Learning Latent Semantic Relations from Clickthrough Data for Query Suggestion

Hao Ma, Haixuan Yang, Irwin King, Michael R. Lyu

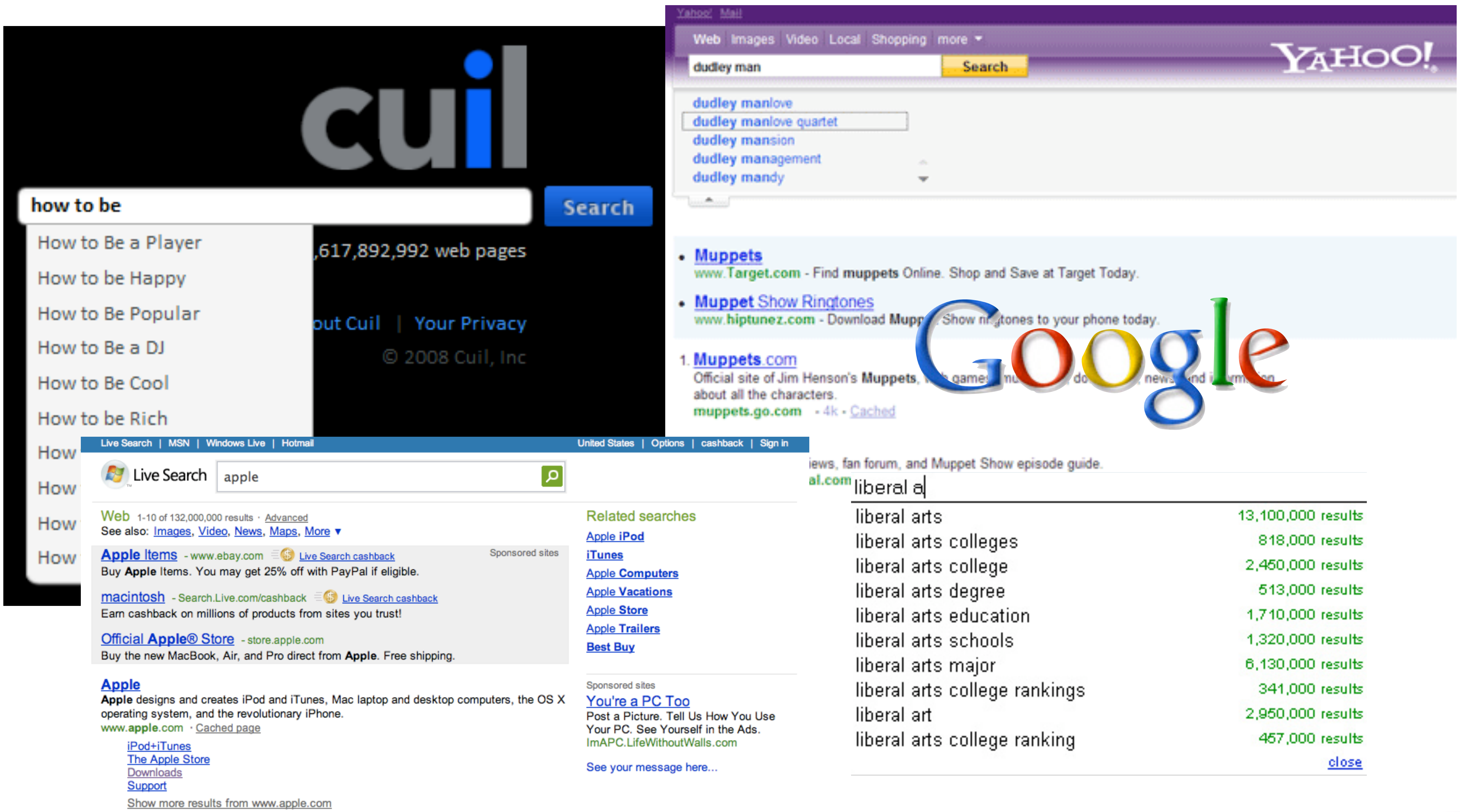
king@cse.cuhk.edu.hk
http://www.cse.cuhk.edu.hk/~king

Department of Computer Science & Engineering
The Chinese University of Hong Kong

Quick! What's another word for Thesaurus?

http://www.blifaloo.com/humor/thesaurus.php

A Better Mousetrap?



Challenges

- Queries contain ambiguous and new terms
  - apple: "apple computer" or "apple pie"?
  - NDCG:?
- Users tend to submit short queries consisting of only one or two words
  - almost 20% one-word queries
  - almost 30% two-word queries

- Users may have little or even no knowledge about the topic they are searching for!

Problems

- Traditional query suggestion
  - local (i.e., search result sets)
  - global (i.e., thesauri) document analysis
- Hard to remove noise in web pages
- Difficult to summarize the latent meaning of documents (ill-posed inverse problem!)

What is Clickthrough Data

- Query logs recorded by search engines

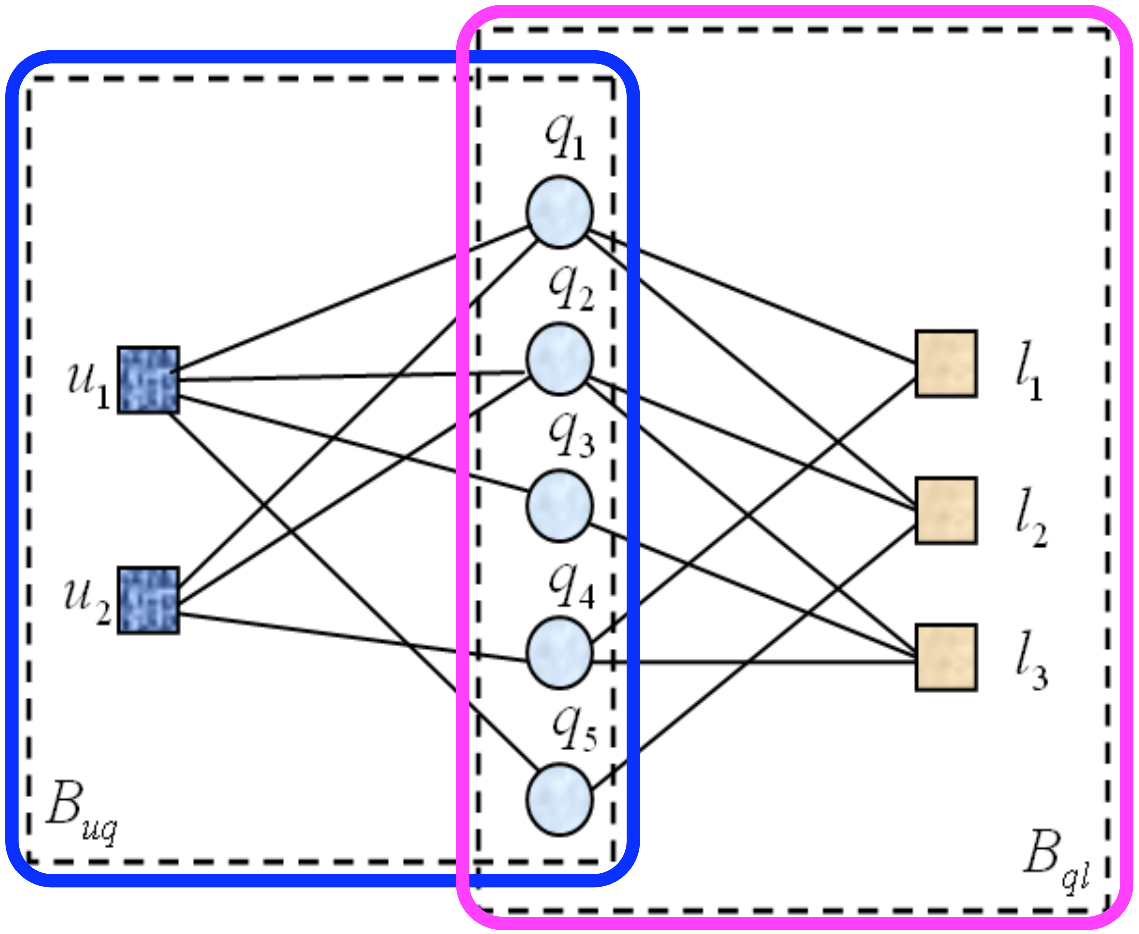
$$\langle u, q, l, r, t \rangle$$

Table 1: Samples of search engine clickthrough data

ID	Query	URL	Rank	Time
358	facebook	http://www.facebook.com	1	2008-01-01 07:17:12
358	facebook	http://en.wikipedia.org/wiki/Facebook	3	2008-01-01 07:19:18
3968	apple iphone	http://www.apple.com/iphone/	1	2008-01-01 07:20:36
...	...	...	...	...

- Users' relevance feedback to indicate desired/preferred/target results

Joint Bipartite Graph



$B_{uq} = (V_{uq}, E_{uq})$
$V_{uq} = U \cup Q$
$U = \{u_1, u_2, ..., u_m\}$
$Q = \{q_1, q_2, ..., q_n\}$
$E_{uq} = \{(u_i, q_j)|$ there is an edge from u_i to $q_j\}$ is the set of all edges.
The edge (u_i, q_j) exists in this bipartite graph if and only if a user u_i issued a query q_j.

$B_{ql} = (V_{ql}, E_{ql})$
$V_{ql} = Q \cup L$
$Q = \{q_1, q_2, ..., q_n\}$
$L = \{l_1, l_2, ..., l_p\}$
$E_{ql} = \{(q_i, l_j)|$ there is an edge from q_i to $l_j\}$ is the set of all edges.
The edge (q_j, l_k) exists if and only if a user u_i clicked a URL l_k after issuing an query q_j.

Key Points

- Two-level latent semantic analysis

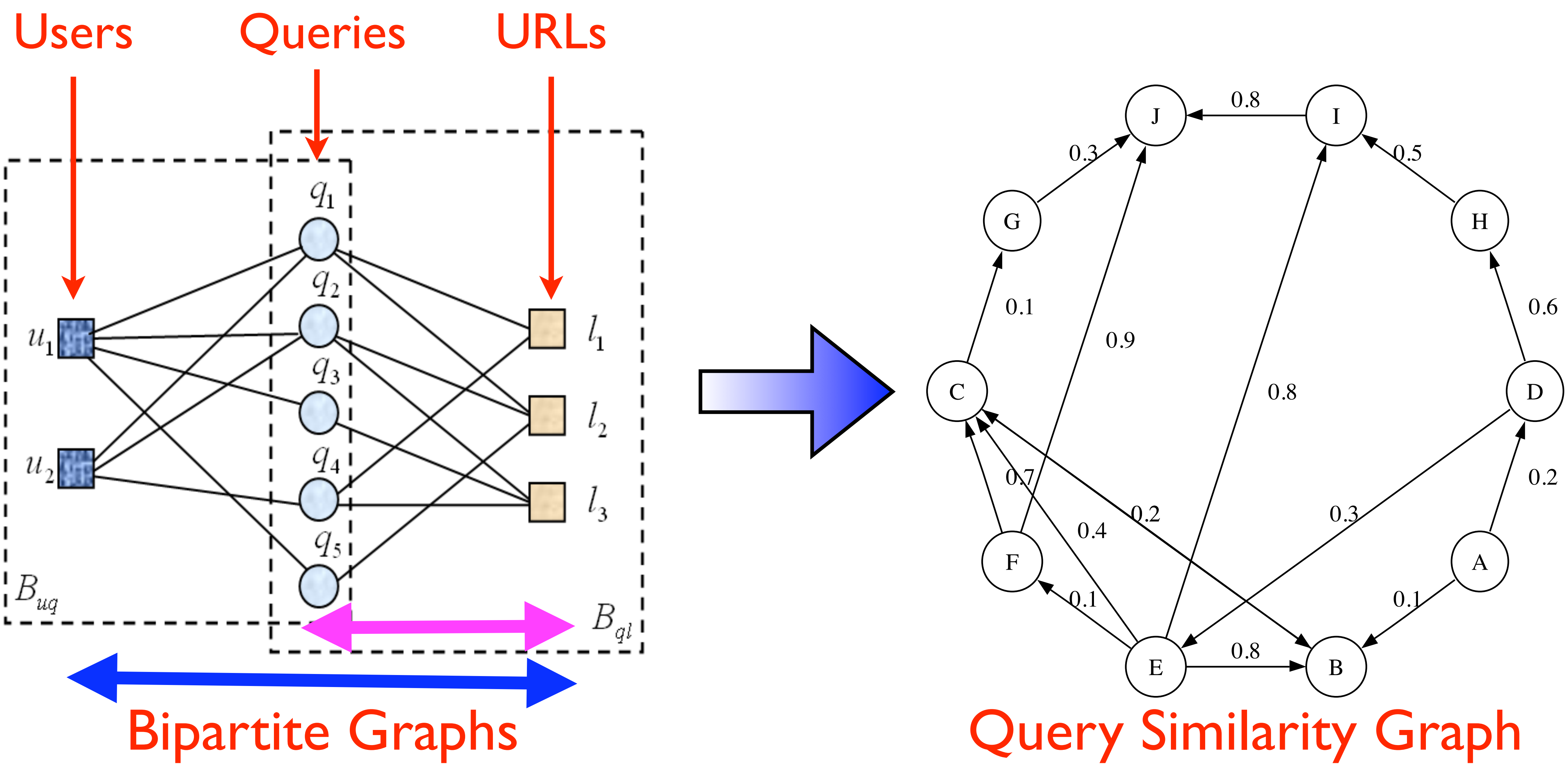
Level 1

- Consider the use of a joint user-query and query-URL bipartite graphs for query suggestion

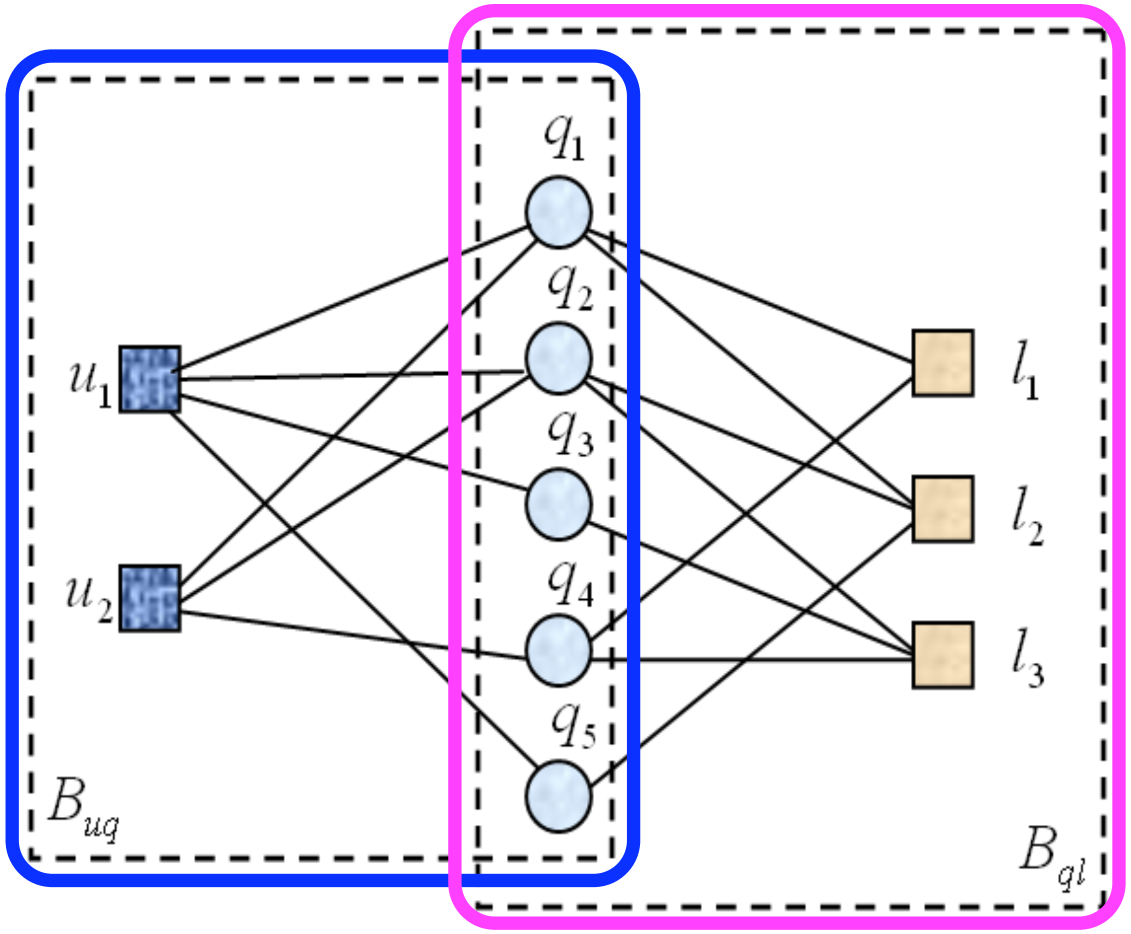
- Use matrix factorization for learning query features in constructing the Query Similarity Graph

Level 2

- Use heat diffusion for similarity propagation for query suggestions



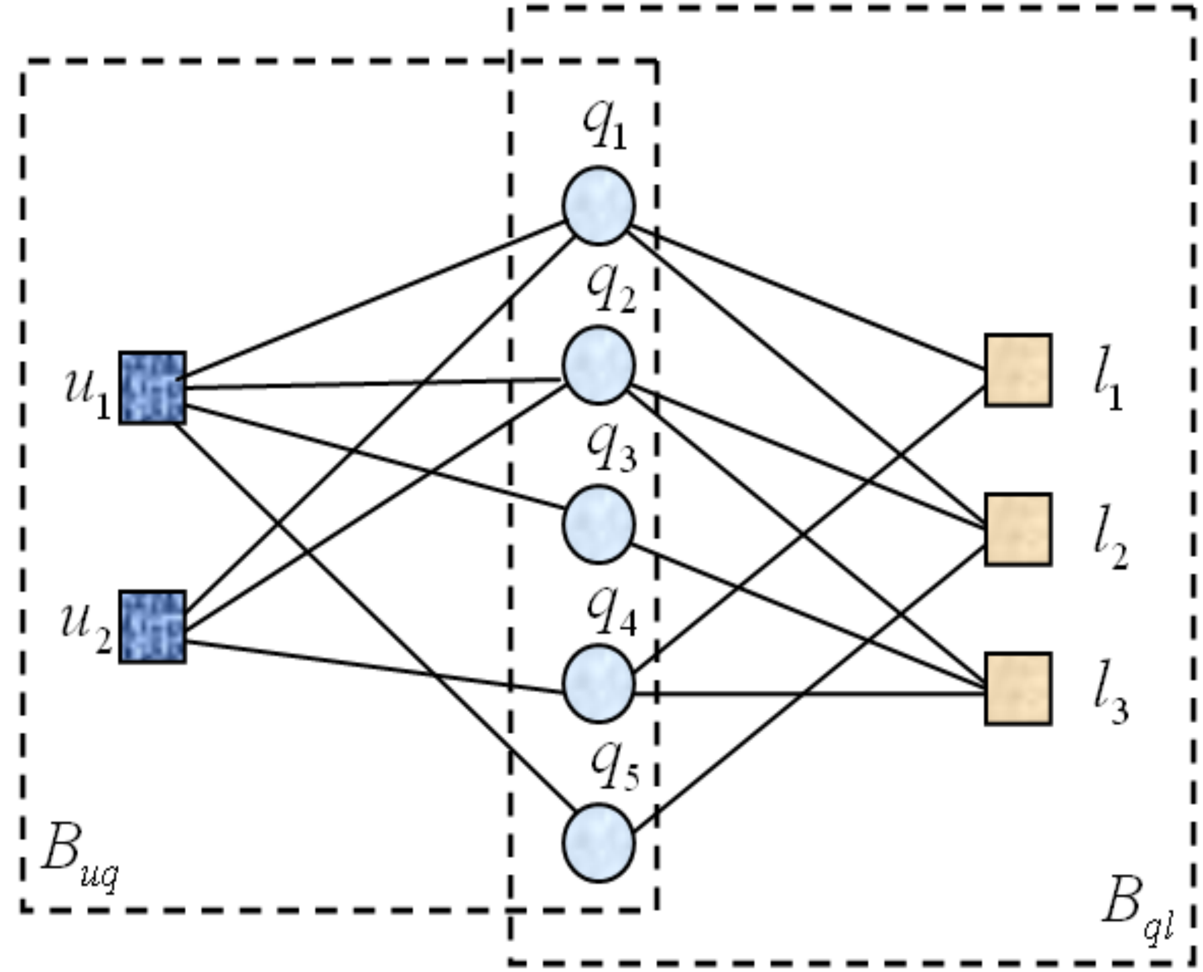
- Queries are issued by the users, and which URLs to click are also decided by the users
- Two distinct users are similar if they issued similar queries
- Two queries are similar if they are issued by similar users



r^*_{ij}	Normalized weight, how many times u_i issued q_j
s^*_{jk}	Normalized weight, how many times q_j is linked to l_k
U_i	L-dimensional vector of user u_i
Q_j	L-dimensional vector of query q_j
L_k	L-dimensional vector of URL l_k

$$\mathcal{H}(R,U,Q) = \min_{U,Q} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I^R_{ij}(r^*_{ij} - g(U_i^T Q_j))^2 + \frac{\alpha_u}{2}\|U\|_F^2 + \frac{\alpha_q}{2}\|Q\|_F^2$$

$$\mathcal{H}(S,Q,L) = \min_{Q,L} \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{p} I^S_{jk}(s^*_{jk} - g(Q_j^T L_k))^2 + \frac{\alpha_q}{2}\|Q\|_F^2 + \frac{\alpha_l}{2}\|L\|_F^2$$



$$\mathcal{H}(S,R,U,Q,L) = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{p} I_{jk}^{S}(s_{jk}^{*} - g(Q_j^T L_k))^2 + \frac{\alpha_r}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(r_{ij}^{*} - g(U_i^T Q_j))^2 + \frac{\alpha_u}{2}\|U\|_F^2 + \frac{\alpha_q}{2}\|Q\|_F^2 + \frac{\alpha_l}{2}\|L\|_F^2,$$

- A local minimum can be found by performing gradient descent in U_i, Q_j and L_k

Gradient Descent Equations

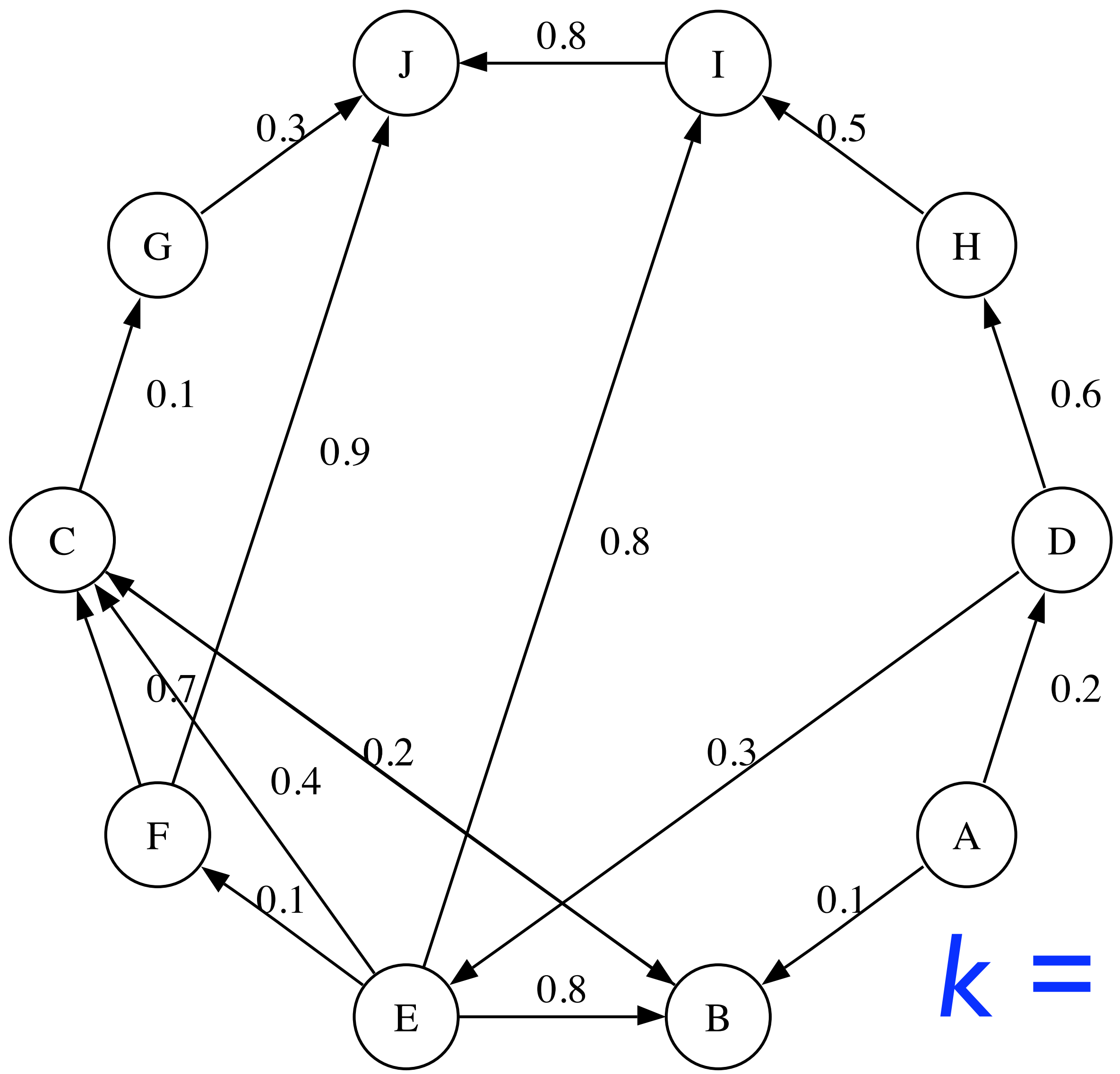
$$\frac{\partial \mathcal{H}}{\partial U_i} = \alpha_r \sum_{j=1}^{n} I_{ij}^{R} g'(U_i^T Q_j)(g(U_i^T Q_j) - r_{ij}^{*}) Q_j + \alpha_u U_i,$$

$$\boxed{\begin{aligned} \frac{\partial \mathcal{H}}{\partial Q_j} &= \sum_{k=1}^{p} I_{jk}^{S} g'(Q_j^T L_k)(g(Q_j^T L_k) - s_{jk}^{*}) L_k \\ &+ \alpha_r \sum_{i=1}^{m} I_{ij}^{R} g'(U_i^T Q_j)(g(U_i^T Q_j) - r_{ij}^{*}) U_i + \alpha_q Q_j, \end{aligned}}$$

$$\frac{\partial \mathcal{H}}{\partial L_k} = \sum_{j=1}^{n} I_{jk}^{S} g'(Q_j^T L_k)(g(Q_j^T L_k) - s_{jk}^{*}) Q_j + \alpha_l L_k,$$

Only the Q matrix, the queries' latent features,
is being used to generate the query similarity graph!

Query Similarity Graph



$k = 4$

- Similarities are calculated using queries' latent features
- Only the top-*k* similar neighbors (terms) are kept

Similarity Propagation

- Based on the Heat Diffusion Model
- In the query graph, given the heat sources and the initial heat values, start the heat diffusion process and perform *P* steps
- Return the Top-*N* queries in terms of highest heat values for query suggestions

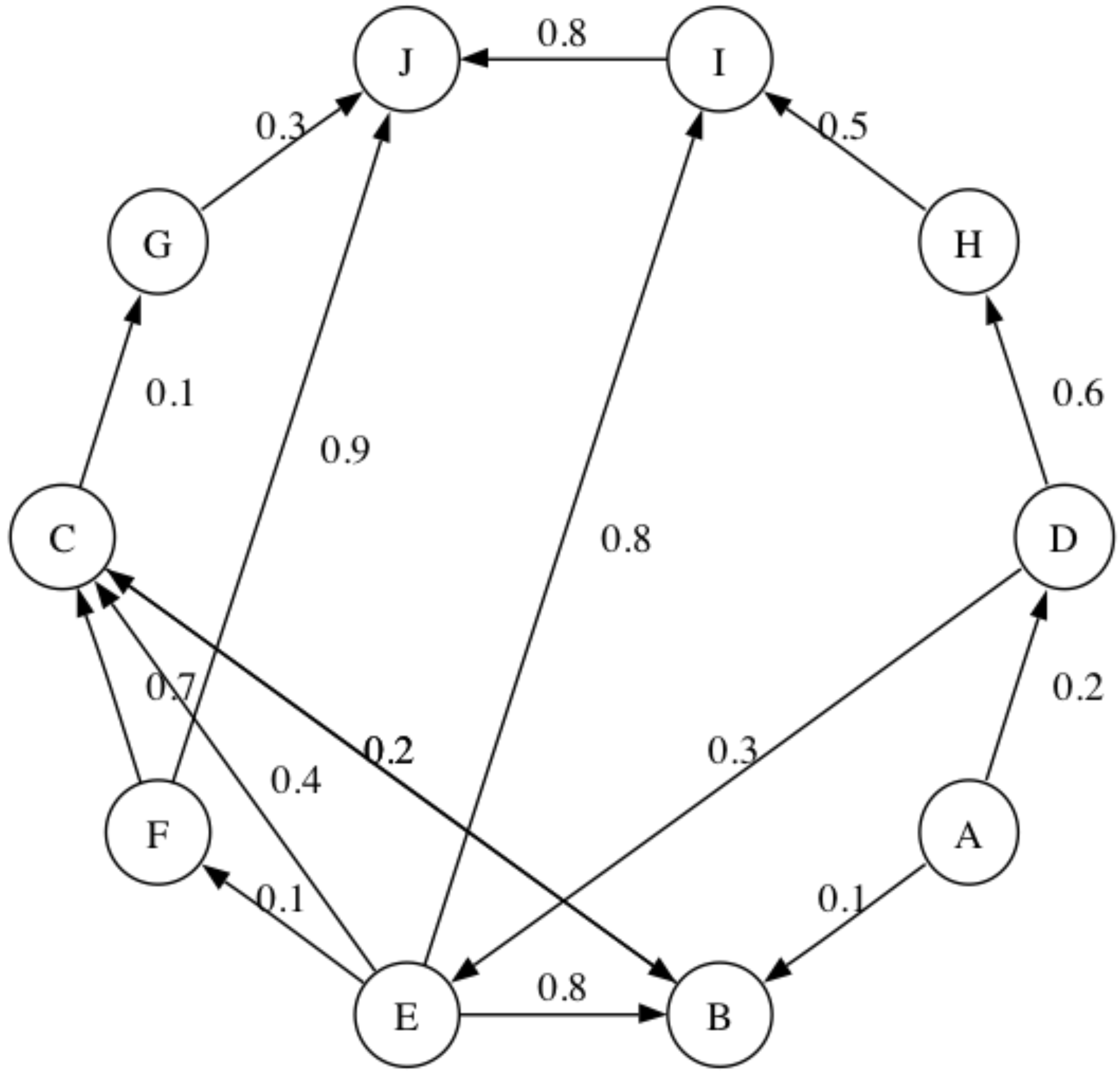
Heat Diffusion Model

- Heat diffusion is a physical phenomena
- Heat flows from high temperature to low temperature in a medium
- Heat kernel is used to describe the amount of heat that one point receives from another point
- The way that heat diffuse varies when the underlying geometry

$$\rho C_P \frac{\partial T}{\partial t} = Q + \nabla \cdot (k \nabla T)$$

ρ	Density
C_P	Heat capacity and constant pressure
$\frac{\partial T}{\partial t}$	Change in temperature over time
Q	Heat added
k	Thermal conductivity
∇T	Temperature gradient
$\nabla \cdot \mathbf{v}$	Divergence

Heat Diffusion Process



Similarity Propagation Model

$$\frac{f_i(t+\Delta t)-f_i(t)}{\Delta t} = \alpha\left(-\frac{\tau_i}{d_i} f_i(t)\sum_{k:(q_i,q_k)\in E} w_{ik} + \sum_{j:(q_j,q_i)\in E}\frac{w_{ji}}{d_j} f_j(t)\right) \qquad (1)$$

$$\mathbf{f}(1) = e^{\alpha \mathbf{H}}\mathbf{f}(0) \qquad (2)$$

$$H_{ij} = \begin{cases} w_{ji}/d_j, & (q_j, q_i) \in E, \\ -(\tau_i/d_i)\sum_{k:(i,k)\in E} w_{ik}, & i = j, \\ 0, & \text{otherwise.} \end{cases} \qquad (3)$$

$$\mathbf{f}(1) = e^{\alpha \mathbf{R}}\mathbf{f}(0), \quad \boxed{\mathbf{R} = \gamma \mathbf{H} + (1-\gamma)\mathbf{g}\mathbf{1}^T} \qquad (4)$$

α	Thermal conductivity
d_i	Heat value of node i at time t
$f_i(t)$	Heat value of node i at time t
w_{ik}	Weight between node i and node k
$\mathbf{f}(0)$	Vector of the initial heat distribution
$\mathbf{f}(1)$	Vector of the heat distribution at time 1
τ_i	Equal to 1 if node i has outlinks, else equal to 0
γ	Random jump parameter, and set to 0.85
$\mathbf{g}$	Uniform stochastic distribution vector

Discrete Approximation

- Compute $e^{\alpha \mathbf{R}}$ is time consuming
- We use the discrete approximation to substitute

$$\mathbf{f}(1) = \left(\mathbf{I} + \frac{\alpha}{P}\mathbf{R}\right)^{P} \mathbf{f}(0)$$

- For every heat source, only diffuse heat to its neighbors within *P* steps
- In our experiments, *P* = 3 already generates fairly good results

Query Suggestion Procedure

- For a given query *q*

1. Select a set of *n* queries, each of which contains at least one word in common with *q*, as heat sources
2. Calculate the initial heat values by

$$f_{\hat{q}_i}(0) = \frac{|\mathcal{W}(q) \cap \mathcal{W}(\hat{q}_i)|}{|\mathcal{W}(q) \cup \mathcal{W}(\hat{q}_i)|}$$

q = "Sony"
"Sony" = 1
"Sony Electronics" = 1/2
"Sony Vaio Laptop" = 1/3

3. Use $\mathbf{f}(1) = e^{\alpha \mathbf{R}} \mathbf{f}(0)$ to diffuse the heat in graph
4. Obtain the Top-*N* queries from $\mathbf{f}(1)$

Physical Meaning of α

- If set α to a large value
  - The results depend more on the query graph, and more semantically related to original queries, e.g., travel => lowest air fare
- If set α to a small value
  - The results depend more on the initial heat distributions, and more literally similar to original queries, e.g., travel => travel insurance

Experimental Dataset

Data Source	Clickthrough data from AOL search	After Pre-Processing
Collection Period	March 2006 to May 2006 (3 months)	
Lines of Logs	19,442,629	
Unique user IDS	657,426	192,371
Unique queries	4,802,520	224,165
Unique URLs	1,606,326	343,302
Unique words		69,937

Query Suggestions

Table 2: Examples of LSQS Query Suggestion Results ($k = 50$)

Testing Queries	Suggestions				
	$\alpha = 10$			$\alpha = 1000$	
	Top 1	Top 2	Top 3	Top 4	Top 5
michael jordan	michael jordan shoes	michael jordan bio	pictures of michael jordan	nba playoff	nba standings
travel	travel insurance	abc travel	travel companions	hotel tickets	lowest air fare
java	sun java	java script	java search	sun microsystems inc	virtual machine
global services	ibm global services	global technical services	staffing services	temporary agency	manpower professional
walt disney land	world of disney	disney world orlando	disney world theme park	disneyland grand hotel	disneyland in california
intel	intel vs amd	amd vs intel	pentium d	pentium	centrino
job hunt	jobs in maryland	monster job	jobs in mississippi	work from home online	monster board
photography	photography classes	portrait photography	wedding photography	adobe elements	canon lens
internet explorer	ms internet explorer	internet explorer repair	internet explorer upgrade	microsoft com	security update
fitness	fitness magazine	lifestyles family fitness	fitness connection	womens health magazine	family fitness
m schumacher	schumacher	red bull racing	formula one racing	ferrari cars	formula one
solar system	solar system project	solar system facts	solar system planets	planet jupiter	mars facts
sunglasses	replica sunglasses	cheap sunglasses	discount sunglasses	safilo	marhon
search engine	audio search engine	best search engine	search engine optimization	song lyrics search	search by google
disease	grovers disease	liver disease	morgellons disease	colic in babies	oklahoma vital records
pizzahut	pizza hut menu	pizza coupons	pizza hut coupons	papa johns pizza coupon	papa johns
health care	health care proxy	universal health care	free health care	great west healthcare	uhc
flower delivery	global flower delivery	online florist	flowers online	send flowers	virtual flower
wedding	wedding guide	wedding reception ideas	wedding decoration	unity candle	centerpiece ideas
astronomy	astronomy magazine	astronomy pic of the day	star charts	space pictures	comet

Comparisons

Table 3: Comparisons between LSQS and SimRank

	Top 1	Top 2	Top 3	Top 4	Top 5
jaguar					
LSQS	jaguar cat	jaguar commercial	jaguar parts	jaguarundi	leopard
SimRank	american black bear	bottlenose dolphin	leopard	margay	jaguarundi
apple					
LSQS	apple computers	apple ipod	apple diet	apple vacations	apple bottom
SimRank	ipod troubleshooting	apple quicktime	apple ipods	apple computers	apple software

Table 4: Accuracy Comparisons

Accuracy	**LSQS**	**SimRank**
By Experts	0.8413	0.7101
By ODP	0.6823	0.5789

ODP, Open Directory Project, see http://dmoz.org

Impact of Parameter *k*

To test the extend of similarity needed

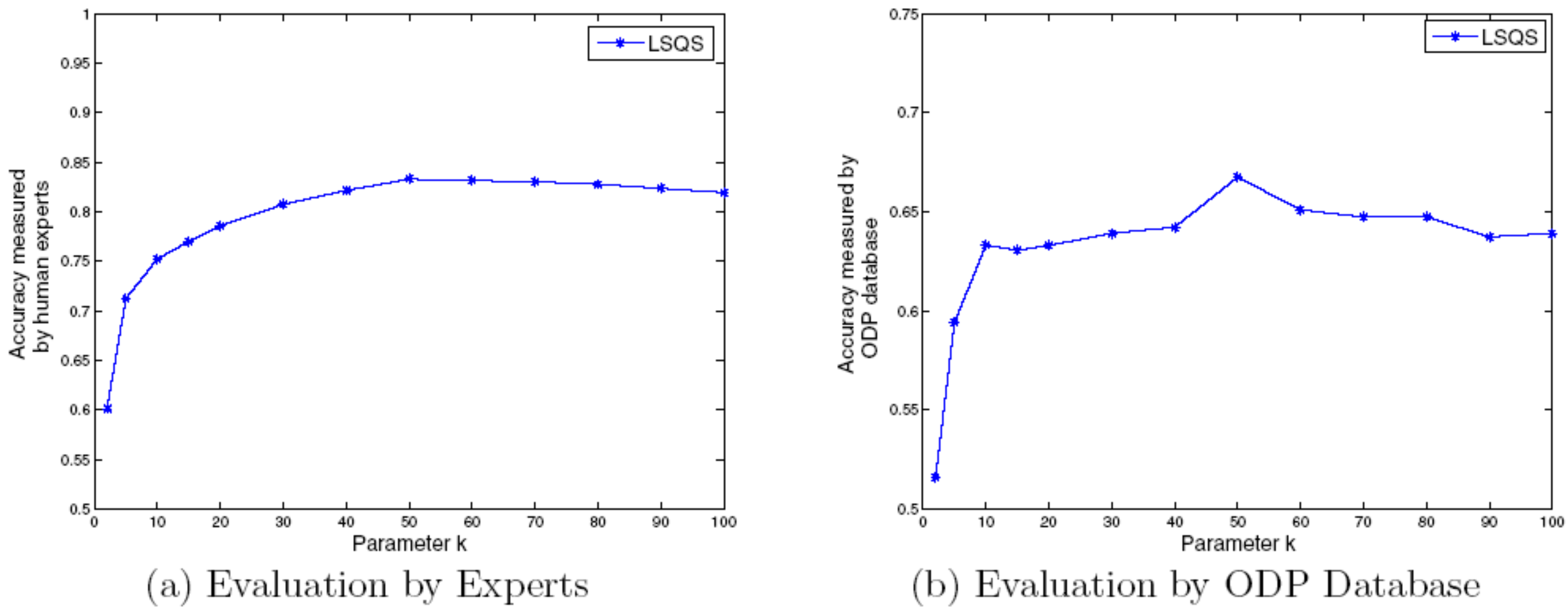
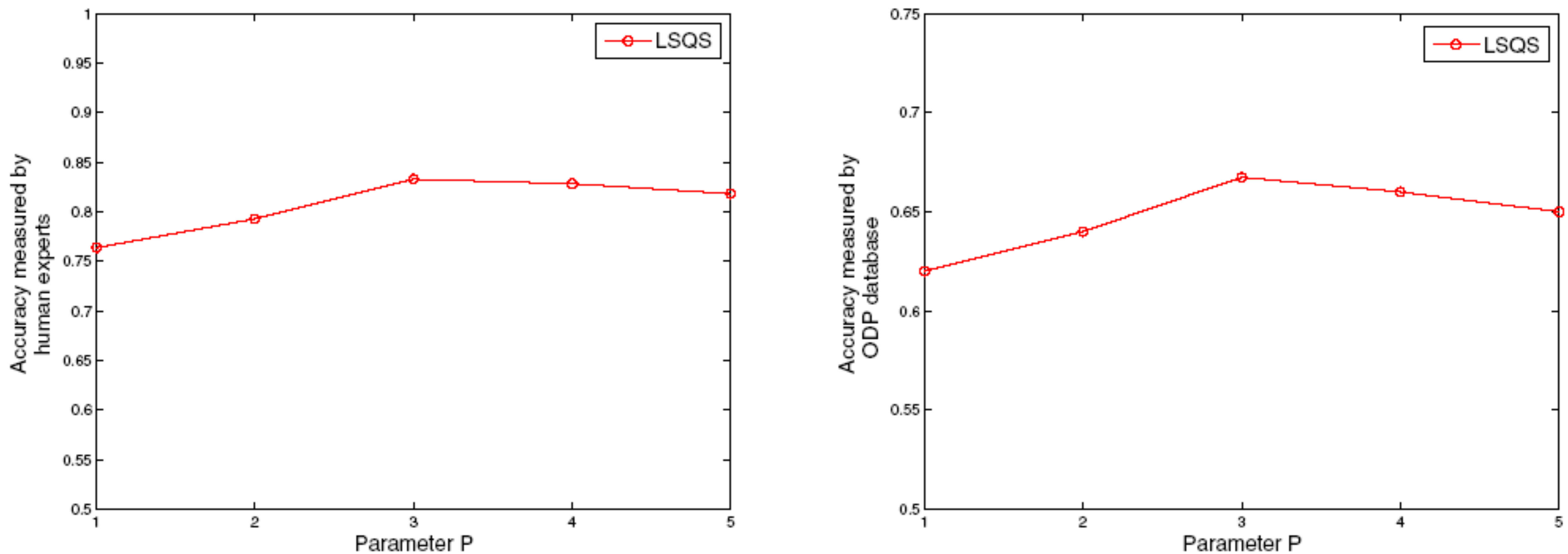


Figure 2: Impact of Parameter k ($P = 3$)

Impact of Parameter *P*

To test the propagation influence

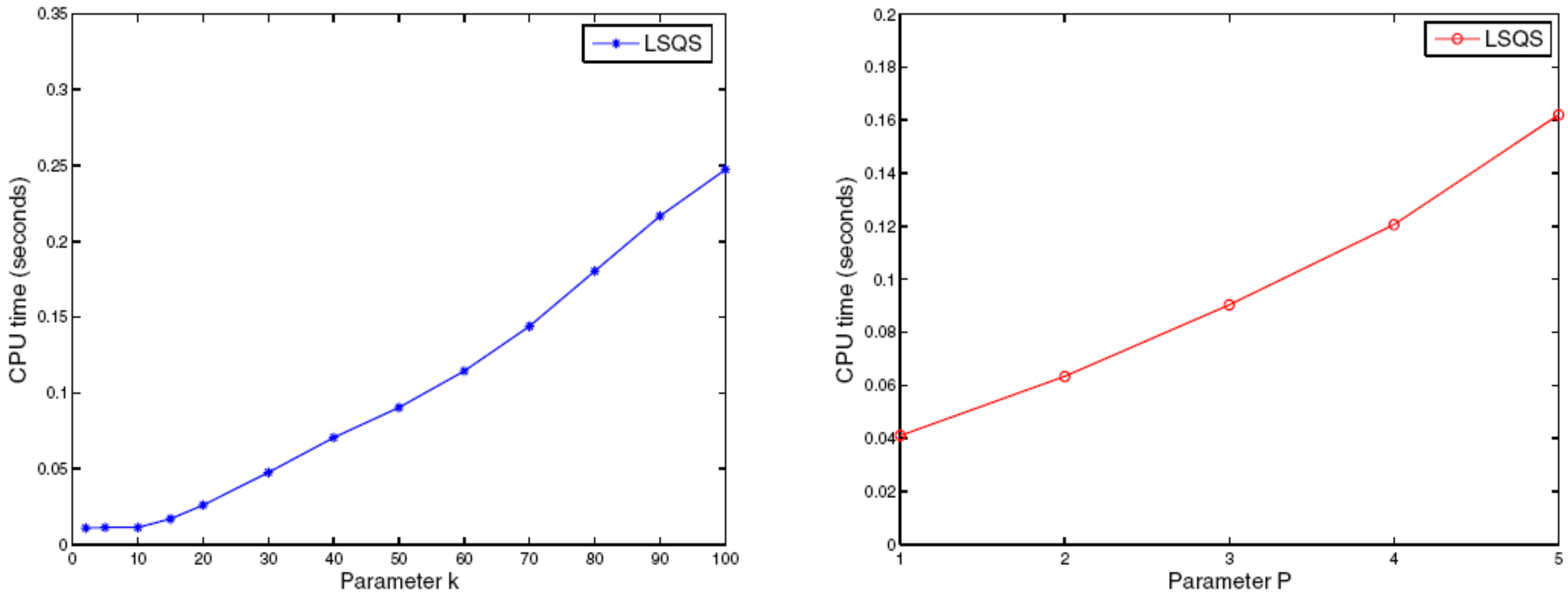


(a) Evaluation by Experts

(b) Evaluation by ODP Database

Figure 3: Impact of Parameter P ($k = 50$)

Efficiency Analysis



(a) $P = 3$ (b) $k = 50$

Figure 4: Efficiency Analysis

Complexity Analysis

- Complexity of the gradient descent calculation of function $\mathcal{H}$ is

$$\frac{\partial \mathcal{H}}{\partial U},\ \frac{\partial \mathcal{H}}{\partial Q},\ \text{and}\ \frac{\partial \mathcal{H}}{\partial L} = O(\rho_R d),\ O(\rho_R d + \rho_S d),\ \text{and}\ O(\rho_S d)$$

- Complexity of the heat diffusion method is

$$O(h \cdot k^3)$$

Conclusion

- Propose an offline novel **joint matrix factorization** method using **user-query** and **query-URL bipartite graphs** for learning query features
- Propose an online diffusion-based **similarity propagation** and **ranking method** for query suggestion
- To investigate how *rank*, *refinement*, and *temporal* information can be used effectively for query suggestion

Q & A

http://www.cse.cuhk.edu.hk/~king