

September, 2010

# SQL QueRIE Recommendations: a query fragment-based approach

Jayad Akbarnejad, *San José State University*

Magdalini Eirinaki, *San José State University*

Suju Koshy, *San José State University*

Duc On, *San José State University*

Neoklis Polyzotis, *University of California, Santa Cruz*

# SQL QueRIE Recommendations: A Query Fragment-based Approach

Javad Akbarnejad

Magdalini Eirinaki

Suju Koshy

Duc On

Neoklis Polyzotis



**SAN JOSÉ STATE**  
UNIVERSITY



# Motivation



- Scientific disciplines use relational DBMS for storage and retrieval of information
  - Biologists (e.g. UCSC Genome, BMRB)
  - Astronomers (e.g. Skyserver)
  - Chemists (e.g. PubChem)
- DBs are accessible online by users with diverse information needs
- Typical users do interactive exploration

## Motivation (cont'd)



- Typical users are not SQL experts
- Scientific datasets increase in size
- Users may miss interesting information
  - They do not write the “right” query
  - They are not aware of all parts of the database

**Our goal:** Assist users in finding useful information

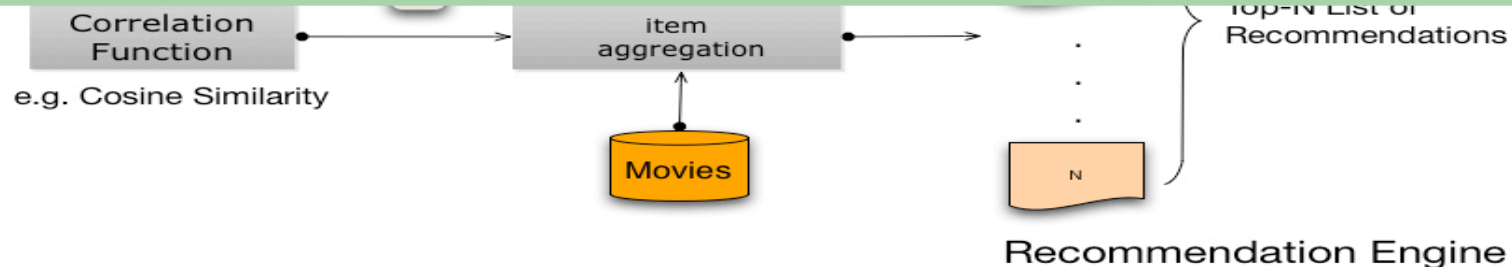
# Web Collaborative Filtering

## Example: Movie Recommendations

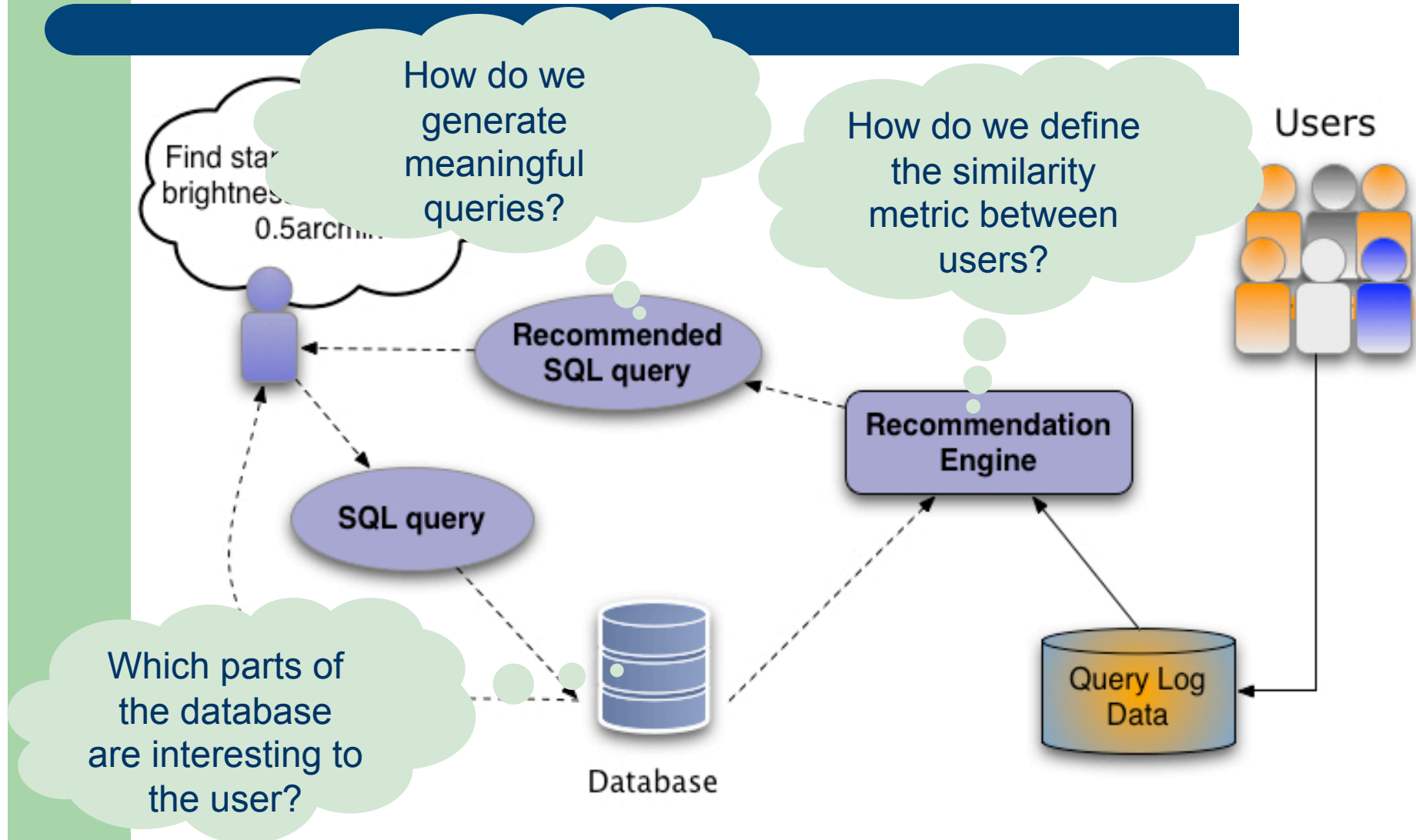
If Alice and Bob **both like movie X** and Alice **likes movie Y**  
**then**  
Bob is likely to be interested in **seeing movie Y**



If Alice and Bob **both query data X** and Alice **queries data Y**  
**then**  
Bob is likely to be interested in **querying data Y**



# System Architecture

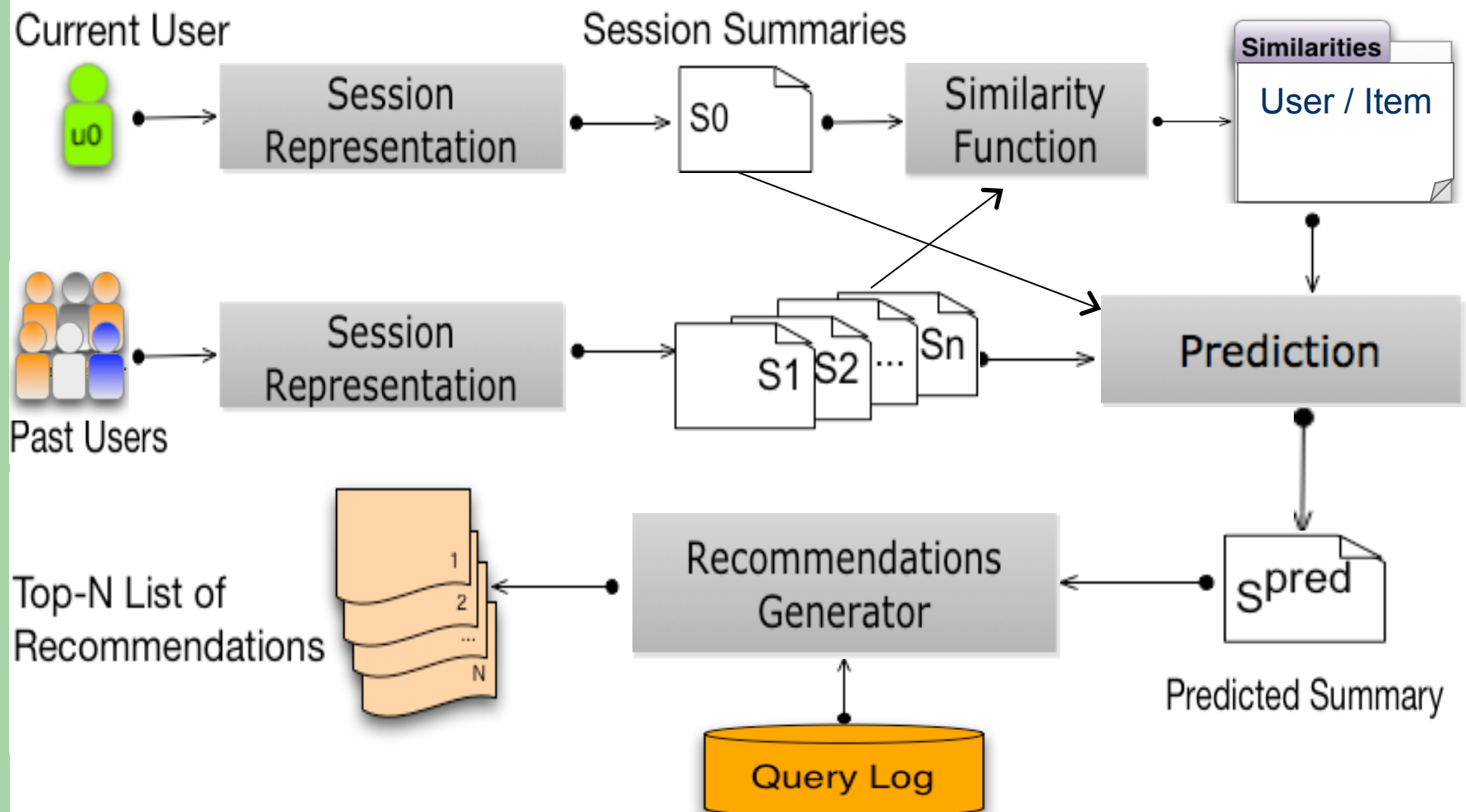


# Roadmap

---

- Introduction
- QueRIE Recommendation Framework
- Experiments
- QueRIE Prototype
- Conclusion

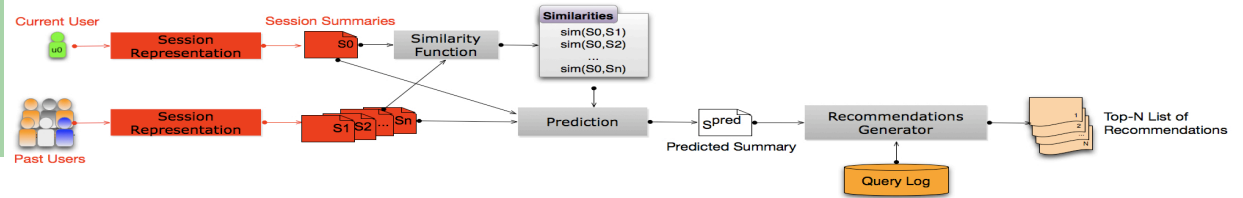
# QueRIE Conceptual Framework





# QueRIE Recommendation Engines

1. Tuple-based recommendations [SSDBM09, ICDM09]
  - Sessions represented by the tuples “touched” by respective queries
  - User-based similarity: 2 users are similar if they explore the same parts of the DB
  - Predict which parts of DB will interest the user and recommend queries that “touch” them
2. Query fragment-based recommendations



# Session Representation

Relations:  $R(\underline{a}, b, c)$   
 $S(\underline{d}, e, \underline{f})$



$Q_1$ : SELECT R.a, R.b FROM R WHERE R.b = 2

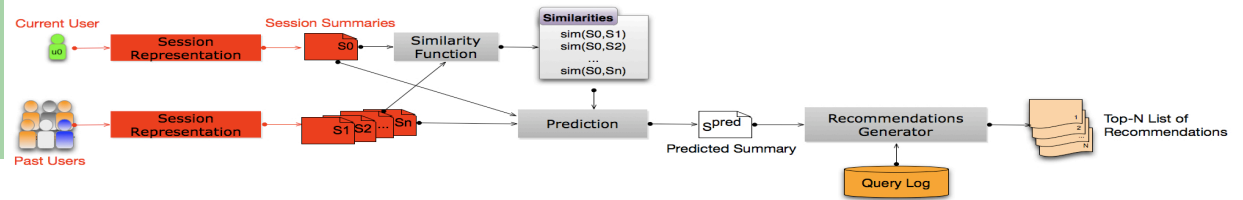
$Q_2$ : SELECT R.a, R.b, S.e FROM R, S WHERE R.a = S.f AND R.b < 3



Query parsing & relaxation

$Q_1$ : SELECT R.a, R.b FROM R WHERE R.b EQU NUM

$Q_2$ : SELECT R.a, R.b, S.e FROM R, S WHERE R.a EQU S.f  
 AND R.b COMPARE NUM



## Session Representation (cont'd)



$Q_1$ : SELECT R.a, R.b FROM R WHERE R.b EQU NUM

$Q_2$ : SELECT R.a, R.b, S.e FROM R, S WHERE R.a EQU S.f  
AND R.b COMPARE NUM

$QF = \{R, S, \dots, R.a, R.b, S.e, \dots, R.b \text{ EQU NUM}, R.b \text{ COMPARE NUM}, R.a \text{ EQU S.f}\}$

### Binary Scheme

$Q_1 = \langle 1, 0, \dots, 1, 1, 0, \dots, 1, 0, 0 \rangle$

$Q_2 = \langle 1, 1, \dots, 1, 1, 1, \dots, 0, 1, 1 \rangle$

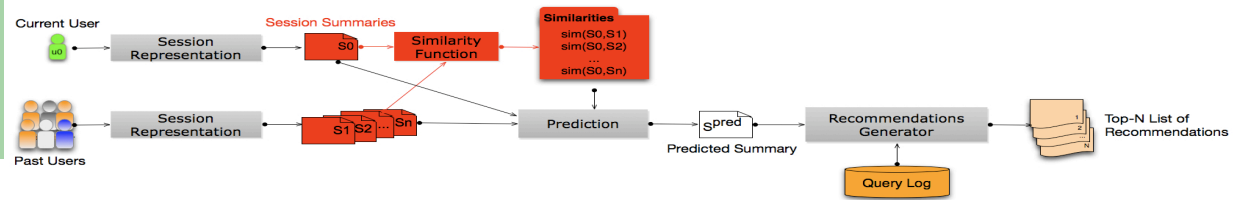
$S_0 = \langle 1, 1, \dots, 1, 1, 1, \dots, 1, 1, 1 \rangle$

### Weighted Scheme

$Q_1 = \langle 1, 0, \dots, 1, 1, 0, \dots, 1, 0, 0 \rangle$

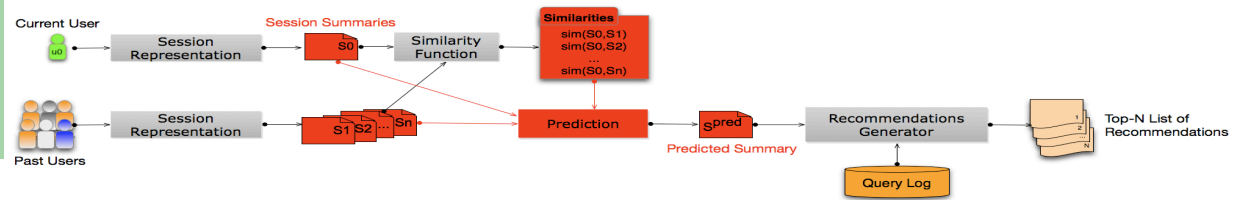
$Q_2 = \langle 1, 1, \dots, 1, 1, 1, \dots, 0, 1, 1 \rangle$

$S_0 = \langle 2, 1, \dots, 2, 2, 1, \dots, 1, 1, 1 \rangle$



# Session Similarity

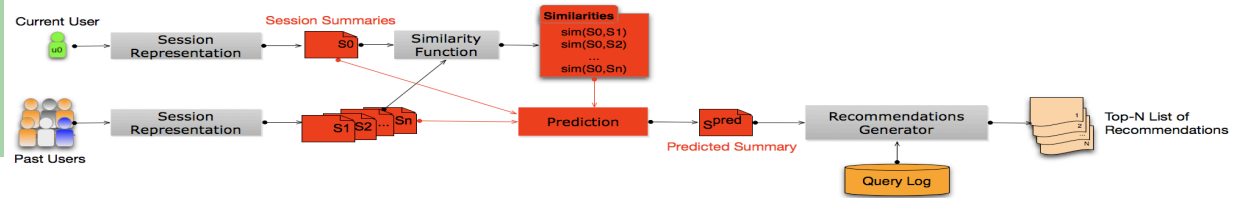
- Based on the item-based approach
  - Construct *fragment x fragment* similarity matrix offline
  - More efficient than the user-based approach
- Vector-space similarity functions can be used
- High similarity means that the query fragments co-appear frequently in sessions
  - => the active user might also like to use them



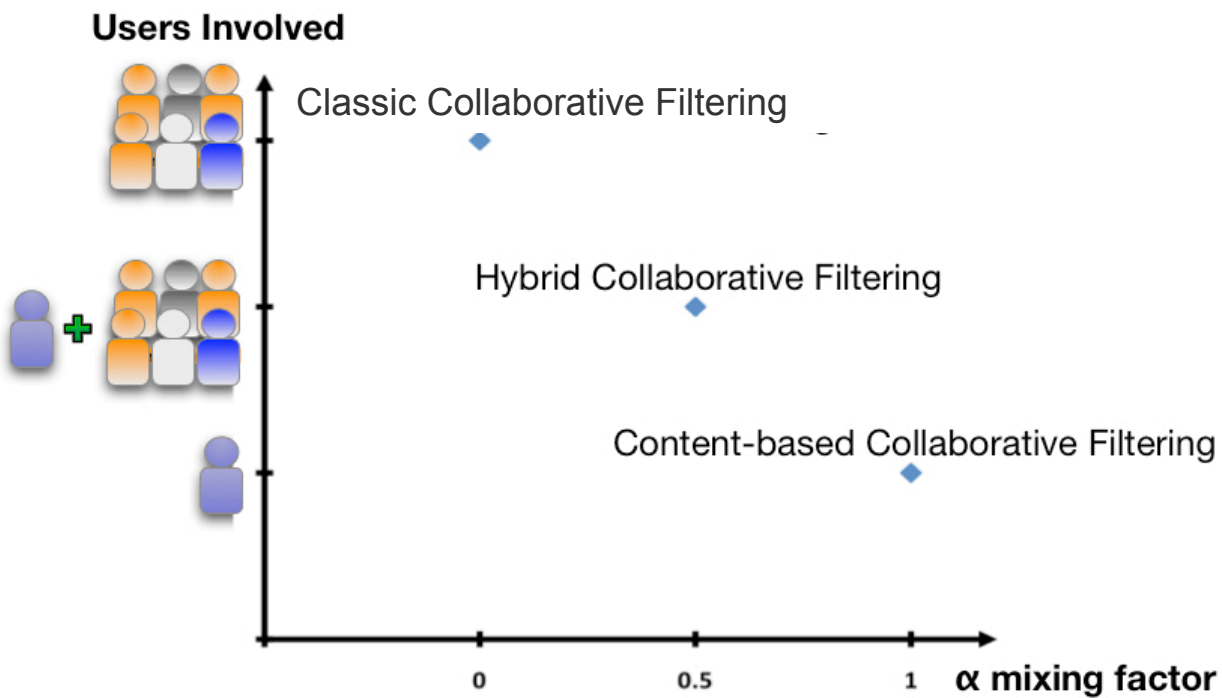
# Prediction

- For each fragment  $\phi$ , select top-k similar fragments  $\rho \in R$
- Then compute “predicted summary”:

$$S_0^{pred}[\phi] = \frac{\sum_{\rho \in R} S_0[\rho] * sim(\rho, \phi)}{\sum_{\rho \in R} sim(\rho, \phi)}$$

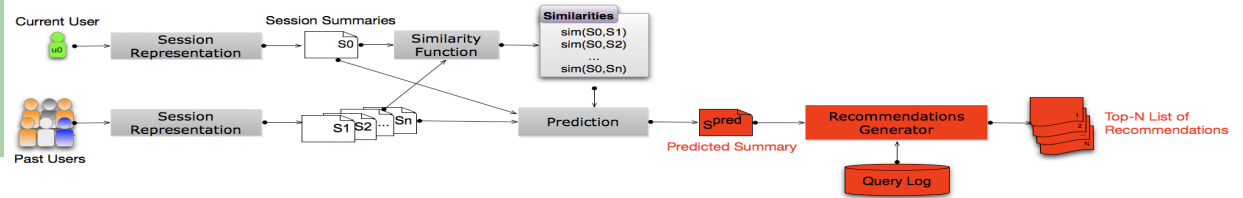


# Prediction – the $\alpha$ factor



$S_0^{pred}$  contains:

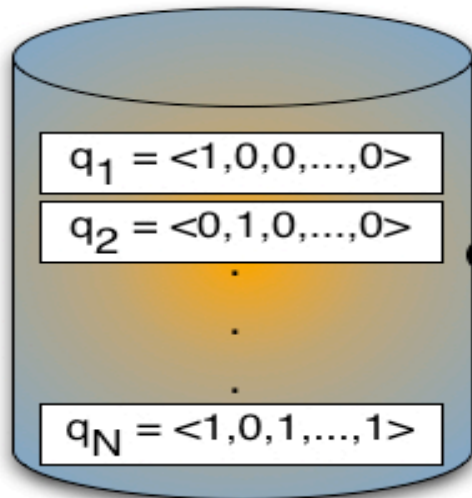
- Only other users' fragments
- Both  $S_0$  and other users' fragments
- Only  $S_0$  fragments



# Recommendations Generator



Use queries of past users



Query Log Data

$$S^{\text{pred}} = \langle 1, 0, 0, \dots, 0 \rangle$$

Top-n fragments

Similarity Function  
 $(u^{\text{pred}}, q_i)$

$$\begin{aligned} \text{rank}(q_1) &= \text{sim}(u^{\text{pred}}, q_1) \\ \text{rank}(q_2) &= \text{sim}(u^{\text{pred}}, q_2) \end{aligned}$$

$$\text{rank}(q_N) = \text{sim}(u^{\text{pred}}, q_N)$$

Return Top- m  
Queries

# Roadmap

- Introduction
- QueRIE Recommendation Framework
- Experiments
- QueRIE Prototype
- Conclusions



# Experimental Setup

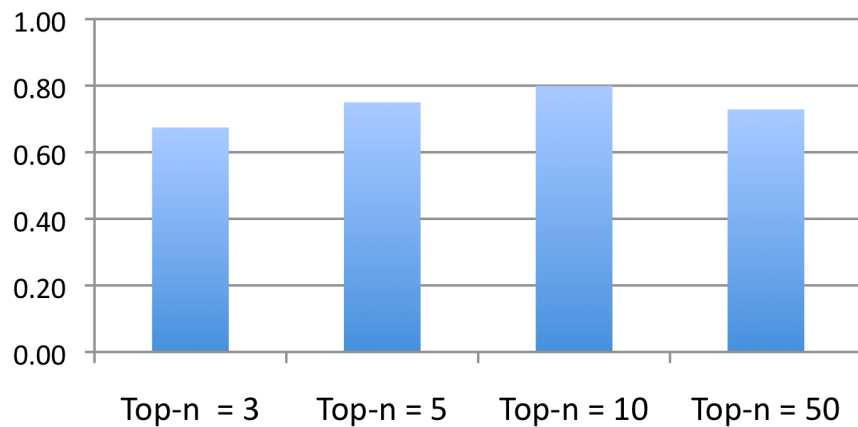
- SkyServer Dataset

#Sessions	180
#Distinct Queries	1400
#Distinct query fragments	755
#Non-zero pair-wise fragment similarities	30436
Avg. number of queries per session	9.3
Min. number of queries per session	3

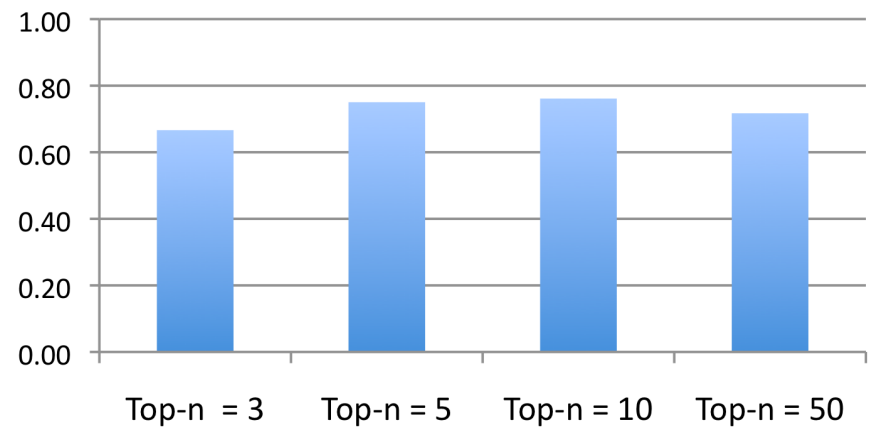
- Validation method: Holdout Set
- Evaluation Metrics: Precision, Recall, F-Score

# Experimental evaluation – top-n

**Average Precision**



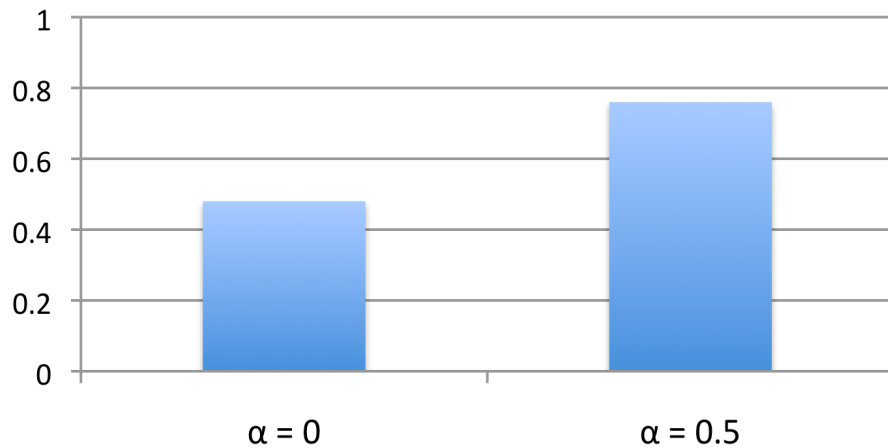
**Average F-Score**



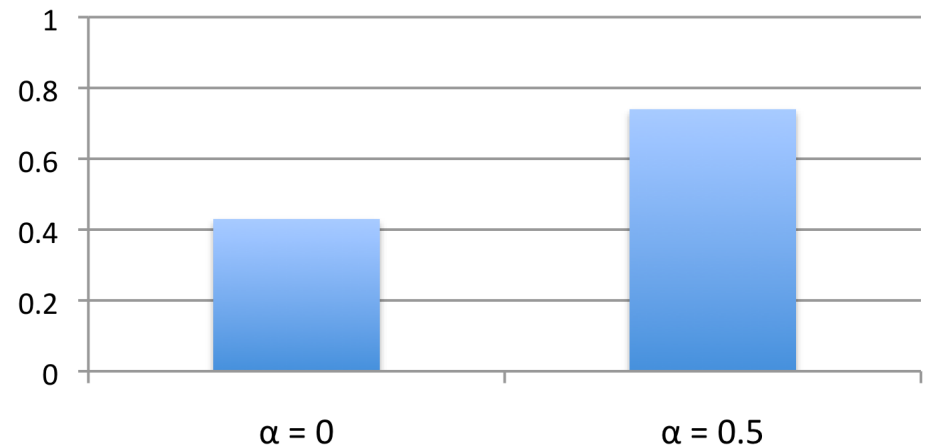
- Precision and recall drop for large  $n$ .
- More fragments with low similarity included in the mix

# Experimental Evaluation - $\alpha$

Average Precision



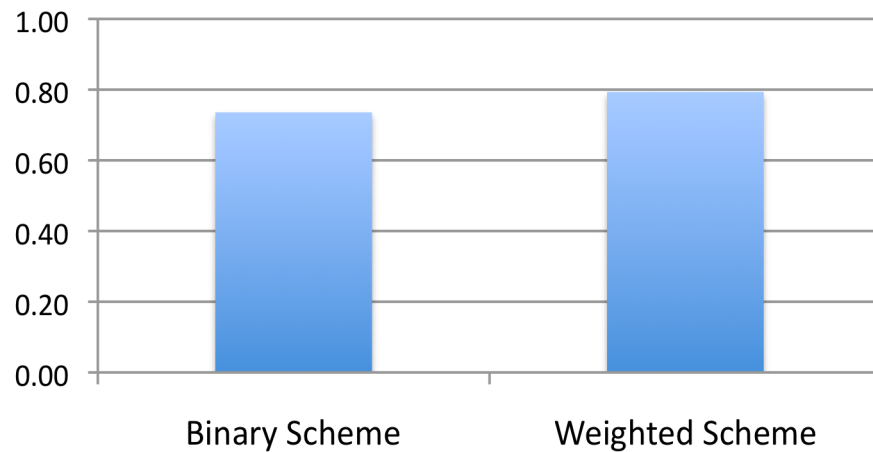
Average F-score



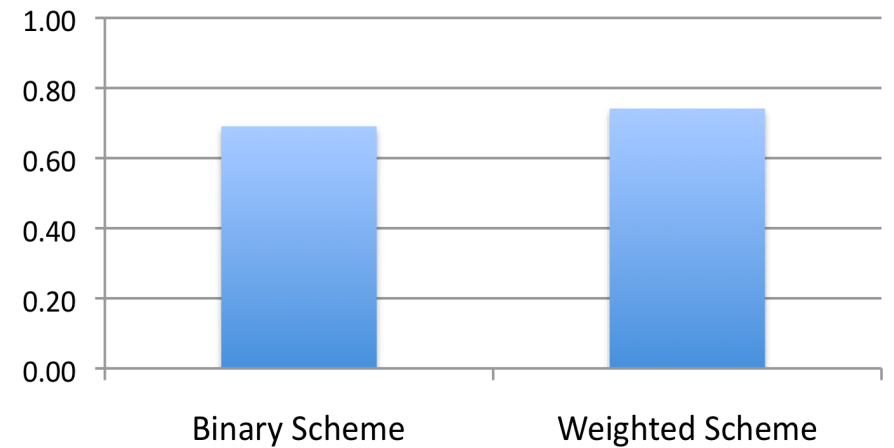
- Including user's current session fragments is beneficial
- Expansion/Restructuring of posted queries

# Experimental Evaluation – Weighting Scheme

**Average Precision**



**Average F-score**




Weighted scheme slightly outperforms the binary

# Roadmap

---

- Introduction
- QueRIE Recommendation Framework
- Experiments
- **QueRIE Prototype**
- **Conclusions**

# QueRIE Prototype



## Query Recommendations for Interactive Data Exploration

September 12, 2010

- Welcome
- Schema Browser
- Show My History
- Recommendation Details
- Administration
- Test Harness
- Logout

### Query Results

Please provide your Query here:

```
select top 1000 * from field where  
fieldid=0x08280ab2802c0000
```

Submit

**Query Results:**

fieldID	skyVersion	run	rerun	camcol	field	nObjects	nChild	nGalaxy	nStars	numStars_u	numStars_g	numStars_r	numStars_i	numStars_z	nCR_u	nCR_g	nCR_r	nCR_i	nCR_z	nBrightC
587731513142673408	1	2738	40	4	44	1103	328	668	274	472	757	757	757	735	139	304	164	164	141	28

Recommended Queries:

- select top 1000 \* from frame where fieldid=0x08280ab2802c0000
- select top 1000 \* from frame where fieldid=0x08280ab2802c0000
- select top 1000 \* from photoobj where objid=0x08280ab2802c0111

# QueRIE Prototype (cont'd)



## Query Recommendations for Interactive Data Exploration

September 12, 2010

Welcome

Schema Browser

Show My History

Recommendation Details

Administration

Test Harness

Logout

### Recommendation Details

#### Recommendations:

1.	Current active session is 61468
2.	1. Queries in active session: select top 1000 * from field where fieldid=0x08280ab2802c0000
3.	Top predicted items: 7735 7736 7737 7739 7740
4.	Top predicted items names: T16 FRAME.* C16_0 EQU HEXNUM PHOTOOBJ.* CV17_0 EQU HEXNUM
5.	Recommendation queries are
6.	Recommendation Query 1 select top 1000 * from frame where fieldid=0x08280ab2802c0000
7.	Session ID for above Query 45
8.	Recommendation Query 2 select top 1000 * from photoobj where objid=0x08280ab2802c0111
9.	Session ID for above Query 45

# QueRIE Prototype

- Demo @ VLDB
  - Session: Data Extraction, Integration and Mining
  - Tue & Wed, 2 – 3:30 PM
  - Lyrebird room



# Conclusions

- Non-expert users need help in exploring databases
- Query recommendations can be an effective tool in guiding exploration
- Collaborative filtering provides a natural method to generate recommendations
- Experiments show promising results on real-world datasets
- Ongoing & Future Work:
  - Comparison of two recommendation engines
  - Extend for form-based queries

# Thank you !



Questions