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

Search
stanford graduate

Search
silicon valley company

pages
page reading

Search
Larry Page
Jerry Yang
Dick Price
David Filo
...

found

Paypal
HP
Yahoo!
Google
...

pages

<Jerry Yang, Yahoo!> confirmed as a right answer
Entity-Relationship Query

SELECT x, y
FROM PERSON AS x, COMPANY AS y
WHERE x:["stanford" "graduate"] ... p1
AND y:["silicon valley"] ........... p2
AND x,y:["found"] ............... p3

- Entity-Centric
- Declarative
- Structured
- Keyword-based
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

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)
The predicate score is aggregated from contexts / evidence

<table>
<thead>
<tr>
<th>answer</th>
<th>x</th>
<th>y</th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>Ranking score</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Jerry Yang</td>
<td>Yahoo!</td>
<td>0.8</td>
<td>0.7</td>
<td>0.8</td>
<td><strong>0.448</strong></td>
</tr>
<tr>
<td>t2</td>
<td>Larry Page</td>
<td>Google</td>
<td>0.6</td>
<td>0.5</td>
<td>0.6</td>
<td><strong>0.180</strong></td>
</tr>
<tr>
<td>t3</td>
<td>Scott McNealy</td>
<td>Cisco</td>
<td>0.9</td>
<td>0.8</td>
<td>0.2</td>
<td><strong>0.144</strong></td>
</tr>
<tr>
<td>t4</td>
<td>Bill Gates</td>
<td>IKEA</td>
<td>0.3</td>
<td>0.1</td>
<td>0.2</td>
<td><strong>0.006</strong></td>
</tr>
</tbody>
</table>

0.8 * 0.7 * 0.8 = 0.448
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

$x: ["Stanford" "graduate"] \ldots p_1$

$s_1: \text{Stanford University graduates Jerry Yang} \text{ and } \ldots$

prox(Jerry Yang, s1) = (1 + 1 + 2) / 5 = 0.8

Higher proximity indicates more reliable evidence

Baseline 2 [PROX]: weight each contexts by proximity

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

\[ f(\text{sgx}) = \frac{4+2}{4+2+3} = 0.67 \]

Frequent patterns indicate reliable evidence.

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

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

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)

\[ F_p(t) = \sum_{o \in O_p} \sum_{s \in \Phi_p(t, o)} \text{credit}(o, s) \]
### Predication Scoring (cont.)

<table>
<thead>
<tr>
<th>answer</th>
<th>x</th>
<th>y</th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>Ranking score</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Jerry Yang</td>
<td>Yahoo!</td>
<td>0.8</td>
<td>0.7</td>
<td>0.8</td>
<td>0.448</td>
</tr>
<tr>
<td>t2</td>
<td>Larry Page</td>
<td>Google</td>
<td>0.6</td>
<td>0.5</td>
<td>0.6</td>
<td>0.180</td>
</tr>
<tr>
<td>t3</td>
<td>Scott McNealy</td>
<td>Cisco</td>
<td>0.9</td>
<td>0.8</td>
<td>0.2</td>
<td>0.144</td>
</tr>
<tr>
<td>t4</td>
<td>Bill Gates</td>
<td>IKEA</td>
<td>0.3</td>
<td>0.1</td>
<td>0.2</td>
<td>0.006</td>
</tr>
</tbody>
</table>

**Bounded Cumulative Model (BCM):** integrating all features

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

- 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

```sql
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>
<thead>
<tr>
<th>Query</th>
<th>COUNT</th>
<th>MEX</th>
<th>PROX</th>
<th>CM</th>
<th>BCM</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-11</td>
<td>0.891</td>
<td>0.911</td>
<td>0.920</td>
<td>0.920</td>
<td>0.920</td>
<td>0.904</td>
</tr>
<tr>
<td>Multi-6</td>
<td>0.880</td>
<td>0.918</td>
<td>0.932</td>
<td>0.954</td>
<td>0.958</td>
<td>0.927</td>
</tr>
<tr>
<td>All-17</td>
<td>0.886</td>
<td>0.913</td>
<td>0.924</td>
<td>0.932</td>
<td>0.933</td>
<td>0.912</td>
</tr>
</tbody>
</table>

**nDCG on INEX17**

<table>
<thead>
<tr>
<th>Query</th>
<th>COUNT</th>
<th>MEX</th>
<th>PROX</th>
<th>CM</th>
<th>BCM</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-11</td>
<td>0.756</td>
<td>0.812</td>
<td>0.843</td>
<td>0.844</td>
<td>0.842</td>
<td>0.779</td>
</tr>
<tr>
<td>Multi-6</td>
<td>0.772</td>
<td>0.820</td>
<td>0.852</td>
<td>0.885</td>
<td>0.894</td>
<td>0.809</td>
</tr>
<tr>
<td>All-17</td>
<td>0.762</td>
<td>0.815</td>
<td>0.846</td>
<td>0.859</td>
<td>0.860</td>
<td>0.790</td>
</tr>
</tbody>
</table>

**MAP on INEX17**

<table>
<thead>
<tr>
<th>Query</th>
<th>COUNT</th>
<th>MEX</th>
<th>PROX</th>
<th>CM</th>
<th>BCM</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-16</td>
<td>0.917</td>
<td>0.943</td>
<td>0.947</td>
<td>0.953</td>
<td>0.954</td>
<td>0.923</td>
</tr>
<tr>
<td>Multi-12</td>
<td>0.800</td>
<td>0.812</td>
<td>0.836</td>
<td>0.844</td>
<td>0.878</td>
<td>0.781</td>
</tr>
<tr>
<td>ALL-28</td>
<td>0.867</td>
<td>0.887</td>
<td>0.899</td>
<td>0.906</td>
<td>0.922</td>
<td>0.862</td>
</tr>
</tbody>
</table>

**nDCG on OWN28**

<table>
<thead>
<tr>
<th>Query</th>
<th>COUNT</th>
<th>MEX</th>
<th>PROX</th>
<th>CM</th>
<th>BCM</th>
<th>ER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-16</td>
<td>0.758</td>
<td>0.825</td>
<td>0.838</td>
<td>0.858</td>
<td>0.853</td>
<td>0.760</td>
</tr>
<tr>
<td>Multi-12</td>
<td>0.579</td>
<td>0.620</td>
<td>0.660</td>
<td>0.684</td>
<td>0.748</td>
<td>0.521</td>
</tr>
<tr>
<td>ALL-28</td>
<td>0.681</td>
<td>0.738</td>
<td>0.762</td>
<td>0.783</td>
<td>0.808</td>
<td>0.658</td>
</tr>
</tbody>
</table>
Experiment: Results

**precision-at-\(k\)**

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

![Graphs showing precision-at-\(k\) for BCM and EntityRank](image)
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

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

Demo

http://idir.uta.edu/erq