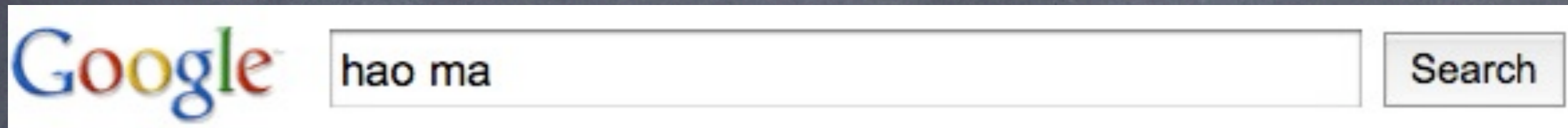


Diversifying Query Suggestion Results

Query Suggestion

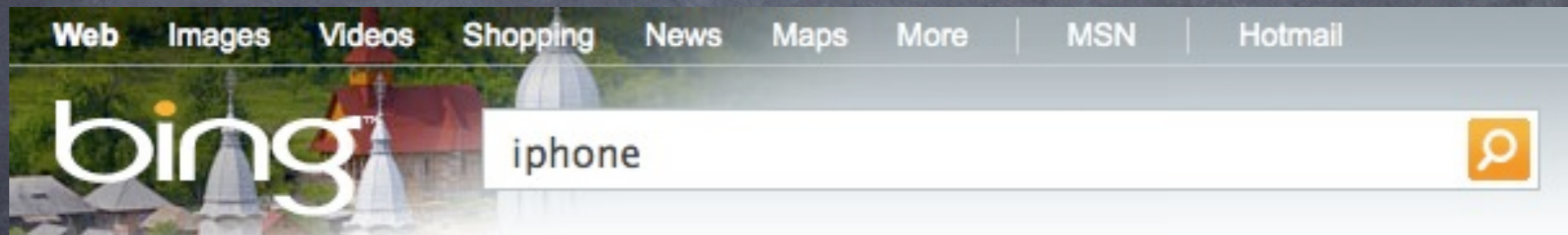


Searches related to **hao ma**

[hao ma cuhk](#)

[ni hao ma](#)

[ni hao ma means](#)



Related Searches

[iPhone Price](#)

[iPhone Cingular](#)

[iPhone Apps](#)

[iPhones Instructions](#)

[iPhone AT&T](#)

[iPhone Taiwan](#)

[Unlock iPhone](#)

[Buy iPhone](#)

Query Suggestion

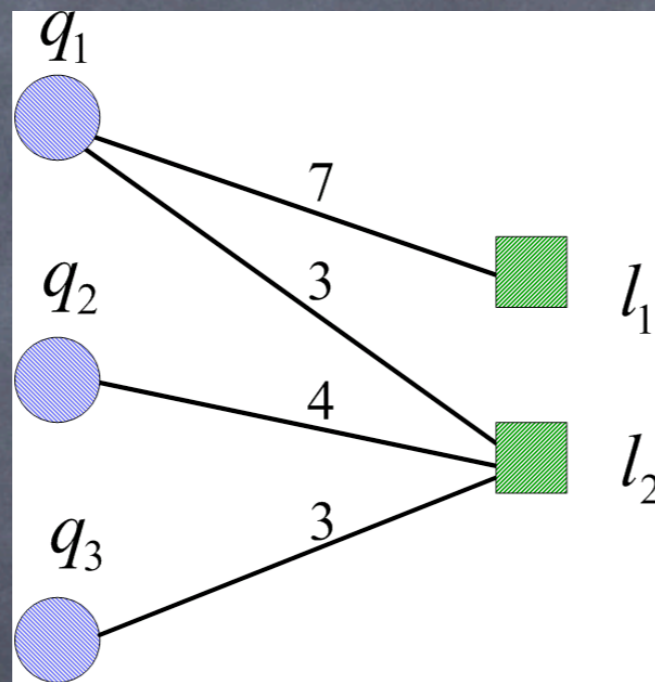
The screenshot shows the Yahoo search engine interface. At the top left is the Yahoo logo. To its right are navigation links: Web, Images, Video, Local, Shopping, News, and More. The search input field contains the text 'apache'. Below the input field is a dropdown menu with suggestions: 'apache indians', 'apache web server', 'ski apache', 'apache junction az', and 'apache corp'. To the right of these suggestions is a section titled 'Explore related concepts:' with links to 'Apache Software Foundation', 'Chiricahua', 'Western Apache', and 'Apache web server'.

The screenshot shows the Yahoo search engine interface. At the top left is the Yahoo logo. To its right are navigation links: Web, Images, Video, Local, Shopping, News, and More. The search input field contains the text 'apple'. Below the input field is a dropdown menu with suggestions: 'apple ipod', 'applebee's', 'apple iphone', 'apple store', and 'apple trailers'.

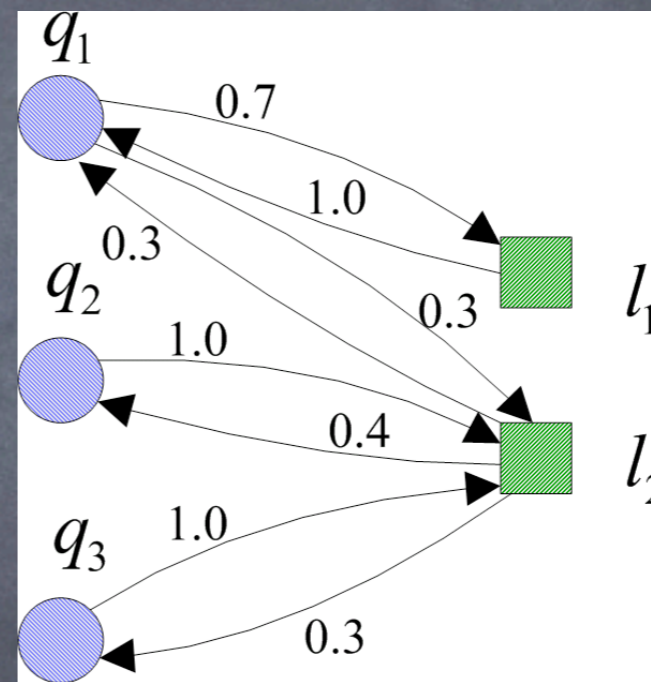
Query Suggestion

- We need to diversify query suggestions
 - Satisfy different users
 - Boost long tail queries

Query Suggestion



(a)

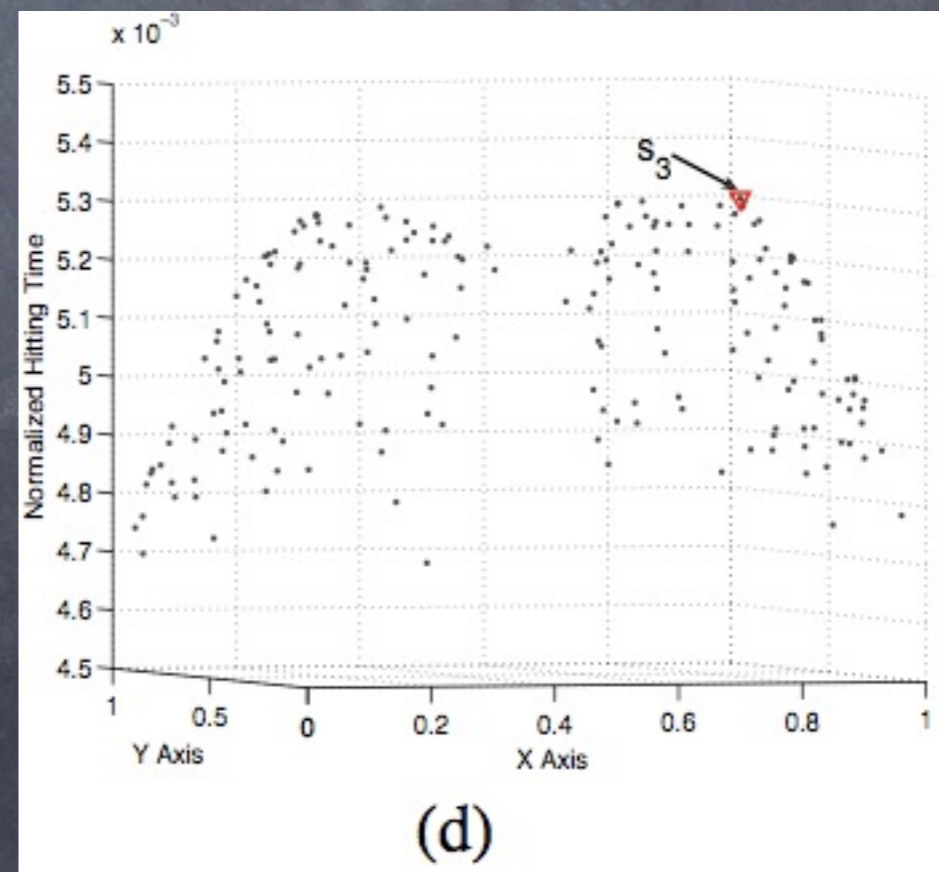
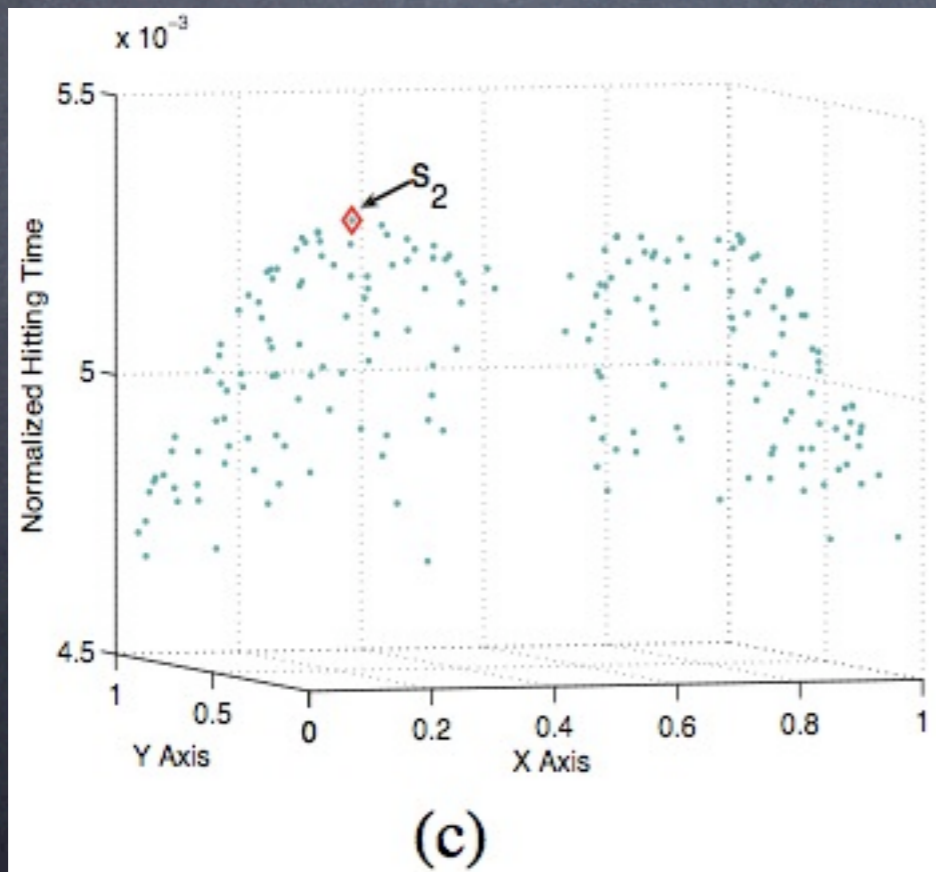
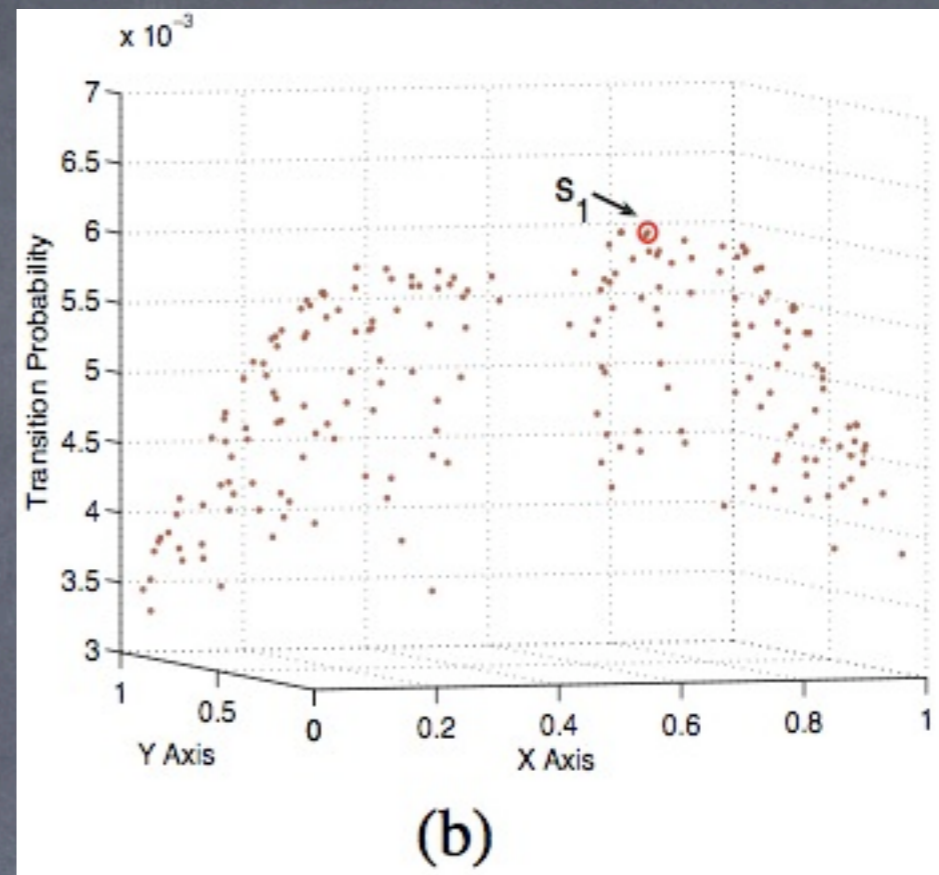
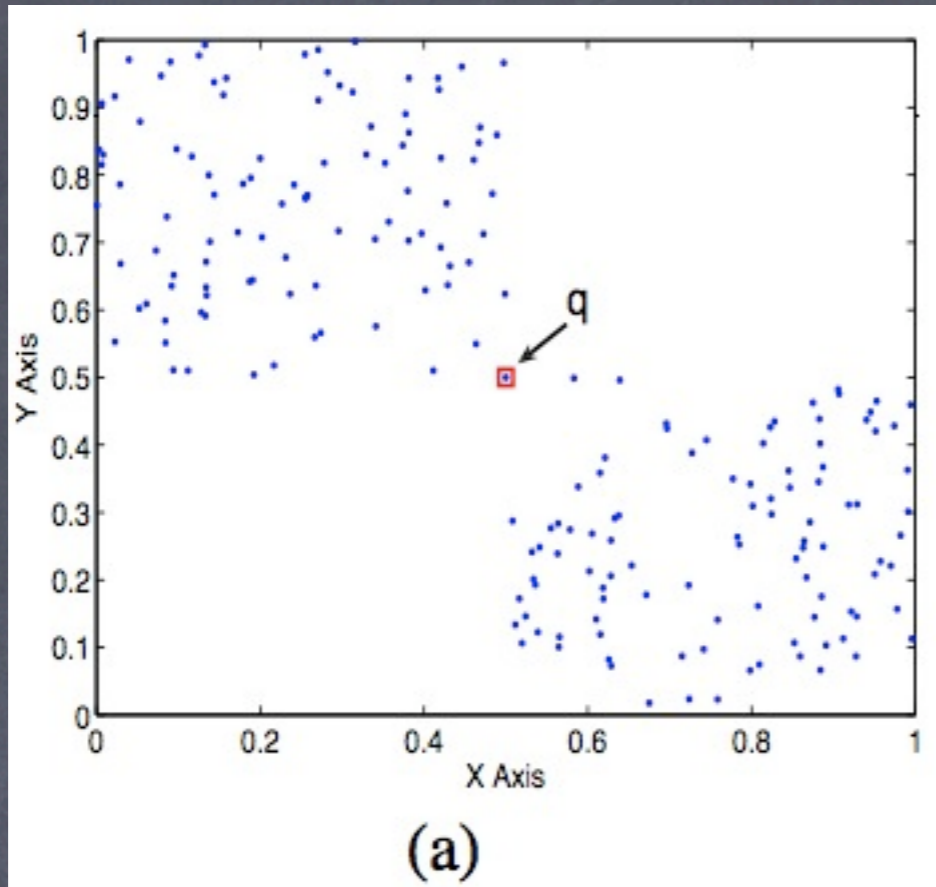


(b)

Figure 1: Example for Query Suggestion. (a) Query-URL Bipartite Graph. (b) Converted Query-URL Bipartite Graph

Diversifying

- Determine the first query suggestion
 - Random walk
- Determine the rest query suggestions
 - Hitting time analysis



Algorithm 1: Diversifying Query Suggestion Algorithm

Input: Transition probability matrix \mathbf{P} of a click bipartite graph \mathcal{G} and a query node $q \in \mathcal{Q}$

Output: A ranked list of K suggested queries

- 1: Given the query node q , form a $1 \times n$ vector v (n is the total number of nodes), with the entry of query node q equal to 1 while other entries equal to 0.
- 2: Perform a t -step random walk and get a new vector $v^* = v\mathbf{P}^t$. Get the top query node s_1 with the highest value in vector v^* (notice that, this node should be in the query node set \mathcal{Q}).
- 3: Add query nodes q and s_1 into a subset \mathcal{S} , and repeat the following statements until K queries are suggested:
 - (a) For all queries except the ones in the subset \mathcal{S} , iterate

$$h_l(i|\mathcal{S}) = 1 + \sum_{j \in \mathcal{N}(i)} p_0(i, j) h_{l-1}(j|\mathcal{S})$$

for a predefined number of l iterations starting with $h_0(i|\mathcal{S}) = 0$.

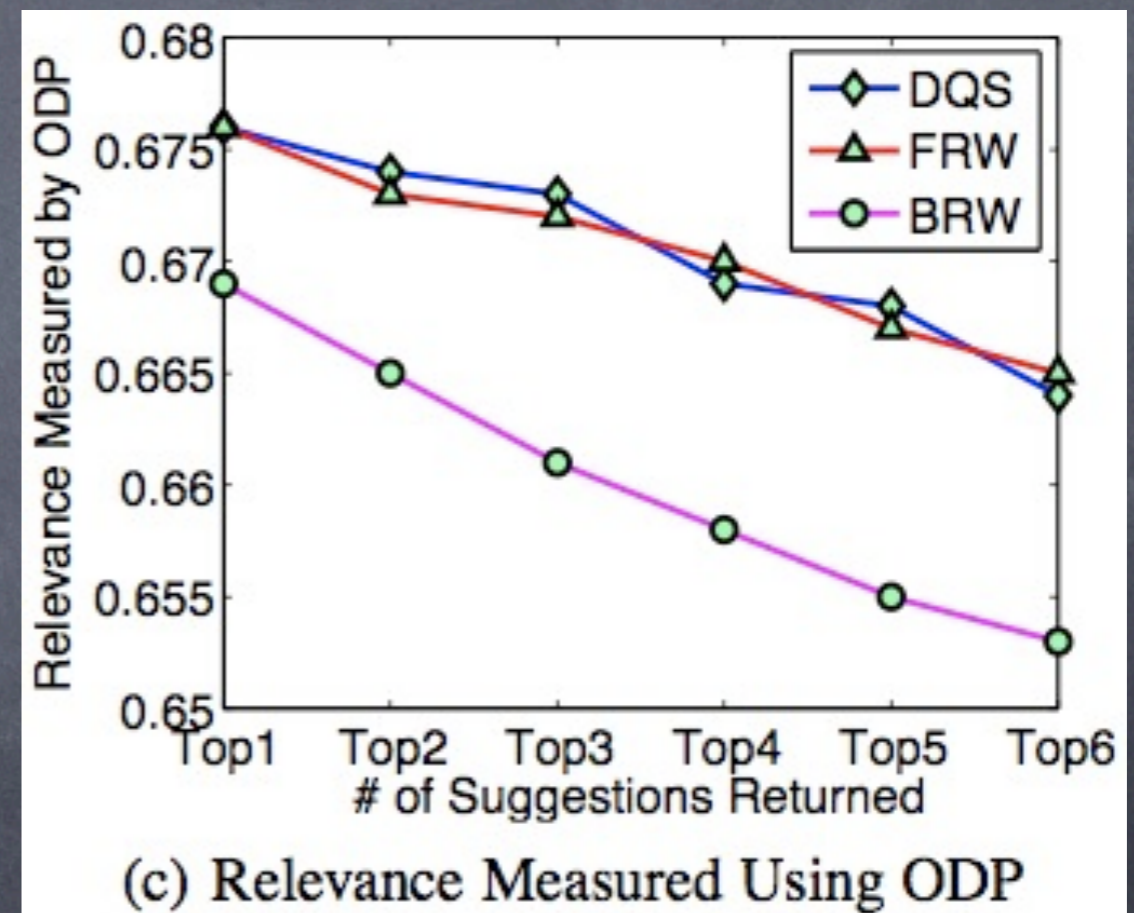
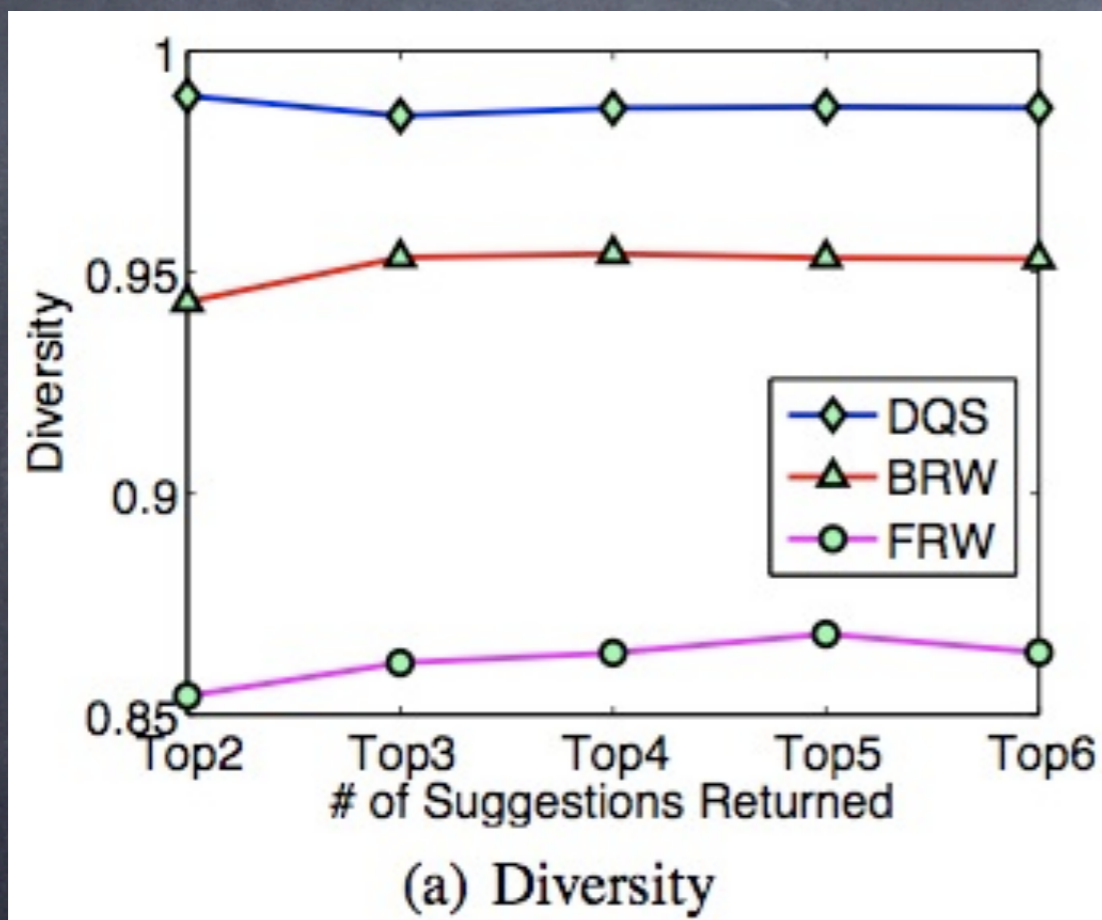
- (b) Pick next suggested query as $\operatorname{argmax}_i h_l(i|\mathcal{S})$, and add it into the subset \mathcal{S} .
-

Some Results

Table 1: Query Suggestion Comparisons between DQS and other Methods

Query = nike			Query = mp3		
DQS	FRW	BRW	DQS	FRW	BRW
nike shoes nike yoga jordans basketball shoes nike id ronaldinho	nike shoes nike shox niketown nikes nike outlet nike town	nike clothing shoes nike nike sandals nikes nike logo nike clothes	winamp aol music walmart music mp3 download creative cdnow	winamp mp3 players free music downloads limewire free music sound click	best mp3 player mp3 files lame mp3 lame lame mp3 encoder lame encoder
Query = sony			Query = nikon		
DQS	FRW	BRW	DQS	FRW	BRW
sony psp pictures ps3 sony ericsson psp games sony canada	sony psp psp sony tv sony vaio sony style sonystyle	sony singapore sony customer service sony corporation sony vaio drivers sony e sony service	nikon cameras canon ritz camera rifle scopes nikon d300 wolf camera	nikon cameras nikon usa nikon camera nikonusa nikon lenses ritz camera	nikon products nikonusa nikon lenses nikon usa nikon digital slr nikon digital

Evaluation



Thanks!
Q&A