Query Interpretation and Representation

(for Searching the Web of Objects)

Soumen Chakrabarti

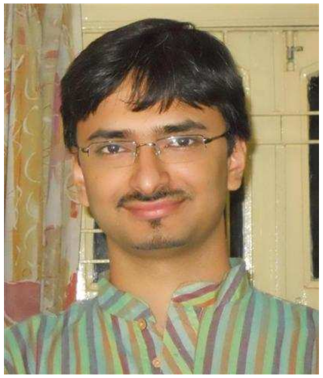
IIT Bombay

http://goo.gl/fwpG5

Acknowledgment

Mandar Joshi
IIT Bombay

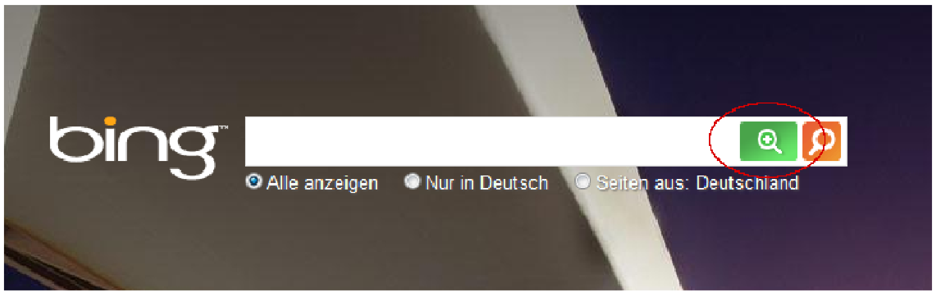
Monojit Choudhuri
Microsoft Research



Rishiraj Saha Roy
IIT Kharagpur

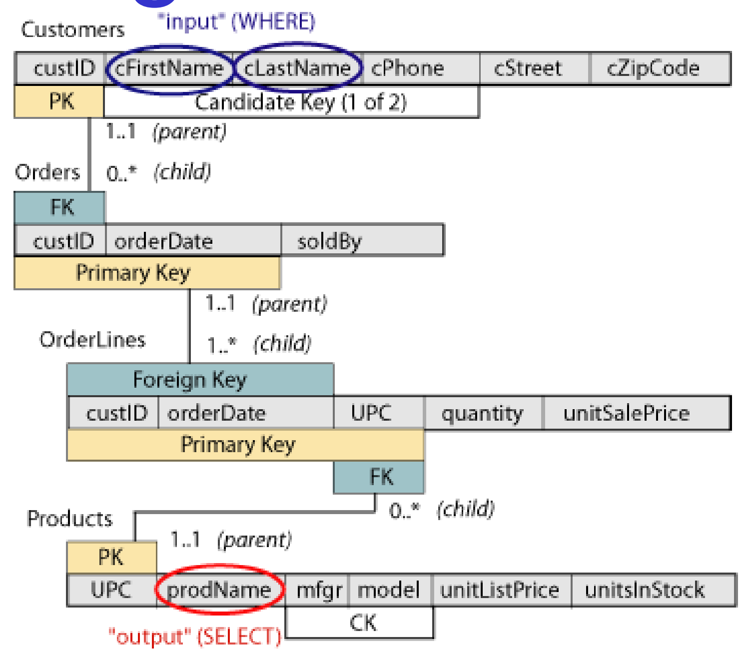
Matthias Hagen
Universität Weimar

The query understanding crisis in IR



Web search

- No or chaotic schema
- Query = “telegraphic” token sequence
- *Purpose* of each token in the query unknown



Relational databases

- Tables, rows, columns
- Primary, foreign keys
- Variables, constraints, aggregators

Several SIGIR workshops on query understanding

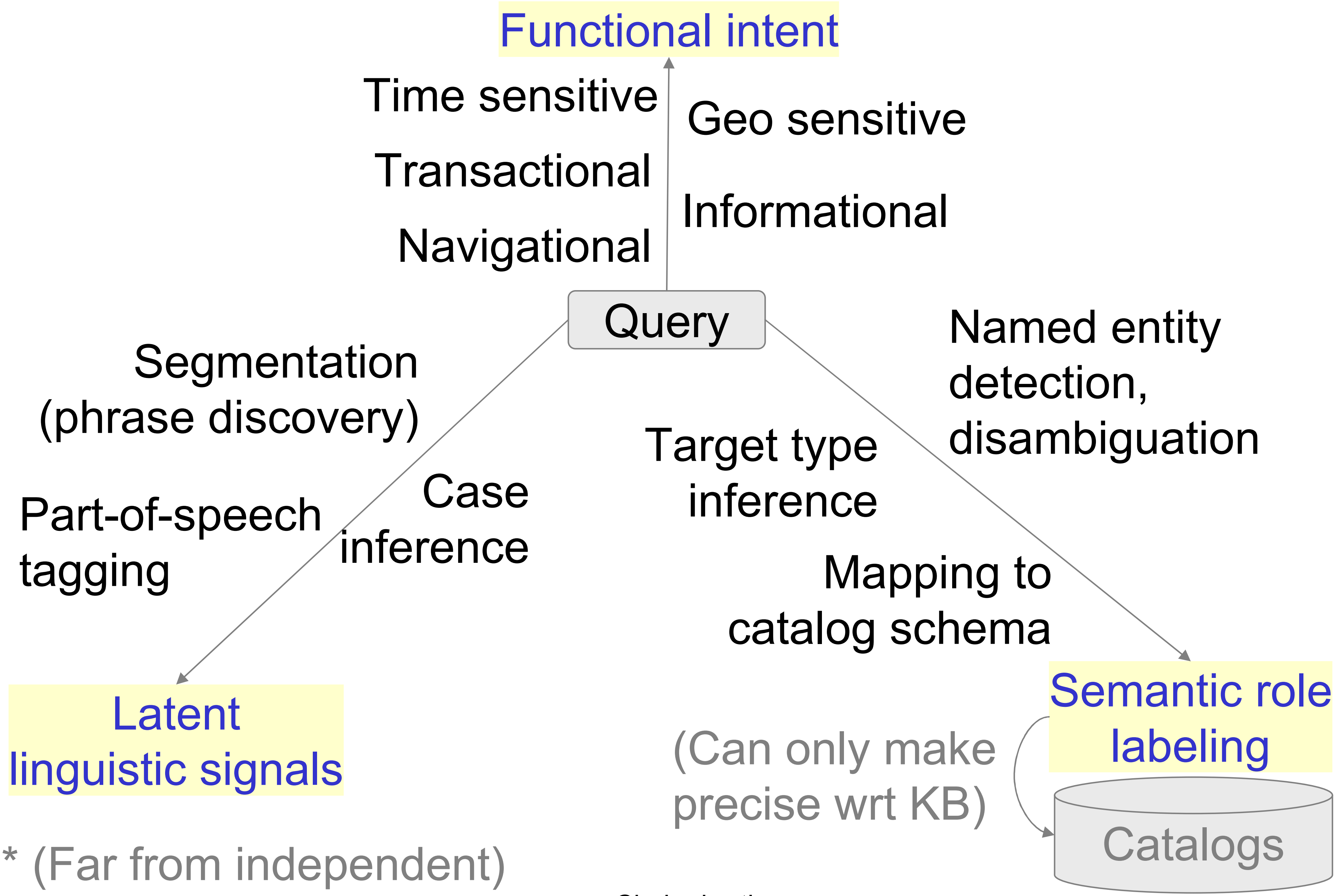
“Telegraphic” Web queries

mother teresa images	woodrow wilson president university
4 minutes lyric	dolly clone institute
black swan summary	hermitage museum bank river
mario kart guide	crysis mods
condemned screenshots	losing baseball world series 1998

- Few function or relational words
- Relatively free word order
  - dolly clone institute ≈ institute dolly clone
- No or rare capitalization
- Rare to find quoted phrases

Rather different from playing Jeopardy or question answering

Query understanding dimensions*



* (Far from independent)

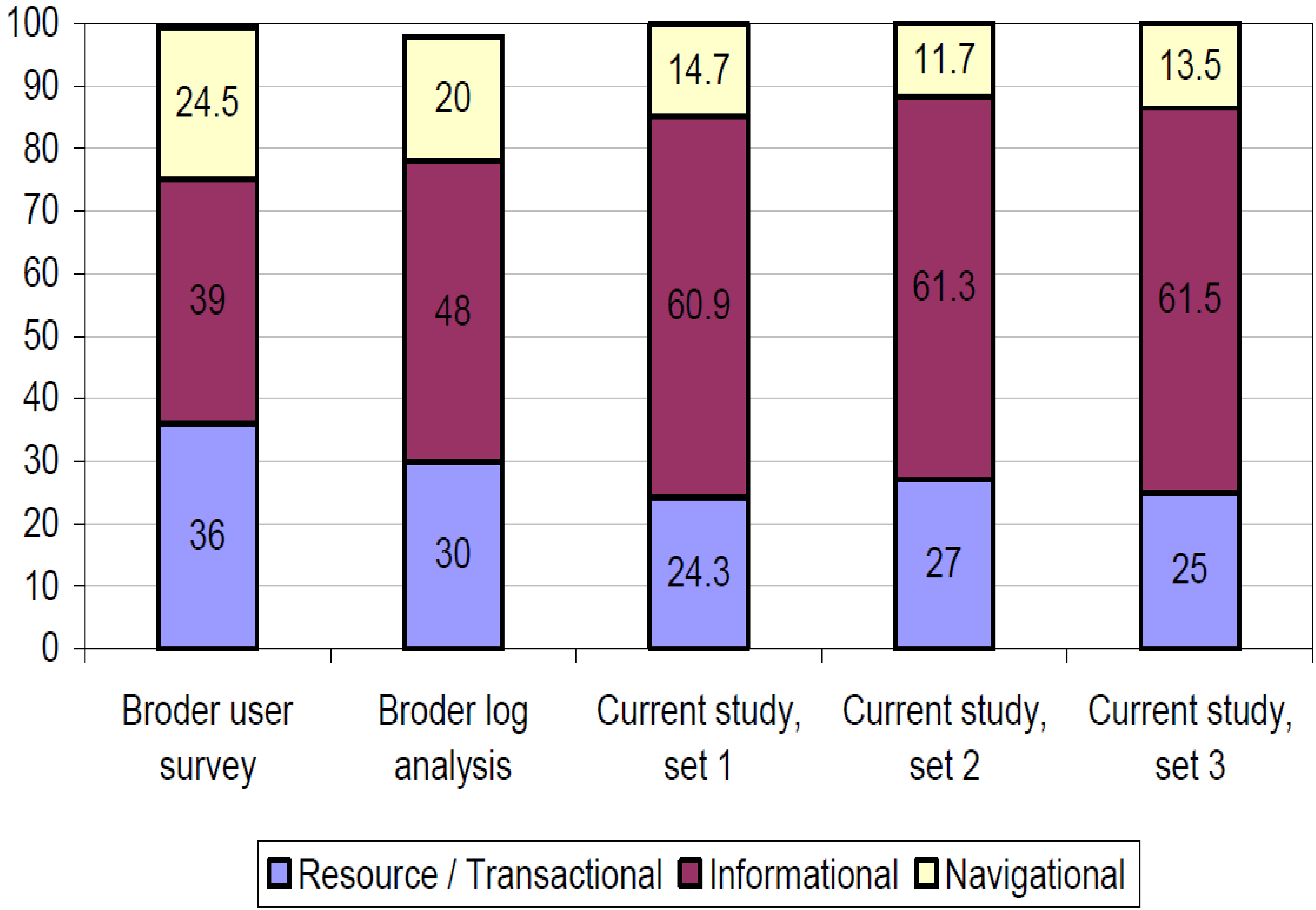
Functional intent

http://goo.gl/fwpG5

Kinds of functional intent

- Broder’s original intent categorization
- Navigational, where the user has a particular Web page in mind, but does not know the exact URL, e.g. wikipedia home page
- Informational, where the user seeks some information that is assumed to be present in one or more Web page
- Transactional, where the user wants to perform some Web based activity or transaction, like shopping or downloads

2004 population estimates

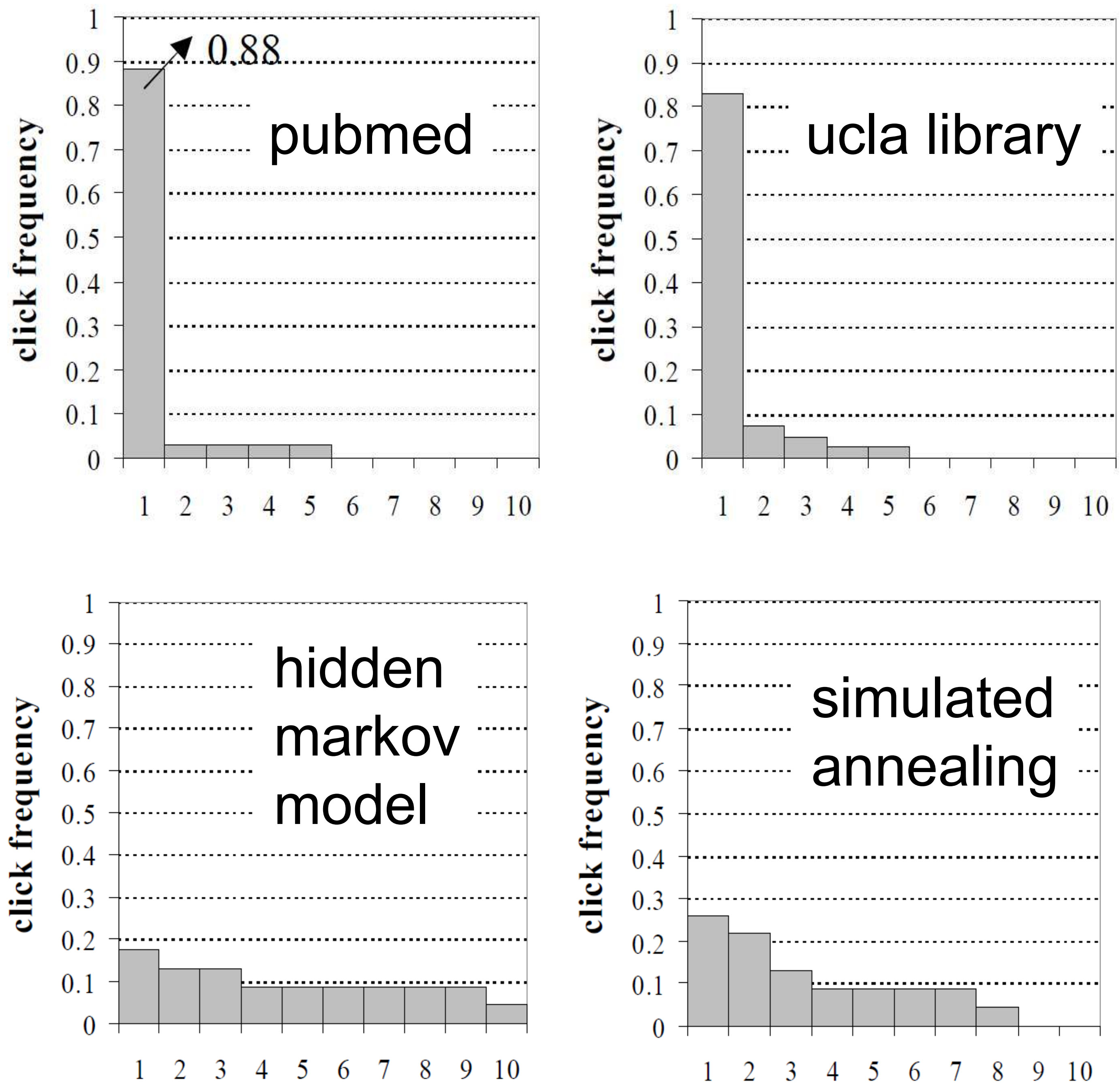


Why useful to classify?

- Search engines use hundreds of features to rank response pages
- Page content match with query, anchor text match with query, PageRank, past clicks…
- Best combination usually fitted by machine learning
- Best combination varies considerably by query type
  - PageRank and clicks more important for navigational queries

Past click behavior

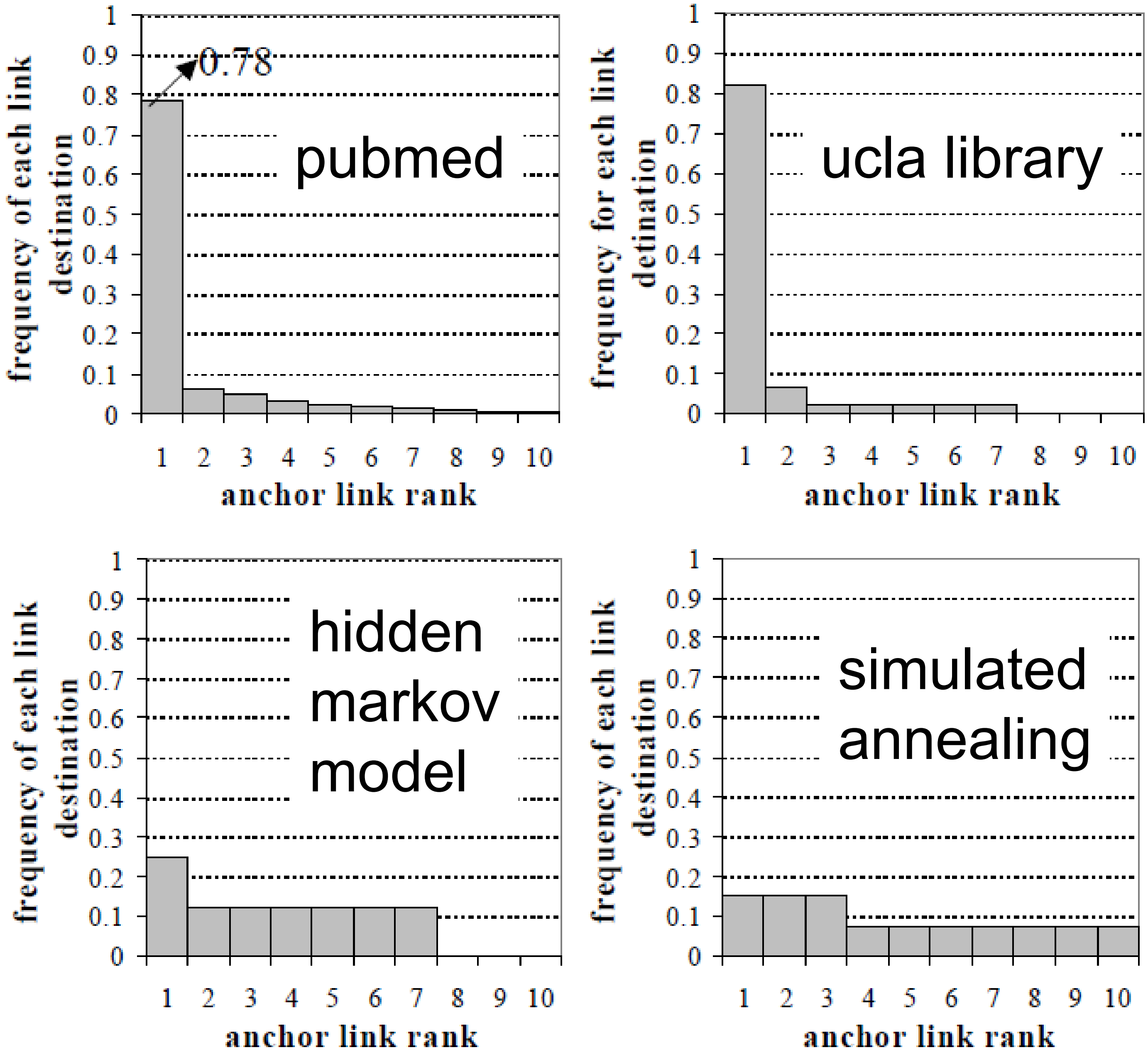
- Sort response URLs in decreasing order of clicks from all users
- Normalize counts to add up to 1
- Skewed⇒ navigational, flat⇒ informational



Anchor-link distribution

- Most instances of anchor texts *pubmed* or *bofa* will associate with links www.ncbi.nlm.nih.gov and www.bankofamerica.com
- This will not hold for anchor text *hidden markov model* or *simulated annealing*
- As with click distribution, compute skew in top-ranked URLs
- Large skew $\Rightarrow$ navigational,
  flat $\Rightarrow$ informational

Anchor-link skew results



Combination of features leads to 90% accurate classification of queries

Latent linguistic signals

http://goo.gl/fwpG5

Query segmentation

- **`times square dance`**
- **`two man power saw`**
- Good reconstruction of quoted phrases
  - Speeds up posting merges
  - Reduces the number of candidates to score
  - Improves scoring function (proximity reward)
- Bad reconstruction damages ranking quality
- Clues and data we can harness
  - N-gram statistics from query logs and corpus
  - Phrase dictionaries, clicks

What makes $w_1 w_2$ a phrase?

n two-word sliding windows		w_2 found $n_{01}+n_{11}$ times
	$n-n_{01}-n_{10}-n_{11}$	Not w_1 followed by w_2, n_{01} times
w_1 found $n_{10}+n_{11}$ times	w_1 followed by not w_2, n_{10} times	w_1 followed by w_2, n_{11} times

- Is n_{11}/n large compared to $\frac{n_{01}+n_{11}}{n} \frac{n_{10}+n_{11}}{n}$
- Contingency table probabilities better explained as
  - Two coins, joint probabilities as products of marginals
    - Two parameters
  - Or one four-sided die, dependent random variables?
    - Three parameters; is the additional complexity justified by data?

Best independent model

- Two coins with head (word present) probabilities $p_1,\ p_2$
- To maximize the probability of observing counts

$n_{00}=n-n_{01}-n_{10}-n_{11}$	n_{01}
n_{10}	n_{11}

  we should choose

$$p_1^* = \frac{n_{11}+n_{10}}{n} \qquad p_2^* = \frac{n_{11}+n_{01}}{n}$$

- Calculate $H_0 =$

$$\sum_{i,j\in\{0,1\}} n_{ij} \log\{(1-p_1^*)^{1-i}(p_1^*)^i(1-p_2^*)^{1-j}(p_2^*)^j\}$$

Best dependent model; likelihood ratio

- This time there are three independent parameters
- We should choose $p^*_{ij} \propto n_{ij}$
- Calculate H, largest log probability for dependent case; $H \geq H_0$

Null model

Alternative model

- $H - H_0$ is an indication of how strong the association is between the two words
- If large, likely compound word or phrase
- Query log or corpus?

Limitations

- **`new york times square dance`**
- Decision for **`york times`** made independently of decision for **`times square`**
- Threshold based; no global perspective
- Phrase for one user not for another?
- No connection to knowledge bases like Wikipedia/Freebase
- Does not exploit entity-action patterns
- No cognizance of retrieval/ranking performance (more about this later)

Encoding segments

- If query is $w_1\ w_2 \ldots\ w_n$
- Can choose to either insert a separator or not in each of $n-1$ gaps
- Therefore 2^{n-1} possible segmentations
  - Each gap makes a binary decision
- Also represented as $s_1 s_2 \ldots s_m$
- Each s_j is a segment of one or more words
- A segment may or may not be quoted
- Two steps: segmentation then quoting

Supervised segmentation

- Bergsma+Wang 2007, early influential work
- SVM trained using manually segmented queries
  - Labor intensive
  - Will also look at unsupervised techniques
- Binary classification at each gap
  - Decision boundary features
  - Context features
  - Dependency features

Decision boundary features

- Indicator features $\ldots x_{L2} x_{L1} x_{L0} | x_{R0} x_{R1} x_{R2} \ldots$

Name	Description
is-L0-the, is-R0-the	Is the token *the*?
is-L0-free, is-R0-free	Is the token *free*?
are-POS-pL-pR	Is POS(x_{L0})=pL and POS(x_{R0})=pR? (One feature for each pL, pR pair)
is-i-from-left, is-i-from-right	Is x_{L0} placed i tokens from left; is x_{R0} placed i tokens from right end of query?

Mutual information features

$$\mathrm{MI}(x_{L0}, x_{R0}) = \frac{\Pr(x_{L0} x_{R0})}{\Pr(x_{L0}) \Pr(x_{R0})}$$

- Written in log form, constant plus…

$$\log C(x_{L0} x_{R0}) - \log C(x_{L0}) - \log C(x_{R0})$$

- … where C is count of number of pages
- Instead of hardwiring this, fire three features

$$\log C(x_{L0} x_{R0}),\ \log C(x_{L0}),\ \log C(x_{R0})$$

- And let classifier find best weighted combination

Boundary (statistical) features at w|x

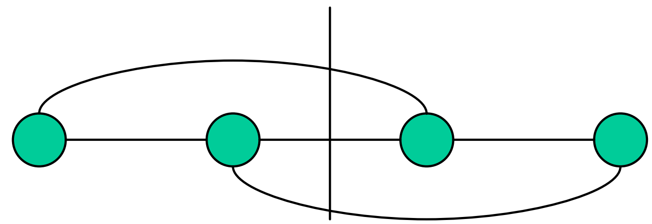
Name	Description
web-count	Doc frequency of x in Web corpus
pair-count	… “w x” …
definite	… “the w x” …
collapsed	… wx … (concatenated)
and-count	… “w and x” …
qcount1	Count of x in query log
qcounts2	Count of “w x” in query log

Context features

- **`bank loan amortization schedule`**
  - `bank` **`loan amortization`** `schedule`
  - **`bank loan`** **`amortization schedule`**
- Competing strengths of association
  - Why not a global segmentation? (Coming up)
- Add more features
  - Include x_{L1} and x_{R1}
  - Also include corpus frequencies of 2, 3-grams if available

Dependency features

- **`female`** `bus` **`driver`**
- *Female* is associated with *driver*, not *bus*
- Therefore, include as new features the pairwise counts between
  - x_{L0} and x_{R1}
  - x_{L1} and x_{R0}



- (Modeling dependencies over a longer range did not improve performance)

Evaluation of query segmentation

- **Query level accuracy** is the ratio of correctly segmented queries to the total number of queries
- A decision has to be made at every term boundary whether to insert a segmentation break or not; **break level accuracy** = # correct decisions ÷ # total decisions
- Output is treated as a set of segments. Therefore, one can compute the precision, recall, and F1 measures with respect to these sets as **segment-level scores**
- How about IR performance (coming up)

Bergsma-Wang results

Feature Type	Feature Span	Test Set	
		Seg-Acc	Qry-Acc
MI	Decision-Boundary	68.0	26.6
Basic	Decision-Boundary	71.7	29.2
Basic	Decision-Boundary, Context	80.2	52.0*
Basic	Decision-Boundary, Context, Dependency	81.1	53.2
All	Decision-Boundary	84.3	57.8*
All	Decision-Boundary, Context	86.3	63.8*
All	Decision-Boundary, Context, Dependency	85.8	61.0

Segment-level Query-level

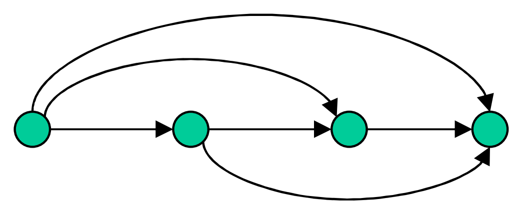
- Of 35M AOL queries, those with clicks
- POS tagged, at least 4 tokens, 1500 sampled
- Manually annotated; agreement not great (~50%)
- On test subsets with agreement, algorithm shows higher accuracy

Unsupervised segmentation

- Tan+Peng 2008, reduces manual labor
- Say we are given an oracle that returns a probability $\Pr(s)$ given any span s of tokens
- Query $w_1,\ldots,w_n$ segmented to $S = s_1,\ldots,s_m$
- Approximation

$$\Pr(S) = \Pr(s_1)\Pr(s_2|s_1)\cdots\Pr(s_m|s_{1:m-1})$$

$$\approx \prod_{s_i \in S} \Pr(s_i)$$



- Related to shortest path in Monge graphs; dynamic programming gives max prob segmentation

Dynamic program

$$\underbrace{w_1, \ldots, w_{j-1}}_{\text{Recurse}}, \underbrace{w_j, \ldots, w_i}_{\text{Last segment}}$$

- For $i = 1, 2, \ldots, n$, find best segmentation (probability) $B[i]$ up to i^{th} word
- Last segment can be from any j to i
- Explore possible js exhaustively
- Probability is $B[j-1]\Pr(w_j, \ldots, w_i)$
- Can keep track of segments as usual
- Can extend to top-k segmentations with $O(nkm\log(km))$ work

Probability oracle

- Input: q-gram raw counts up to modest $q=5$
  - Extend to larger q via inclusion-exclusion bounds
- Probability of one word = count of word $\div$ count of all possible words
- Probability of a phrase = ?
  - Count of all possible phrases?
  - Pr("york times") > Pr("new york times")
- Solution: semi-supervised learning
  - Distill corpus into counts of maximal corpus matches of query q-grams
  - Iteratively learn probabilities and re-segment

Semisupervised re-segmentation

- Corpus distilled to $\mathcal{D} = \{(x, c(x)) : x \in Q\}$
- From this, (re)estimate phrase probabilities

$$\theta = \arg\max \Pr(\mathcal{D}|\theta) \Pr(\theta)$$

Regularization term controls power of θ

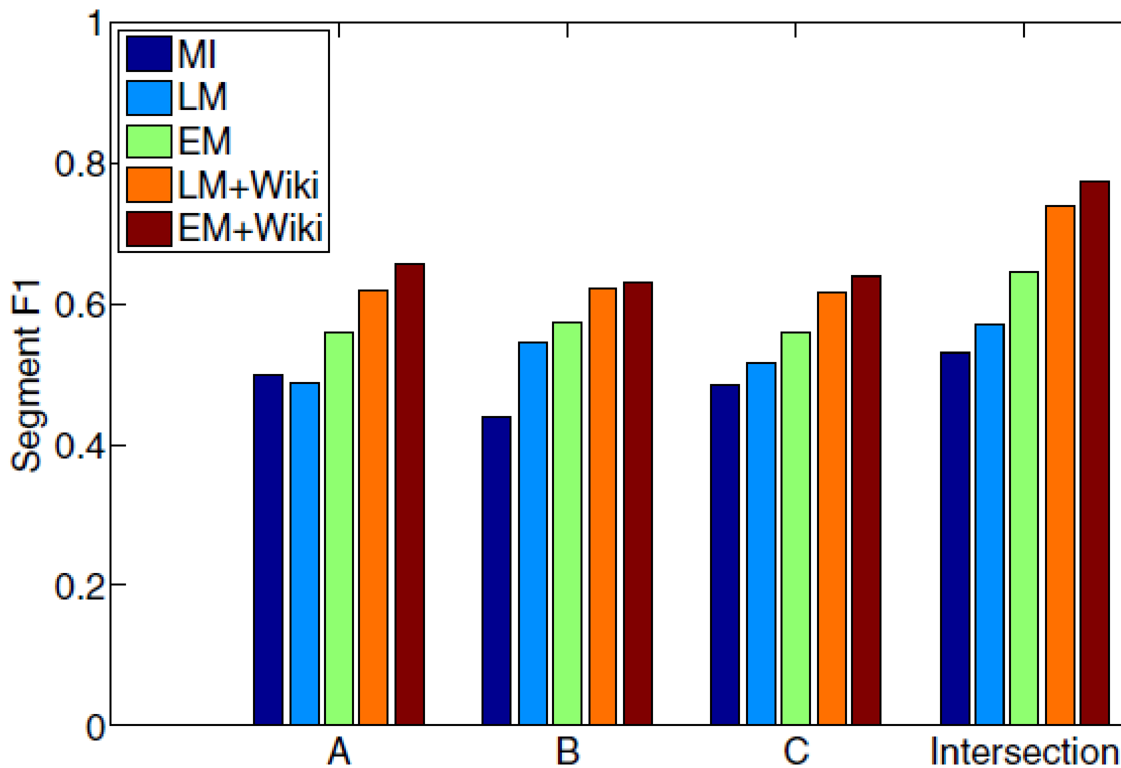
$$= -\arg\min \log \Pr(\theta) + \log \Pr(\mathcal{D}|\theta)$$

$$\log \Pr(\mathcal{D}|\theta) = \sum_{x \in Q} c(x) \log \Pr(x|\theta)$$

$$+ \boxed{\left[N - \sum_{x \in Q} |x| c(x)\right] \log \left[1 - \sum_{x \in Q} \Pr(x|\theta)\right]}$$

Counts of phrases not in the query

[TP 2008] sample results



(Three annotators ABC)

Better for cases where annotators agreed

- Also used a separate language model from Wikipedia titles
- LM = without iterative EM reestimation

Simpler system [HPSB 2011]

- [TP 2008] arguably non-trivial to implement, considerable computation costs
- Turns out the following merit score for a segment does very well
- Has no need for normalization to probabilities

$$score(S) = \begin{cases} \sum_{s \in S, |s| \geq 2} \boxed{|s|^{|s|}} \cdot freq(s) & \text{if } freq(s) > 0 \text{ for all } s \in S, |s| \geq 2 \\ \boxed{-1} & \text{else.} \end{cases}$$

Prevents recognition of phrases with zero count support

Offsets for the natural (raw frequency) penalization of longer phrases

Some justification for $|s|^{|s|}$

$\lvert s\rvert$-grams	Unique Entries	Median Freq.	$\frac{\text{2-gram Freq.}}{\lvert s\rvert\text{-gram Freq.}}$	$\lvert s\rvert^{\lvert s\rvert}$
2-grams	2 409 063	3 461 030	1	4
3-grams	5 431 544	78 733	44	27
4-grams	8 073 863	7 356	470	256
5-grams	10 000 000	1 129	3 065	3 125

Annotator	Segment Length					
	1	2	3	4	5	6
A	451	699	74	14	2	
B	351	541	113	77	9	2
C	426	588	100	51	5	1
Agree	151	318	31	9		

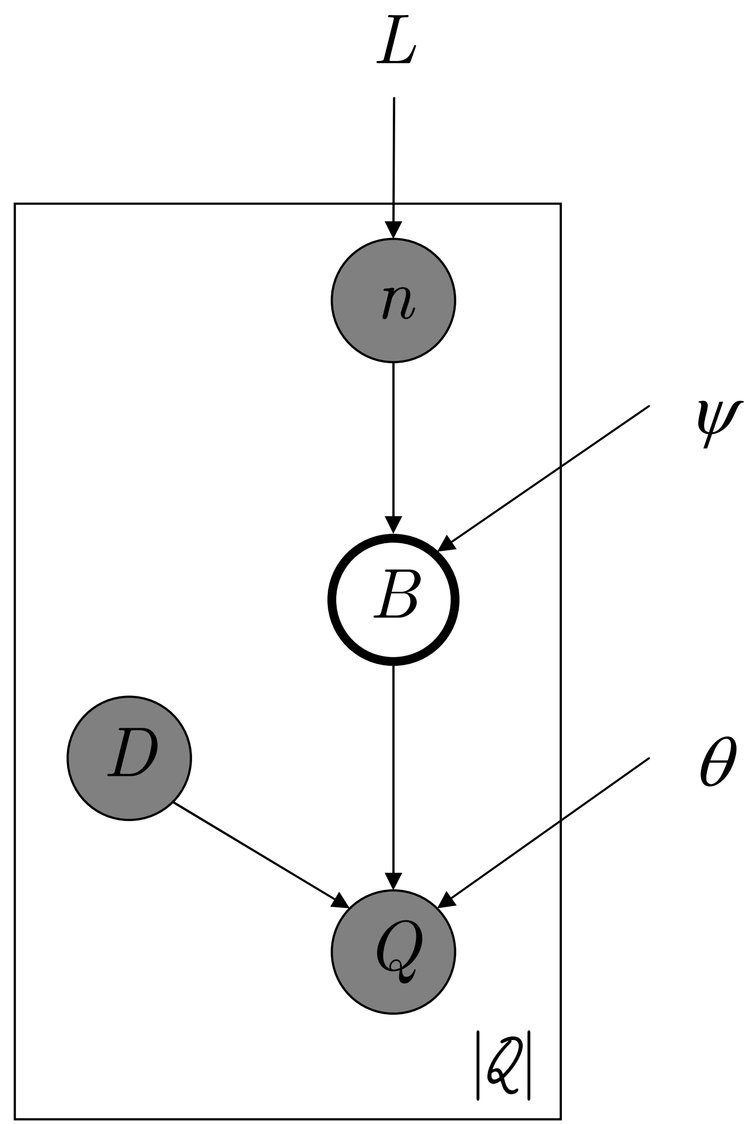
Exploiting clickthrough [LHZW 2011]

- Unsupervised technique
- Title and text from clicked docs provide clues for query segmentation

bank of america invesment	bank of america associate banking investments homepage
	bank of america investment services inc investments overview
credit card bank of america	bank of america credit cards contact us overview
	credit cards overview find the right bank of america credit card for you

[LHZW 2011] generative process

- Pick query length n using length distribution (params) L
- Pick segment partition from $\Pr(B|n,\psi)$
- Given n, B, use segment unigram model (and clicked documents D) to generate each segment
- Inference goal: estimate B after seeing n, D, Q



Events and modeled probabilities

Pick segment lengths

Pick words in segments

$$\Pr(B|Q,n,\psi,\theta,D) \propto \Pr(B|n,\psi)\Pr(Q|B,\theta,D)$$

$$\Pr(B|n,\psi) \propto \prod_{m=1}^{M} \Pr(|S_m(B)|\,|\psi)$$

Length comp's indep.

...modeled as...

$$= \prod_{m=1}^{M} \exp(-|S_m(B)|^f)$$

$$\Pr(Q|B,\theta,D) = \prod_{m=1}^{M} \Pr(Q_m(B)|\theta,D)$$

Text in each span indepdent of others

Now use smoothed (bigram) language model

Sample results

	MI	[TP 2008]	[LHZW 2011]
Query accuracy	.343	.671	.682
Break accuracy	.728	.871	.855
Segment precision	.510	.767	.770
Segment recall	.550	.782	.788
Segment F1	.530	.774	.779

Also in [LHZW 2011]

- Integrated relevance model for retrieval
- End to end ranking performance
- Better than [FP 2008] but worse than clairvoyant

Exploiting only query logs [MSGLC 2011]

- Query q_i has length ℓ_i tokens
- Given an arbitrary n-token span M in it
- Is it a statistically significant multi-word unit?
  - Do its constituents appear together more frequently than they would under a *bag-of-words* (null) model?
- $(\ell_i - n + 1)$ positions where M could be placed
- Other tokens can be permuted $(\ell_i - n)!$ ways
- Event "M occurs in q_i" ($X_i = 1$) has probability

$$P_i = \frac{(\ell_i - n + 1)(\ell_i - n)!}{\ell_i!} = \frac{(\ell_i - n + 1)!}{\ell_i!}$$

Deviation analysis and segmentation

- $X = \Sigma_i X_i$ is the modeled number of occurrences of M, with expectation $\Sigma_i P_i$
- Say observed frequency is k out of N queries
- Using Hoeffding's inequality, $\frac{2(k - \sum_i P_i)^2}{N}$ is a surprise value
- Large value means more likely to be MWE
- Use dynamic programming to find segmentation having largest sum of segment surprise values
- Can mix in Wikipedia titles easily

The end goal of query segmentation

- IR performance, strangely neglected
  - Except [LHZW 2011] and [SGCL 2012]
  - May be *sensitive to ranking quirks* of search engine
- Segmentation followed by quoting
  - Alternatively, ignore quotes on single tokens
- Segmentation algorithm characterized by clairvoyant best-performing quotation

Segmented query	Quoted versions
	we are the people song lyrics
	we are the people "song lyrics"
	we are "the people" song lyrics
we are | the people | song lyrics	we are "the people" "song lyrics"
	"we are" the people song lyrics
	"we are" the people "song lyrics"
	"we are" "the people" song lyrics
	"we are" "the people" "song lyrics"

[SGCL 2012] experiments and results

- 500 queries from Bing Australia, May 2010
- Unsegmented *worse than* best algorithms *comparable to* humans *worse than* best clairvoyant

	Unsegmented	Hagen+	Mishra+	Human-A	Human-B	Human-C	Clairvoyant
NDCG@5	0.688	0.763	0.767	0.77	0.768	0.759	0.825
NDCG@10	0.701	0.767	0.768	0.77	0.768	0.763	0.832
MAP@5	0.882	0.942	0.945	0.944	0.942	0.936	0.958
MAP@10	0.865	0.921	0.923	0.923	0.921	0.916	0.944
MRR@5	0.538	0.649	0.65	0.656	0.648	0.632	0.711
MRR@10	0.549	0.658	0.658	0.665	0.656	0.64	0.717

Web of Objects Interpretation

Knowledge bases in search

"Over the next few months, [Google] will also present more facts and direct answers to queries"

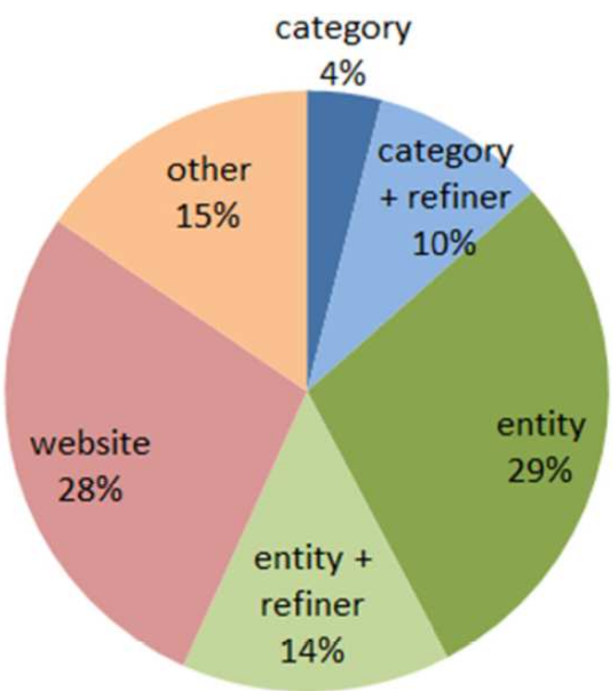
"will better match search queries with a database containing hundreds of millions of "entities"—people, places and things—which the company has quietly amassed in the past two years."

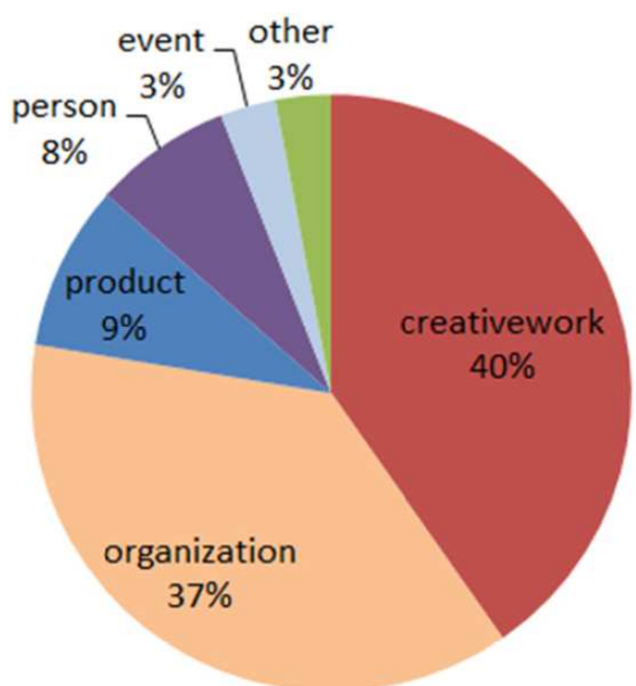
- "Things, not strings"; "knowledge graph" — Google
- "Web of Objects" — Yahoo
- "Snapshot" — Bing
- "Graph search" — Facebook

--- Amit Singhal (Google) to Wall Street Journal, March 2012

The next stage after segmentation

- When a query can be segmented into compound word spans …
- … it is often because said spans mention entities, attributes and types
- The more important question: what is the **purpose** of each query segment?





[LPGKF 2012] study of Bing query logs

Entity recognition in query [GXCL 2009]

- Entity mention in the context of some intent
- Short query, different from entity disambiguation in longer text
- Three random variables of interest
  - t= type of entity: Book, Movie, Game, Song, Organization
  - e = Ambiguous entities: HarryPotter, YMCA
  - n = Left/right context terms: `dvd`, `kindle`, `lyrics`, `phone`
- Context $\{n_1, n_2\}$ expresses intent, help disambiguate alias entities of different types

Example queries

- pics of fight club
- watch gladiator online
- 12 angry men characters
- pc mass effect
- mother teresa images
- 4 minutes lyric
- black swan summary
- new moon
- nineteen minutes synopsis
- all summer long video
- braveheart quote
- american beauty company
- mario kart guide
- crysis mods
- condemned screenshots
- king kong
- blackwater novel
- rehab the song
- umbrella chords
- girlfriend lyrics

Primer: generative ranking models

- A.k.a. probabilistic language models in IR
  - FnT IR monogram by Cheng Zhai
- Query q, response items u (pages, entities)
- From each item u to be ranked, create a probabilistic model M_u that can generate a query
- Then $\Pr(q|M_u)$ is a scoring signal to rank among different competing us
- Note, model goes from response item back to query

GXCL query generation process

$$\Pr(e,n,t) = \Pr(e)\Pr(t|e)\Pr(n|e,t)$$
$$\approx \Pr(e)\Pr(t|e)\Pr(n|t)$$

Choose entity and describe it as (possibly ambiguous) tokens in query

Choose type for entity (mention) --- this often disambiguates it completely

(Only) type determines intent involving the entity, which is expressed as context n

?

Query interpretation means, given a query q, to find

$$\arg\max_{e,n,t} \Pr(e,n,t)\Pr(q|e,n,t)$$

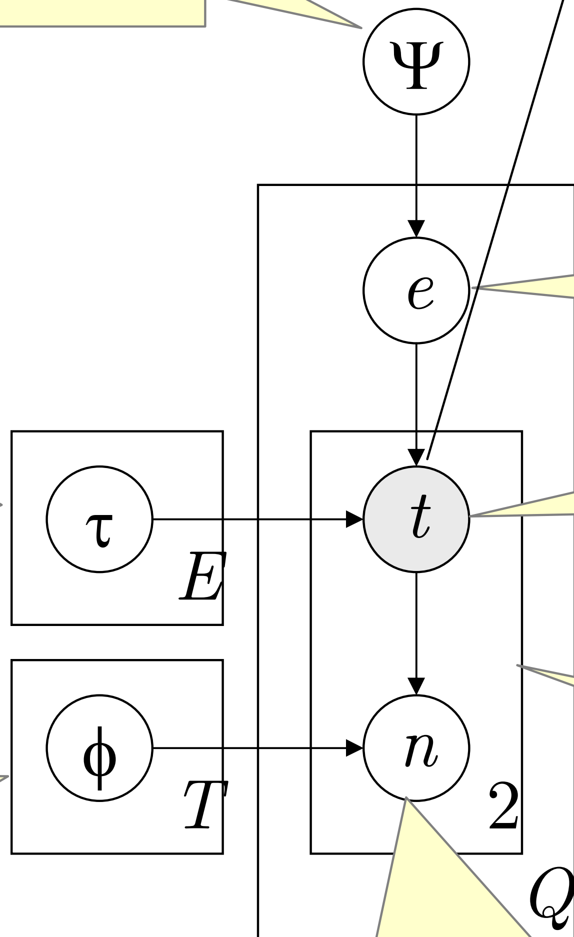
Plate diagram (minor reinterpretation)

t may be unavailable in training data

Distribution from which entity is sampled

For each entity there is a distribution over types (e.g., Harry Potter may be book or movie or game)

For each type there is a distribution over intent words



Choose entity about which to compose query

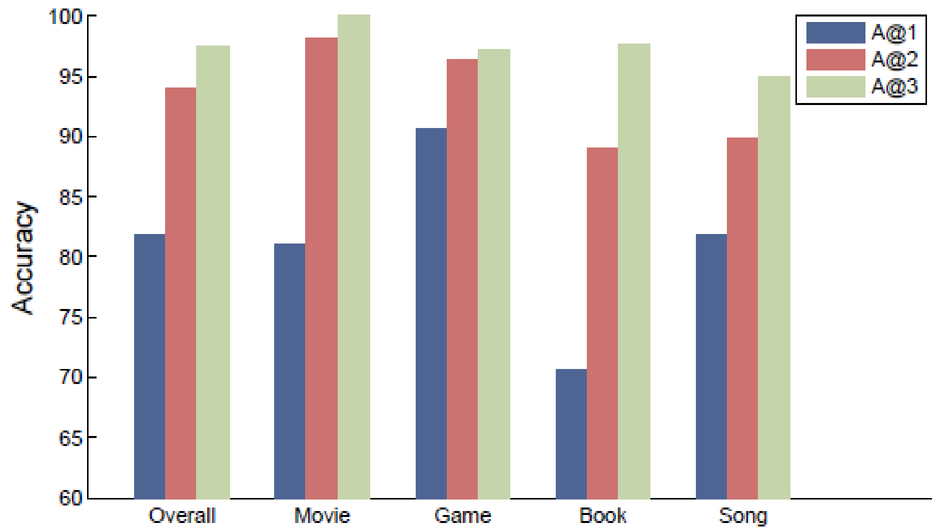
Choose type of entity (why twice?)

For left and right contexts…

Generate left and right intent words given type

For each query…

GXCL sample results



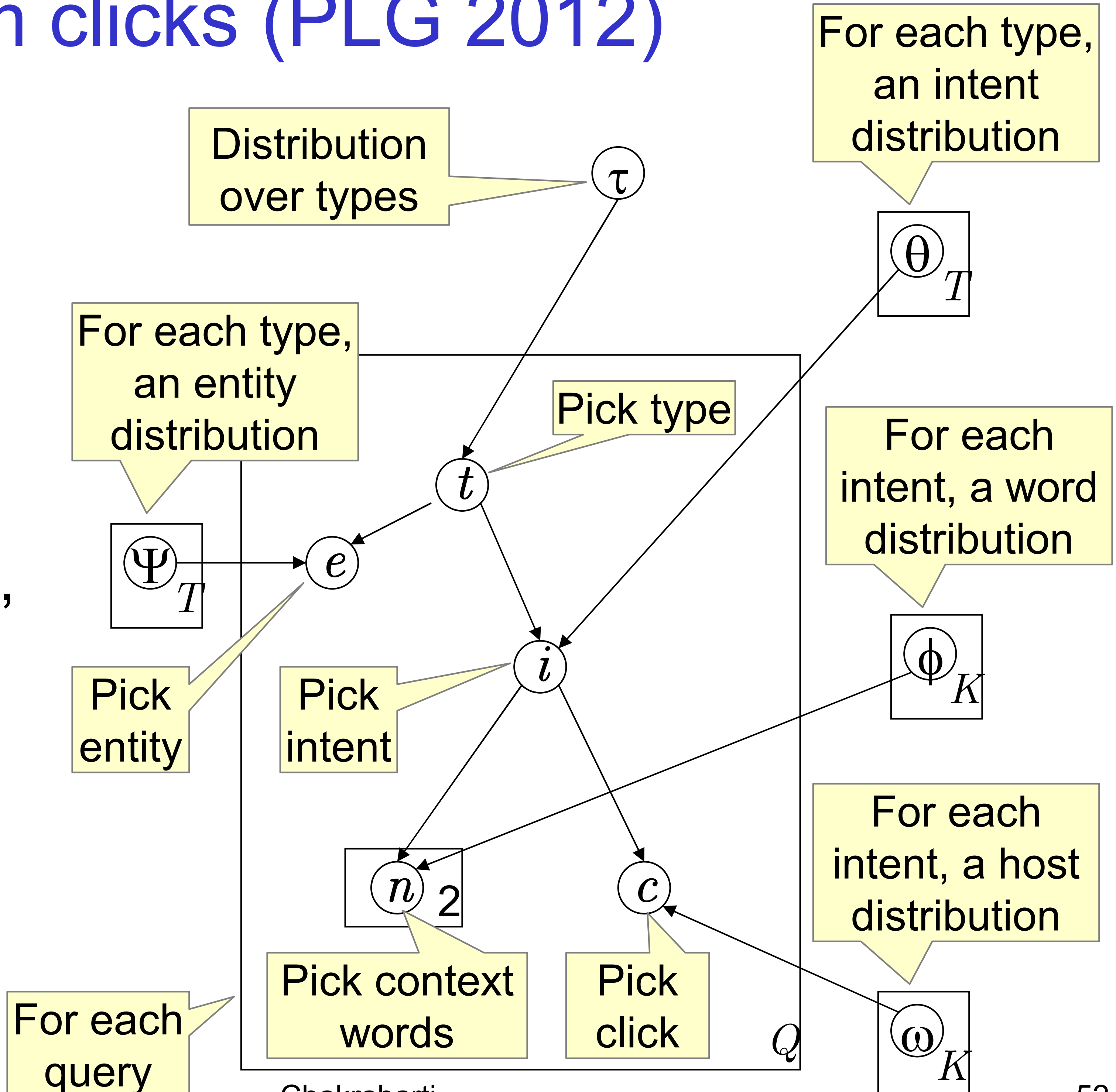
Query-level accuracy on 400 manually segmented and annotated queries

Selected high-probability words for each type

- Movie → movie, photos, soundtrack, pics, wallpaper, cast
- Game → cheats, download, play online, codes
- Book → summary, review, synopsis, quotes, author
- Music → lyrics, video, song

Signal from clicks (PLG 2012)

- Queries governed by latent user intent
- Which influences entity types, choice of query words, and clicked hosts



[PLG 2012] sample results

	Head queries			Tail queries		
	NDCG	MAP	Prec@1	NDCG	MAP	Prec@1
GXCL 2009	0.79	0.71	0.51	0.80	0.73	0.52
PLG 2012	0.87	0.82	0.73	0.80	0.72	0.52

- Millions of Web search queries, 73 types, 135k entities, 40k clicked hosts, 100k context words
- Unlike GXCL, needed automated training
- High-precision string match with Freebase entities
- Trained using EM variant
- 105 head and 98 tail test queries manually annotated
- Better on head (clicks), similar on tail queries

Beyond entity-bearing queries

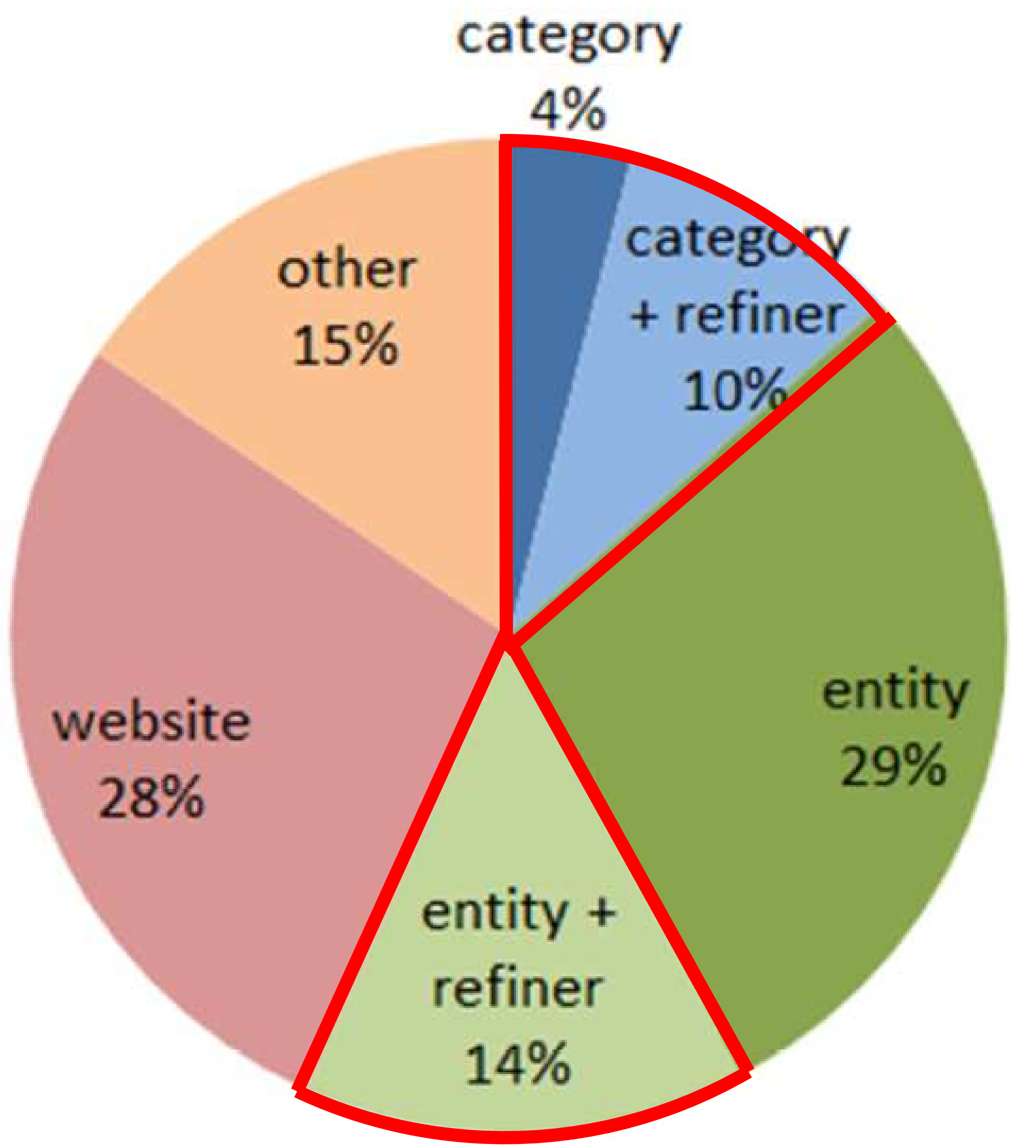
Natural Language Query	Telegraphic Query
Woodrow Wilson was president of which university?	woodrow wilson president university
At what institute was Dolly cloned?	dolly clone institute
Along the banks of what river is the Hermitage Museum located?	hermitage museum bank river
Which team lost the baseball World Series in 1998?	baseball world series 1998 losing team

Note --- each query contains words hinting at a target type

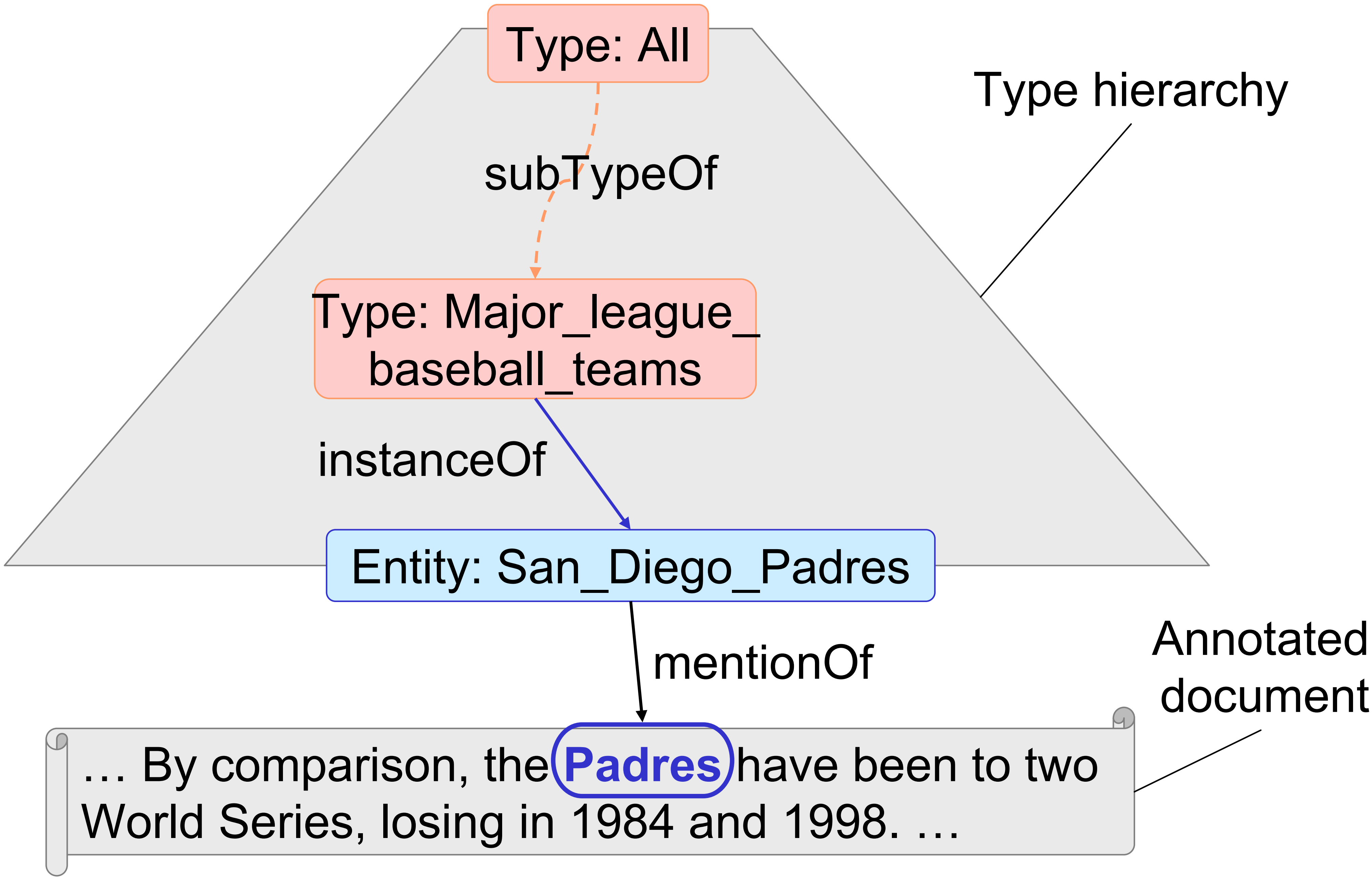
Searching the "Web of things"

At least 14% of Web search queries mention target type or category

Lin et. al., WWW 2012



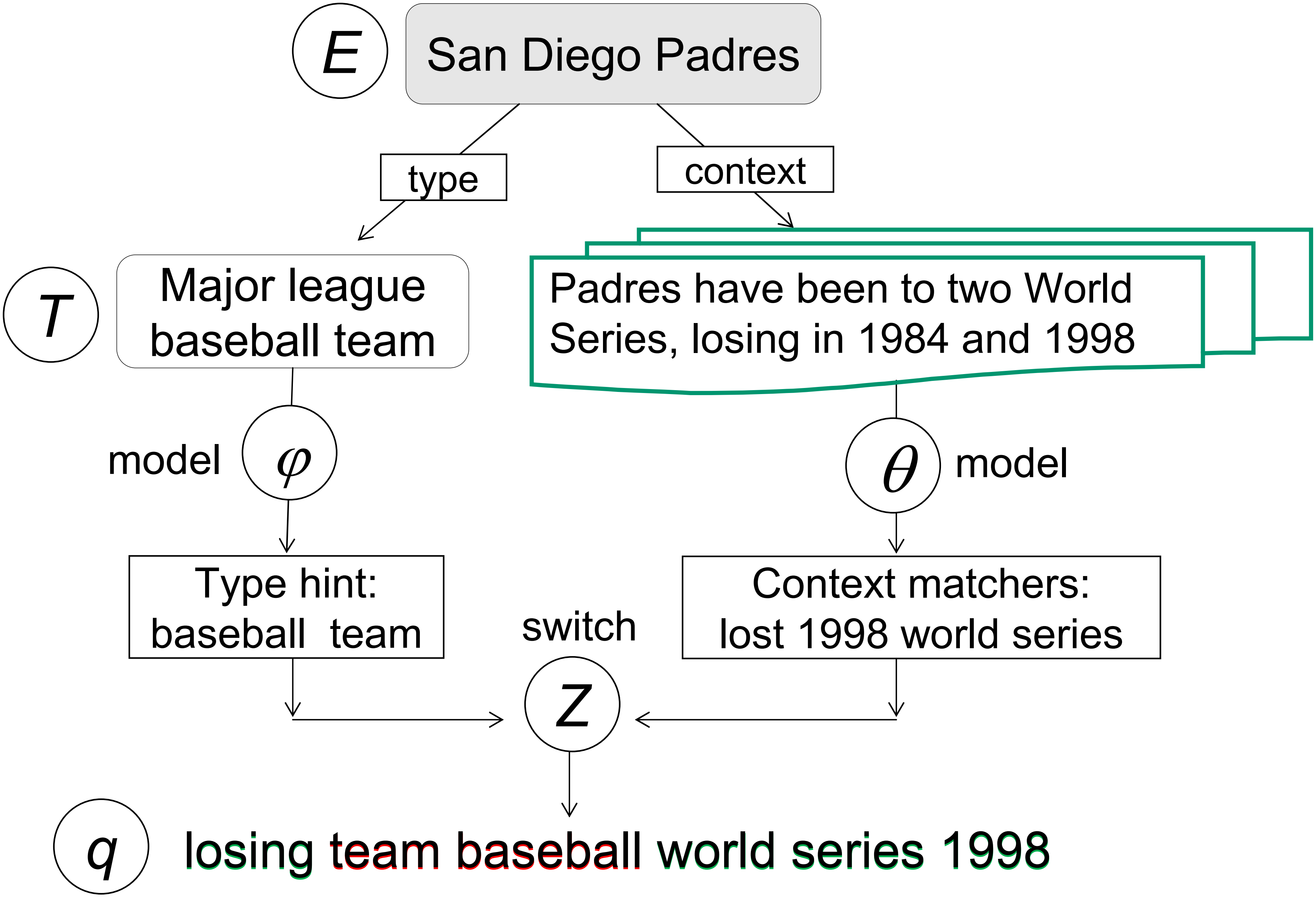
The annotated Web



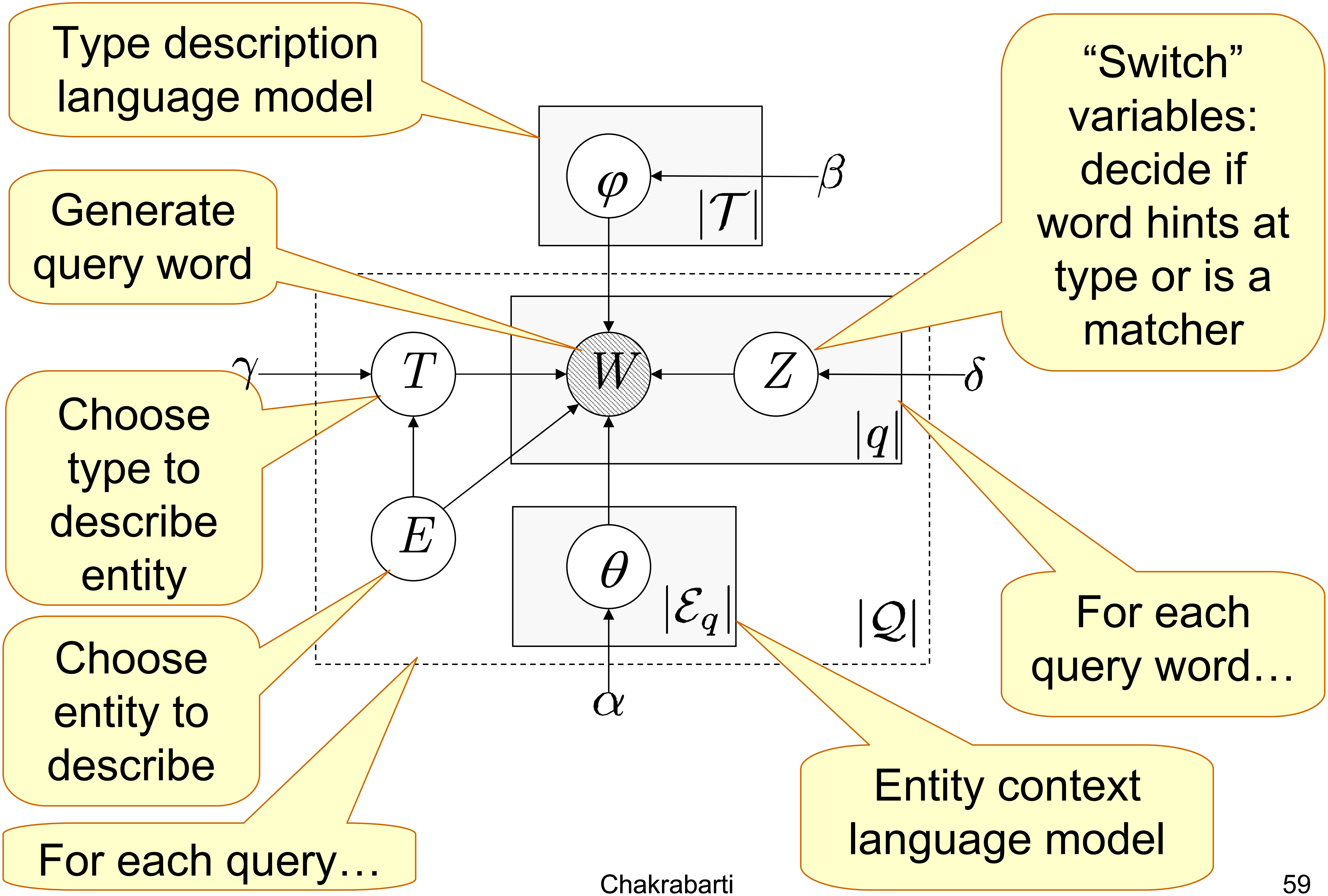
Ranking directed distant supervision

- End goal is to rank entities, not to segment and annotate query
- Although those by-products help
  - Diagnose ranking performance
  - Establish an interpretation dialog with user
- Training input
  - Set of telegraphic queries w/ implicit target types
  - For each query, relevant and irrelevant entities
  - *No manually segmented+annotated queries*
- At test time, only telegraphic query
  - Ranked list of entities, MAP, NDCG, MRR, etc.

Generate query from entity



Generative framework [SC 2013]



Generative objective

Sum over t, z uncertainty

$$
\begin{aligned}
&\Pr(e|\vec{q}) \propto \Pr(e,\vec{q}) \\
&= \Pr(e)\Pr(\vec{q}|e) = \Pr(e)\sum_{t,\vec{z}}\Pr(\vec{q},t,\vec{z}|e) \\
&= \Pr(e)\sum_{t,\vec{z}}\Pr(t|e)\underline{\Pr(\vec{z}|e,t)}\Pr(\vec{q}|e,t,\vec{z})
\end{aligned}
$$

Simplifying approximation

$$
\begin{aligned}
&\approx \Pr(e)\sum_{t,\vec{z}}\Pr(t|e)\underline{\Pr(\vec{z})}\Pr(\vec{q}|e,t,\vec{z}) \\
&= \Pr(e)\sum_{t,\vec{z}}\Pr(t|e)\Pr(\vec{z})\boxed{\Pr(h(\vec{q},\vec{z})|t)}\boxed{\Pr(s(\vec{q},\vec{z})|e)}
\end{aligned}
$$

Hint words depend on t

Selector words depend on e

Discriminative framework

Models entity prior

Compatibility between hint words and type

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

Hints

Matchers

Feature vector given query, entity, type, switches

Models type prior $\Pr(t|e)$

Compatibility between matchers and snippets that mention e

Given q, score of response e is:

$$\max_{t, \vec{z}} \lambda \cdot \phi(q, e, t, \vec{z})$$

Unlike generative, here we score e under the *most favorable* interpretation

Ranking model trained by distant supervision

Discriminative objective

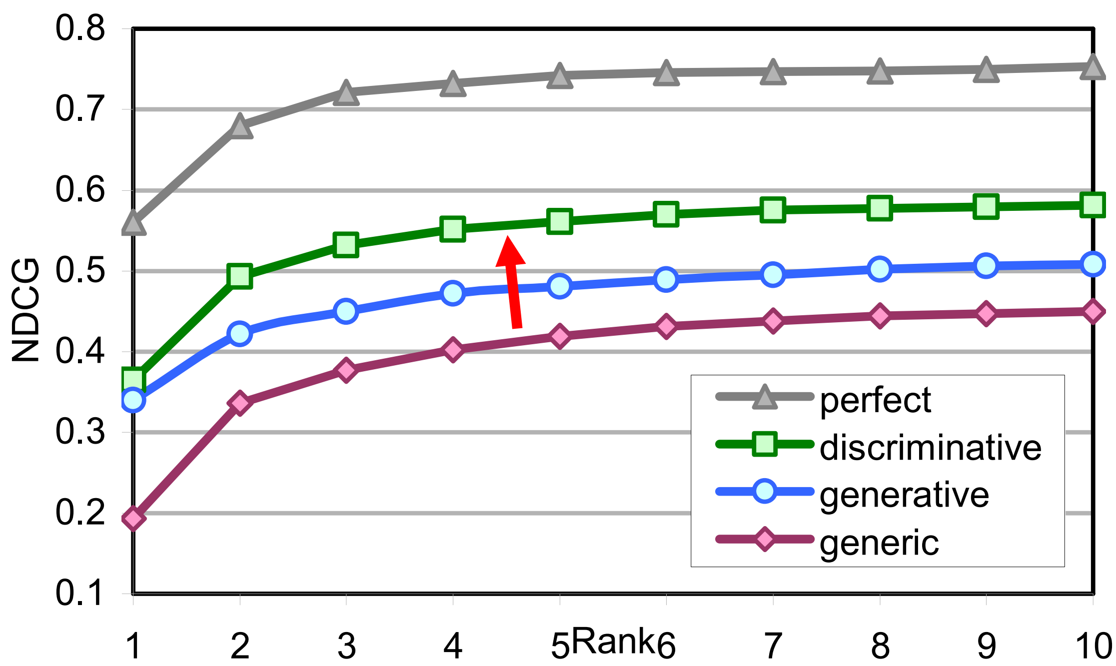
- RankSVM fits λ so that score of each relevant training entity e^+ is larger than score of each irrelevant training entity e^-

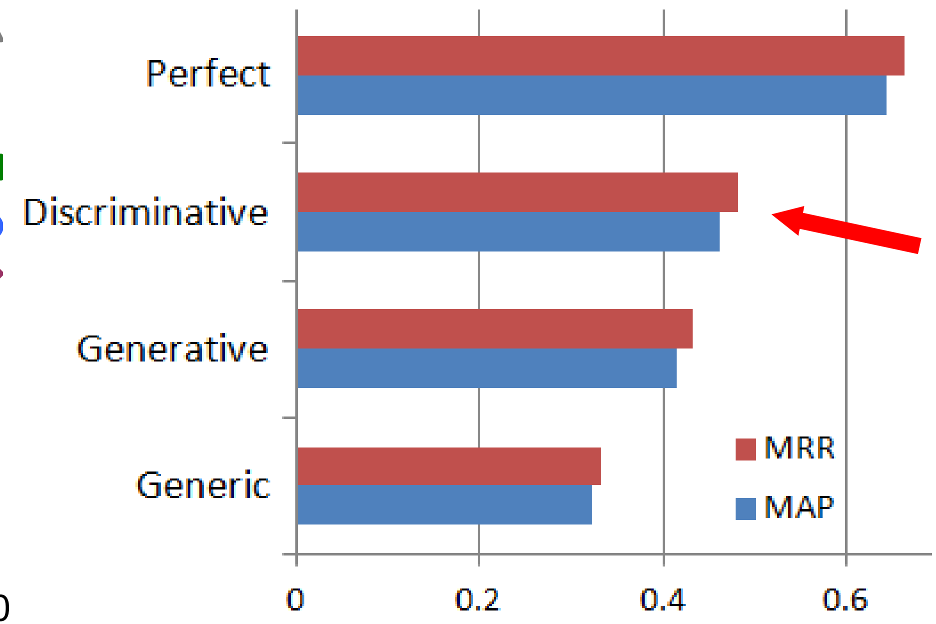
$$\forall q, e^+, e^- : \quad \max_{t,\vec{z}} \lambda \cdot \phi(q, e^+, t, \vec{z}) \geq$$

$$1 + \max_{t,\vec{z}} \lambda \cdot \phi(q, e^-, t, \vec{z})$$

- Problem: lhs max destroys convexity
  - Appears unavoidable
- Multiple instance learning: replace max with convex combination
- Entropy annealing protocols for optimizing λ

Generic < Generative < Discrim. < “Perfect”

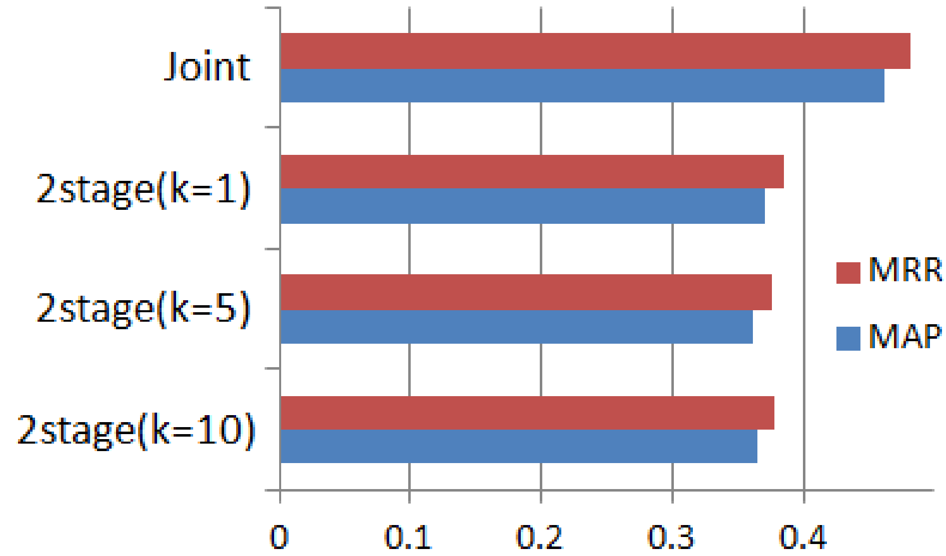


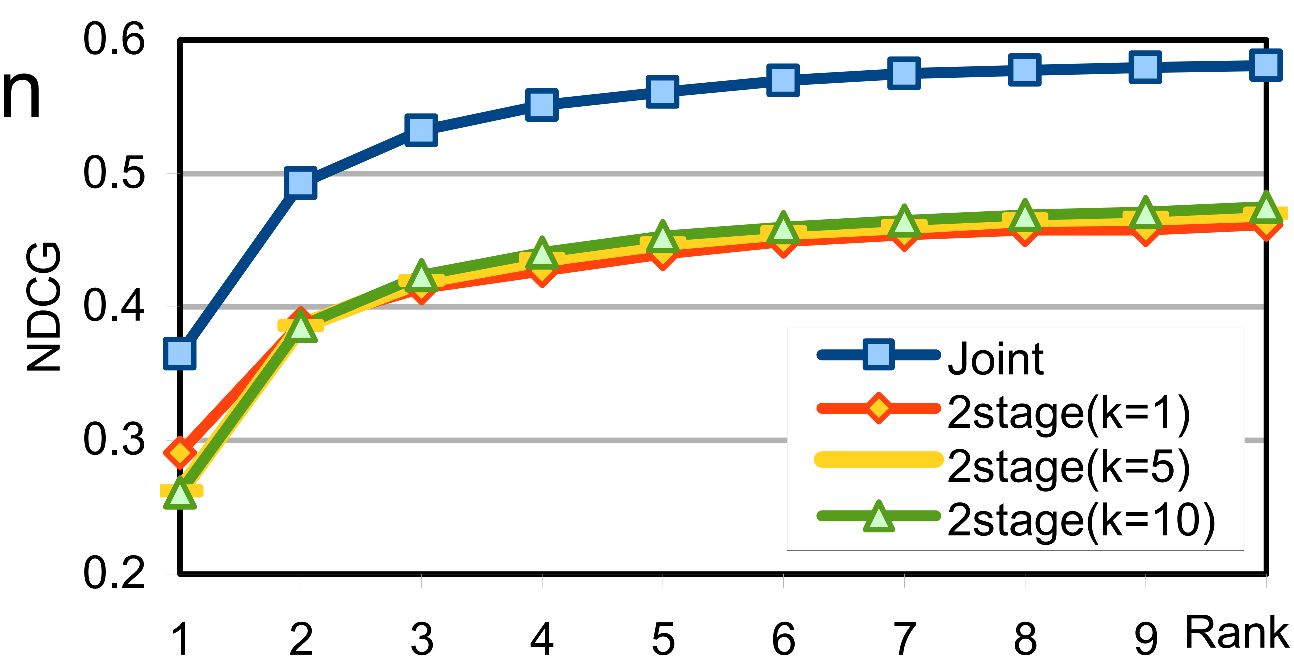


- Generic = root type + keyword query
- “Perfect” = human translated query to semistructured form with type and selectors
- Generative significantly better than generic (lower)
- Discriminative significantly better than generative

Joint better than two-stage

- State of the art target type predictor
  - Does not use corpus information
- Pick top k types to improve type recall
- Launch type-restricted query on annotated corpus
- Significantly worse than joint type prediction and ranking

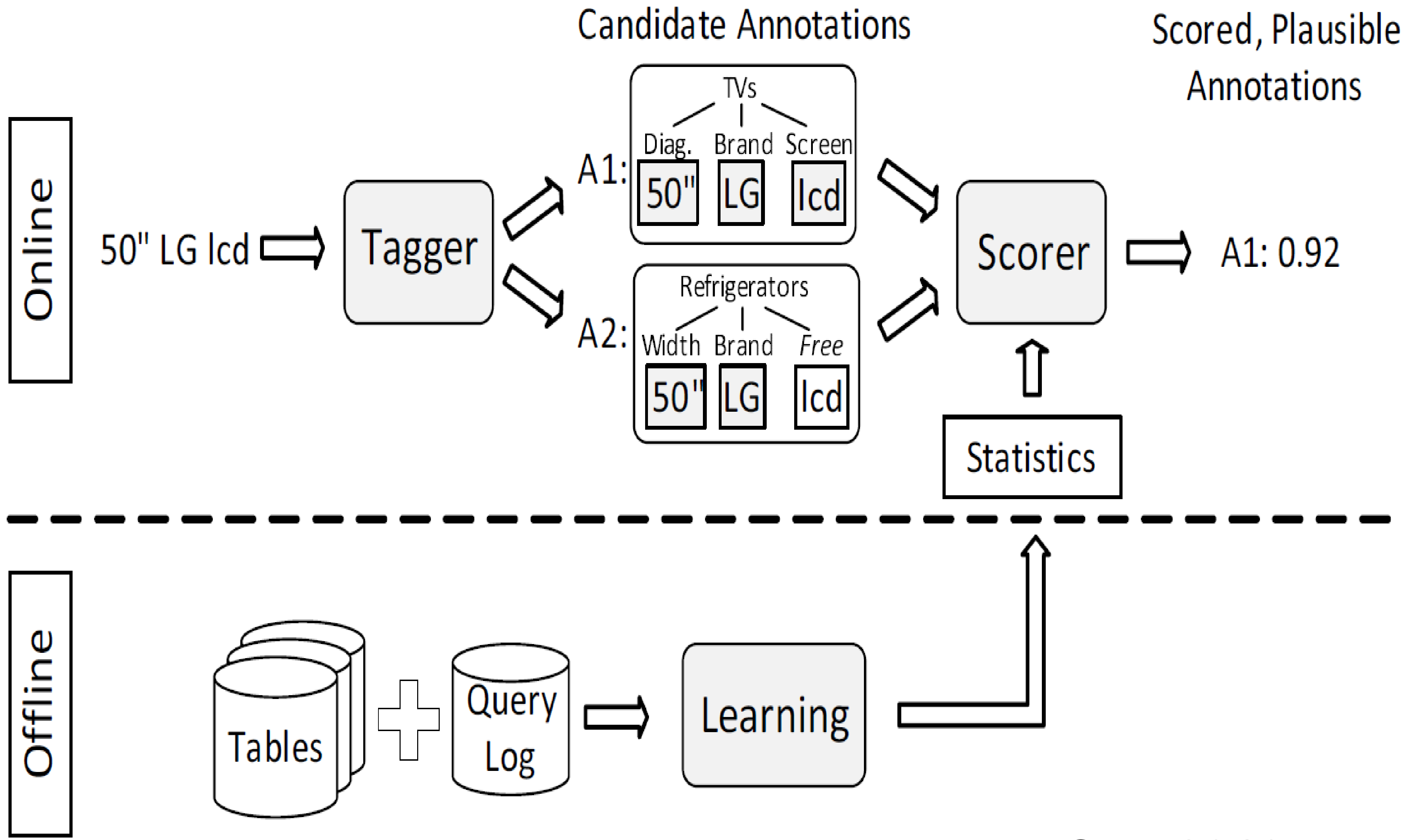




Richer schema [SPT 2010]

- Thus far we have considered these relations
  - Entity hasType Type
  - Type subClassOf Type
  - Entity mentionedAt (document, position)
  - Entity or Type mentionedIn query
- Assumes little, therefore applies broadly
- In some cases, can assume more of the knowledge base
  - E.g., product catalogs
  - Telegraphic queries like `50 in lg lcd`

50 in lg lcd

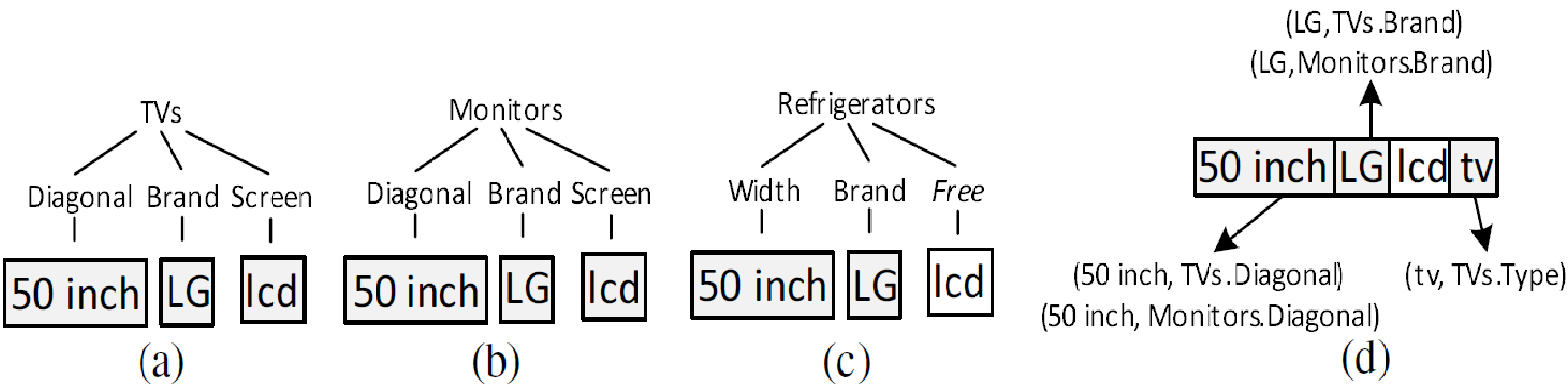


[SPT 2010]

Formal setup

- Tables T $=\{T_1,\ T_2,\ \ldots\ T_\tau\}$
- T.A $=\{T.A_1,\ T.A_2,\ \ldots\ T.A_\alpha\}$ are attributes (columns) of one table
- Domain of values for column c is $T.A_c$.V
- Query seeks row/s from one table
  - Select conditions expressed via keywords
- Query annotation consists of
  - A table T
  - Set $\{t_i,\ T.A_i\}$ (word t_i matches cell in column A_i)
  - Some words may remain unbound or "free"

Example



(a) $S_1 = \langle \text{TVs}, \{(\text{50 inch, TVs.Diagonal}), (\text{LG, TVs.Brand}), (\text{lcd, TVs.Screen})\}, \{\}\rangle$

(b) $S_2 = \langle \text{Monitors}, \{(\text{50 inch, Monitors.Diagonal}), (\text{LG, Monitors.Brand})\}, (\text{lcd, Monitors.Screen}), \{\}\rangle$

(c) $S_3 = \langle \text{Refrigerators}, \{(\text{50 inch, Refrigerators.Width}), (\text{LG, Refrigerators.Brand})\}, \{\text{lcd}\}\rangle$

Goal: *Efficient* generation of the *most plausible* interpretations

Score for plausibility

Prob. of one interpretation

Choose table+attrib template

Generate annotated and free text

$$\Pr(S_i) = \boxed{\Pr(T.\mathcal{A}_i)}\,\Pr(\{\mathcal{AT}_i, \mathcal{FT}_i\}|T.\mathcal{A}_i)$$

$$\approx \Pr(T.\mathcal{A}_i)\,\Pr(\mathcal{AT}_i|T.\mathcal{A}_i)\,\Pr(\mathcal{FT}_i|T.\underline{\mathcal{A}_i})$$

$$= \Pr(T.\mathcal{A}_i)\,\boxed{\Pr(\mathcal{AT}_i|T.\mathcal{A}_i)}\,\Pr(\mathcal{FT}_i|T)$$

Assume conditionally independent

Free text does not depend on specific attributes

- Fraction of table rows in which annotated tokens appear in the specified columns
- Another (naïve Bayes) approximation: word events are independent

Template probability $\Pr(T.\mathrm{A}_i)$

- Probability of a query targeting particular tables and attributes

Query may be generated as free text, or…

$$\Pr(q) = \Pr(OLM)\Pr(\mathcal{FT}_q | OLM) + \sum_{S_i \in \mathcal{S}_q} \Pr(T.\mathcal{A}_i)\Pr(\{\mathcal{AT}_i, \mathcal{FT}_i\} | T.\mathcal{A}_i)$$

… sum over templates

Probability of template

Probability of generating query text from template

Parameter estimation

- Equivalent simpler notation:

Sum over all tables…

Prob. of picking open language model

Prob. of picking j^{th} column subset from i^{th} table

$$\Pr(q) = \beta_q \pi_0 + \sum_{i=1}^{|\mathcal{T}|} \sum_{\mathcal{A}_j \in \mathcal{P}_i} \alpha_{qij} \pi_{ij}$$

Prob. of generating query from open language model

Prob. of generating query from j^{th} column subset from i^{th} table

… sum over all possible combinations of attributes of T_i …

- Laborious to exhaustively annotate queries
- Above solved using EM

Semantic query annotation results

- 50k queries over 7 tables
- SAQ = proposed technique
- Low/med = reject thresholds
- IG = Intelligent greedy: maximize part of query explained by table (and not OLM)
- 0, 1, 5: Acceptance threshold for cover

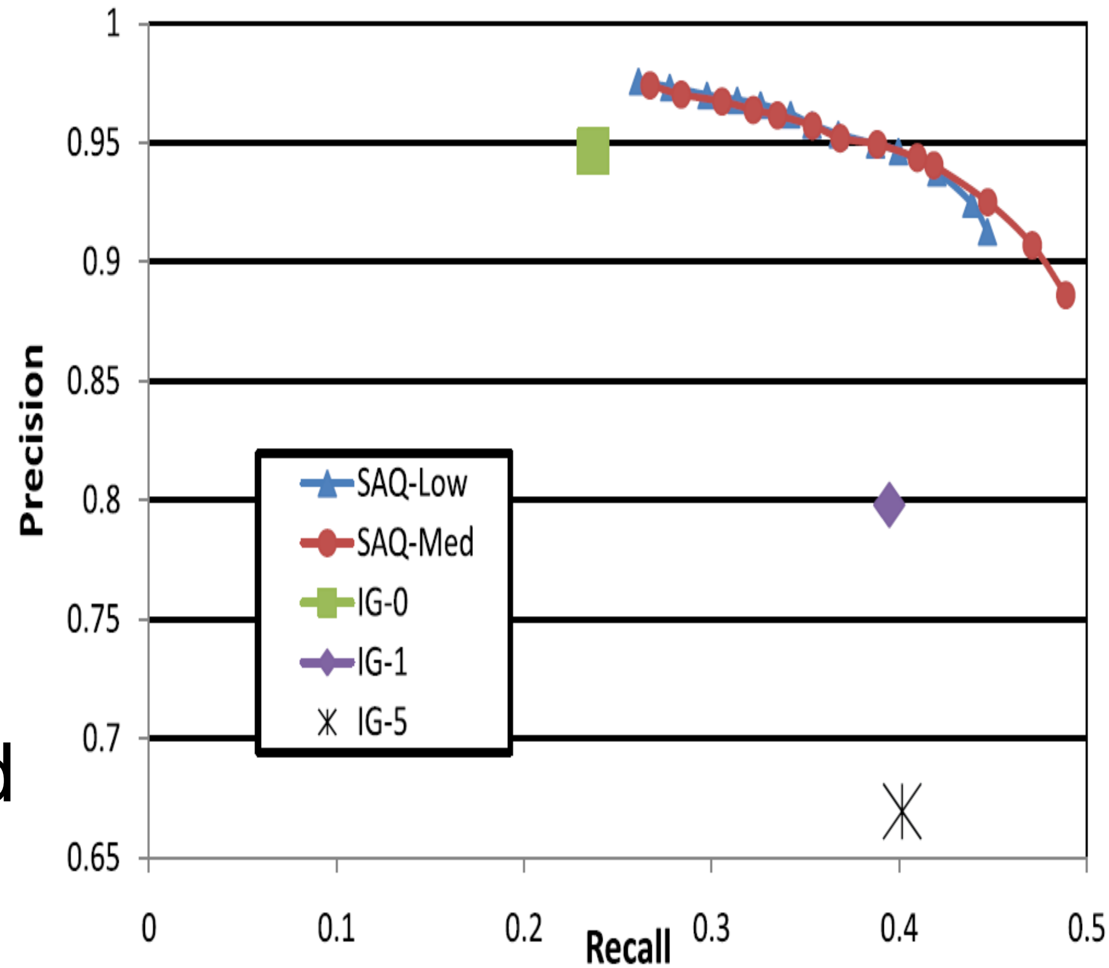


Table confusion matrix

Predicted→ Actual↓	Cameras	Camcorders	Lenses	Accessories	OLM
Cameras	92%	2%	4%	2%	0%
Camcorders	4%	96%	0%	0%	0%
Lenses	2%	0%	94%	4%	0%%
Accessories	13%	3%	3%	81%	0%
OLM	7%	2%	0%	1%	90%

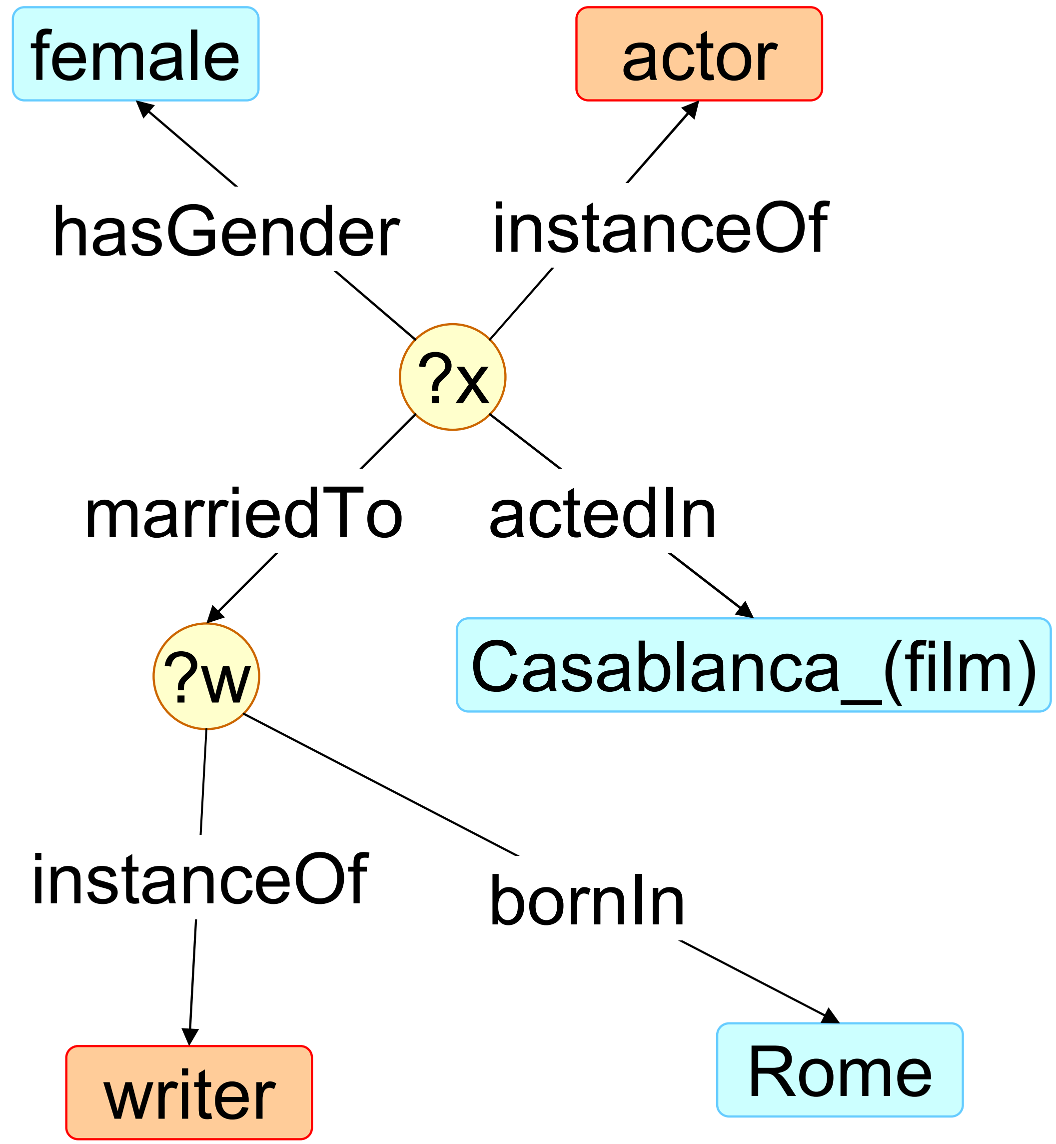
Interpreting queries on linked data

- Linked open data reaching the stage where we should be able to answer
  - Which female actor played in Casablanca and is married to a writer who was born in Rome?
- Not that telegraphic
- Based on YAGO, IMDB, Freebase and other s-p-o data

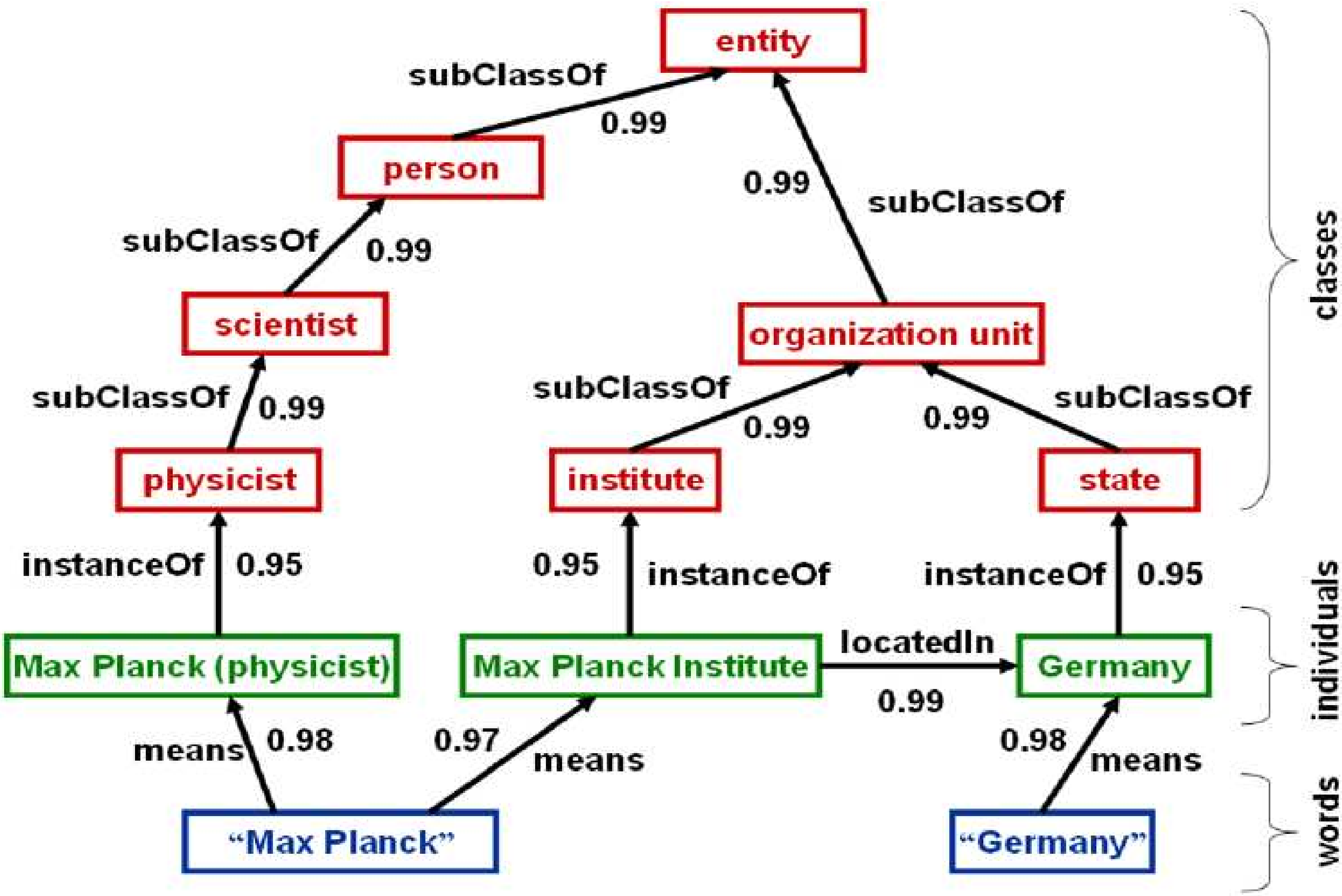
?x hasGender female
?x instanceOf actor
?x actedIn Casablanca_(film)
?x marriedTo ?w
?w instanceOf writer
?w bornIn Rome

Interpreting queries on linked data

- Linked open data reaching the stage where we should be able to answer
  - Which female actor played in Casablanca and is married to a writer who was born in Rome?
- Execute query graph fragment on linked data graph



A sample view of linked data graph



Workshops on question answering on linked data (QALD)
Workshop on interacting with linked data

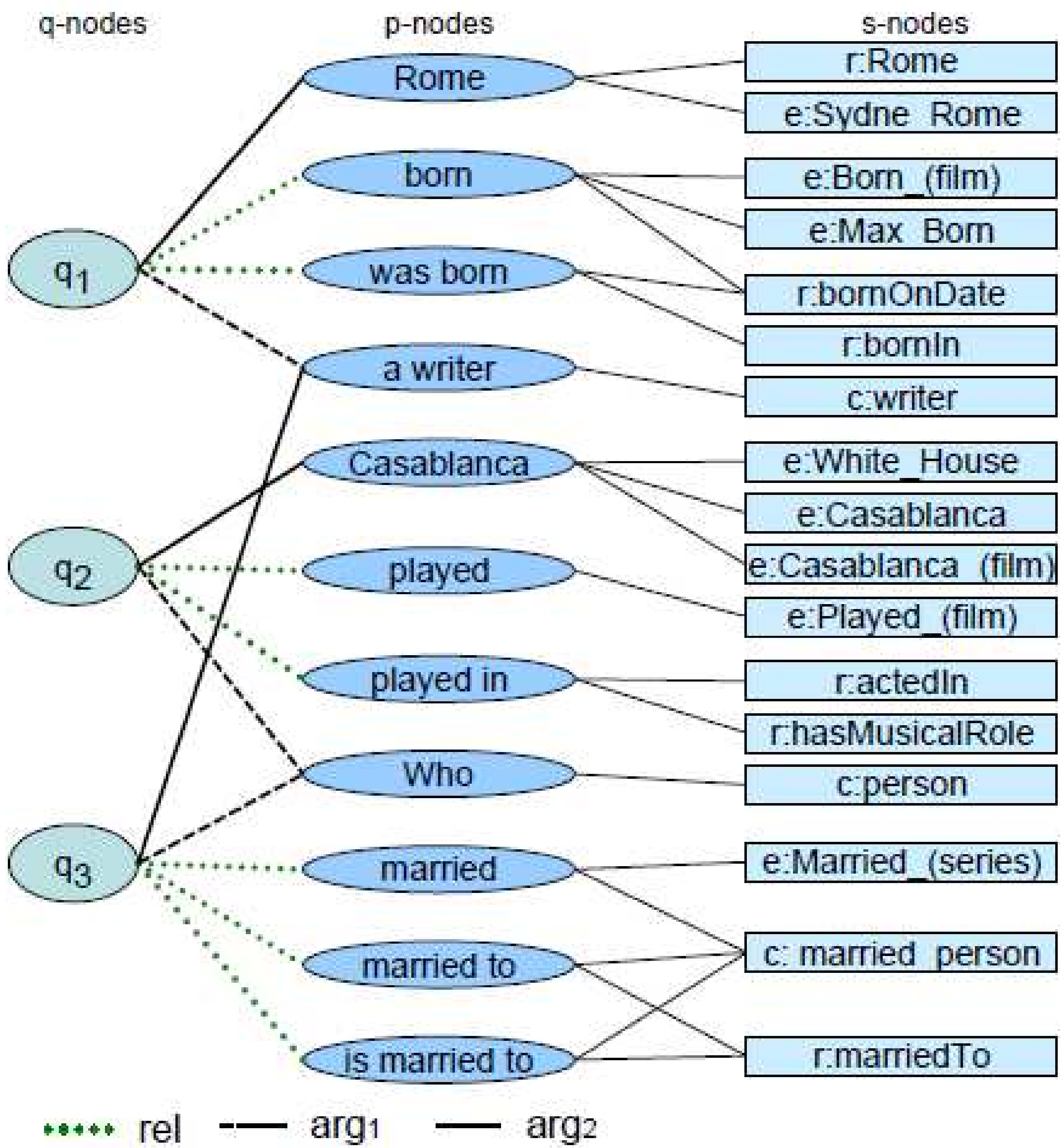
Supporting data

- Hierarchy of types, attached entities, as in telegraphic query interpretation discussed earlier
- Sample descriptions of both entities and types
  - {‘Rome’,‘eternal city’} $\rightarrow$ *Rome*
  - {‘Casablanca’} $\rightarrow$ *Casablanca_(film)*
  - {‘play’,‘star in’,‘act’,‘leading role’} $\rightarrow$ `actedIn`
  - {‘married’, ‘spouse’,‘wife’} $\rightarrow$ `marriedTo`
- In general, many-to-many mapping

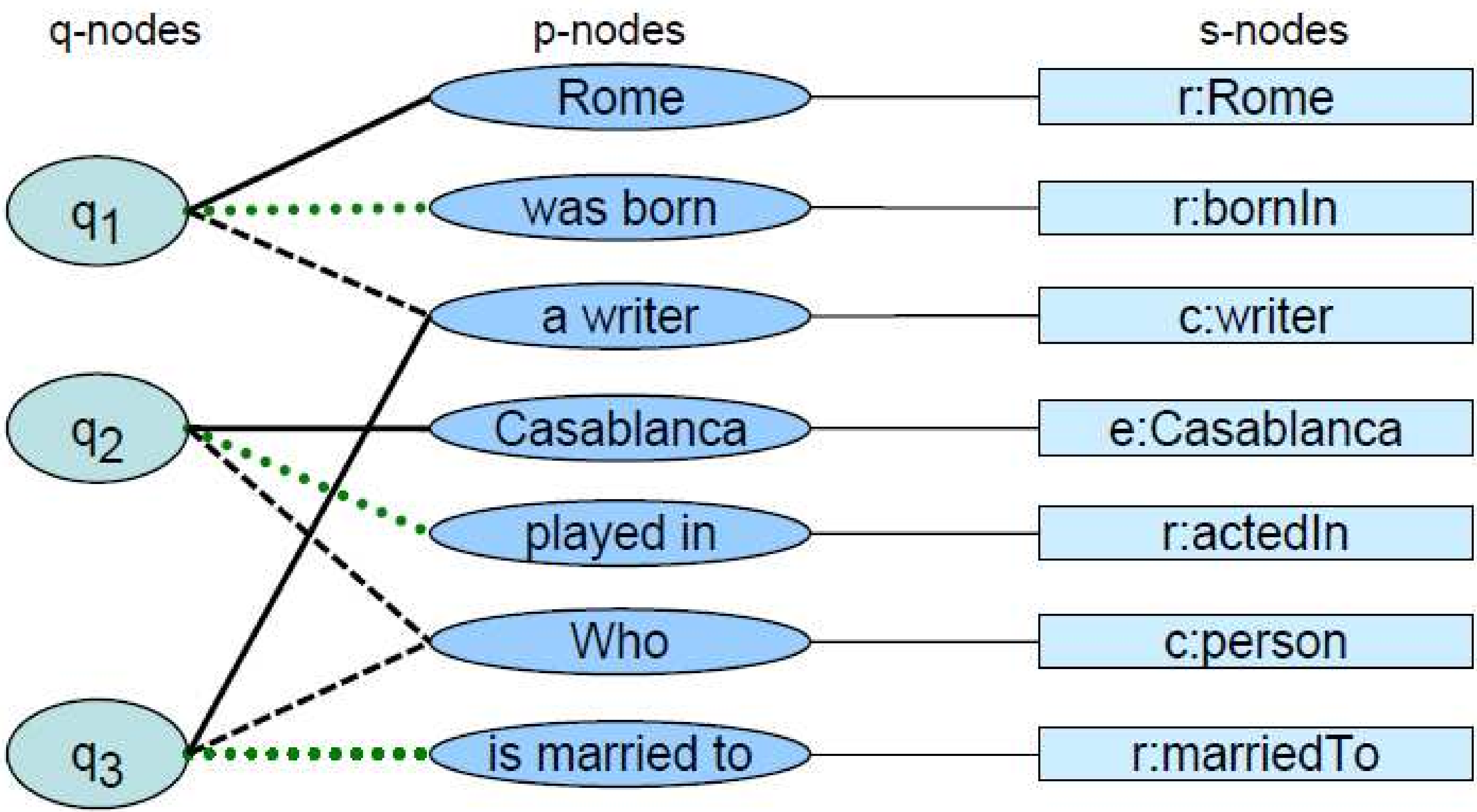
Interpretation stages [WEBRTW 2012]

- Detecting phrases corresponding to semantic items like *who*, *played in*, *movie*, *Casablanca*
- Mapping phrases to semantic items
  - *played in* may mean `actedIn` or `playedForTeam`
  - *Casablanca* may mean *`Casablanca_(film)`* or *`Casablanca,_Morocco`*
- Triple generation and joint disambiguation
  - If *Casablanca* means film, prefer `actedIn` over `playedForTeam`
- Variable grouping (*which* vs. *who*) and query generation

Joint disambiguation, input

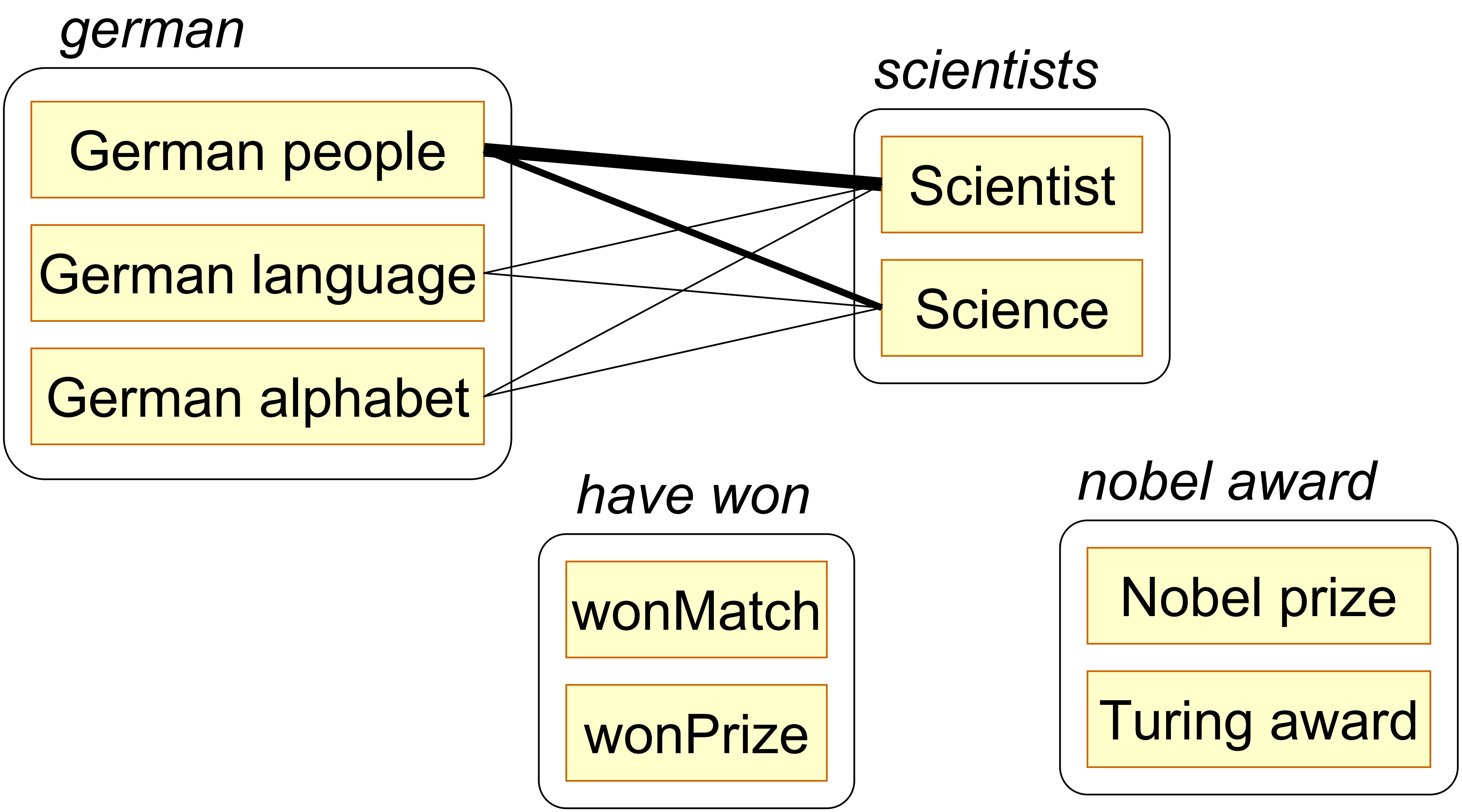


Joint disambiguation, output



Keyword-based structured queries [PIW 2010]

german scientists who have won nobel award



High level solution approach

- Represent local choices using decision variables
  - X_i = phrase i is matched to some semantic node
  - Y_{ij} = phrase i is matched to semantic node j
  - Z_{kl} = semantic nodes (entities or types) $k,\ l$ are both matched (so their mutual coherence matters in the objective)
  - … and many other structural constraints
- Use a general mixed integer linear program to set the decision variables
- Execute best interpretation of query

Geospatial intent

Associating queries and documents with locations

- Lat-long coordinates: an important special case of an "entity"
- Easily observed user attribute during search
- But not easy to associate with Web page
  - Not necessarily predicted well by location from where page is hosted
  - Better: locations from which page is *accessed*
- If both possible, can help personalize search
  - Include location-based features during ranking

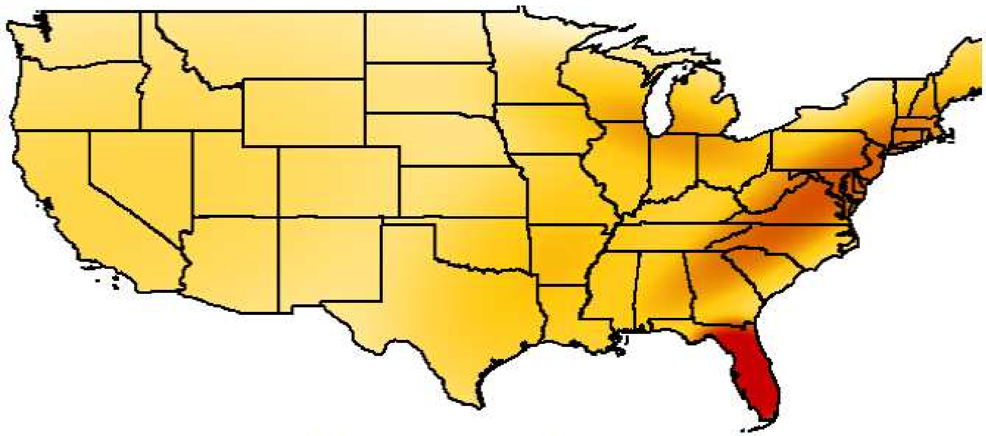
Access location heat-map [BRWY 2011]

- Commercial Web browser add-on records
  - Browser IP address → visitor lat-long
  - URL of pages visited
- Data collected over three months
- Any URL with more than 50 visits included
- A location may be "hot" for a URL because of two factors
  - High population density
  - Actual interest in the URL
  - How to separate these effects?

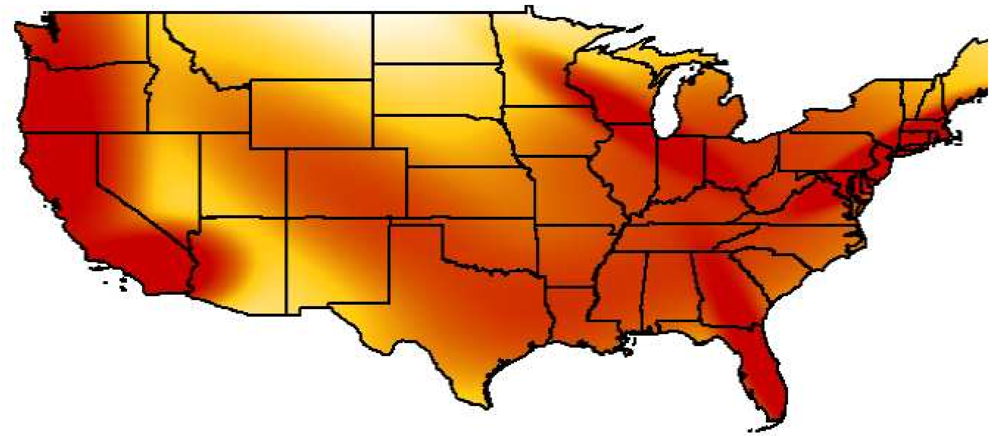
Mixture of 2d Gaussians

- Kernel density estimation, n components
- i^{th} component has 2d location μ_i, relative weight w_i, and covariance Σ_i
  - Large w_i near large population densities
- Probability (density) of a location given a URL is
$$\Pr(x|\text{URL}) = \sum_{i=1}^{n} w_i \mathcal{N}(x|\mu_i, \Sigma_i)$$
- $\mathcal{N}$ represents a 2d Gaussian distribution
- For each URL, fit $\{w_i, \mu_i, \Sigma_i\}$ using training data (set of access lat-long) $\{x\}$
- Also fit a model to union of all locations

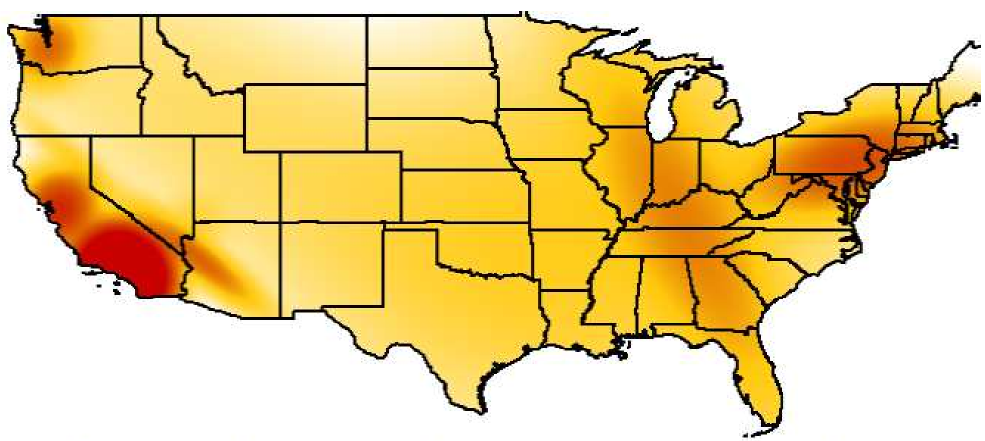
Sample heat-maps



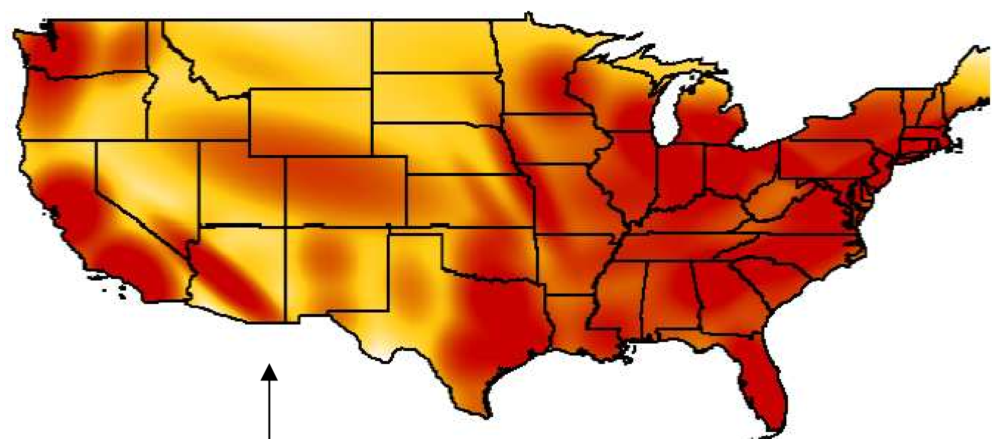
(a) Sarasota Memorial Health, http://smh.com/



(b) Sydney Morning Herald, http://smh.com.au/



(c) Los Angeles Times: Reviews and Recommendations
http://findlocal.latimes.com/

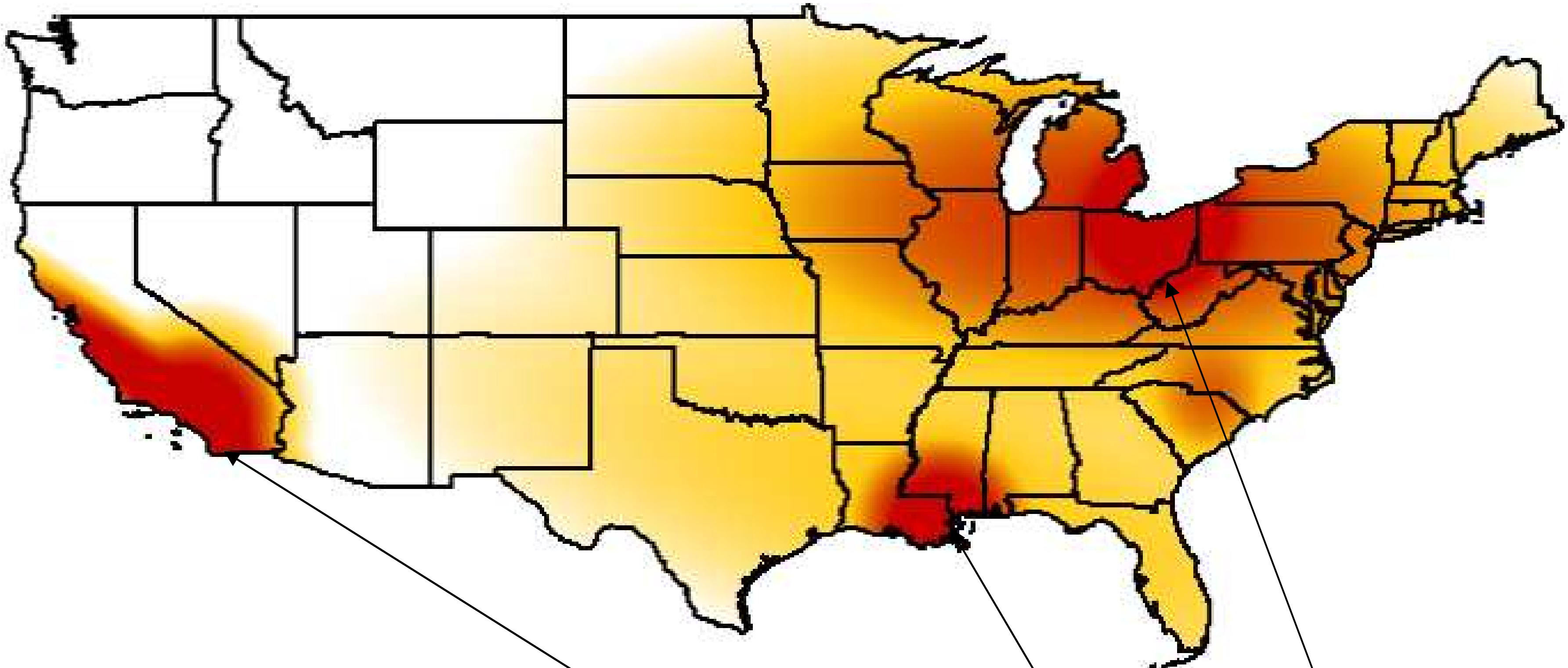


Corroboration: Background model closely reflects population density

Extending from URLs to queries

- Collect locations of all users who issued a particular query
- Associate resulting kernel density with query
  - (Approx) entry of fitted density large $\Rightarrow$ low location sensitivity
  - Large divergence from background model $\Rightarrow$ high location sensitivity
- No smoothing across queries; exact match required
  - May lose out on tail queries (sparse data)
  - Possibly richer models lurking here that integrate spatial density with language models?
- In any case … now we have location heat maps for each query and URL

Heat map for query `rta bus schedule`



Density peaks around California, Louisiana, Ohio

Use of heat maps in ranking

- KL divergence between heat maps of query and URL

$$\mathrm{KL}(M_u || M_q) = \int_x \Pr(x|M_u) \log \frac{\Pr(x|M_u)}{\Pr(x|M_q)} dx$$

- Feature depending on (user location, URL)

$$\Pr(\mathrm{URL}|\mathrm{loc}) = \frac{\Pr(\mathrm{URL}) \Pr(\mathrm{loc}|\mathrm{URL})}{\Pr(\mathrm{loc})}$$

$$\propto n_{\mathrm{URL}} \Pr(\mathrm{loc}|M_{\mathrm{URL}})$$

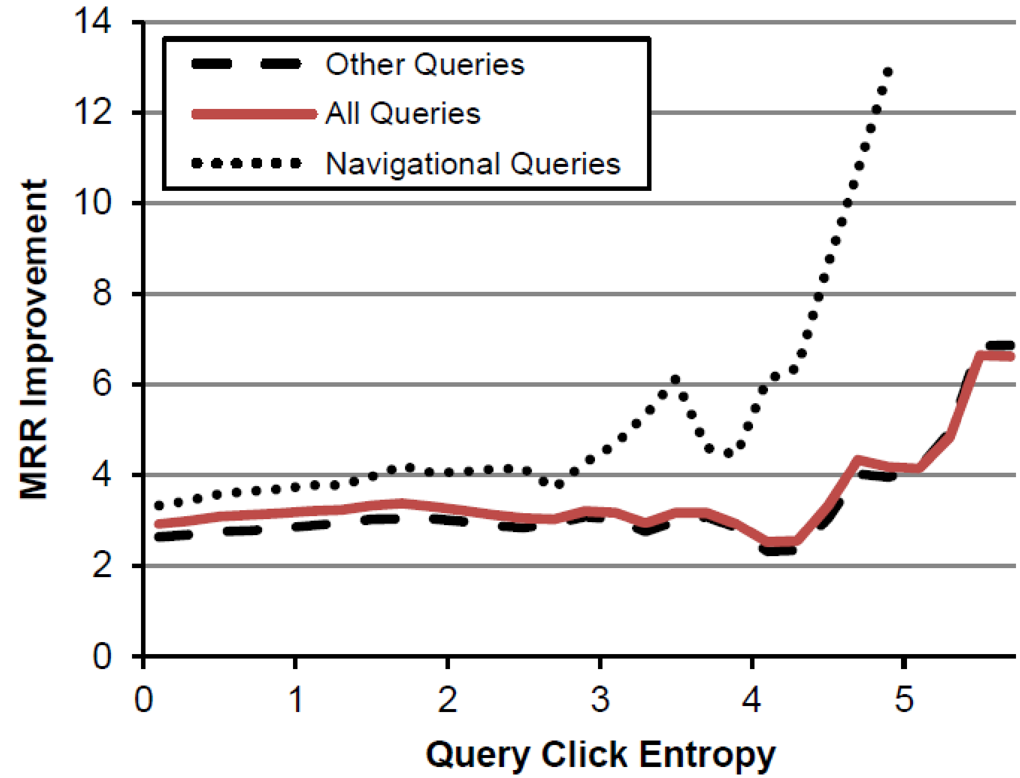
Overall viewing frequency of URL

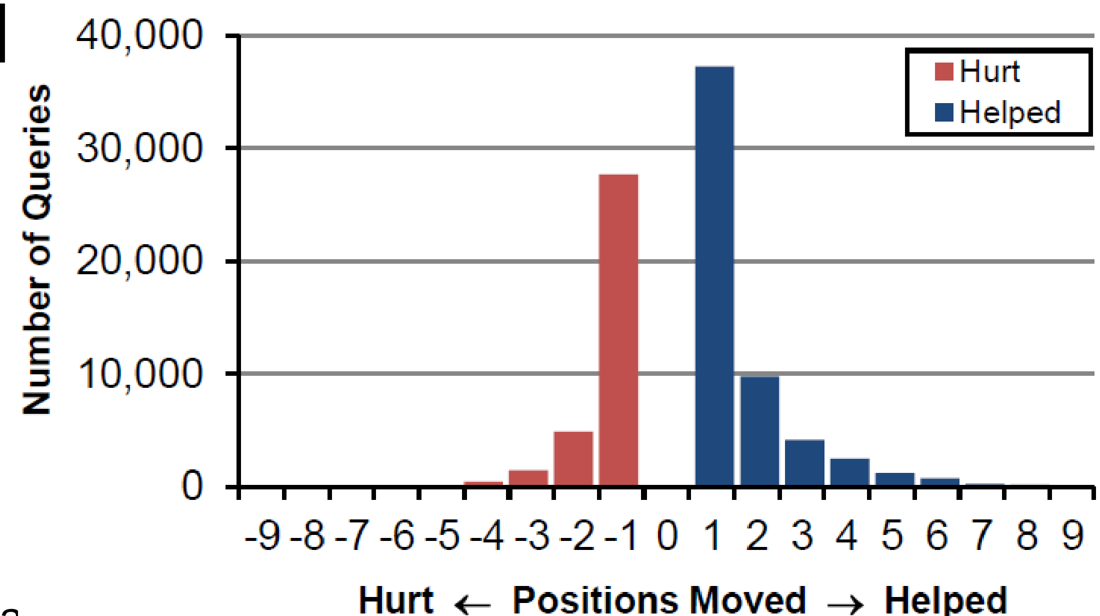
Gaussian mixture model heat map for URL

- Input to feature-based learning to rank algo

Results of heat map based reranking

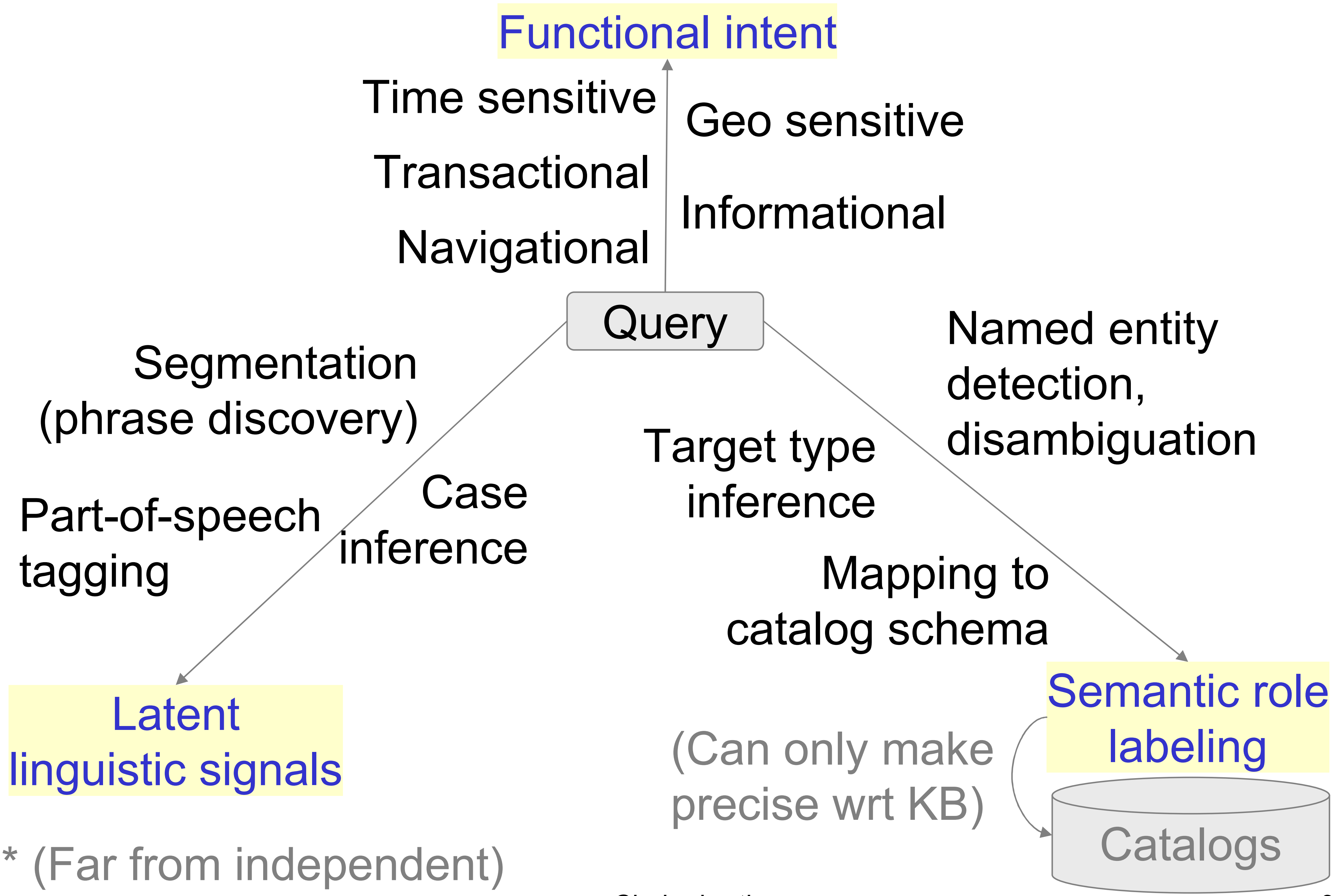
- Navigational queries see best improvements
- Gains directly related to click entropy (many different URLs clicked by different users)
- Reranking within top 10 causes both losses and gains
- Further analysis of failure cases?





Concluding remarks

Query understanding dimensions*



* (Far from independent)

Summary of techniques studied

- Broad query intent classification
- Segmentation and phrase detection
- Entity disambiguation in queries
- Interpreting targeted-type queries
- Semantic annotation per tables and columns
- Interpreting to queries over RDF stores
- The special case of geospatial intent

Key takeaways

- In the beginning, there was TFIDF cosine
- No need to cast queries into any schema
- ~2001—2008: information extraction and integration, entity and relation discovery
- Query habits have not changed (much)
- Bridge needed between telegraphic unstructured queries and increasingly structured corpus + knowledge base combo
- Verticals are low-hanging fruit, but also helps to push tail queries

Final comments

- Source relatively unstructured (plain text) $\Rightarrow$ not much interpretation to do $\Rightarrow$ unified retrieval and ranking
- Source somewhat structured (text annotated with entity mention spans) $\Rightarrow$ range of choices
  - First interpret then execute
  - More unified joint interpretation and ranking
- Triple store curated from unstructured source (YAGO): almost exclusively 2-phase