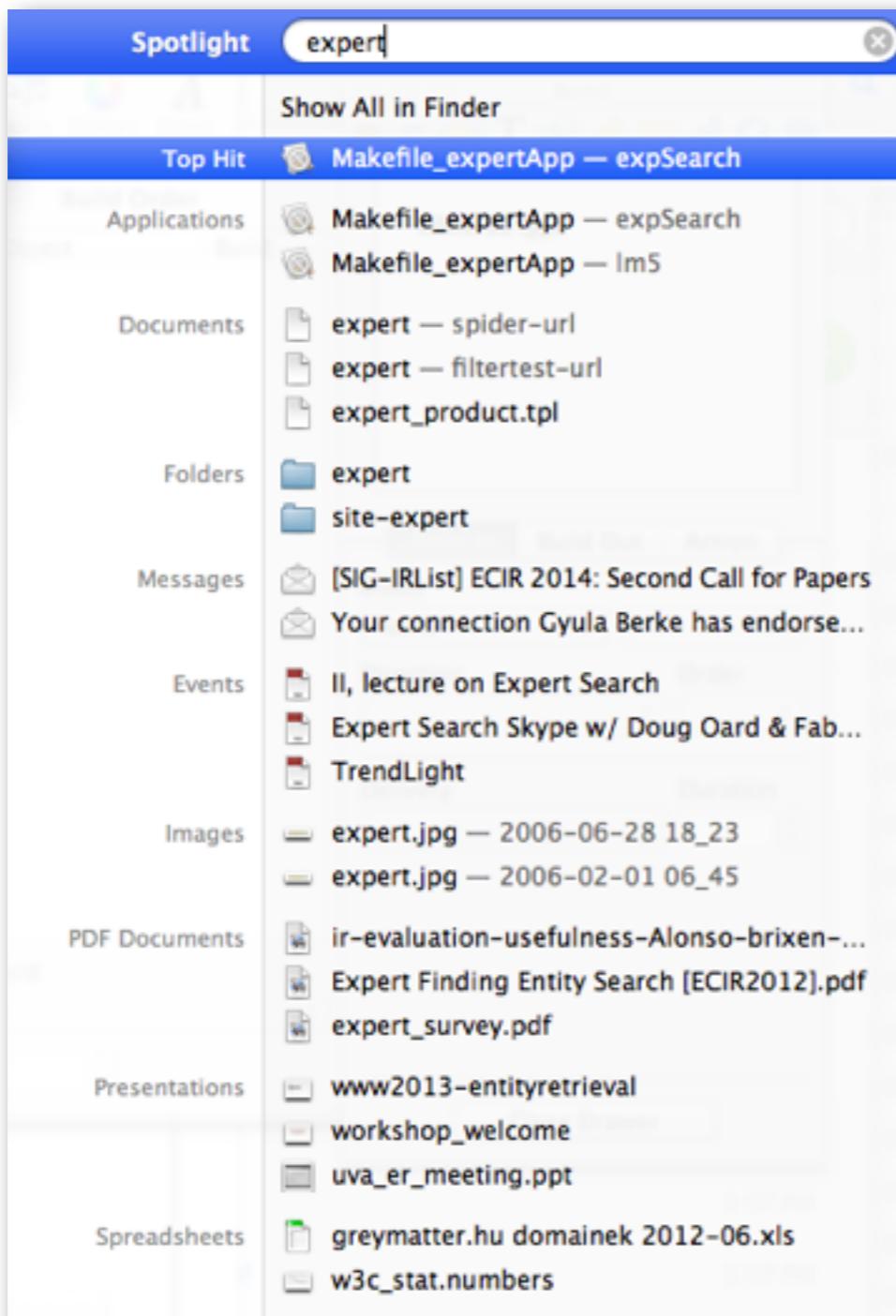
Organize Projects better!
Norway's leading project tool. Super simple! Try for free until August!

Executive MBA in London
Network with professionals from 135+ countries. Join Hult in Sept 2013.

Google AdWords

Administrative Information Management Advisor

For a handful of types grouping results by entity type is a viable solution



But what about very many types? which are typically hierarchically organized

The image shows a composite screenshot of two web pages illustrating hierarchical organization.

Left Side (Amazon.com):

- Header:** Amazon logo, "Join Prime", "Krisztian's Amazon.com", "Today's Deals", "Gift Cards", "Sell", "Help".
- Search Bar:** "Shop by Department", "Search All".
- Top Right:** "FREE TWO-DAY SHIPPING FOR COLLEGE STUDENTS", "Hello, Krisztian", "Your Account", "Join Prime", "Cart (0)", "Wish List".
- Left Sidebar:**
 - EARTH'S BIGGEST SELECTION**
 - Unlimited Instant Videos**: Prime Instant Video, Learn More About Amazon Prime, Amazon Instant Video Store, Your Video Library, Watch Anywhere.
 - MP3s & Cloud Player**: MP3 Music Store, Music on Kindle Fire, Cloud Player for PC, Cloud Player for Web, Cloud Player for Android, Cloud Player for iOS, Cloud Player for Home.
 - Amazon Cloud Drive**: Your Cloud Drive, Get the Desktop App, Cloud Drive Photos for Android, Cloud Drive Photos for iPhone, Learn More About Cloud Drive.
 - Kindle**: Kindle, Kindle Paperwhite, Kindle Paperwhite 3G, Kindle E-reader Accessories, Kindle Books, Newsstand.
- Right Content Area:** A detailed list of categories under "Books & Media".

Right Side (Wikipedia: Category:Main topic classifications):

- Header:** "Category:Main topic classifications", "From Wikipedia, the free encyclopedia".
- Top Right:** "Create account", "Log in".
- Content:**
 - This is a list of Wikipedia's major topic classifications. These are used throughout Wikipedia to organize the presentation of links to articles on its various reference systems, including Wikipedia's lists, portals, and categories.
 - An alternative top-level category to begin navigation of articles, based on a smaller number of initial thematic classifications, is [Category:Fundamental categories](#).
 - The top level of Wikipedia's overall [category system](#), including both articles and other project pages, is [Category:Contents](#).
- Subcategories:** This category has the following 25 subcategories, out of 25 total.
- Alphabetical Index:** A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T.
- Media:** Wikimedia Commons has media related to: [Topics](#).
- Footnote:** Categories: Articles, Hidden categories: Commons category with local link same as on Wikidata | Container categories.

Challenges

- Users are not familiar with the type system
 - (Often) user input is to be treated as a hint, not as a strict filter
- Type system is imperfect
 - Inconsistencies
 - Missing assignments
 - Granularity issues
 - Entities labeled with too general or too specific types
- In general, categorizing things can be hard
 - E.g. is King Arthur “British royalty”, “fictional character”, or “military person”?

Two settings

- Target type(s) are provided by the user
 - keyword++ query
- Target types need to be automatically identified
 - keyword query

Target type(s) are provided faceted search, form fill-in, etc.

The image consists of three separate screenshots arranged vertically. The top screenshot shows a vertical sidebar menu from the eBay website, listing categories like Fashion, Parts & accessories, Electronics, Collectibles & art, Home & garden, Women's Clothing, Jewelry & watches, and Daily deals. The middle screenshot shows a Mac OS X desktop search interface with a dropdown menu open, showing various file types such as Any, Application, Document, Executable, Folder, Image, Movie, Music, PDF, Presentation, Text, and Other. The bottom screenshot shows a vertical list of product categories on a website, including Departments (Grocery & Gourmet Food, Energy Drinks), Clothing & Accessories (Men's Keyrings & Keychains, Novelty T-Shirts, Novelty & Special Use Clothing, Men's Fashion Hoodies & Sweatshirts), Automotive (Racing Apparel, Motorcycle Protective Coats & Vests, Decals, Motorcycle & ATV Helmets, Motorcycle & ATV Graphics, Towing Winches, Key Chains), Tools & Home Improvement (Wall Stickers & Murals, Diversion Safes), Sports & Outdoors (Sports Fan Clothing, + See more...), and Computers & Accessories (USB Flash Drives, + See All 33 Departments).

Departments
Grocery & Gourmet Food
Energy Drinks

Clothing & Accessories
Men's Keyrings & Keychains
Novelty T-Shirts
Novelty & Special Use Clothing
Men's Fashion Hoodies & Sweatshirts

Automotive
Racing Apparel
Motorcycle Protective Coats & Vests
Decals
Motorcycle & ATV Helmets
Motorcycle & ATV Graphics
Towing Winches
Key Chains

Tools & Home Improvement
Wall Stickers & Murals
Diversion Safes

Sports & Outdoors
Sports Fan Clothing
+ See more...

Computers & Accessories
USB Flash Drives
+ See All 33 Departments

INEX Entity Ranking track

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



Movies with eight or more Academy Awards
best picture oscar british films american films

Titanic (1997 film)

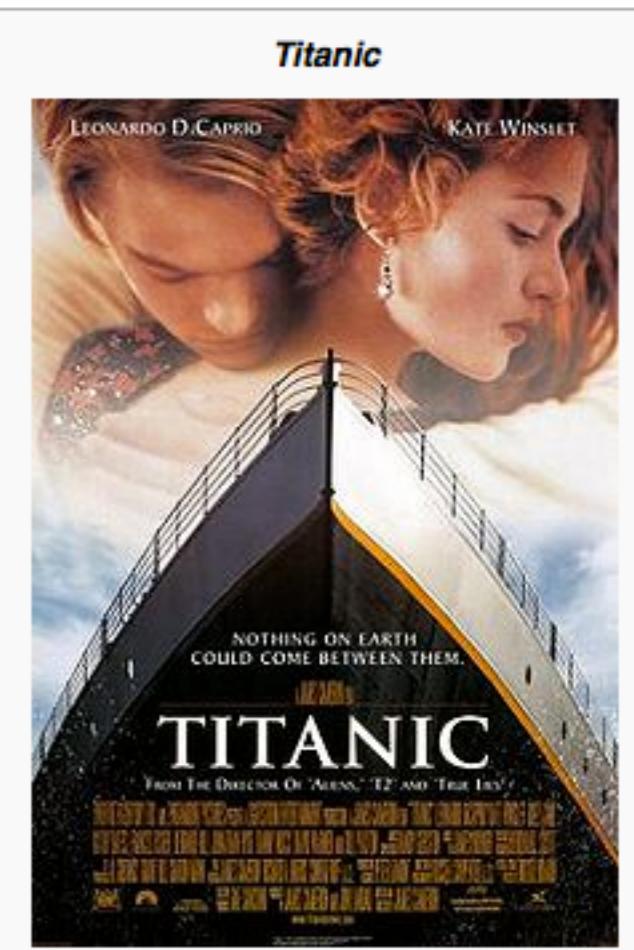


From Wikipedia, the free encyclopedia

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]



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
 - Types added as a separate metadata field

Modeling terms and categories

[Balog et al. 2011]

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

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

Term-based representation

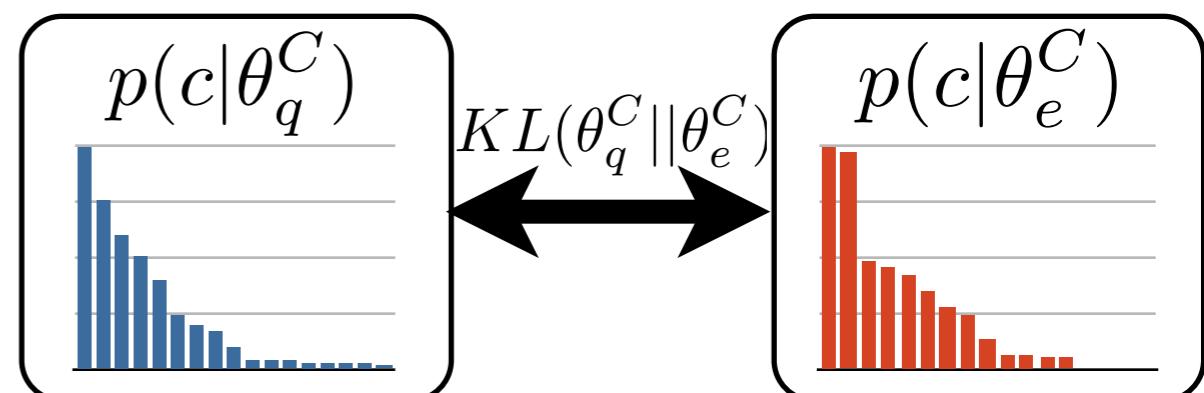
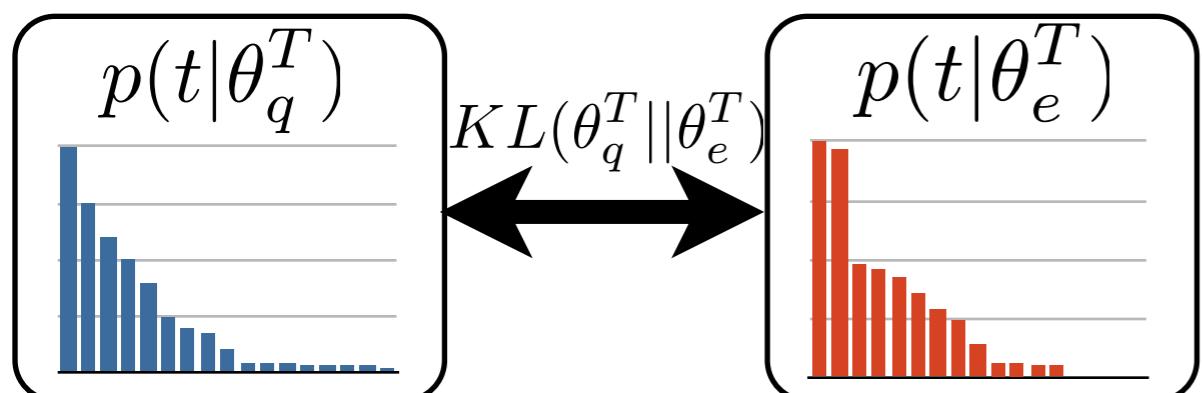
Query model

Entity model

Category-based representation

Query model

Entity model



Advantages

- Transparent combination of term-based and category-based information
- Sound modeling of uncertainty associated with category information
- Category-based feedback is possible (analogously to the term-based case)

Expanding target types

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

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

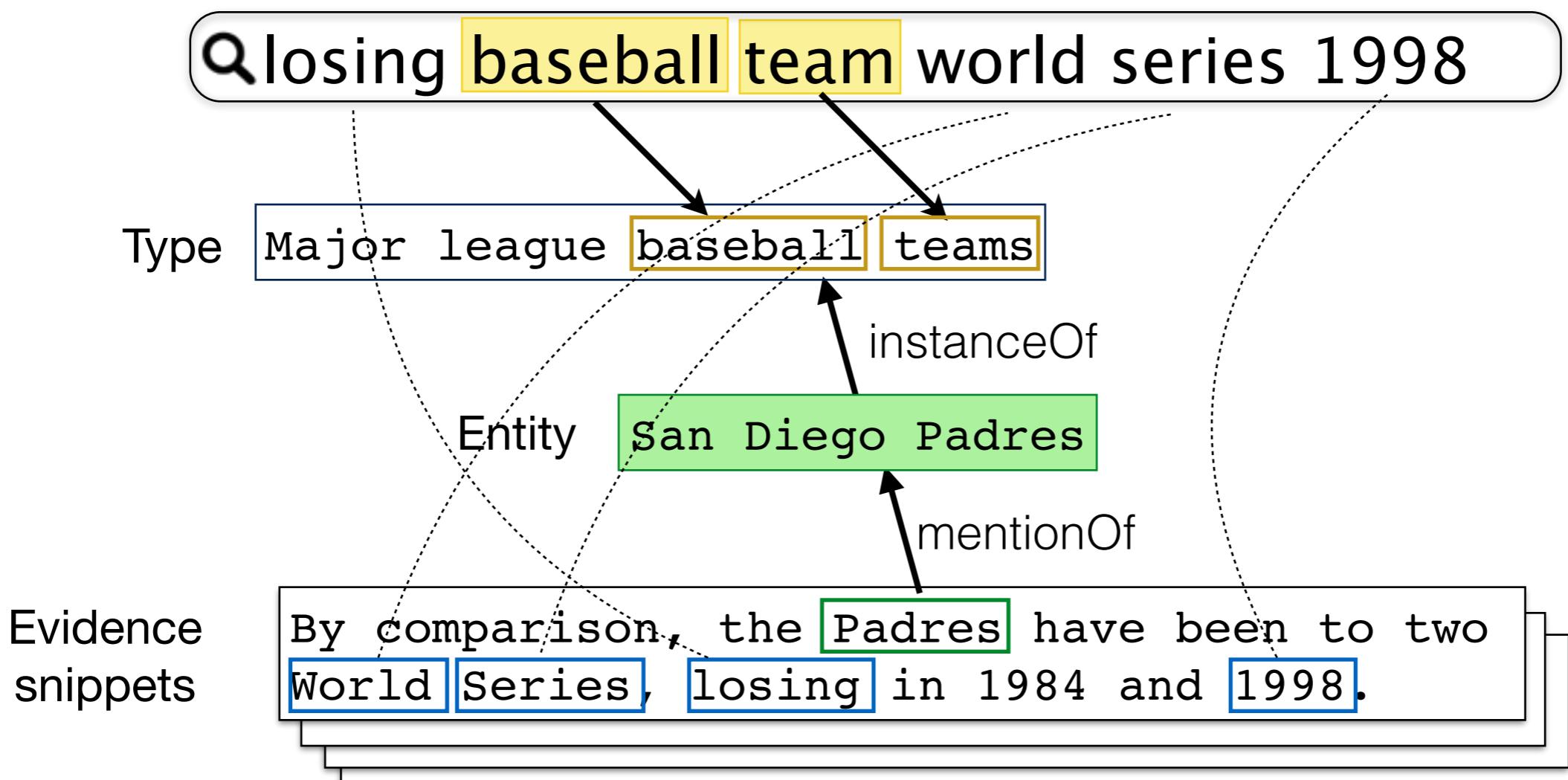
- Types of top ranked entities [**Vallet & Zaragoza 2008**]
- Types can be ranked much like entities [**Balog & Neumayer 2012**]
 - Direct term-based vs. indirect entity-based representations (“Model 1 vs. Model 2”)
 - Hierarchical case is difficult

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 ($2^{|q|}$)



Generative approach

Generate query from entity

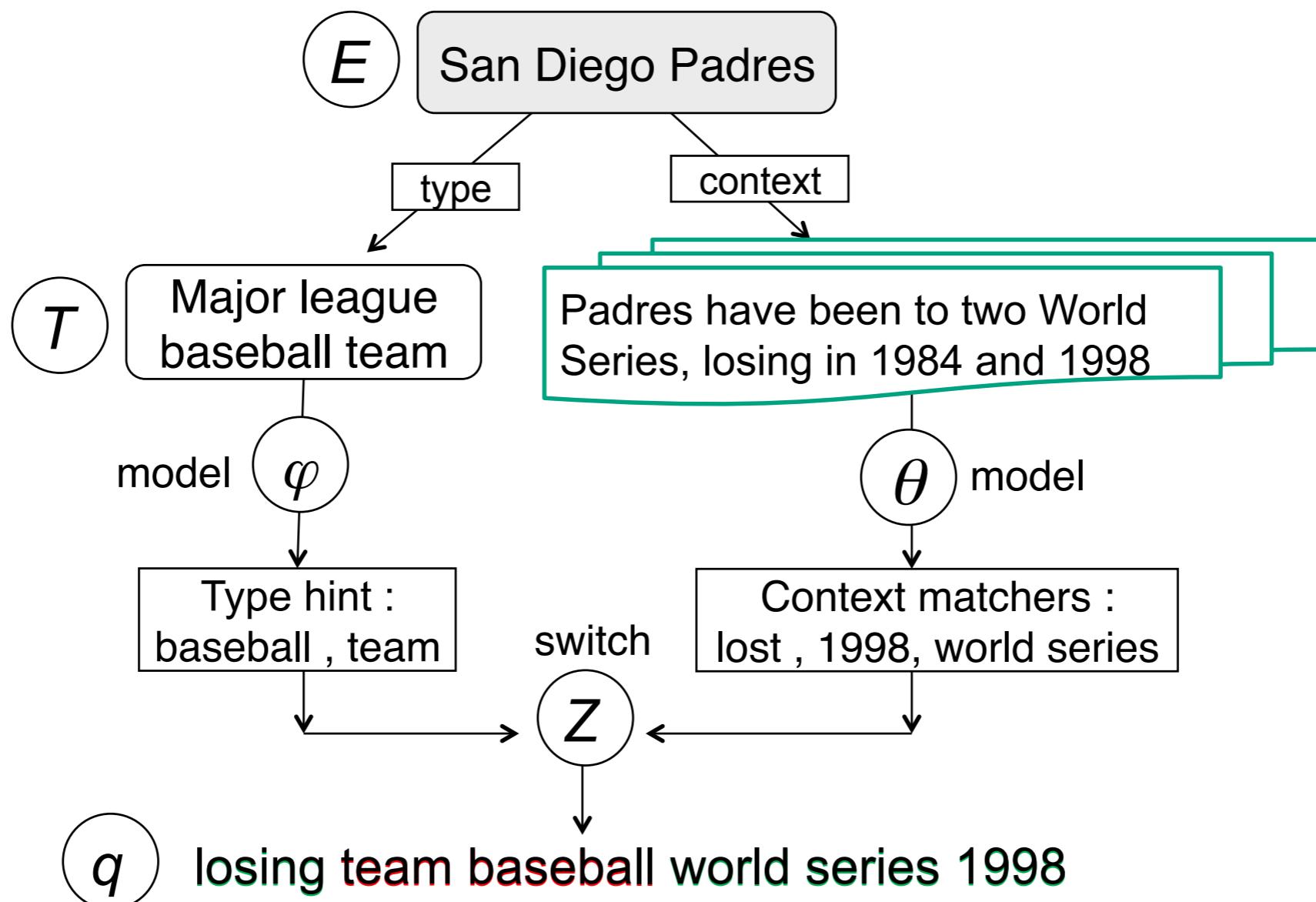
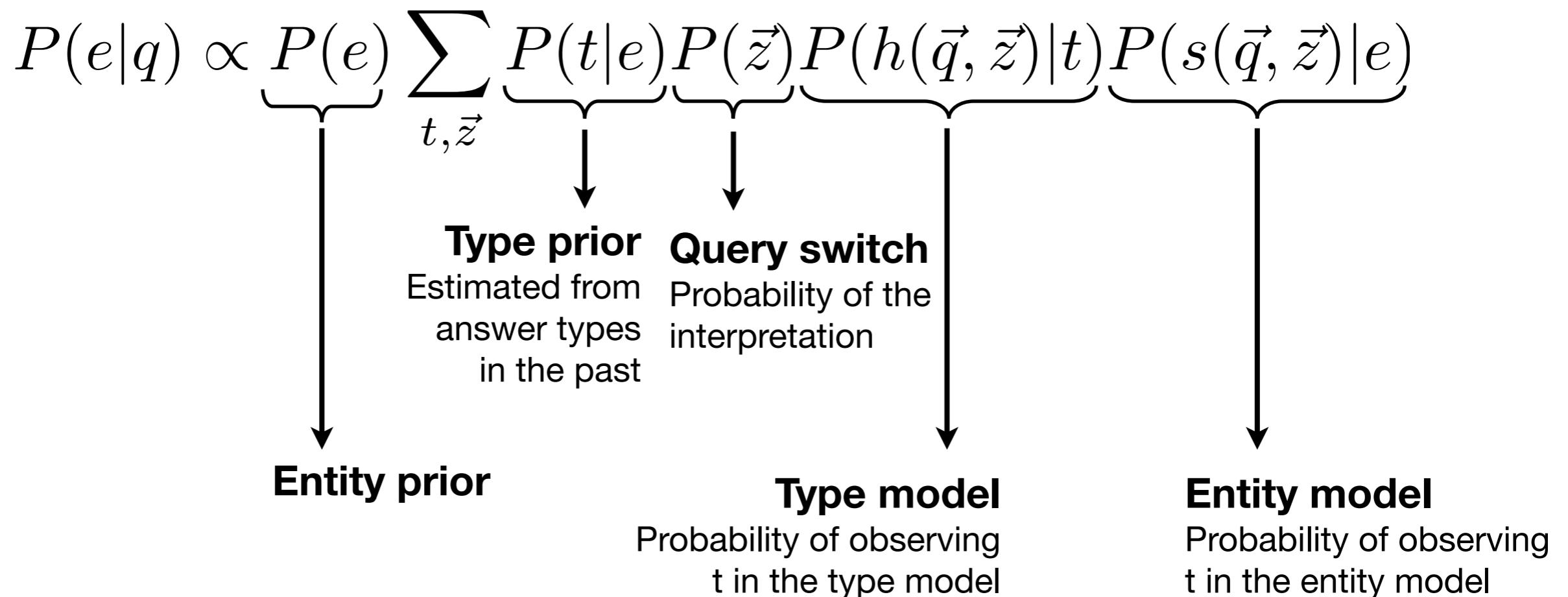


Figure taken from Sawant & Chakrabarti (2013). Learning Joint Query Interpretation and Response Ranking. In WWW '13. (see [presentation](#))

Generative formulation



Discriminative approach

Separate correct and incorrect entities

q : losing team baseball world series 1998

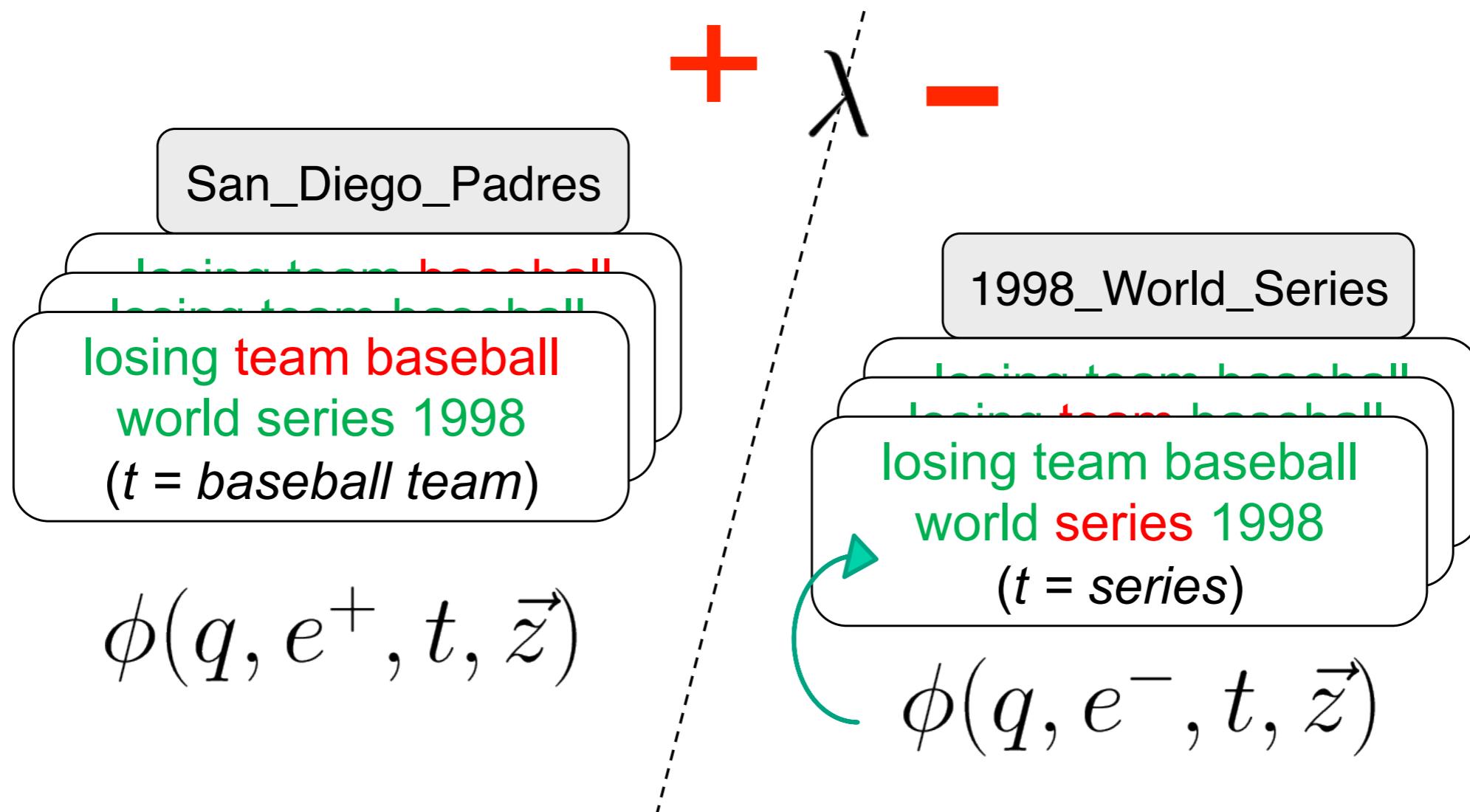
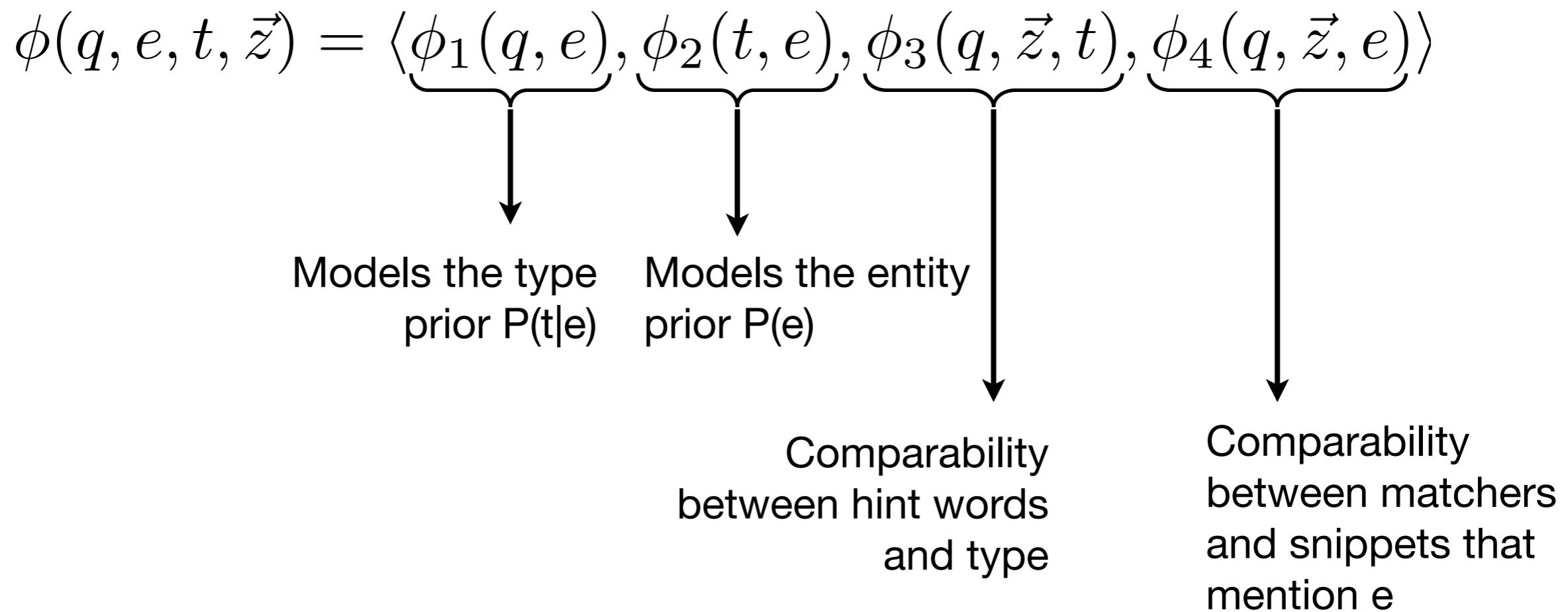
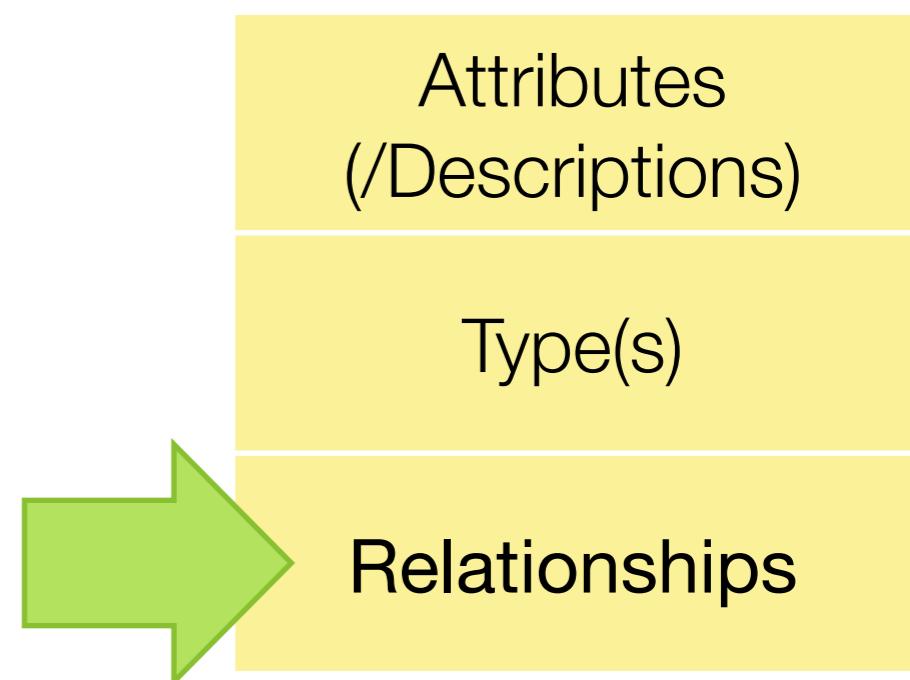


Figure taken from Sawant & Chakrabarti (2013). Learning Joint Query Interpretation and Response Ranking. In WWW '13. (see [presentation](#))

Discriminative formulation



Entity relationships



Related entities

Google Search Krisztian Balog Notification bell

Web Images Maps Shopping News More Search tools User profile icon

About 8,700,000 results (0.41 seconds)

Kimi Raikkonen - Lotus
Flag 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 ˈmotiəs ˈræikːənen]; 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
kimiraikkonen.com/



More images

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|

Search icon

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

Google

tom cruise wives

Search icon

Krisztian Balog

Bell icon

+ Share

Profile icon

[Web](#) [Images](#) [Maps](#) [Shopping](#) [More](#) [Search tools](#)

[Profile](#) [Feedback](#) [Settings](#)

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](#)

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

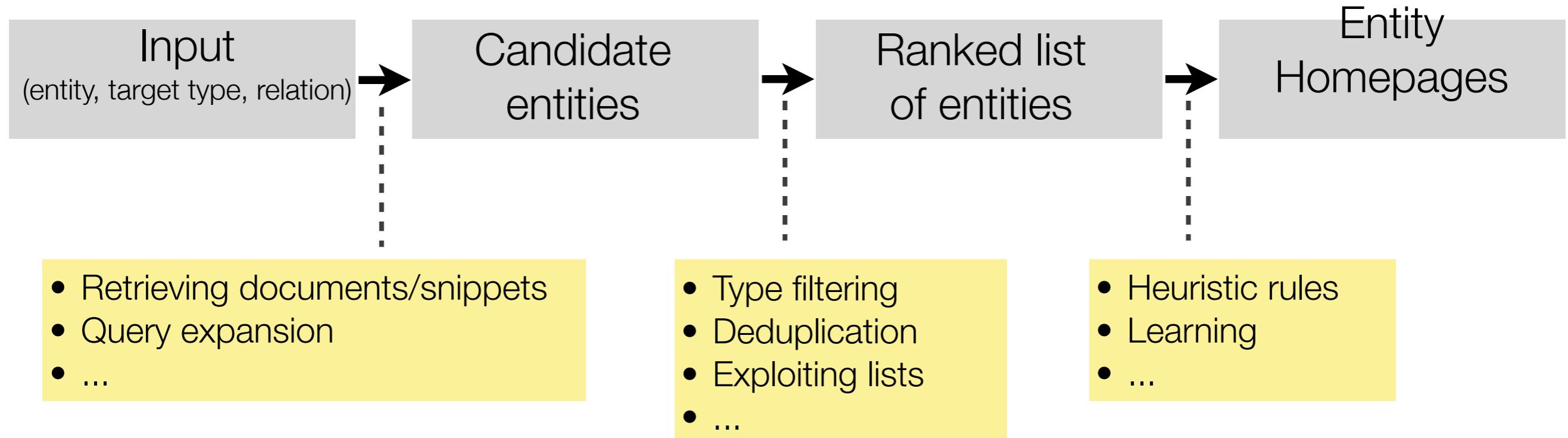
Example information needs

🔍 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

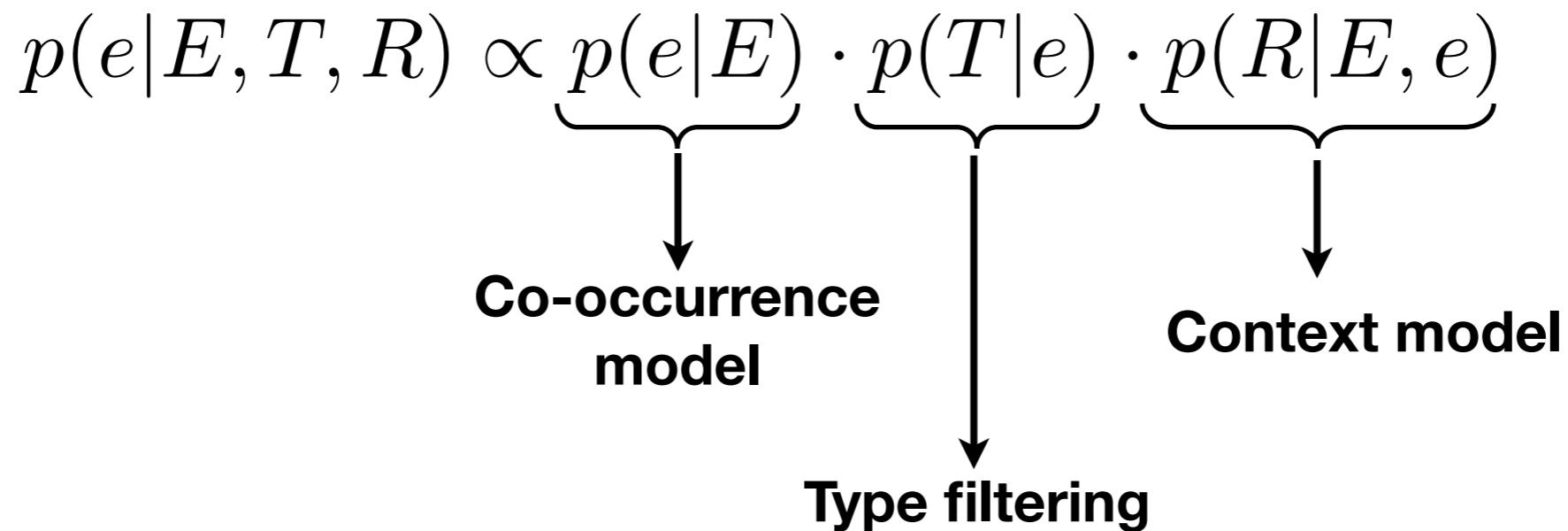
A typical pipeline



Modeling related entity finding

[Bron et al. 2010]

- Three-component model



Wrapping up

- 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!