International Workshop on Search and Mining User-Generated Contents
@ International Conference on Information and Knowledge Management
Toronto, Ontario, Canada, 2010

Entity-Relationship Query over Wikipedia

Xiaonan Li [1], Chengkai Li [1], Cong Yu [2]

[1] University of Texas at Arlington
[2] Yahoo! Research

Motivation

A business analyst is investigating the development of Silicon Valley. She is looking for:

Silicon Valley Companies **founded by** Stanford graduates

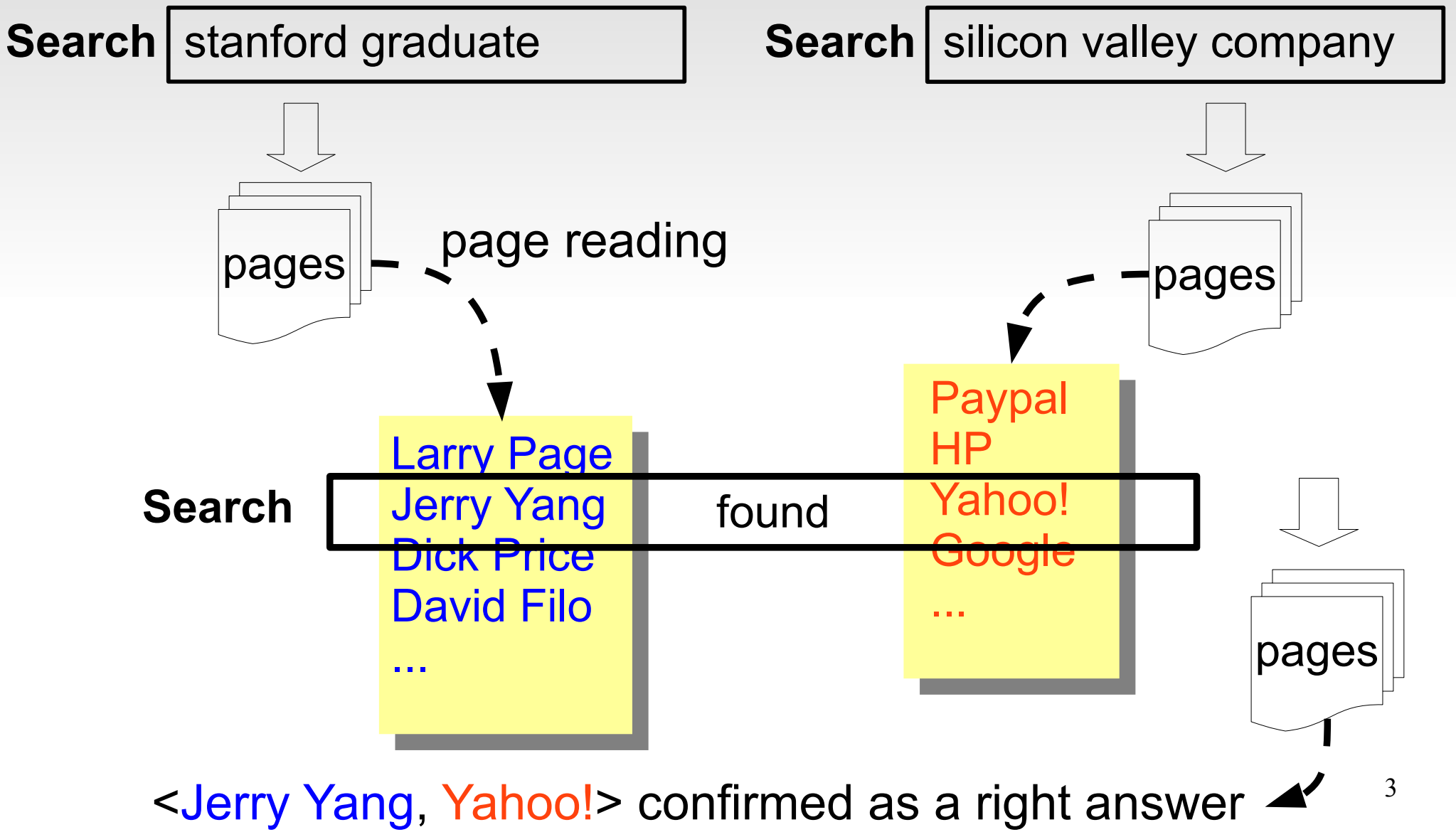
Yahoo! — **Jerry Yang**

Yahoo! — **David Filo**

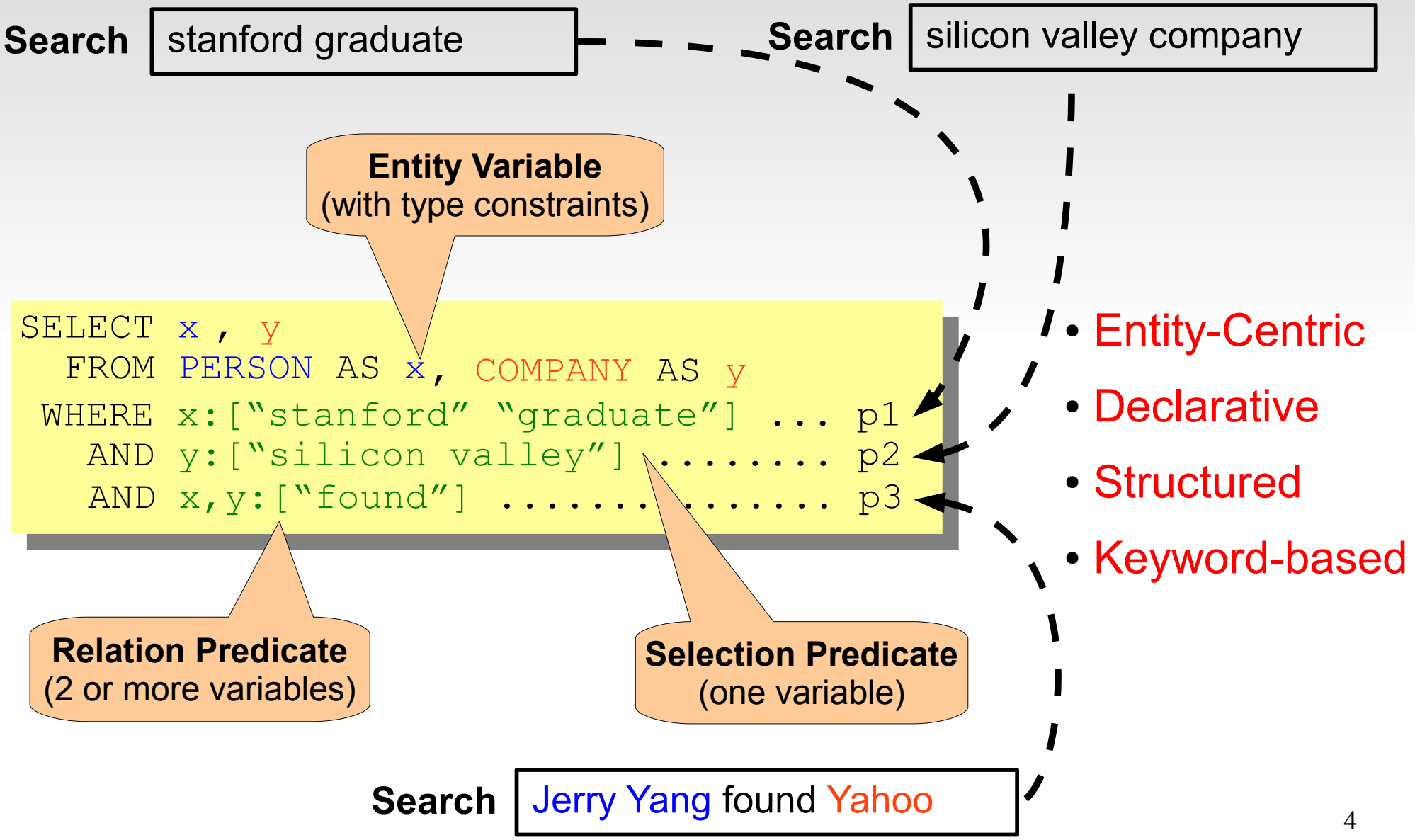
Google — **Larry Page**

HP — **David Packard**

Pain with Search Engine



Entity-Relationship Query



Query Answer

<x:Jerry Yang, y:Yahoo!> is a query answer, **if**

`x:["Stanford" "graduate"] ... p1`

Stanford University graduates Jerry Yang and …

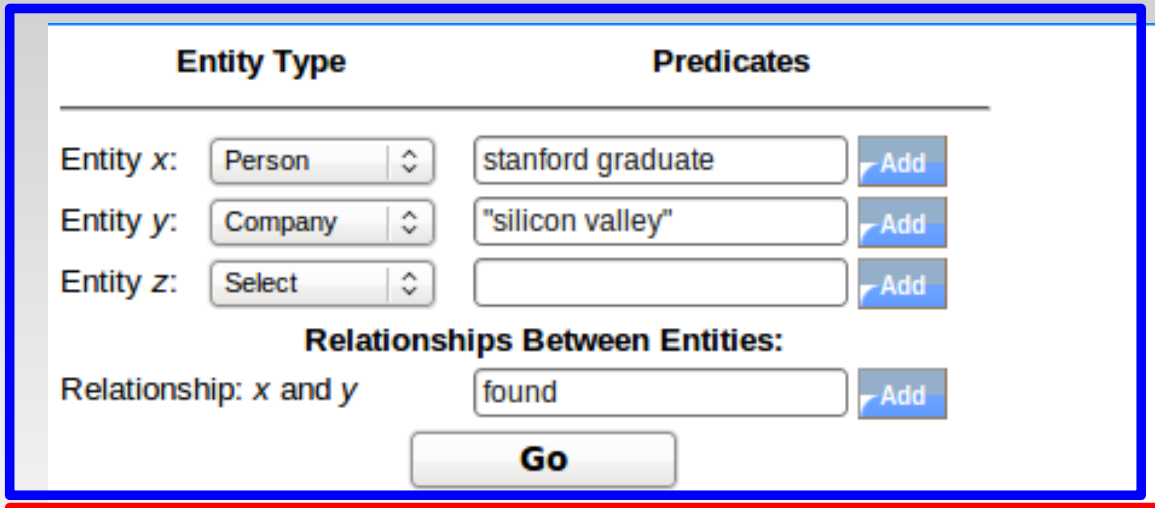
`y:["Silicon Valley"] ..... p2`

… a senior manager at Yahoo! in Silicon Valley.

`x,y:["found"] ..... p3`

Jerry Yang co-founded Yahoo!.

co-occurrence contexts as evidence



Type in the query

Browse the answers

Total Answers: **25** x=18 y=14 Time 22 ms

1 Jerry Yang Yahoo!

- In January 1994, Stanford graduate students Jerry Yang and David Filo created a website named Jerry's Guide to the World Wide Web. (see all 6)
- Imran is the son of Nuzhat Khan and Anil Pal, who works as a senior manager at Yahoo in Silicon Valley. (see all 7)
- Jerry Yang co- founded Yahoo (see all 3)

2 Scott McNealy Sun Microsystems

- On February 12, 1982 Vinod Khosla, Andy Bechtolsheim, and Scott McNealy , all Stanford graduate students, founded Sun Microsystems. (see all 4)
- The AmBAR was founded in 2002 by a group of experienced technology entrepreneurs and business professionals from the Silicon Valley companies and venture capital firms such as Sun Microsystems , Intel Capital, and Draper Fisher Jurvetson. (see all 8)
- Vinod Khosla, a fellow graduate of Stanford who was an early employee at Daisy Systems Corporation convinced Bechtolsheim along with Scott McNealy to found Sun Microsystems in order to build the Sun1/ 100 workstation. (see all 3)

Ranking

0.8 * 0.7 * 0.8 = 0.448

answer	x	y	p1	p2	p3	Ranking score
t1	Jerry Yang	Yahoo!	0.8	0.7	0.8	**0.448**
t2	Larry Page	Google	0.6	0.5	0.6	**0.180**
t3	Scott McNealy	Cisco	0.9	0.8	0.2	**0.144**
t4	Bill Gates	IKEA	0.3	0.1	0.2	**0.006**

The **predicate score** is aggregated from contexts / evidence

Predicate Scoring

Baseline 1 [COUNT]: number of co-occurrence contexts

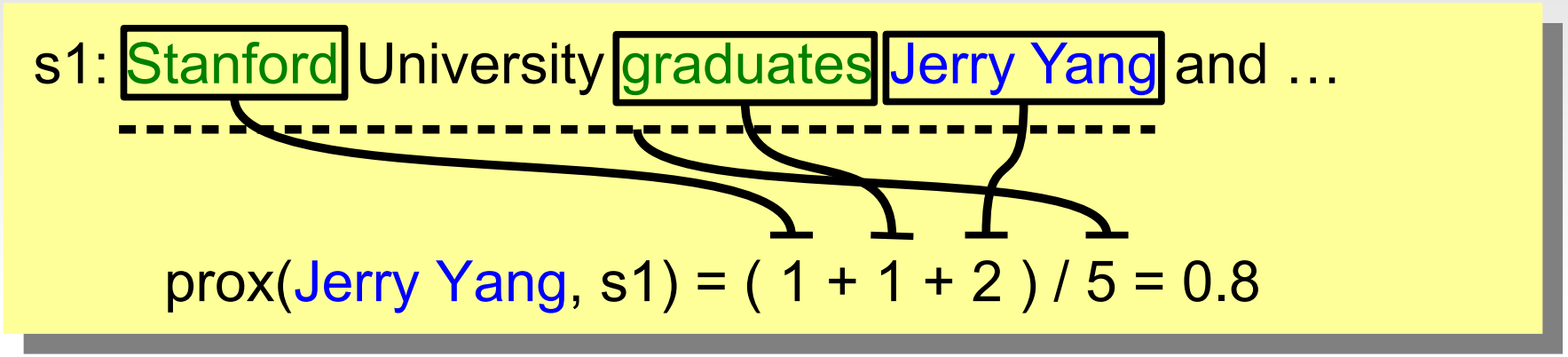
$$F_p(t) = \sum_{s \in \Phi_p(t)} 1 = |\Phi_p(t)|$$

However, contexts are different from each other.
We exploit **positional features** for refined evaluation of contexts.

proximity | ordering pattern | mutual exclusion

Feature 1: Proximity

`x:["Stanford" "graduate"] ... p1`



Higher proximity indicates more reliable evidence

Baseline 2 [PROX]: weight each contexts by proximity

$$F_p(t) = \sum_{s \in \Phi_p(t)} prox(t, s)$$

Feature 2: Ordering Pattern

- x~PERSON, s~Stanford, g~graduate
- 6 patterns: xsg,xgs, sxg, gxs, sgx, gsx

Stanford University graduates Jerry Yang and … (4 times)

… Stanford graduates Larry Page … (2 times)

f(sgx)=(4+2)/(4+2+3)=0.67

Frequent patterns indicate reliable evidence.

f(sxg)=(3)/(4+2+3)=0.33

A professor at Stanford University, Colin Marlow had a relationship with Cristina Yang before she graduated … (3 times)

Feature 3: Mutual Exclusion

xgs

s4: After Ric Weiland graduated from Stanford University, Paul Allen and Bill Gates hired him in 1975 ...

gsx gsx

*The most proximate entity as **representative***

	Patten	Entities
colliding patterns	xgs	Ric Weiland
	gsx	Paul Allen, Bill Gates

Assumption: **only one of the colliding patterns is effective.**

Which one?

Feature 3: Mutual Exclusion (2)

s4: After Ric Weiland graduated from Stanford University, Paul Allen and Bill Gates hired him in 1975 ...

context (Ric Weiland) = 4
context (Paul Allen) = 2

credit (xgs, s4) = 4 / (4+2) = 0.67
credit (gsx, s4) = 2 / (4+2) = 0.33

The pattern represented by more prominent entity (thus higher credit) is more likely to be effective.

Baseline 3 [MEX]**:** weight each contexts by *credit* (effectiveness)
(*for contexts without collision, the credit(o,s)=1*)

$$F_p(t) = \sum_{o \in O_p} \sum_{s \in \Phi_p(t,o)} credit(o,s)$$

Predication Scoring (cont.)

answer	x	y	p1	p2	p3	Ranking score
t1	Jerry Yang	Yahoo!	0.8	0.7	0.8	**0.448**
t2	Larry Page	Google	0.6	0.5	0.6	**0.180**
t3	Scott McNealy	Cisco	0.9	0.8	0.2	**0.144**
t4	Bill Gates	IKEA	0.3	0.1	0.2	**0.006**

Bounded Cumulative Model (BCM): integrating all features

$$F_p(t) = \sum_{o \in O_p} f(o)\left[1 - \prod_{s \in \Phi_p(t,o)} (1 - prox(t,s)\, credit(o,s))\right]$$

ordering pattern | proximity | mutual exclusion

Experiment: Data Set

- Data Set
  - 2 million Wikipedia articles
  - 10 predefined types (PERSON, COMPANY, NOVEL, etc.)
  - 0.75 million entities (a subset of articles)
  - 100 million entity occurrences (links to entities)
- Two Query Sets
  - INEX17 – *adapted from topics in INEX09 Entity Ranking track*
    - *Single-11, Multi-6*
  - OWN28 – *manually created*
    - *Single-16, Multi-12*

Pride and Prejudice

Pride and Prejudice is a novel by Jane Austen...

Categories: 1913 novels | British novels

State-of-the-art: EntityRank(ER) [Cheng et al. VLDB07]

- A probabilistic model
- Only uses proximity feature (in a different way)
- Only handle queries similar to our single-predicate querys

```
SELECT x
  FROM PERSON AS x
 WHERE x:["stanford" "graduate"]
```

- We use it for computing predicate scores in our structured query model

Experiment: Results
(nDCG and MAP)

- **BCM has the best performance**
- **The advantage even more clear for multi-predicate queries.**

Table 3: MAP and nDCG on INEX17/OWN28

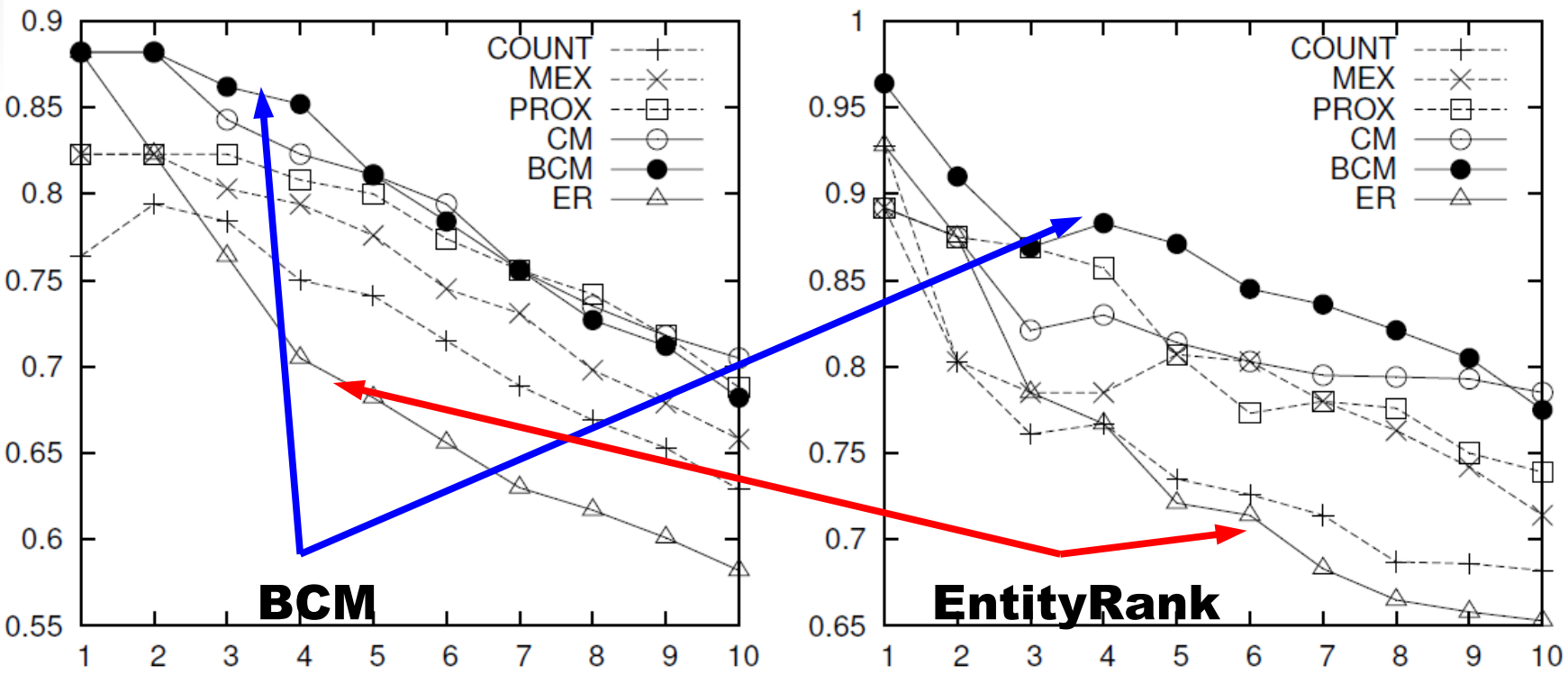
Query	COUNT	MEX	PROX	CM	BCM	ER
nDCG on INEX17						
Single-11	0.889	0.911	0.920	0.920	0.920	0.904
Multi-6	0.880	0.918	0.932	0.954	0.958	0.927
All-17	0.886	0.913	0.924	0.932	0.933	0.912
MAP on INEX17						
Single-11	0.756	0.812	0.843	0.844	0.842	0.779
Multi-6	0.772	0.820	0.852	0.885	0.894	0.809
All-17	0.762	0.815	0.846	0.859	0.860	0.790
nDCG on OWN28						
Single-16	0.917	0.943	0.947	0.953	0.954	0.923
Multi-12	0.800	0.812	0.836	0.844	0.878	0.781
ALL-28	0.867	0.887	0.899	0.906	0.922	0.862
MAP on OWN28						
Single-16	0.758	0.825	0.838	0.858	0.853	0.760
Multi-12	0.579	0.620	0.660	0.684	0.748	0.521
ALL-28	0.681	0.738	0.762	0.783	0.808	0.658

single-predicate queries

multi-predicate queries

Experiment: Results
precition-at-*k*

- BCM is consistently the best
- EntityRank is close to BCM at top 2, but degrades quickly



(a) $k = 10$ on INEX17 (b) $k = 10$ on OWN28

Related Work (1)
IE-Based Approach

- Pre-extract information into database for query
- Still a huge challenge to extract all information
- Un-extracted information is lost and unavailable for query

- [**TextRunner**] Etzioni et al. Open information extraction from the Web. Communication ofACM, 51(12):68–74, 2008.
- [**DBPedia**] Auer et al. DBpedia: A nucleus for a Web of open data. In Int.I Semantic Web Conf., 2007.
- [**Yago**] Suchanek. YAGO: a core of semantic knowledge unifying WordNet and Wikipedia. In WWW, 2007.

Related Work (2)

IR-Based Approach

(ERQ belongs to this approach)

- Search directly in corpus
- All information is pristine and is available for query
- ProxSearch and EntityRank only handle queries resembling our single-predicate query
- CSAW focuses on HTML tables

- [**ProxSearch**] Chakrabarti et al. Optimizing scoring functions and indexes for proximity search in type-annotated corpora. In WWW, 2006.
- [**EntityRank**] Cheng et al. EntityRank: searching entities directly and holistically. VLDB 2007.
- [**CSAW**] Limaye et al. Annotating and Searching Web Tables Using Entities, Types and Relationships. VLDB 2010.

Demo

http://idir.uta.edu/erq