Query Log Analysis for Enhancing Web Search

Salvatore Orlando, University of Venice, Italy
Fabrizio Silvestri, ISTI - CNR, Pisa, Italy

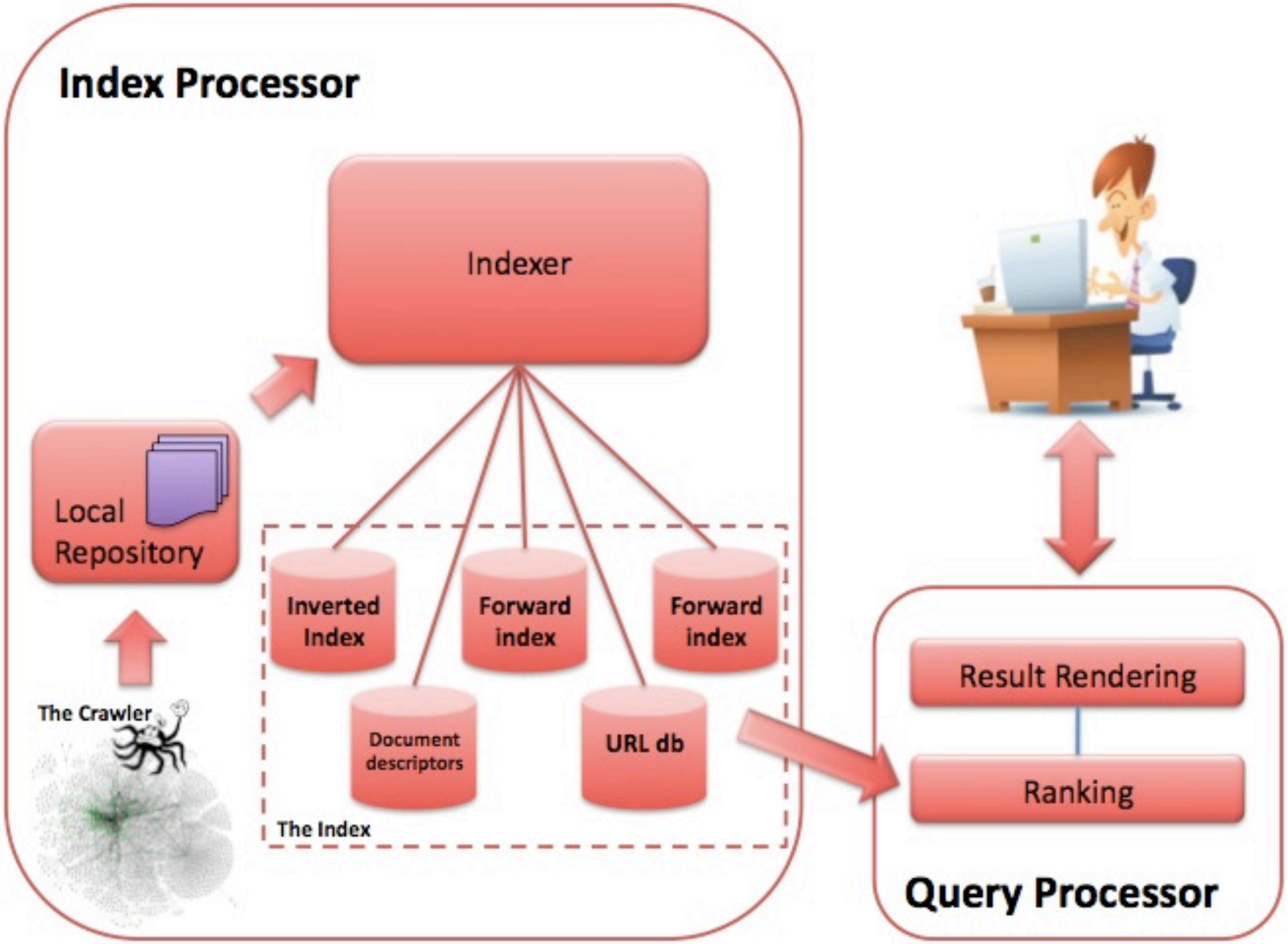
Tutorial at IEEE / WIC / ACM WI/IAT'09
September 15-18, 2009 - Milan, Italy

Query Log Analysis for Enhancing Web Search

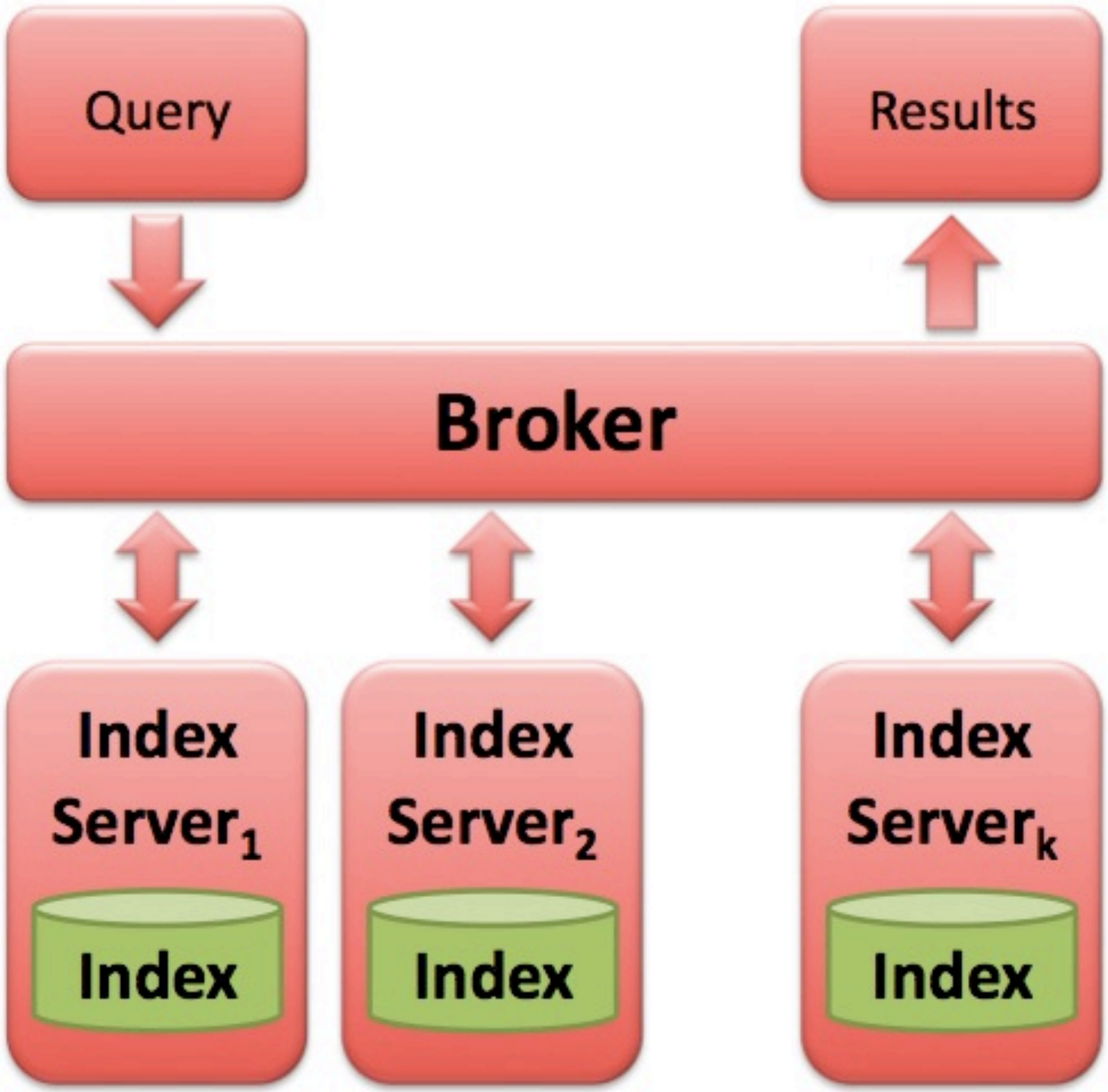
Salvatore Orlando, University of Venice, Italy
Fabrizio Silvestri, ISTI - CNR, Pisa, Italy

Tutorial at IEEE / WIC / ACM WI/IAT'09
September 15-18, 2009 - Milan, Italy

What is a Web Search Engine



Real Web Search Engines



History in Search Engines

Alphonse de Lamartine

Source: Wikipedia

History in Search Engines

Alphonse de Lamartine

Source: Wikipedia

History Teaches Everything... Even the Future!

What is History?

- Past Queries
- Query Sessions
- Click-through Data

Web Mining

- Content:
  - text & multimedia mining
- Structure:
  - link analysis, graph mining
- Usage:
  - log analysis, query mining
- Relate all of the above
  - Web characterization
  - Particular applications

Web Mining

- Content:
  - text & multimedia mining
- Structure:
  - link analysis, graph mining
- Usage:
  - log analysis, query mining
- Relate all of the above
  - Web characterization
  - Particular applications

Dynamic

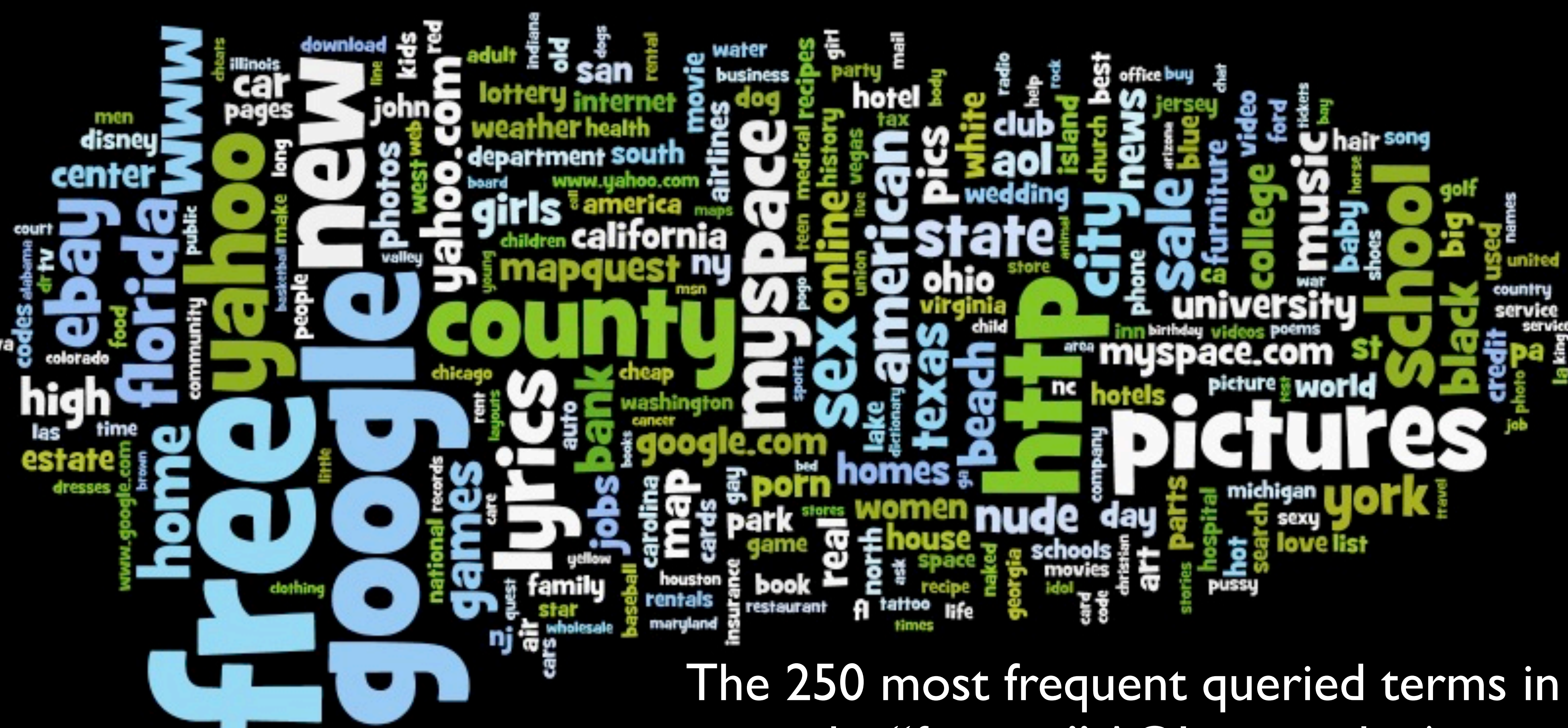
Tutorial Outline

- Query Logs
- Data Mining Techniques for QL Mining
- Enhancing Effectiveness of Search Systems
- Enhancing Efficiency of Search Systems

Tutorial Outline

- Query Logs
    - The Nature of Queries
    - User Actions
- Data Mining Techniques for QL Mining
- Enhancing Effectiveness of Search Systems
- Enhancing Efficiency of Search Systems

What's in Query Logs?



The 250 most frequent queried terms in the "famous" AOL query log!

Thanks to http://www.wordle.net for the tagcloud generator

I Love Alaska!

- http://www.minimovies.org/documentaires/view/ilovealaska
- "I love Alaska tells the story of one of those AOL users. We get to know a religious middle-aged woman from Houston, Texas, who spends her days at home behind her TV and computer. Her unique style of phrasing combined with her putting her ideas, convictions and obsessions into AOL's search engine, turn her personal story into a disconcerting novel of sorts.

  Over a period of three months, a portrait of a woman emerges who is diligently searching for likeminded souls. The list of her search queries read aloud by a voice-over reads like a revealing character study of a somewhat obese middle-aged lady in her menopause, who is looking for a way to rejuvenate her sex life. In the end, when she cheats on her husband with a man she met online, her life seems to crumble around her. She regrets her deceit, admits to her Internet addiction and dreams of a new life in Alaska."

Query Logs Analyzed in the Literature

Query log name	Public	Period	# Queries	# Sessions	# Users
Excite '97	Y	Sep '97	1,025,908	211,063	$\sim 410,360$
Excite '97 (small)	Y	Sep '97	51,473	N.D.	$\sim 18,113$
Altavista	N	Aug 2^{nd} - Sep 13^{th} '98	993,208,159	285,474,117	N.D.
Excite '99	Y	Dec '99	1,025,910	325,711	$\sim 540,000$
Excite '01	Y	May '01	1,025,910	262,025	$\sim 446,000$
Altavista (public)	Y	Sep '01	7,175,648	N.D.	N.D.
Tiscali	N	Apr '02	3,278,211	N.D.	N.D.
TodoBR	Y	Jan - Oct '03	22,589,568	N.D.	N.D.
TodoCL	N	May – Nov '03	N.D.	N.D.	N.D.
AOL (big)	N	Dec 26^{th} '03 – Jan 1^{st} '04	$\sim 100,000,000$	N.D.	$\sim 50,000,000$
Yahoo!	N	Nov '05 – Nov '06	N.D.	N.D.	N.D.
AOL (small)	Y	Mar 1^{st} - May 31^{st} '06	36,389,567	N.D.	N.D.

Caveat Emptor!

Caveat Emptor!

- We will show results from published papers.

Caveat Emptor!

- We will show results from published papers.
- No results have been computed for the purpose of this Tutorial.

Caveat Emptor!

- We will show results from published papers.
- No results have been computed for the purpose of this Tutorial.
- No query logs were harmed during the preparation of this tutorial :-)

Some Popular Terms: Excite and Altavista

query	freq.
Empty Query	2,586
sex	229
chat	58
lucky number generator	56
p****	55
porno	55
b****y	55
nude beaches	52
playboy	46
bondage	46
porn	45
rain forest restaurant	40
f****ing	40
crossdressing	39
crystal methamphetamine	36
consumer reports	35
xxx	34
nude tanya harding	33
music	33
sneaker stories	32

(a) Excite.

query	freq.
christmas photos	31,554
lyrics	15,818
cracks	12,670
google	12,210
gay	10,945
harry potter	7,933
wallpapers	7,848
pornografia	6,893
"yahoo com"	6,753
juegos	6,559
lingerie	6,078
symbios logic 53c400a	5,701
letras de canciones	5,518
humor	5,400
pictures	5,293
preteen	5,137
hypnosis	4,556
cpc view registration key	4,553
sex stories	4,521
cd cover	4,267

(b) Altavista.

Fabrizio Silvestri: **Mining Query Logs: Turning Search Usage Data into Knowledge**.
Foundations and Trends in Information Retrieval. (To Appear).

Topic Distribution: Excite and AOL

Topic	Percentage
Entertainment or recreation	19.9%
Sex and pornography	16.8%
Commerce, travel, employment, or economy	13.3%
Computers or Internet	12.5%
Health or sciences	9.5%
People, places, or things	6.7%
Society, culture, ethnicity, or religion	5.7%
Education or humanities	5.6%
Performing or fine arts	5.4%
Non-English or unknown	4.1%
Government	3.4%

Excite

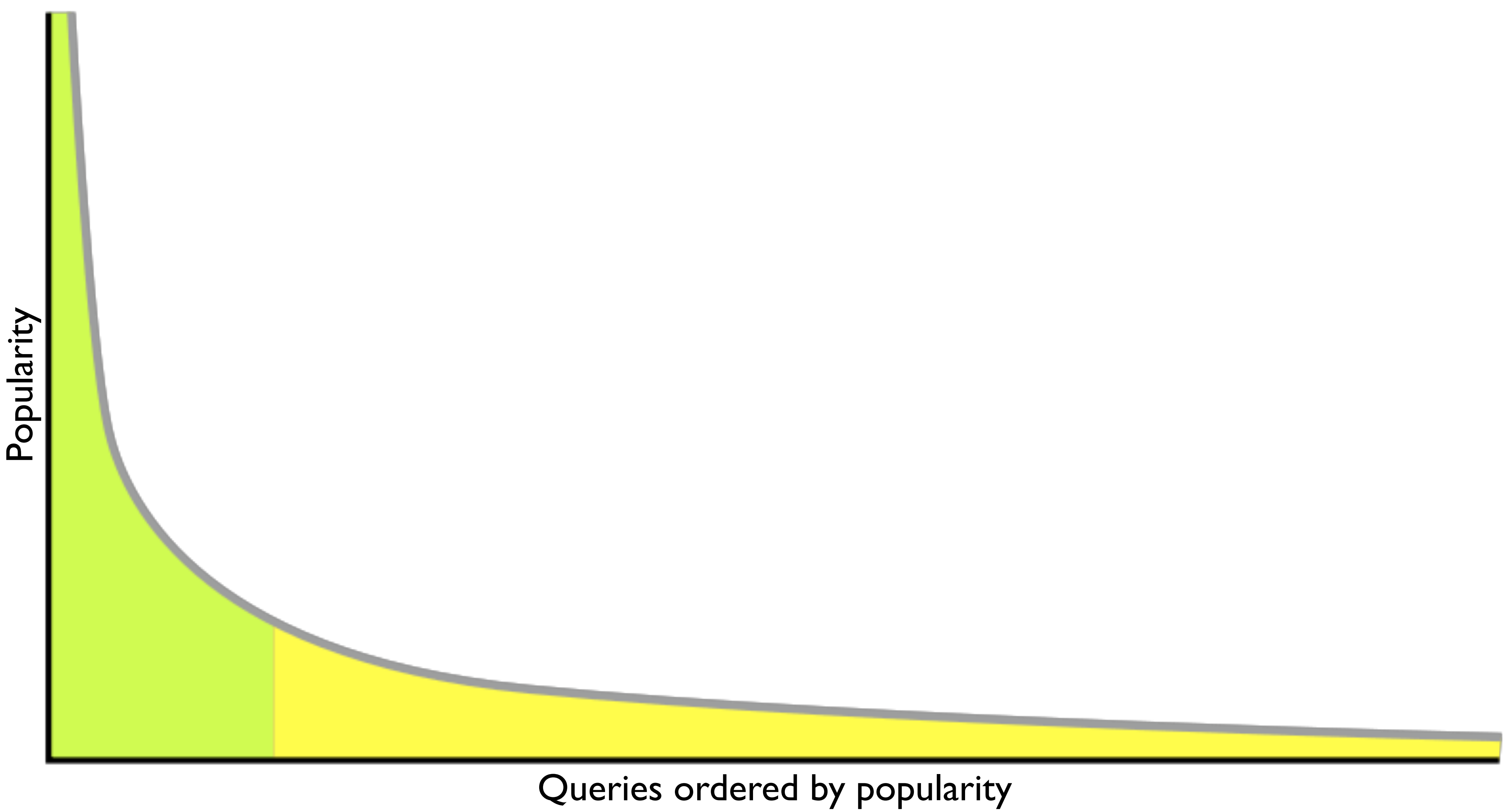
Topic	Percentage
Entertainment	13%
Shopping	13%
Porn	10%
Research & learn	9%
Computing	9%
Health	5%
Home	5%
Travel	5%
Games	5%
Personal & Finance	3%
Sports	3%
US Sites	3%
Holidays	1%
Other	16%

AOL

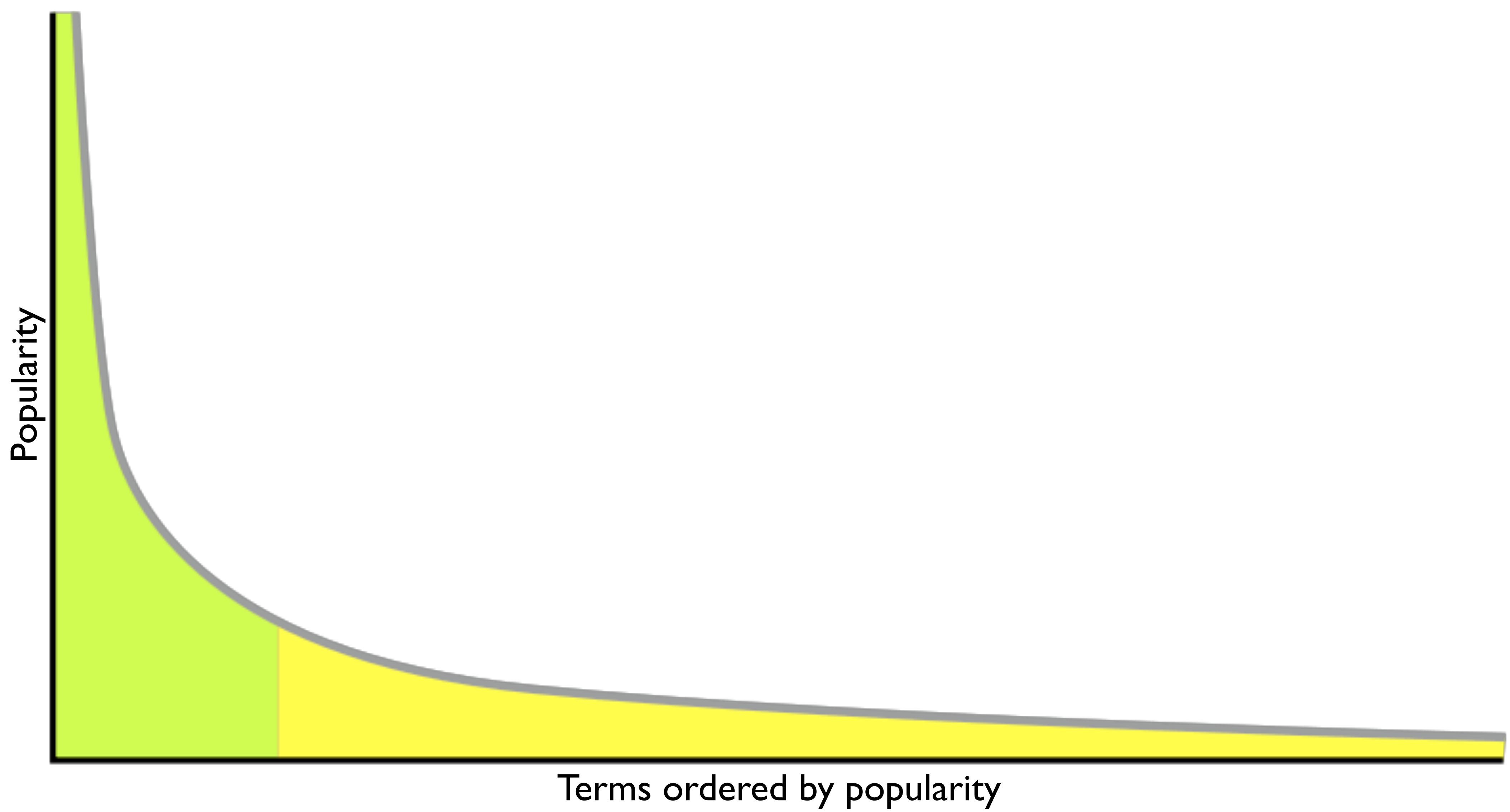
A. Spink, B. J. Jansen, D. Wolfram, and T. Saracevic, "**From e-sex to e-commerce: Web search changes**," Computer, vol. 35, no. 3, pp. 107–109, 2002.

S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman, "**Temporal analysis of a very large topically categorized web query log**," J. Am. Soc. Inf. Sci. Technol., vol. 58, no. 2, pp. 166–178, 2007.

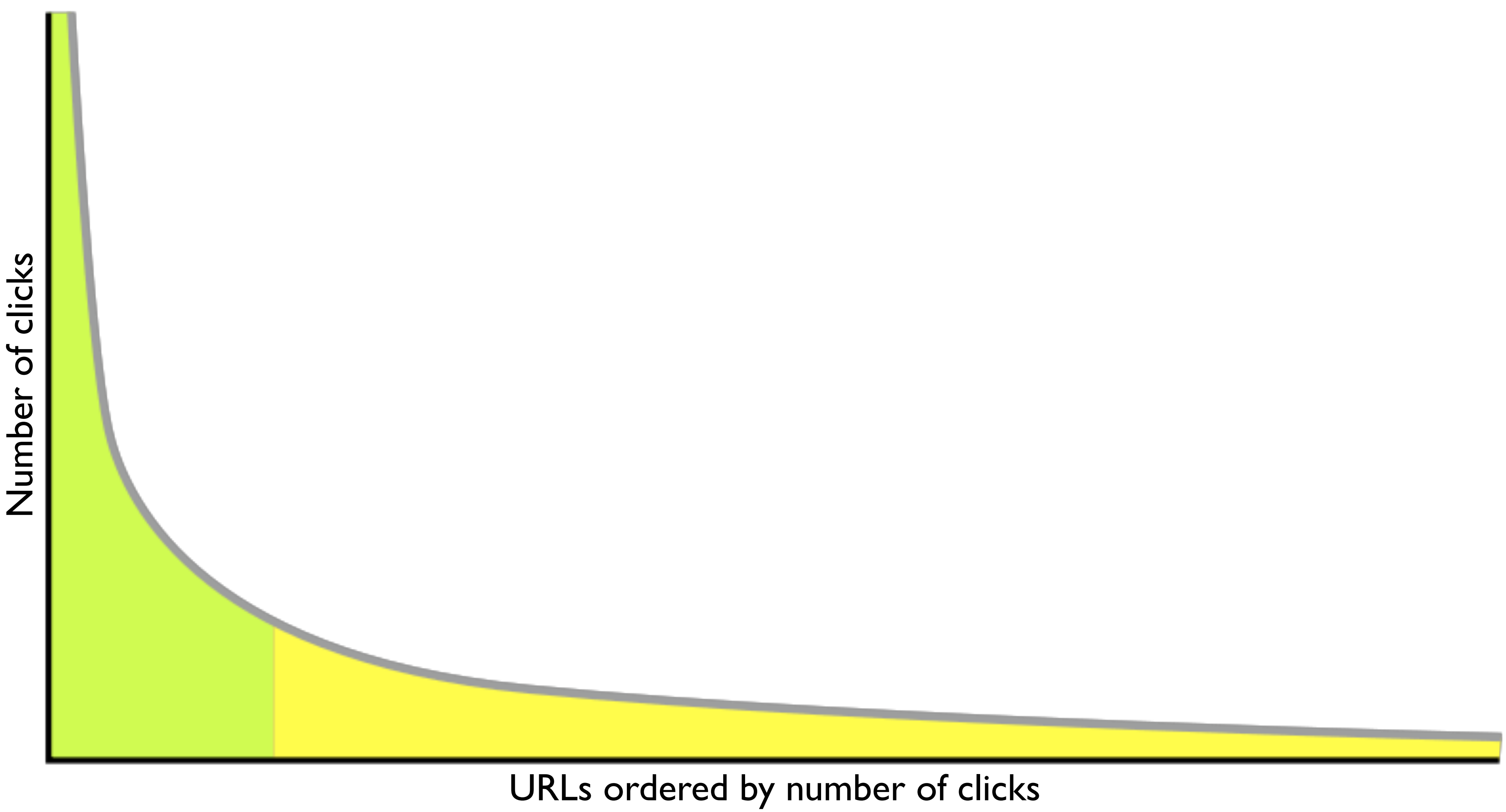
Long Tail Distribution



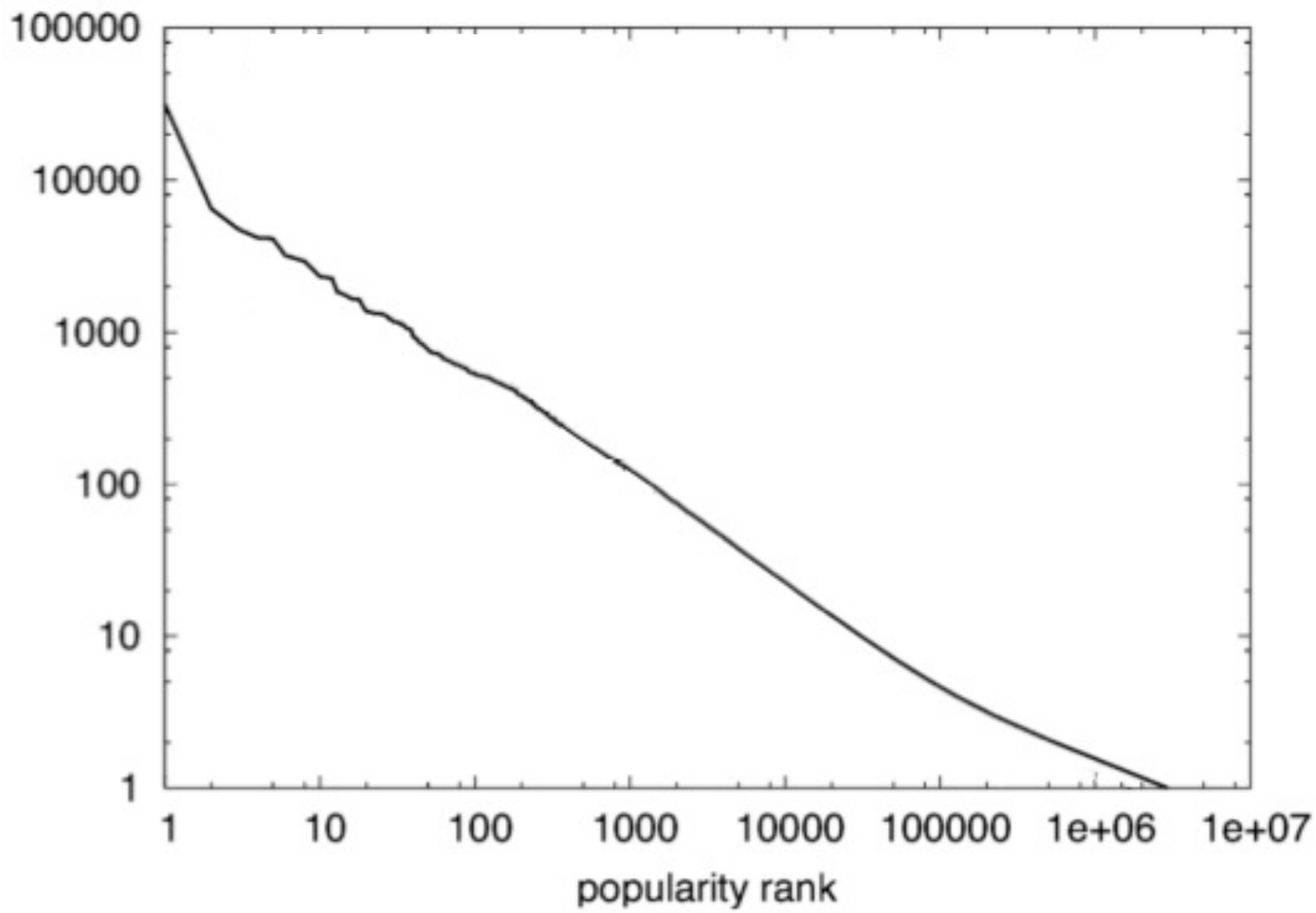
Long Tail Distribution



Long Tail Distribution

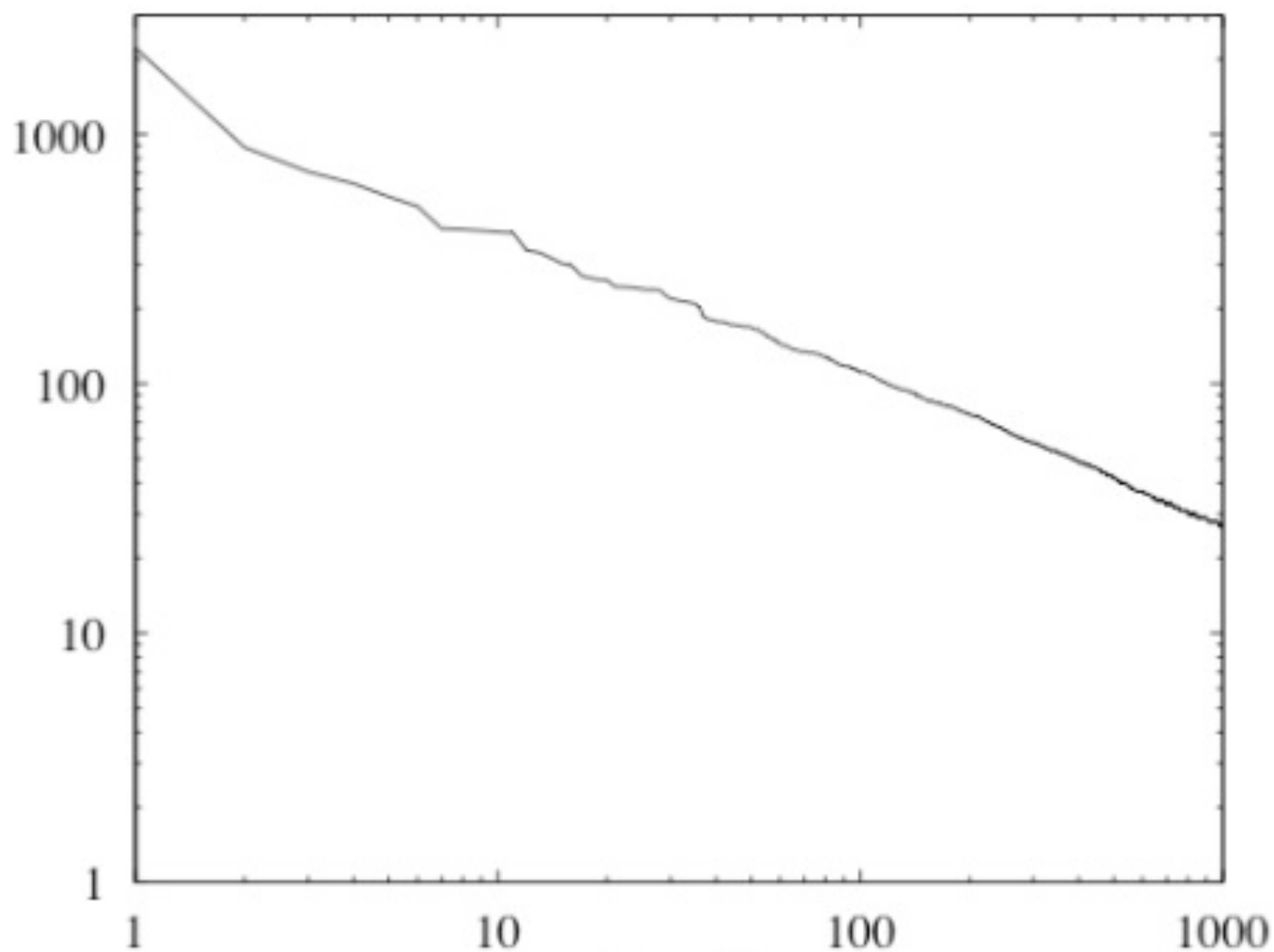


Power-Law In Query Popularity: Altavista



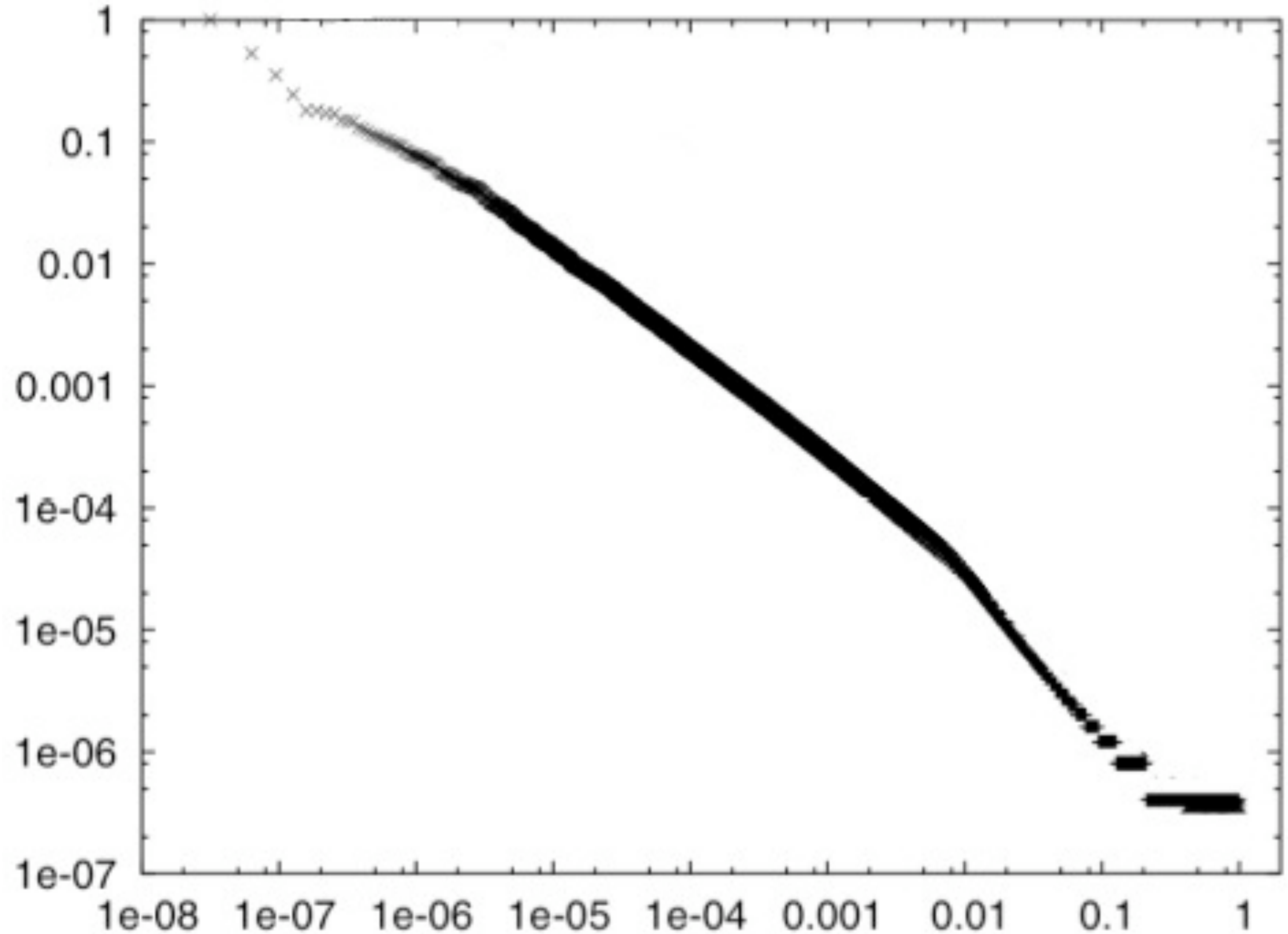
T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

Power-Law In Query Popularity: Excite



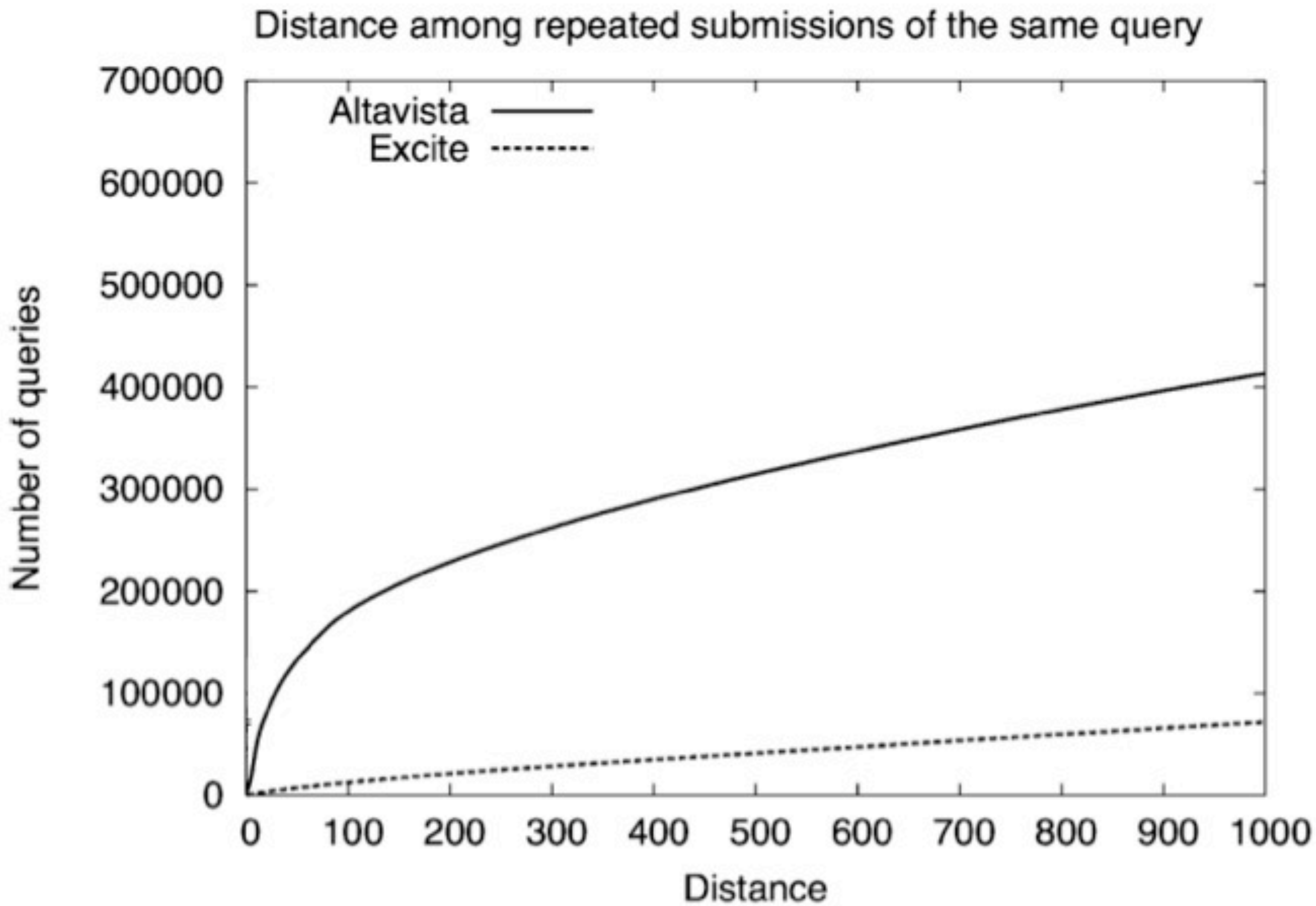
T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

Power-Law In Query Popularity: Yahoo!



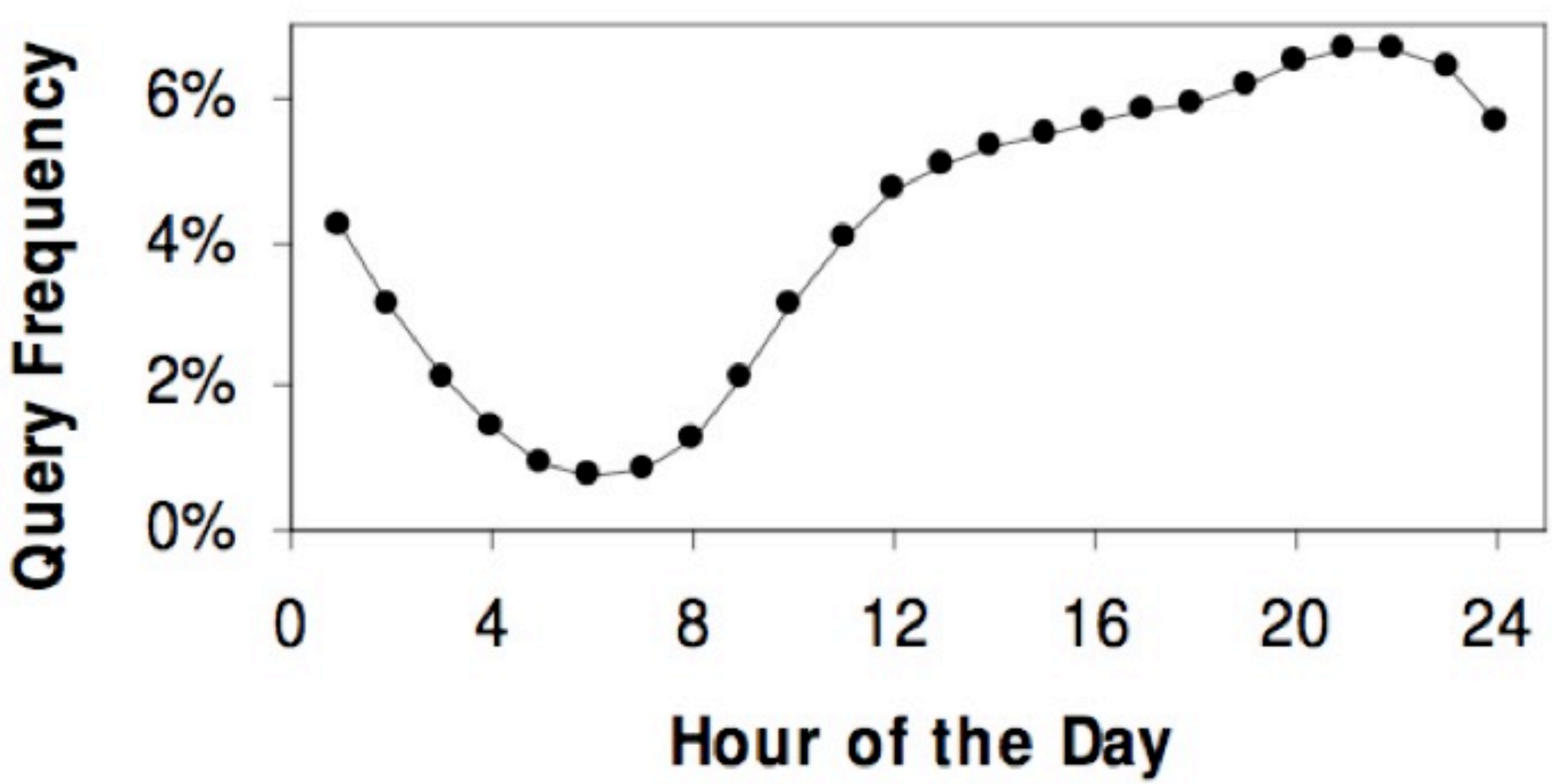
R. Baeza-Yates, A. Gionis, F. P. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri, "**Design trade-offs for search engine caching**," ACM Trans. Web, vol. 2, no. 4, pp. 1–28, 2008.

Query Resubmission



T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

Frequency of Query Submission



S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman, "T**emporal analysis of a very large topically categorized web query log**," J. Am. Soc. Inf. Sci. Technol., vol. 58, no. 2, pp. 166–178, 2007.

Query Statistics: Excite

Characteristic	1997	1999	2001
Mean terms per query	2,4	2,4	2,6
Terms per query			
1 term	26,3%	29,8%	26,9%
2 terms	31,5%	33,8%	30,5%
3+ terms	43,1%	36,4%	42,6%
Mean queries per user	2,5	1,9	2,3

A. Spink, B. J. Jansen, D. Wolfram, and T. Saracevic, "**From e-sex to e-commerce: Web search changes**," Computer, vol. 35, no. 3, pp. 107–109, 2002.

Query Statistics: Excite

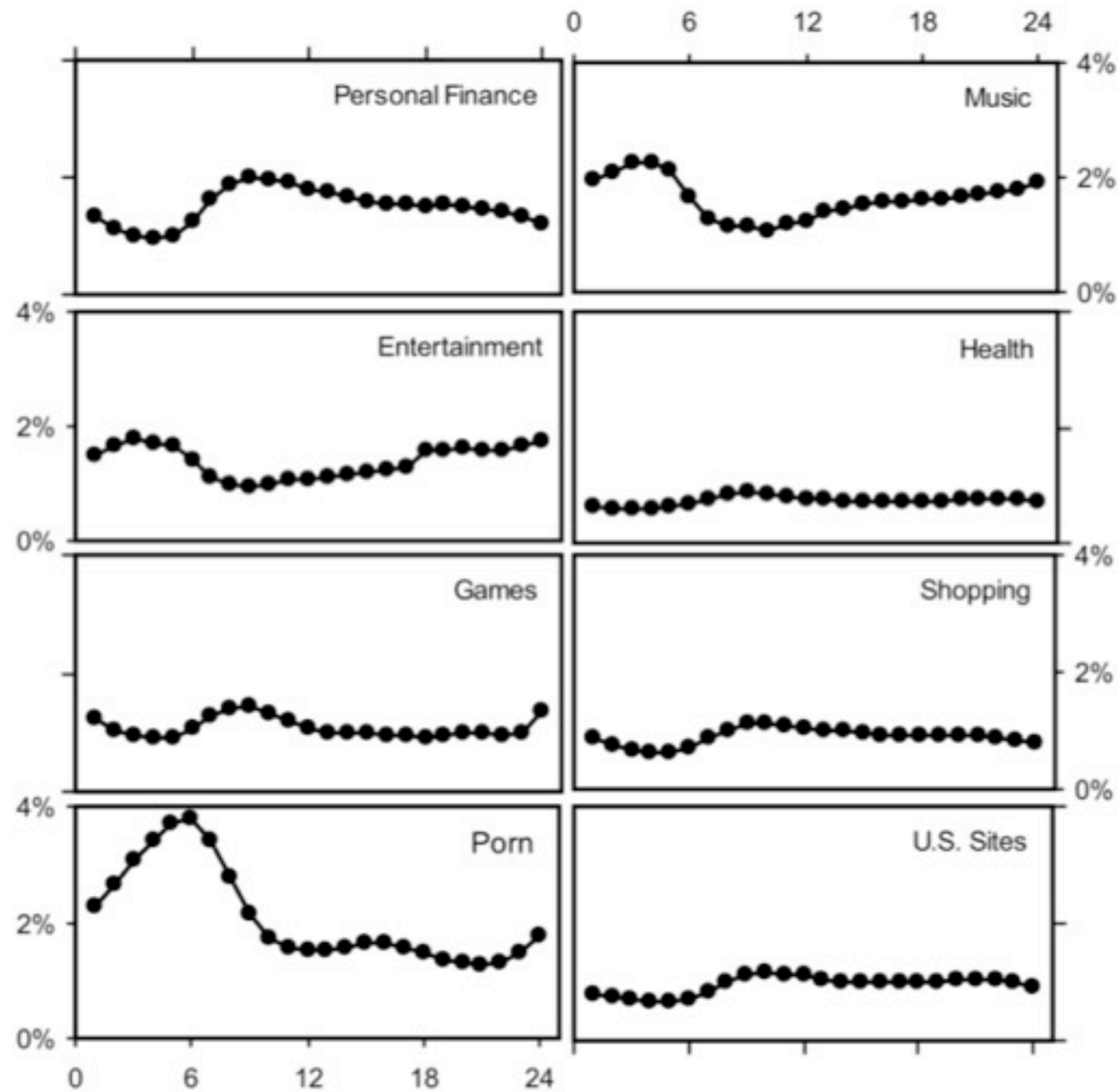
Characteristic	1997	1999	2001
Mean terms per query	2,4	2,4	2,6
Terms per query			
1 term			
2 terms			
3+ terms			
Mean queries per user	2,5	1,9	2,3

In 2008: 2.5 terms per query.

R. Baeza-Yates, A. Gionis, F. P. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri, "**Design trade-offs for search engine caching**," ACM Trans. Web, vol. 2, no. 4, pp. 1–28, 2008.

A. Spink, B. J. Jansen, D. Wolfram, and T. Saracevic, "**From e-sex to e-commerce: Web search changes**," Computer, vol. 35, no. 3, pp. 107–109, 2002.

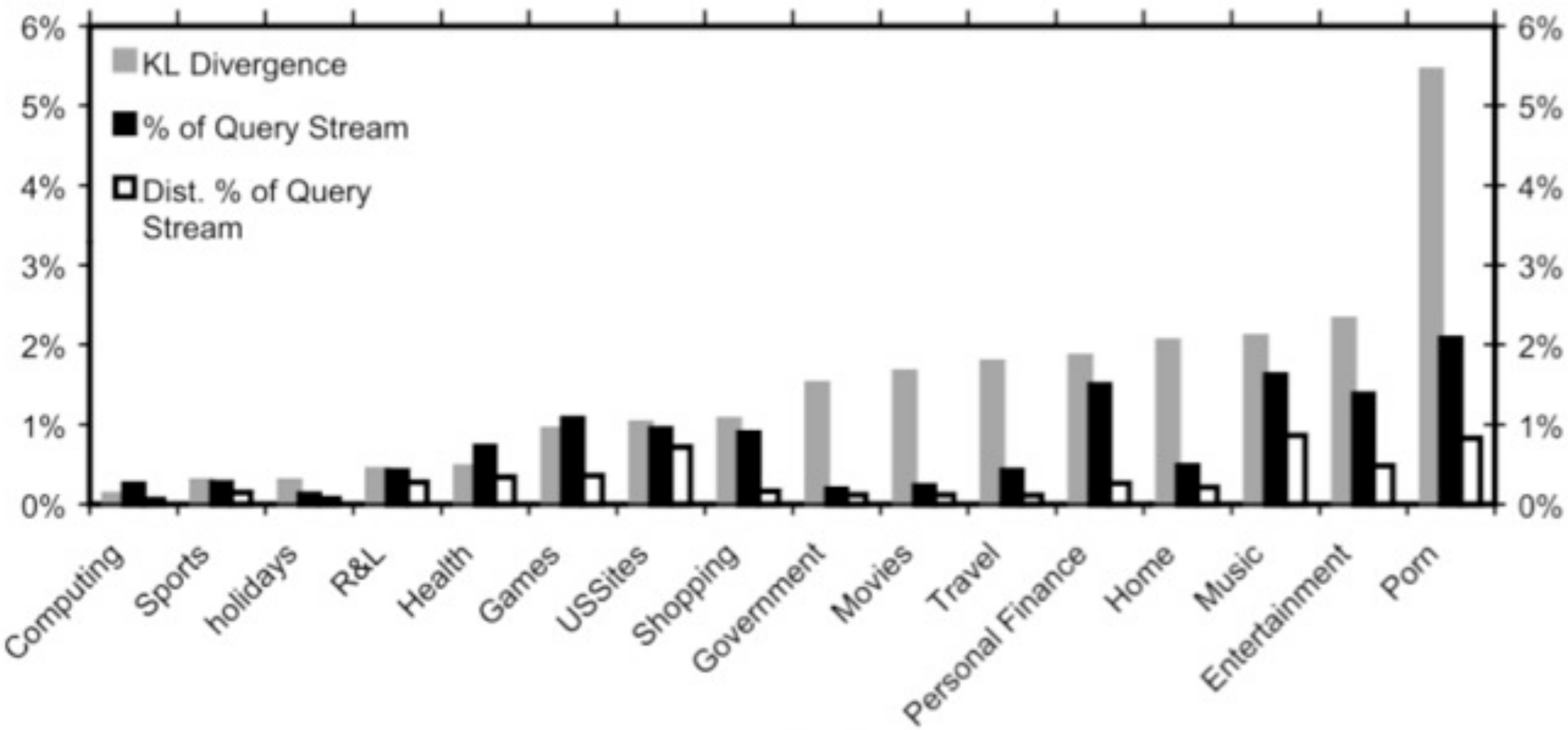
Hourly Topic Distribution



S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman, "T**emporal analysis of a very large topically categorized web query log**," J. Am. Soc. Inf. Sci. Technol., vol. 58, no. 2, pp. 166–178, 2007.

Surprising Topics

- KL-Divergence between the probability distribution of observing a query topic u.a.r. and the actual topic observed.

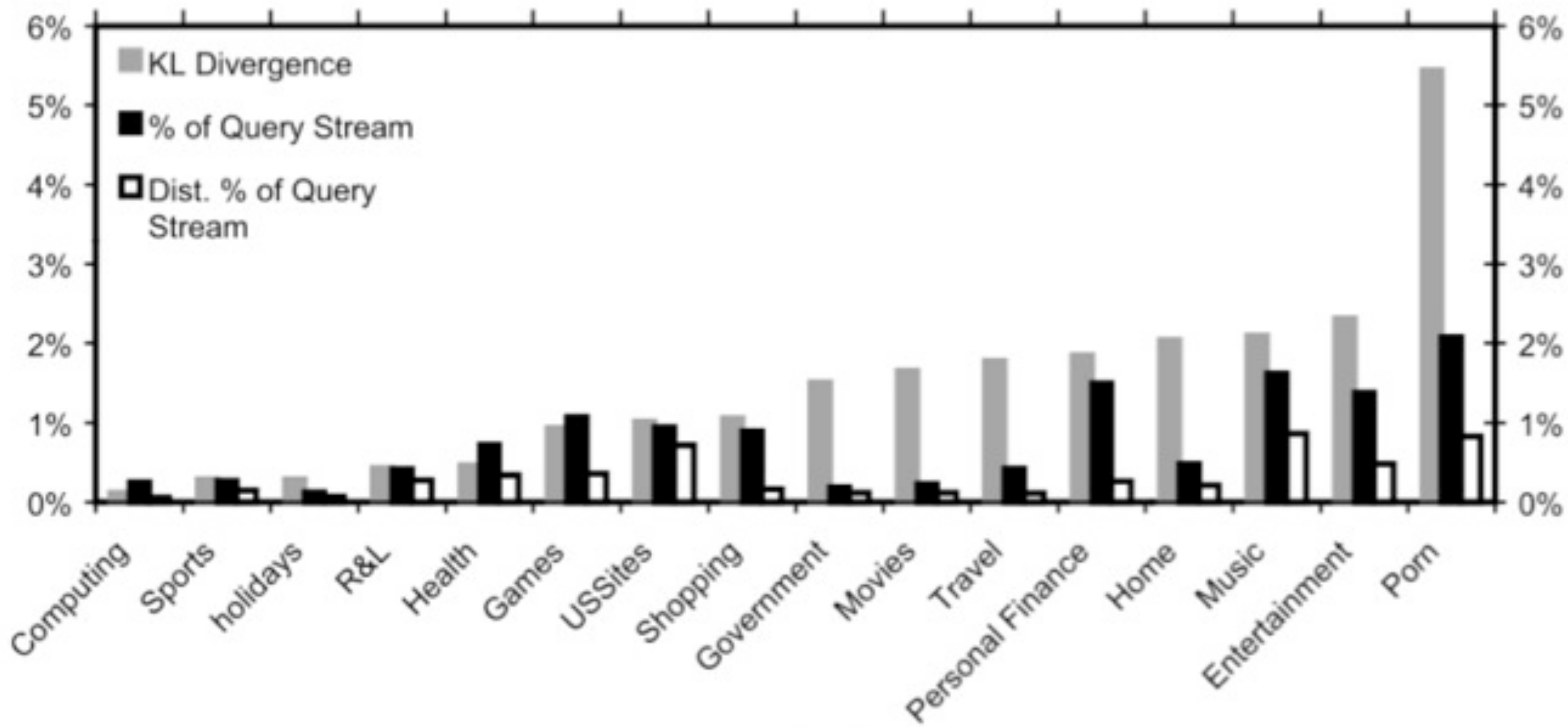


S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman, "T**emporal analysis of a very large topically categorized web query log**," J. Am. Soc. Inf. Sci. Technol., vol. 58, no. 2, pp. 166–178, 2007.

Surprising Topics

$$D\left(p\left(q|t\right)\|p\left(q|c,t\right)\right)=\sum_{q}p\left(q|t\right)\log\frac{p\left(q|t\right)}{p\left(q|c,t\right)}$$

- KL-Divergence betwe[en] ... observing a query topic u.a.r. and the actual topic observed.



S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, and D. Grossman, "T**emporal analysis of a very large topically categorized web query log**," J. Am. Soc. Inf. Sci. Technol., vol. 58, no. 2, pp. 166–178, 2007.

Tutorial Outline

- Query Logs
- Data Mining Techniques for QL Mining
  - “Classical” DM Tasks
  - New Mining Tasks for Query Logs
- Enhancing Effectiveness of Search Systems
- Enhancing Efficiency of Search Systems

Data Mining

- Many Definitions
  - Non-trivial extraction of implicit, previously unknown and potentially useful information from data
  - Exploration & analysis, by automatic or semi-automatic means, of large quantities of data in order to discover meaningful patterns

P.-N. Tan, M. Steinbach, V. Kumar. **“Introduction to Data Mining”**. Pearson Addison-Wesley.
J. Han, M. Kamber. **“Data mining: concepts and techniques”**. Morgan Kaufmann.

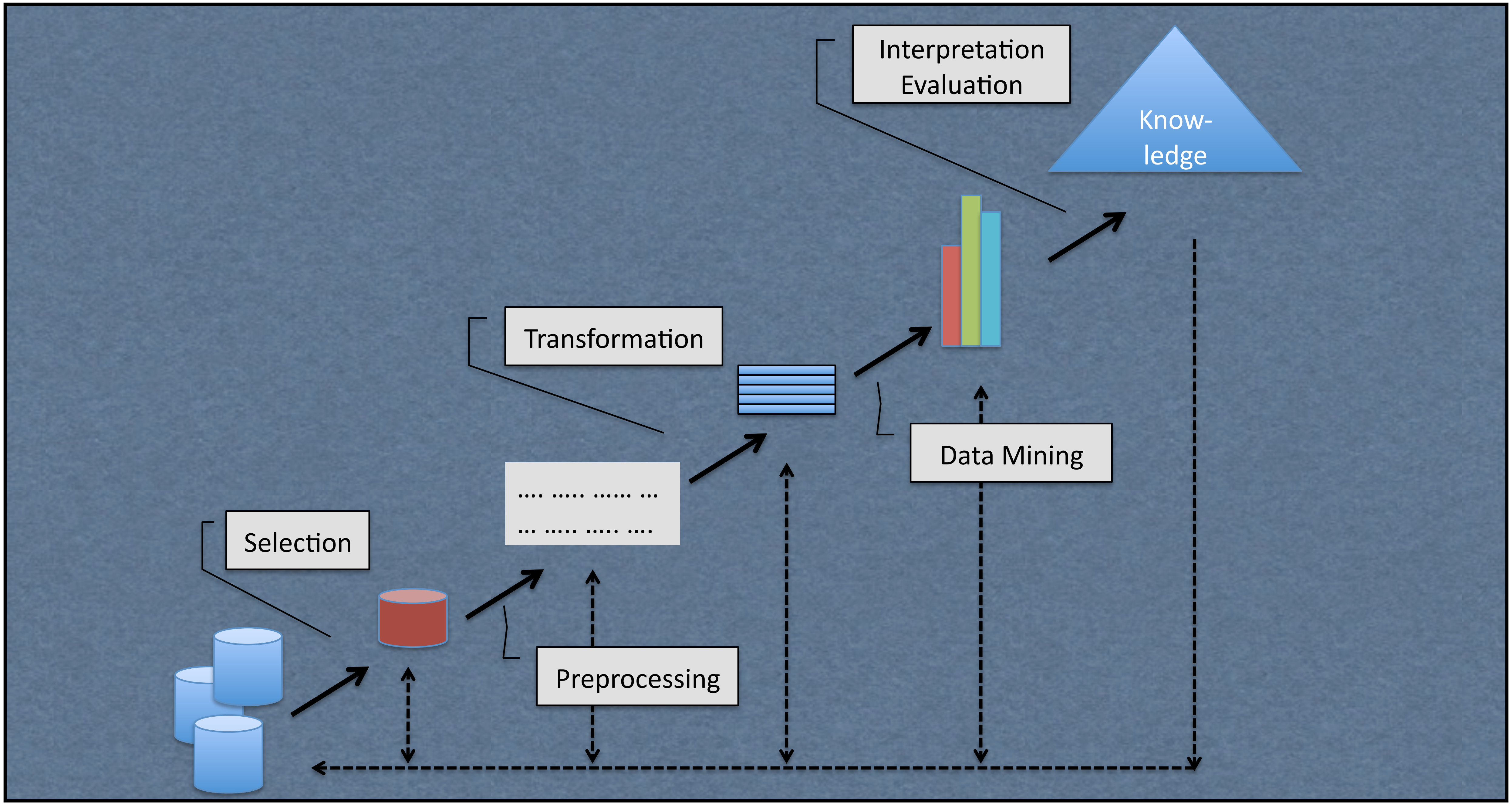
Typical DM tasks

- Given a large collection of documents, all concerning *sport*, build automatically a classifier. Use it to determine all the queries whose topic is *sport* with high probability
- Subdivide the queries in distinct clusters, so that queries in the same cluster are much more similar to each other than to others in different clusters

What is not DM

- Find all the web search queries that include the phrase "September 11"
- Look at a query log, and select all the queries submitted by the same user ID2378

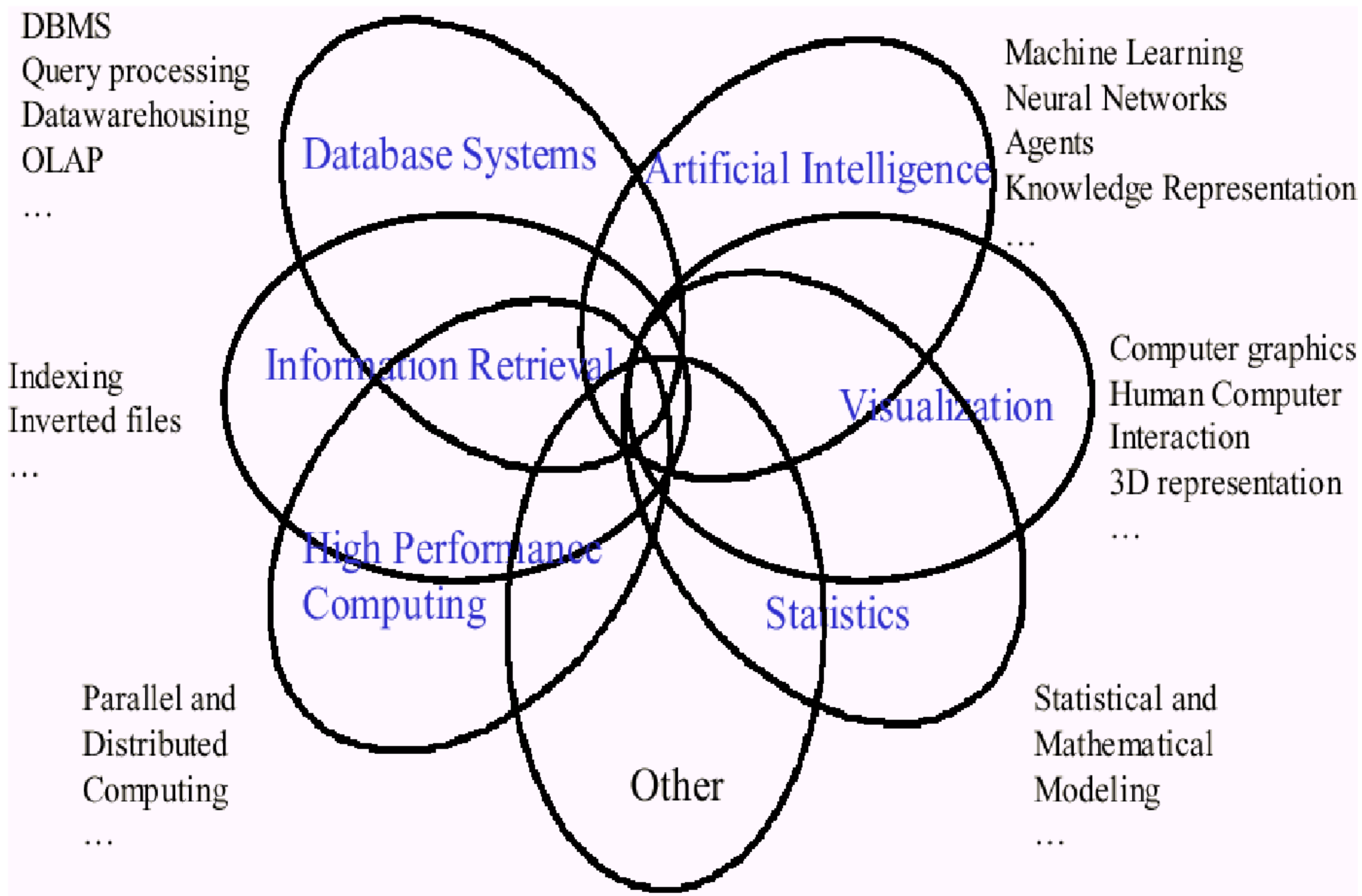
DM is part of a process



The iterative and exploratory Knowledge Discovery process

Origins of DM

- Traditional techniques may be unsuitable
  - data have huge size, high dimensionality, are heterogeneous and distributed



DM tasks

- Prediction Methods
  - Use some variables to predict unknown or future values of other unknown variables
- Description Methods
  - Find human-interpretable patterns that describe the data

DM tasks

- Classification [Predictive]
- Clustering [Descriptive]
- Association Rule Discovery [Descriptive]
- Sequential Pattern Discovery [Descriptive]
- Regression [Predictive]
- Outlier Detection [Predictive]

Web Mining

- DM techniques applied to Web data
- Web content mining
  - From text, image, in general contents of Web pages
- Web structure mining
  - From hyperlink structure (graph) of Web
- Web usage mining
  - From usage data, like logs

R. Kosala. and H. Blockeel, **“Web Mining Research: A Survey”**, SIGKDD Explorations, 2(1):1-15, 2000.
Soumen Chakrabarti, **“Mining the Web”**, Morgan Kaufmann
Bing Liu, **“Web Data Mining”**, Springer, 2007.

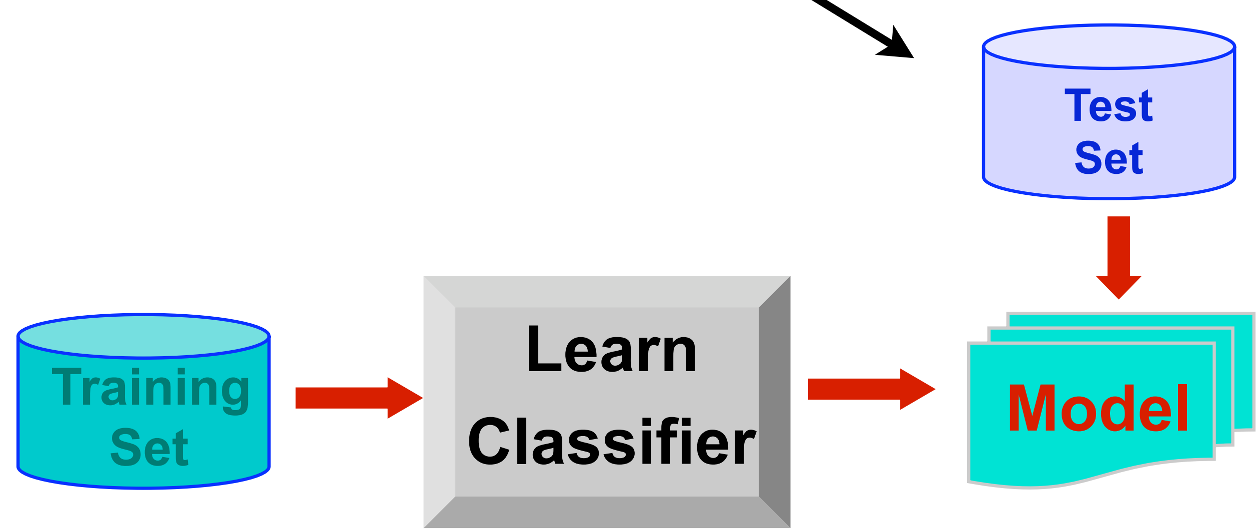
Classification

- Collection of records (training set)
  - One of the attributes is the class
- Find a model for the class attribute
  - A *function* of the values of other attributes.
- Goal of the model
  - Previously unseen records should be assigned a class (classified) as accurately as possible
  - A test set is used to determine the accuracy of the model

Classification example

Qid	Query + snippets	Class
1	soccer, Kaka, ...	Sport
2	Berlusconi, Sartdinia, ...	Politics
3	tennis, Federer, ...	Sport
4	Nicolas, Carla, ...	Politics
5	swimming, Phelps, team, ...	Sport
7	Rugby, New Zeland, cup, ..	Sport
6	Obama, president, elect, ...	Politics
8	hundred, ourdoor, race, ...	Sport
9	beijing, olympic, ...	Sport
10	Brown, bank, ...	Politics

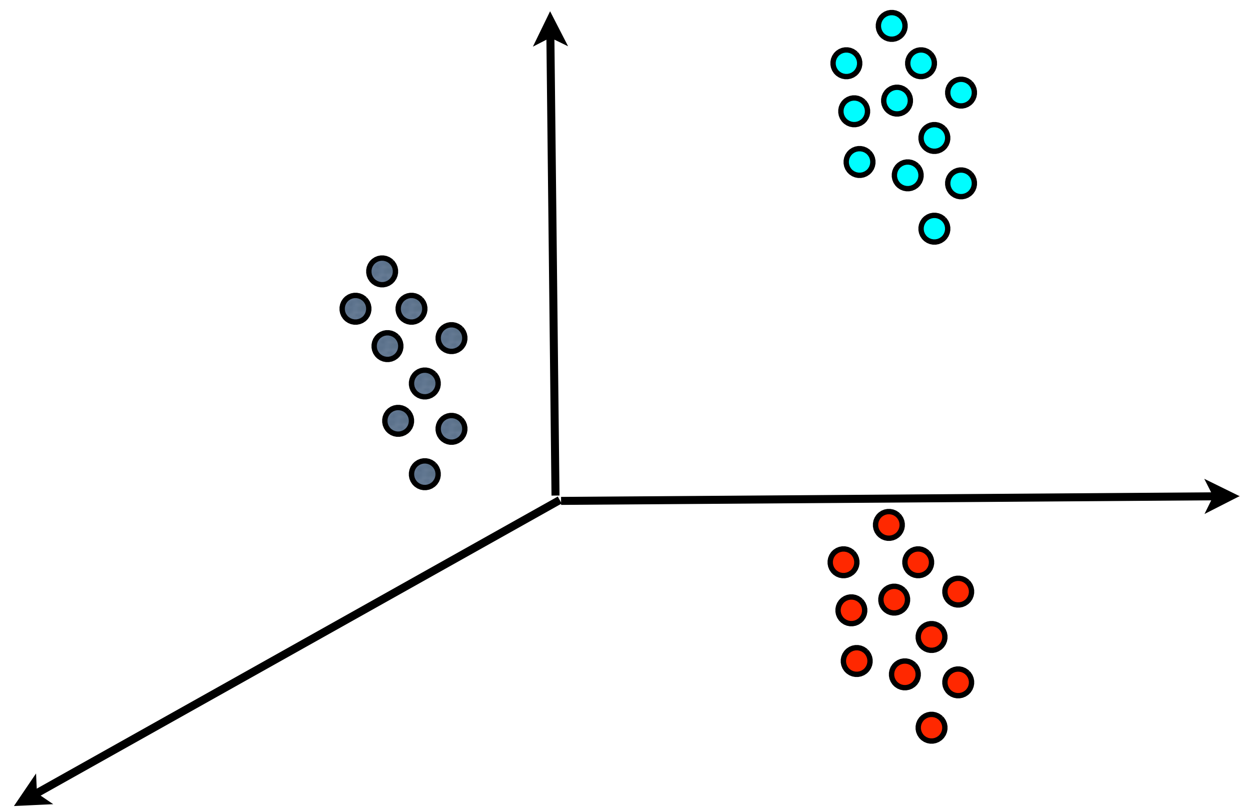
Qid	Query + snippets	Class
1	Pelè, Brasil, ..	?
2	Veltroni, Left, Centre, ...	?
3	Nadal, court, ...	?
4	President, Carla, ...	?
5	football, rugby, ...	?



Clustering

- Given
  - a set of records
  - a similarity measure based on the attributes
- Find clusters such that
  - Data points in one cluster are more similar to one another: MAX intra-cluster similarity
  - Data points in separate clusters are less similar to one another: MIN inter-cluster similarity

Clustering example



- 3D-space: each record is a point corresponding to an R^3 vector
- Similarity measure: Euclidean distance

Clustering of text data

- Each document: term vector with word frequency
- Similarity measure: cosine

	team	coach	pla y	ball	score	game	wi n	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Clustering of queries

- Query are too short text documents
  - Expanded representation for the query "apple pie" by using snippet elements [Metzler et al. ECIR07]

<query>*apple pie*</query>

<title>Applie pie – Wikipedia, the free encyclopedia</title>
<snippet>In cooking, an apple pie is a fruit pie (or tart) in which the principal filling ingredient is apples . Pastry is generally used top-and-bottom, making a double-crust pie, the upper crust of which ...</snippet>
<url>en.wikipedia.org/wiki/Apple_ pie</url>

<title>All About Food – Apple Pies</title>
<snippet>Apple Pie. Recipes. All-American Apple Pie. American Apple Pie. Amish Apple Pie . Apple Cream Pie. Apple Crumble Pie. Apple Pie . Apple Pie in a Brown Bag. Best Apple Pie</snippet>
<url>fp.enter.net/~rburk/pies/ applepie/applepie.htm</url>

<title>Apple Pie Recipe</title>
<snippet>Apple Pie Recipe using apple peeler corer slicer ... Apple Pie Recipe. From Scratch to Oven in 20-Minutes. Start by preheating the oven. By the time it's ...</snippet>
<url>applesource.com/applepierecipe.htm</url>
...

Association rules

TID	items
1	beer, coke, milk
2	water, coke, chips, milk
3	coke, milk
4	milk, bread
5	bread, water, coke

- Example of unveiled rule

  coke $\rightarrow$ milk

  [sup=60%, conf=75%]

- Market basket analysis
  - extract rules from transactional databases
  - each transaction corresponds to a customer basket

Association Rules for Query Expansion

QueryID	words
1	Elect Obama
2	President US Obama
3	Obama President
4	President Sarcozy
5	President Obama

- Example of unveiled rule

  Obama → President

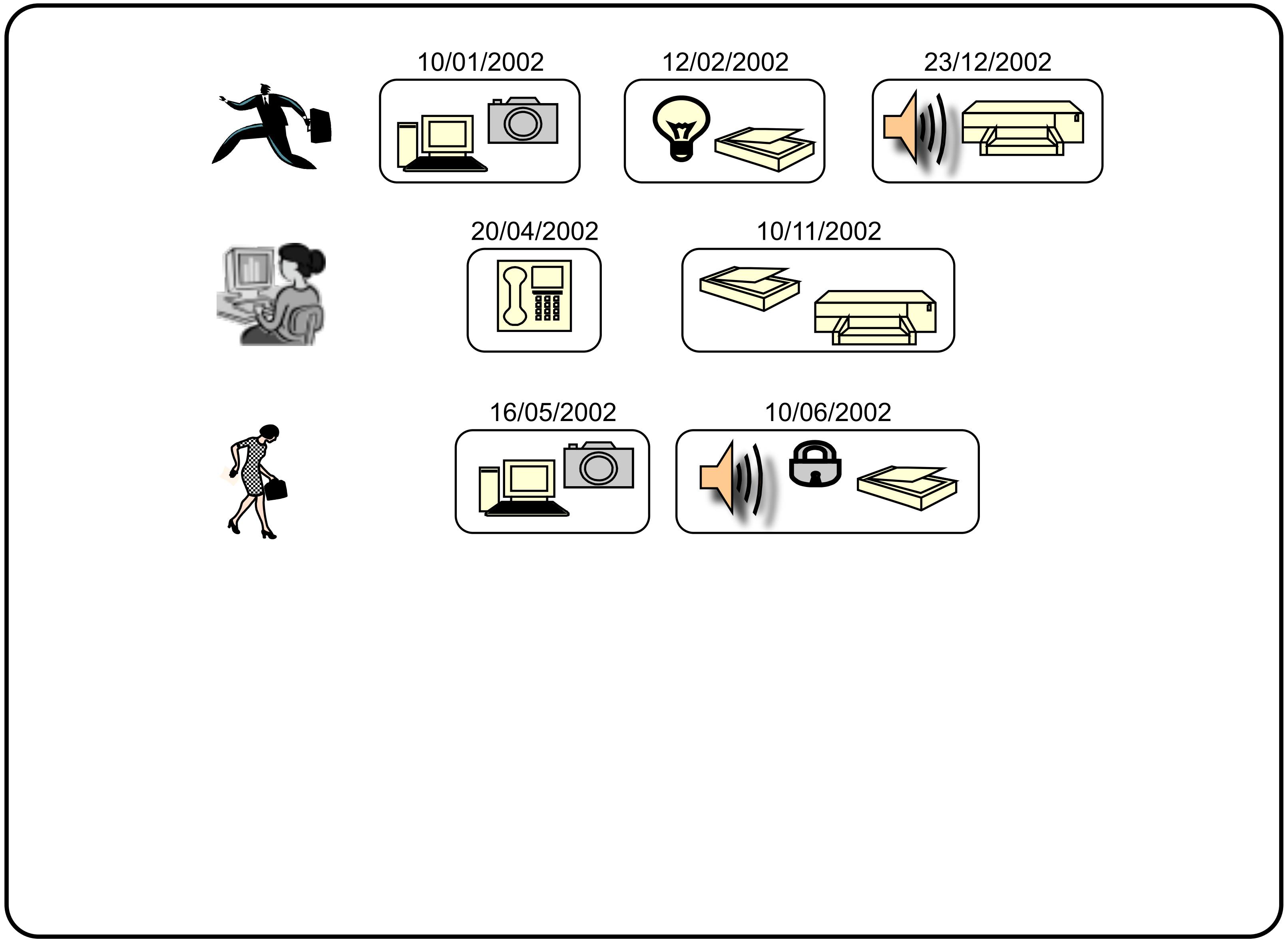
  [sup=60%, conf=75%]

- A user submitting the query “Obama”
- The system suggests to expand the query with the word “President”

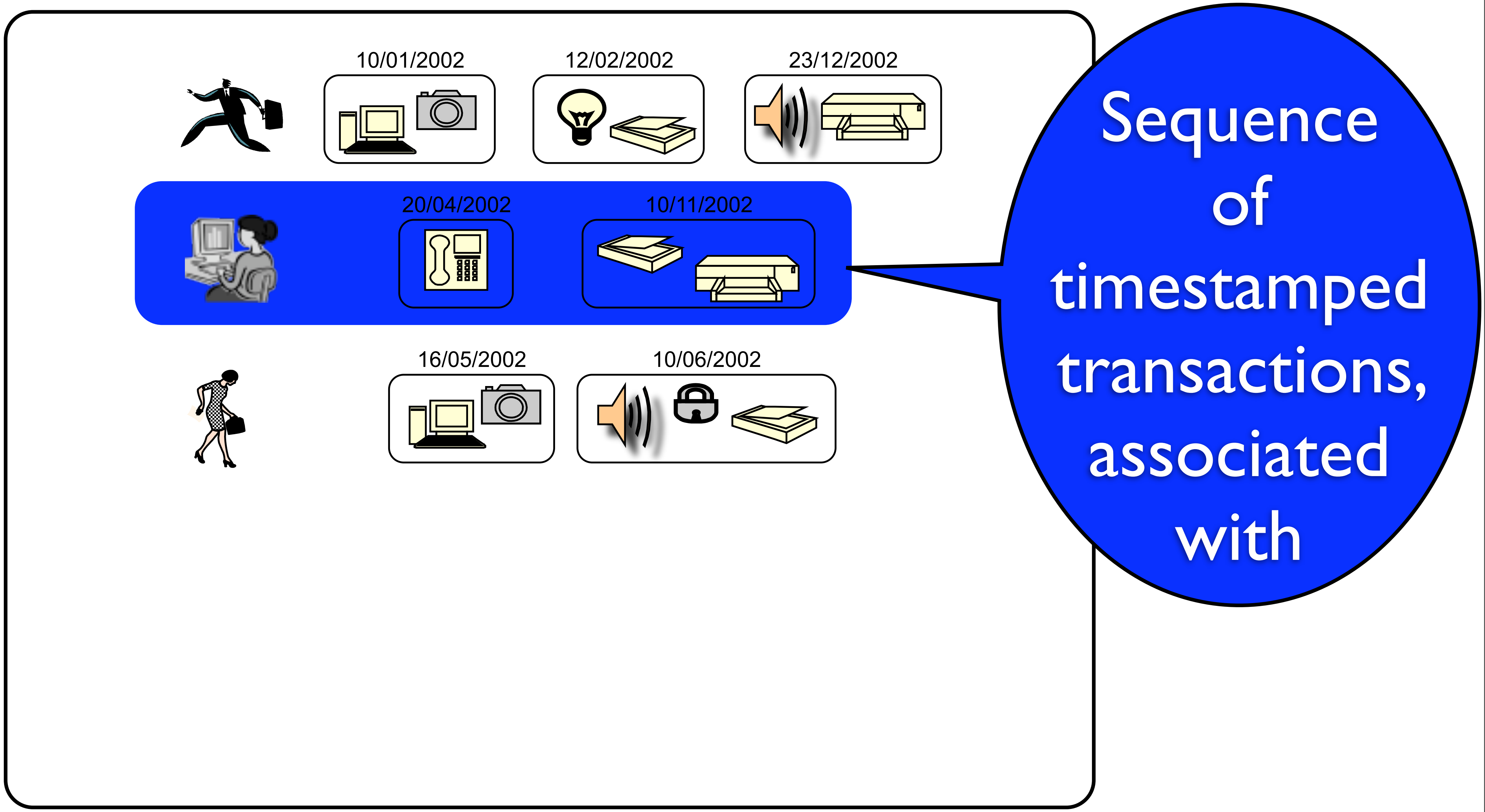
Sequential patterns

- For each user (customer), record sequences of associated events
  - i.e., temporally ordered sequences of events (e.g., sets of bought objects)
- Find frequent rules that predict strong sequential dependencies among different events

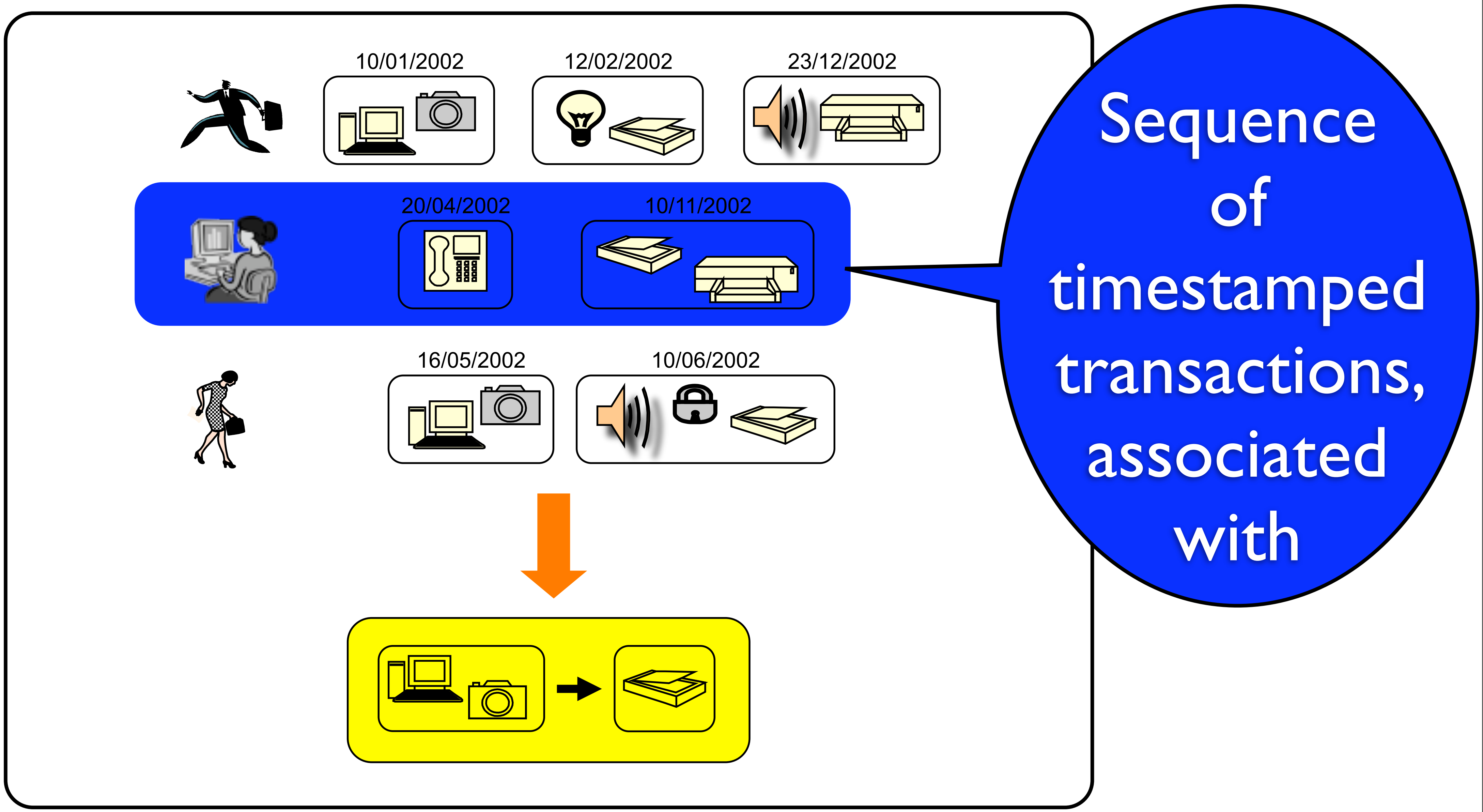
Sequential patterns



Sequential patterns



Sequential patterns



Sequential mining example

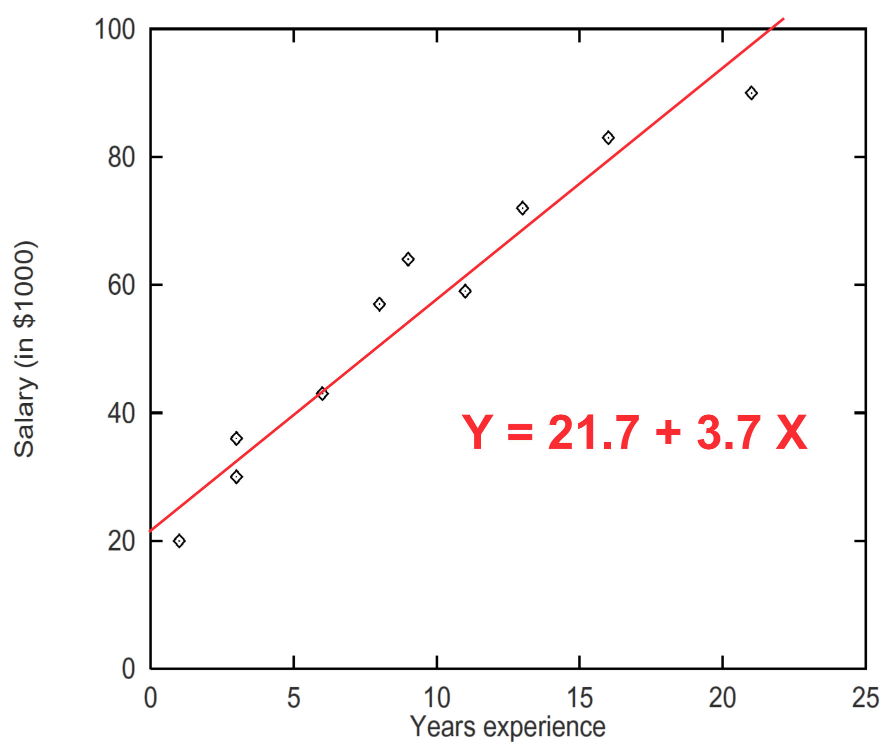
- Sessioned query logs
  - a session is a temporal sequence, where each query can be seen as a transaction of words
- Extract frequent features from each query
  - e.g. single words or pairs of words
- Find frequent sequences, and then association rules
- Use the rules for query suggestions

  ("mortgage loan" → "subprime mortgage") → "lehman brothers"

Predict continuous variables by regression

- Linear regression: $Y = a + b X$
  - Model a variable Y (to be predicted) as a linear function of X
  - Determine coefficients a and b on the basis of the training set

X years experience	Y salary (in \$1000)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83



Regression to learn query ranking

- For a query-document pair *(q; d)*, extract a a feature vector x = [xQ; xD; xQD]
  - examples of features extracted from query-document pair *(q; d)*
    - xQ: e.g., number of terms in query *q*
    - xD: e.g., the language identity of document *d*
    - xQD: e.g., the number of times each term in *q* appears in the anchor-texts of document *d*
  - Assign a numerical grade to each pair *(q; d)* based on the degree of relevance inferred from the query clicktroughs in the logs
  - Use numerical grades as target values for for multivariate regression based on the query-document features

Tutorial Outline

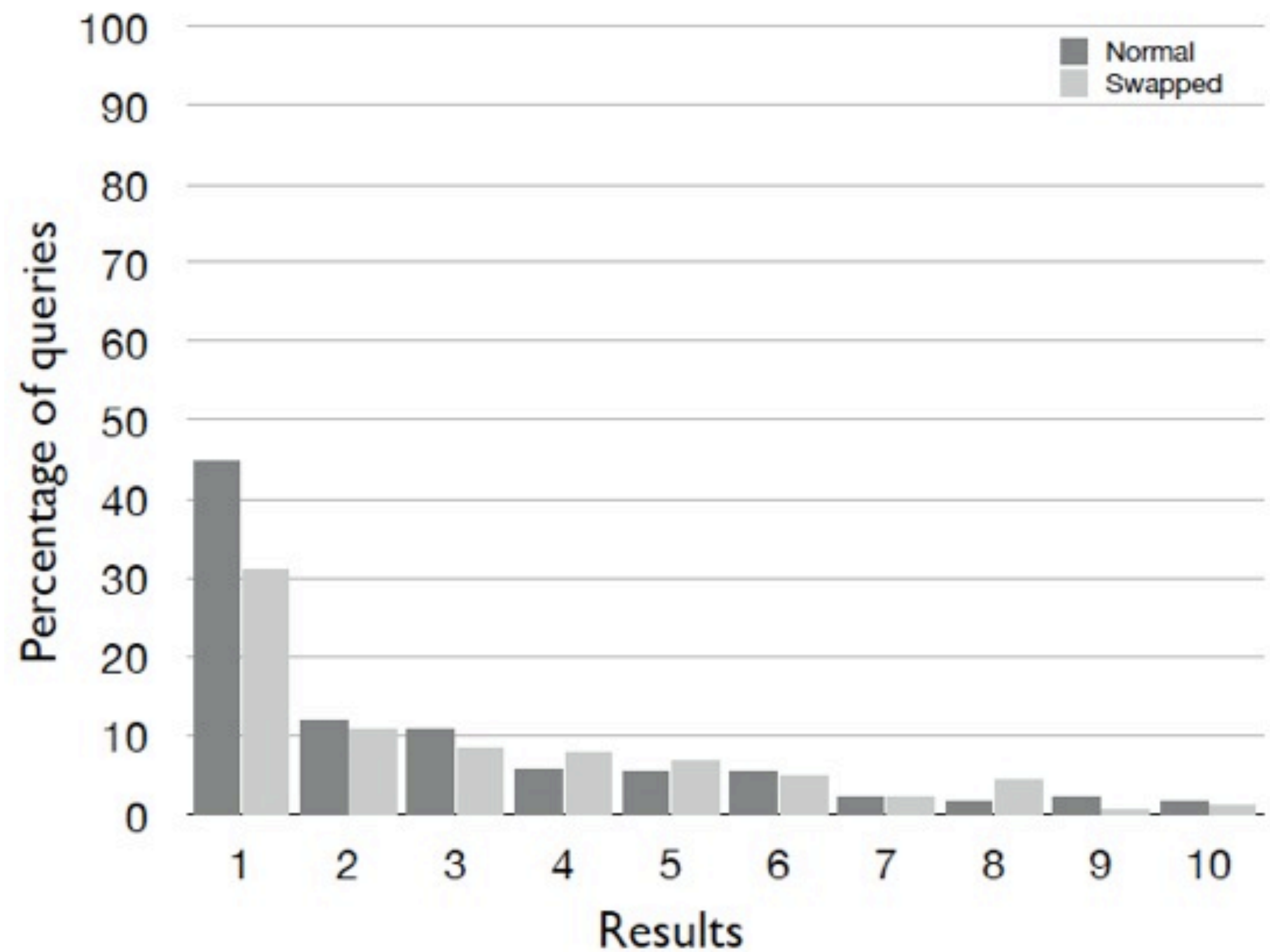
- Query Logs
- Data Mining Techniques for QL Mining
- Enhancing Effectiveness of Search Systems
  - Query Expansion/Suggestion/Personalization
  - Learning to Rank: Ranking SVM
- Enhancing Efficiency of Search Systems

Query Expansion/ Suggestion/Personalization

- Click-through data associated with past queries represent a sort of implicit relevance feedback information
- The challenge is to exploit such information to mine knowledge and use it to improve the effectiveness of the search engines
- The final goal is improve the precision by expanding/suggesting/personalizing queries

Can click-through data be useful relevance feedbacks?

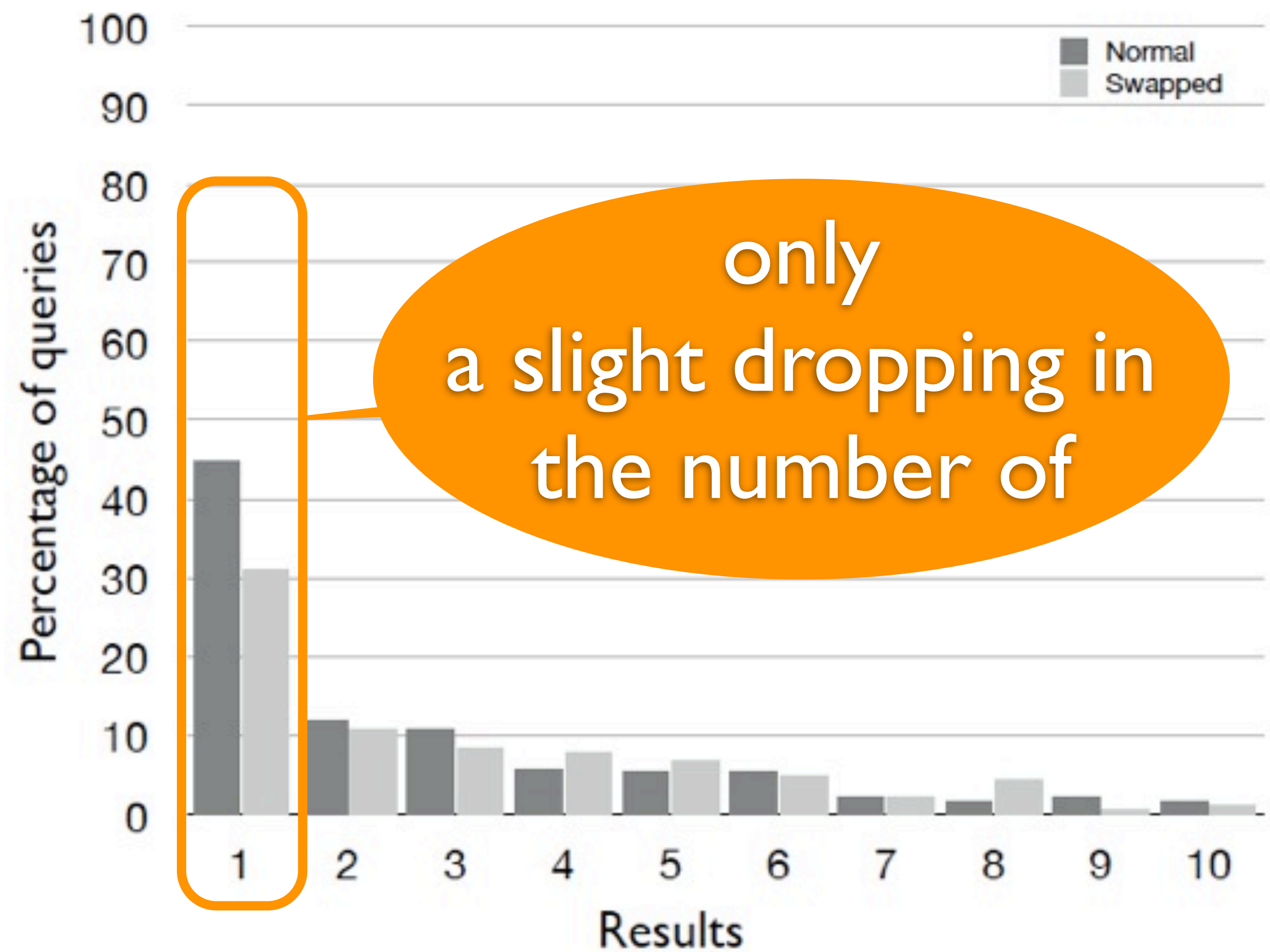
- *Joachims and Radlinski* noted that the top position reported by WSE strongly influence user behavior, beyond snippets
- They registered the number of clicks a given position obtained in two different conditions: normal and swapping the first two top positions



T. Joachims and F. Radlinski, "**Search engines that learn from implicit feedback**", Computer, vol. 40, no. 8, pp. 34-40, 2007.

Can click-through data be useful relevance feedbacks?

- *Joachims and Radlinski* noted that the top position reported by WSE strongly influence user behavior, beyond snippets
- They registered the number of clicks a given position obtained in two different conditions: normal and swapping the first two top positions



T. Joachims and F. Radlinski, "**Search engines that learn from implicit feedback**", Computer, vol. 40, no. 8, pp. 34-40, 2007.

Research issues (1)

- The lack of query logs and well-defined effectiveness metrics may negatively influence the scientific value of research results
  - many times, such logs are not publicly available, and thus experiments may not be reproducible

- The effectiveness of the proposed solutions are often tested on a small group of homogeneous people, e.g., metrics are tested on small human-annotated testbeds

Research issues (2)

- Privacy is nowadays a big concerns of user communities
  - many of the techniques presented need to build user profiles
- Profile-based (i.e. context-based, personalized) search
  - is computationally expensive
  - may prevent the adoption of global techniques aiming at enhancing performance (like caching and collection selection)

Query Expansion

- Queries are short, poorly built, and sometimes mistyped
- *Cui et al.* observed that queries and documents are rather poorly correlated
    - by measuring the gap between the *document vector space* (the most important terms contained in each document according to *if* x *idf*) and the *query vector space* (all the terms contained in the group of queries for which a document was clicked)
    - in most cases, the similarity values are between 0.1 and 0.4, and only a small percentage of documents have similarity above 0.8
- Solution: expanding a query by adding additional terms

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **“Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

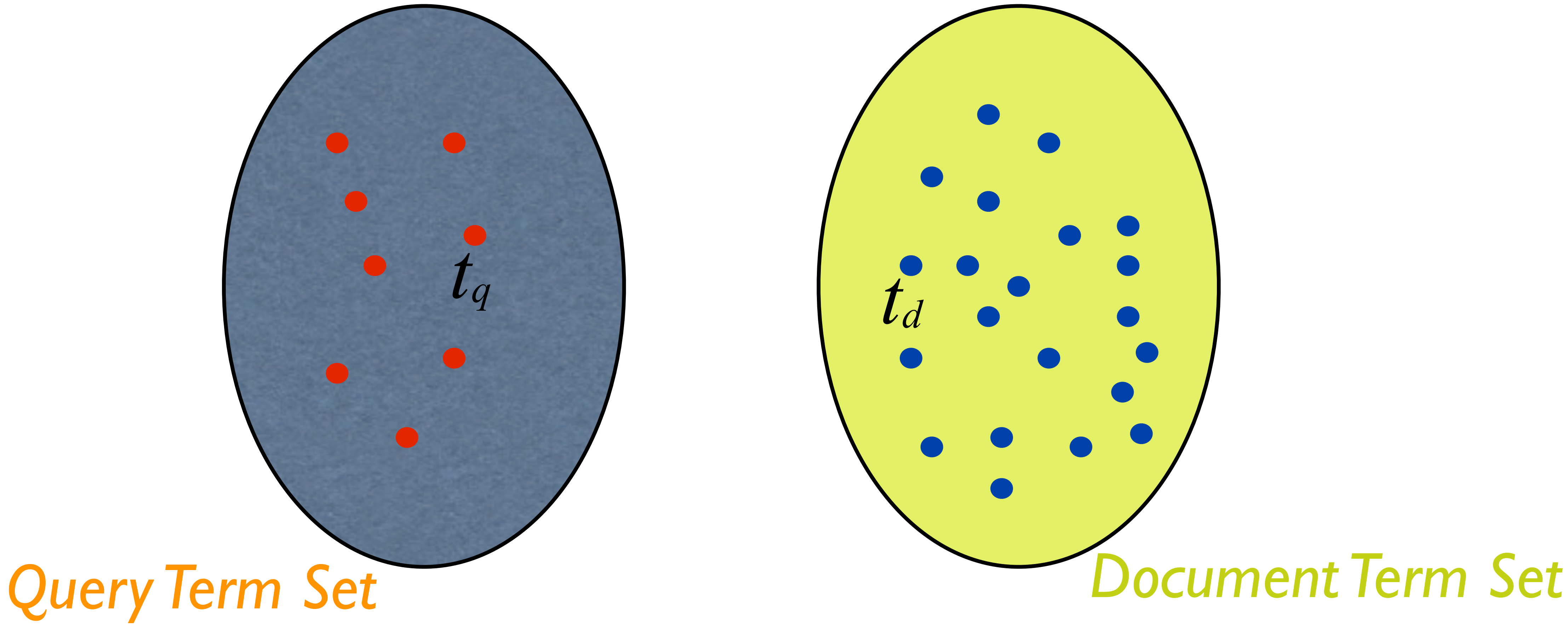
Query Expansion

- In traditional IR systems *query expansion* is a well-known technique
- However, one of the first works making explicit use of past queries to improve effectiveness of query expansion is Fitzpatrick and Dent
  - it builds off-line an *affinity pool* made up of documents retrieved by similar past queries (the TREC queries and databases were used)
  - a submitted query is first checked against the *affinity pool*, and from the resulting top scoring documents, a set of "important" terms is automatically extracted to enrich the query
  - they achieved an improvement of 38.3% in average precision

L. Fitzpatrick and M. Dent, **"Automatic feedback using past queries: social searching?"**. In SIGIR '97, pp. 306-313, ACM, 1997.

Query Expansion

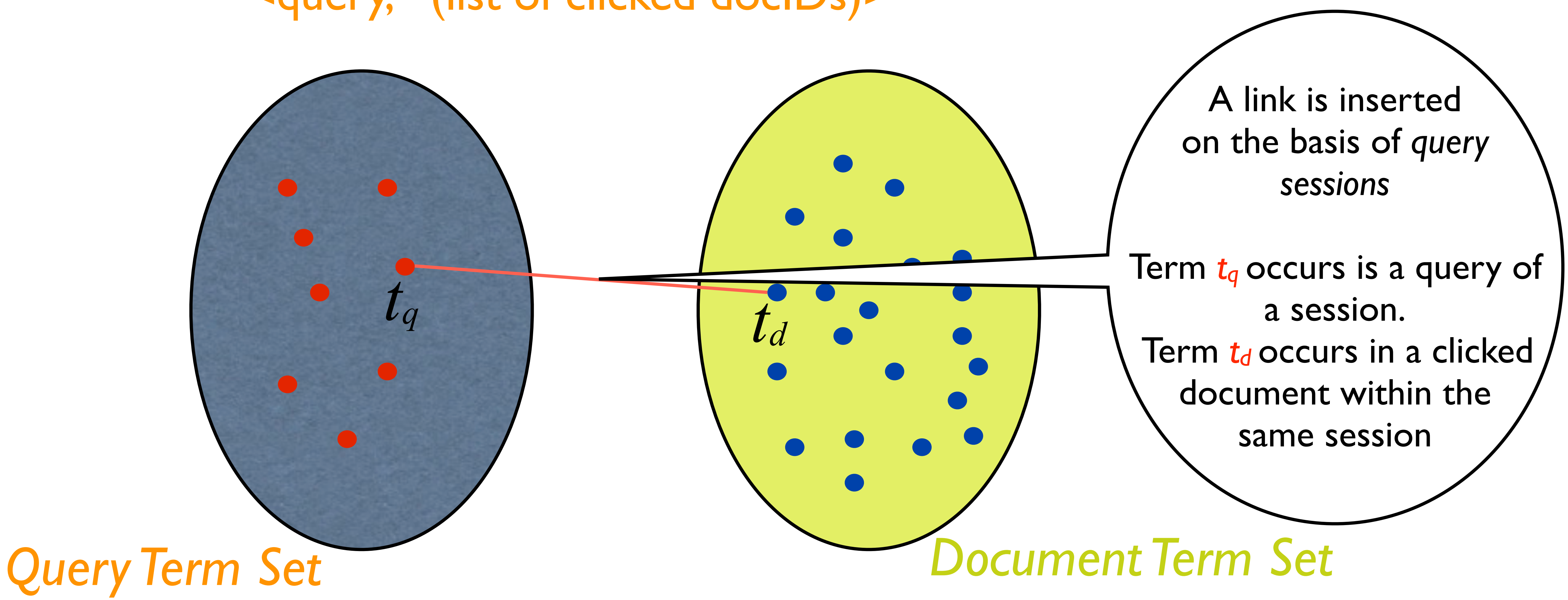
- Cui *et al.* exploited correlations among *terms in clicked documents* and *web search engine queries*
  - *query session* extracted from the query log: <query,   (list of clicked docIDs)>



TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

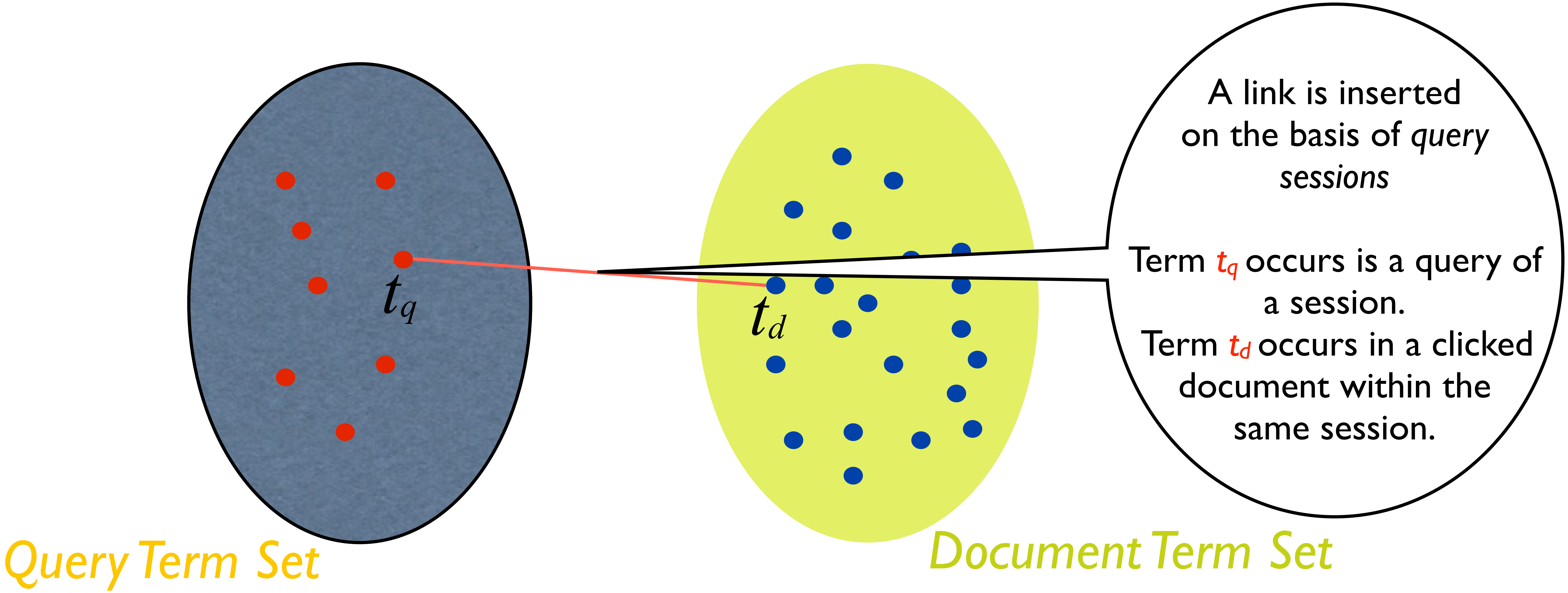
Query Expansion

- Cui *et al.* exploited correlations among *terms in clicked documents* and *web search engine queries*
  - *query session* extracted from the query log: <query, (list of clicked docIDs)>



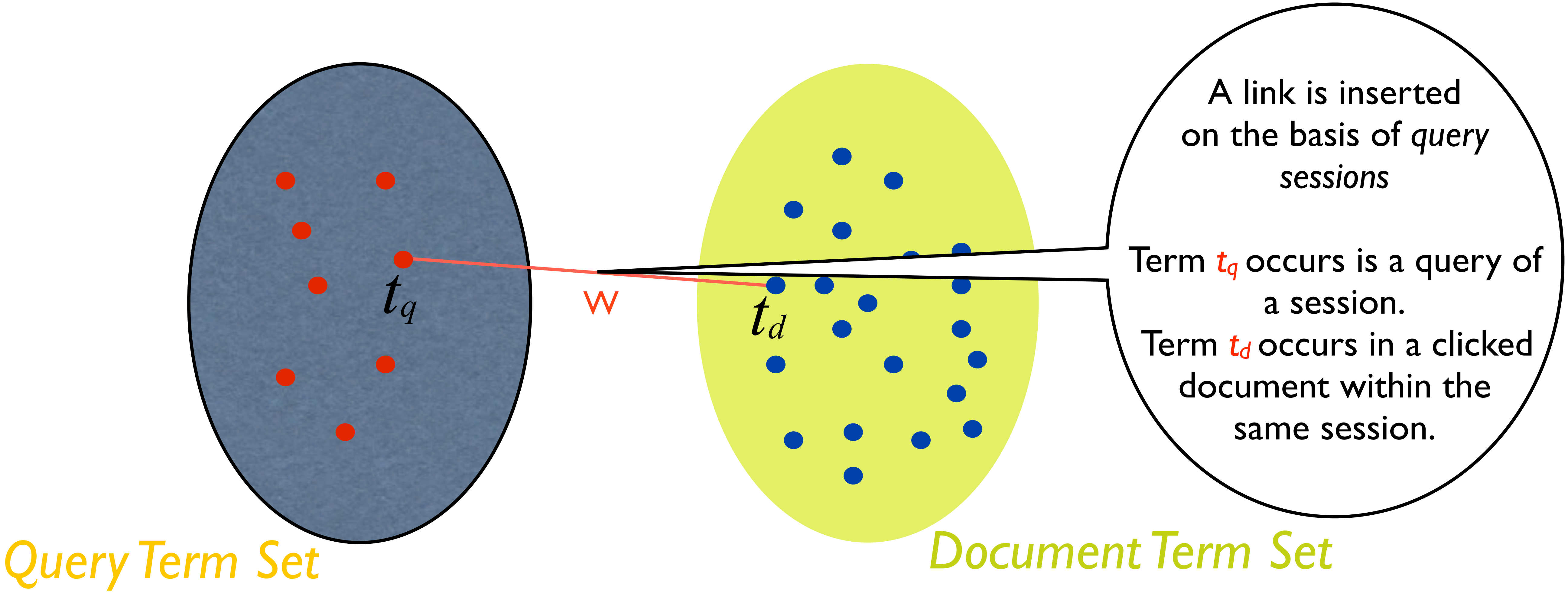
TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion



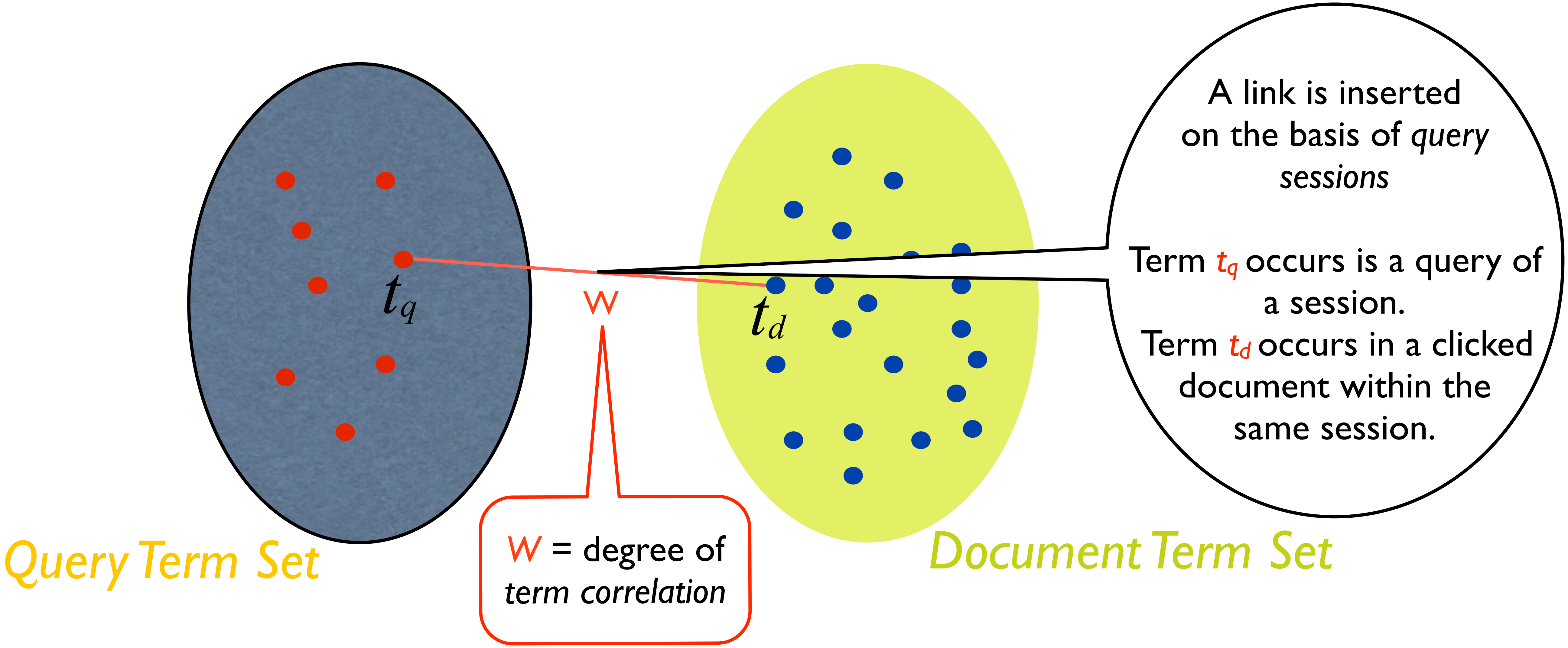
TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion



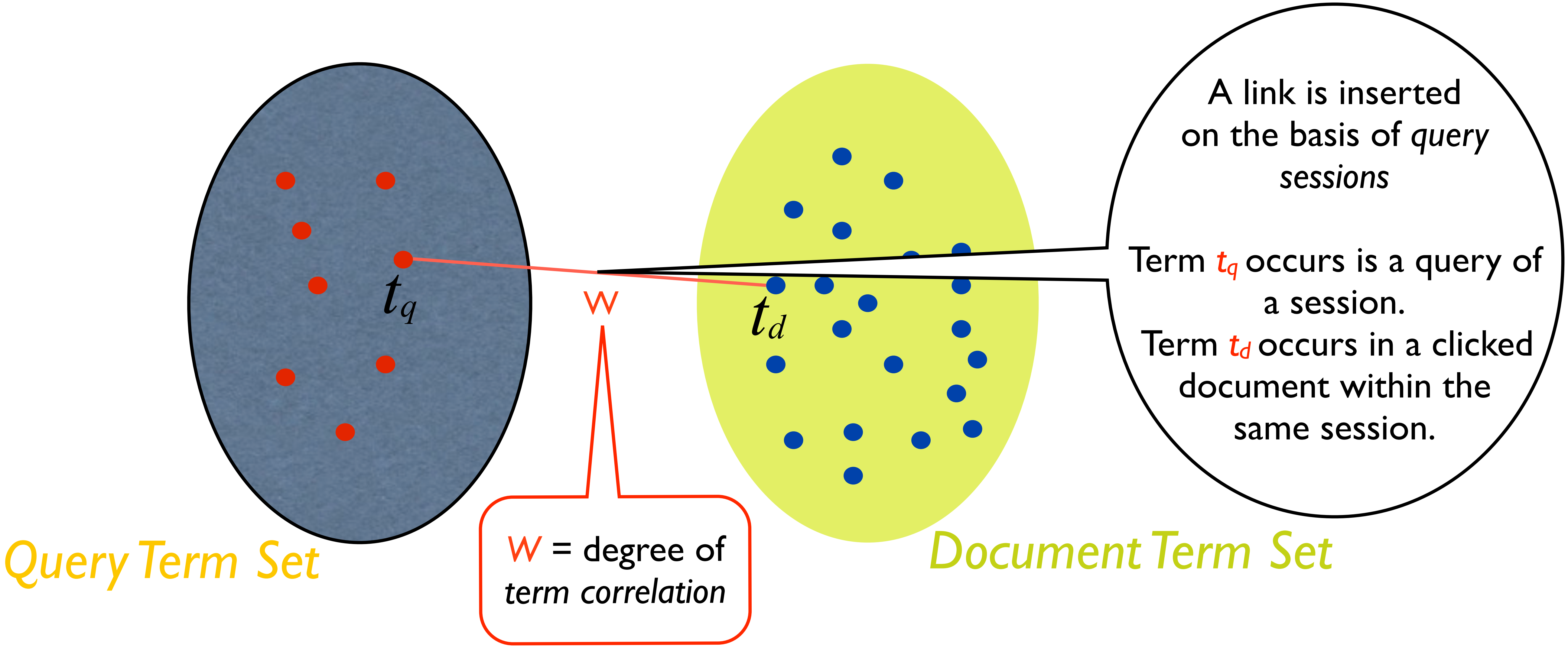
TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion



TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion



- Correlation is given by the conditional probability $P(t_d | t_q)$
  - occurrence of term t_d given the occurrence of t_q in the query

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion

- Term correlation:

$$P\left(t_d|t_q\right) = \sum_{D_i \in S_q} P\left(t_d|D_i\right) \frac{\mathrm{freq}\left(t_q, D_i\right)}{\mathrm{freq}\left(t_q\right)}$$

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion

- Term correlation:

$$P(t_d|t_q) = \sum_{D_i \in S_q} P(t_d|D_i) \frac{\text{freq}(t_q, D_i)}{\text{freq}(t_q)}$$

set of clicked documents for queries containing term t_q

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **“Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion

- Term correlation:

$$P(t_d|t_q) = \sum_{D_i \in S_q} P(t_d|D_i) \frac{\text{freq}(t_q, D_i)}{\text{freq}(t_q)}$$

set of clicked documents for queries containing term t_q

relative weight of term t_d in document D_i

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **“Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion

- Term correlation:

$$P(t_d|t_q) = \sum_{D_i \in S_q} P(t_d|D_i) \frac{\text{freq}(t_q, D_i)}{\text{freq}(t_q)}$$

set of clicked documents for queries containing term t_q

relative weight of term t_d in document D_i

number of queries containing term t_q

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **“Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion

- Term correlation:

$$P(t_d|t_q) = \sum_{D_i \in S_q} P(t_d|D_i)\,\frac{\text{freq}(t_q, D_i)}{\text{freq}(t_q)}$$

set of clicked documents for queries containing term t_q

relative weight of term t_d in document D_i

number of queries containing term t_q

number of queries sessions in which appear both term t_q and document D_i

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **“Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion

- The term correlation measure is then used to devise a *query expansion method*
- It exploits a so-called *cohesion measure* between a query Q and a candidate term t_d for query expansion

$$\mathrm{CoWeight}\left(Q, t_d\right) = \log\left(\prod_{t_q \in Q} P\left(t_d | t_q\right) + 1\right)$$

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion

- The term correlation measure is then used to devise a *query expansion method*
- It exploits a so-called *cohesion measure* between a query Q and a candidate term t_d for query expansion

$$\mathrm{CoWeight}\left(Q, t_d\right) = \log\left(\prod_{t_q \in Q} P\left(t_d | t_q\right) + 1\right)$$

Naïve hypothesis on independence of terms in a query

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion

- The term correlation measure is then used to devise a *query expansion method*
- It exploits a so-called *cohesion measure* between a query Q and a candidate term t_d for query expansion

$$\text{CoWeight}\left(Q, t_d\right) = \log\left(\prod_{t_q \in Q} P\left(t_d | t_q\right) + 1\right)$$

Naïve hypothesis on independence of terms in a query

- The measure is used to build a list of weighted candidate terms
- The *top-k* ranked terms (those with the highest weights) are selected as expansion terms for query Q

TH. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **“Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

Query Expansion

- The log-based method was compared against two baseline methods
  - (a) not using query expansion at all, or
  - (b) using an expansion technique (*local context method*) that does not make use of logs to expands queries
- Indeed, the l*ocal context method* (by *Xu and Croft*) exploits the top ranked documents retrieved for a query to expand the query itself
- A few queries were used for the tests (Encarta and TREC queries, and hand-crafted queries), and the following table summarizes the average results

	Precision
baseline	17%
local context	22%
log-based	30%

H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma, **"Probabilistic query expansion using query logs"**, in WWW '02, pp. 325-332, ACM, 2002.

J. Xu and W. B. Croft, **"Improving the effectiveness of information retrieval with local context analysis"**, ACM Trans. Inf. Syst., vol. 18, no. 1, pp. 79-112, 2000.

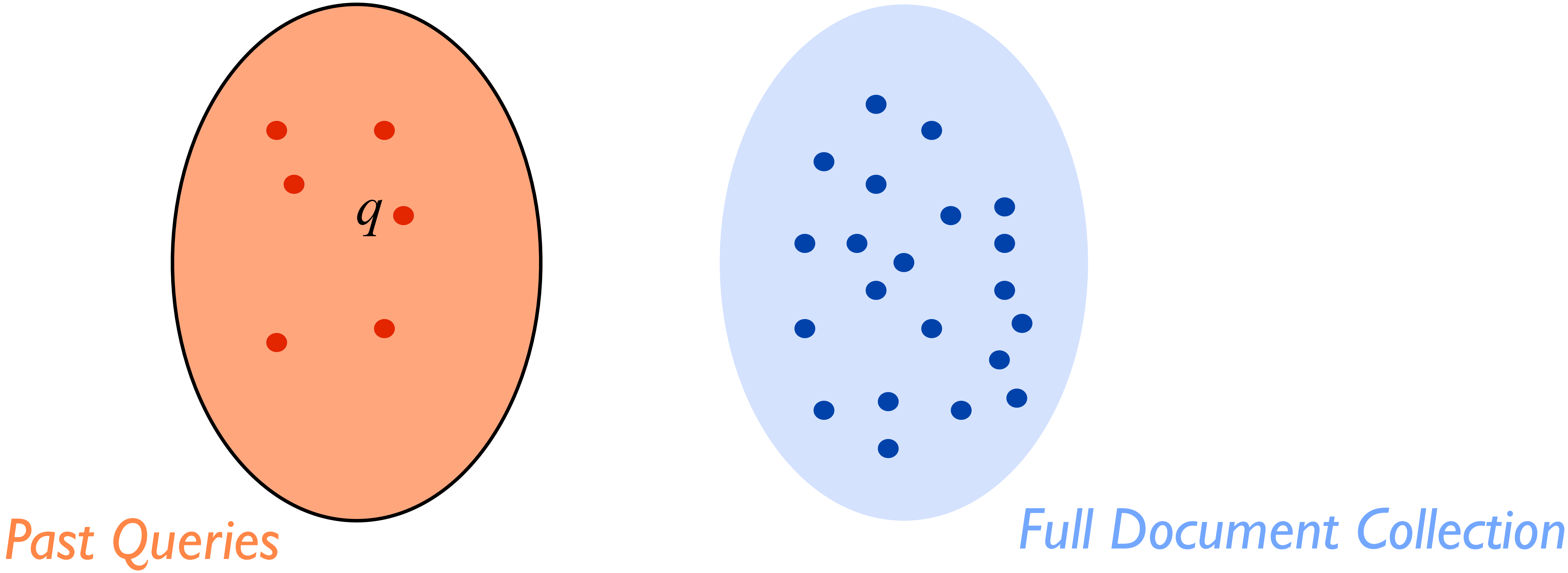
Query Expansion

- *Billerbeck et al.* use the concept of Query Association already proposed by by *Scholer et al.*
  - Past user queries are associated with a document if they share a high statistically similarity

- Past queries associated with a document enrich the document itself
  - All the queries associated with a document can be considered as Surrogate Documents, and can be used as a source of terms for query expansion

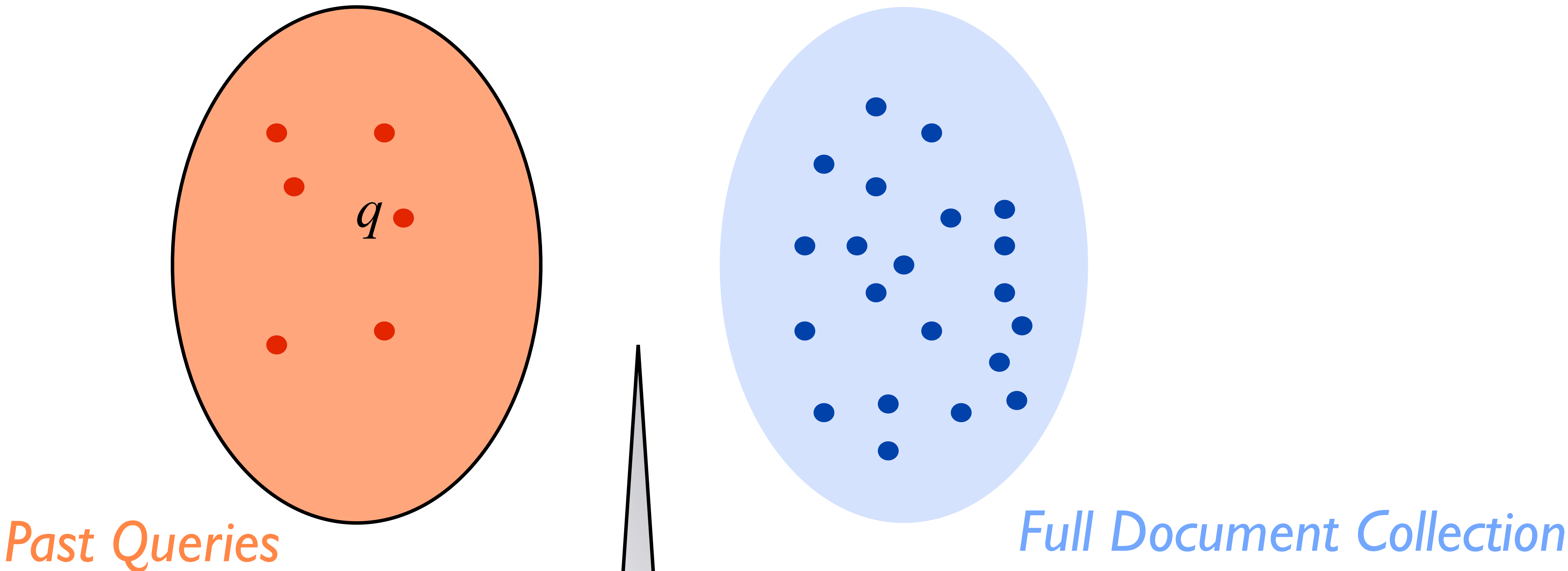
B. Billerbeck, F. Scholer, H. E. Williams, and J. Zobel, **"Query expansion using associated queries"**, in Proc. of the 12th CIKM, pp. 2-9, 2003.
F. Scholer, H.E. Williams. **"Query association for effective retrieval"**, in Proc. of the 11th CIKM, pp. 324–331, 2002.

Query Expansion



F. Scholer, H.E. Williams. **"Query association for effective retrieval"**, in Proc. of the 11th CIKM, pp. 324–331, 2002.

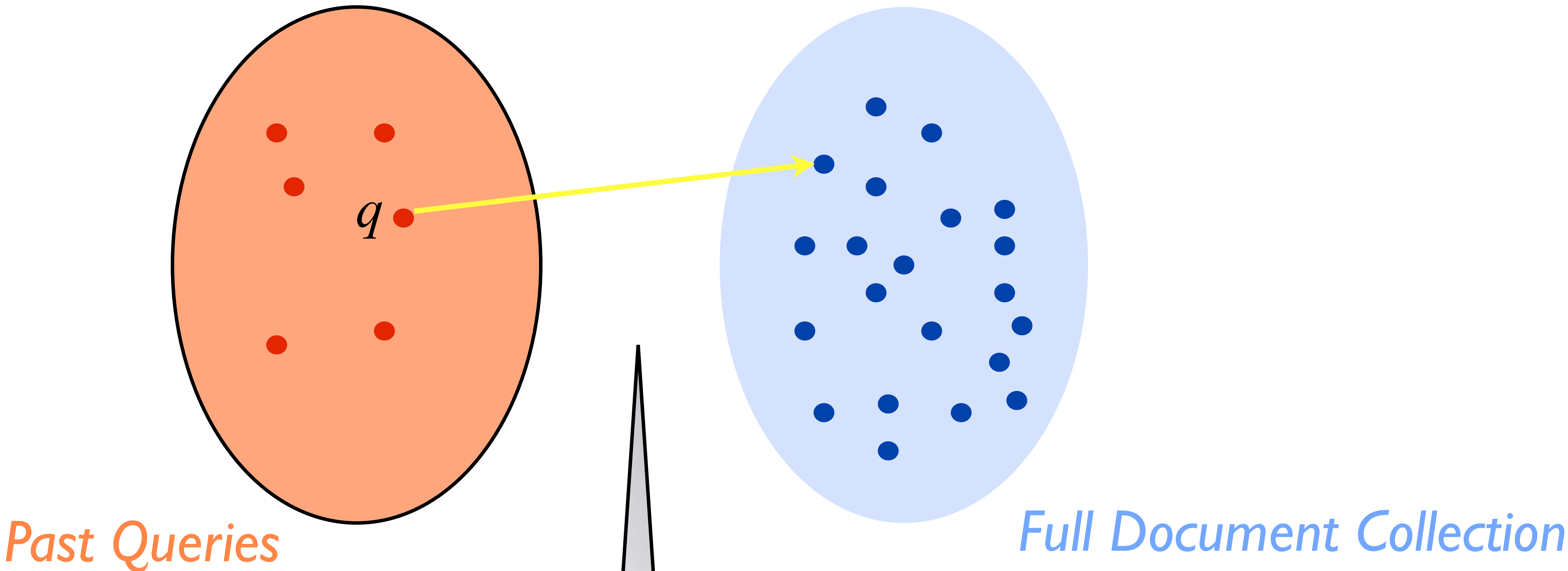
Query Expansion



Each past queries q is naturally associated with the K most relevant *documents* returned by a search engine

F. Scholer, H.E. Williams. **"Query association for effective retrieval"**, in Proc. of the 11th CIKM, pp. 324–331, 2002.

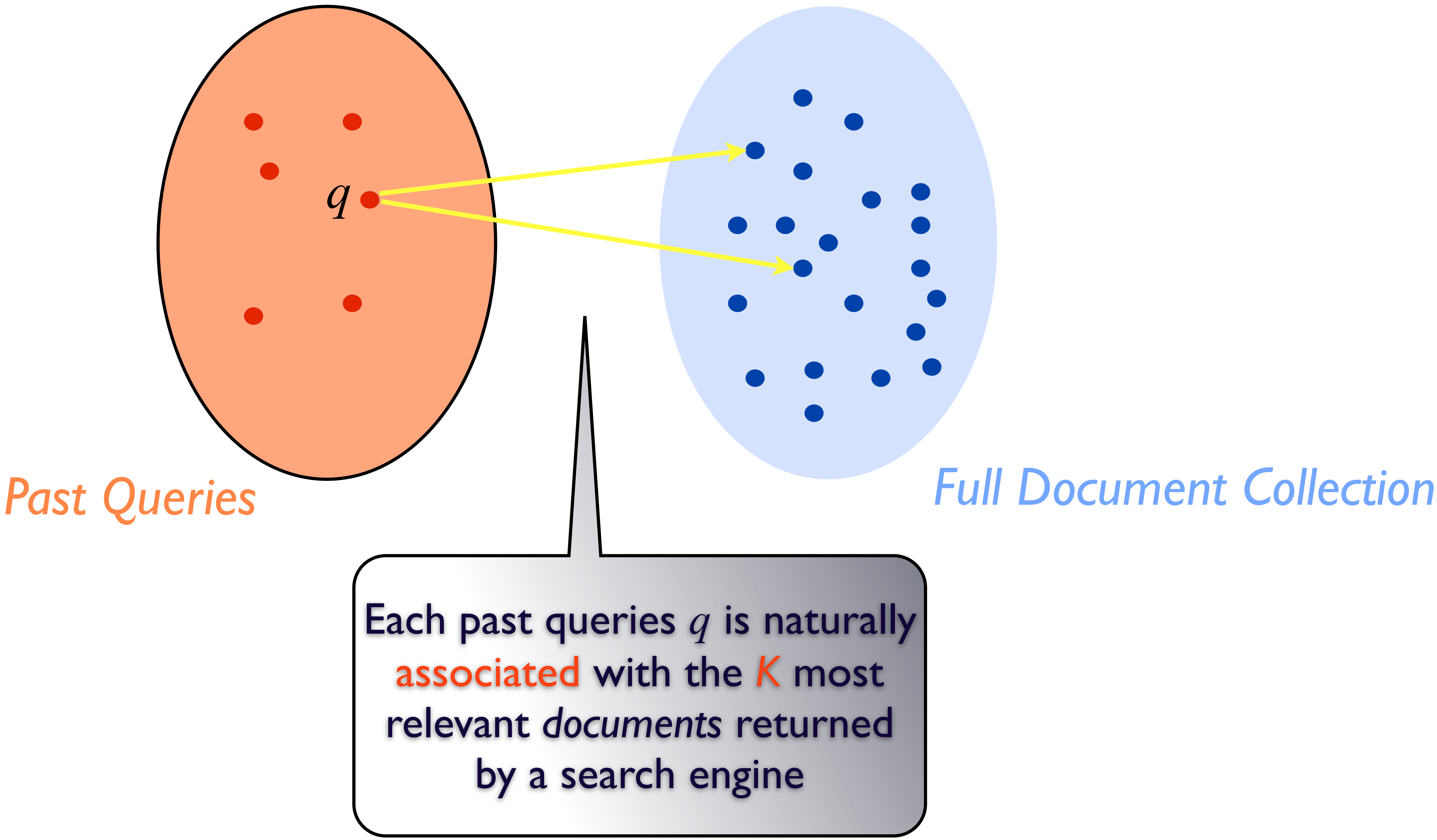
Query Expansion



Each past queries q is naturally associated with the K most relevant *documents* returned by a search engine

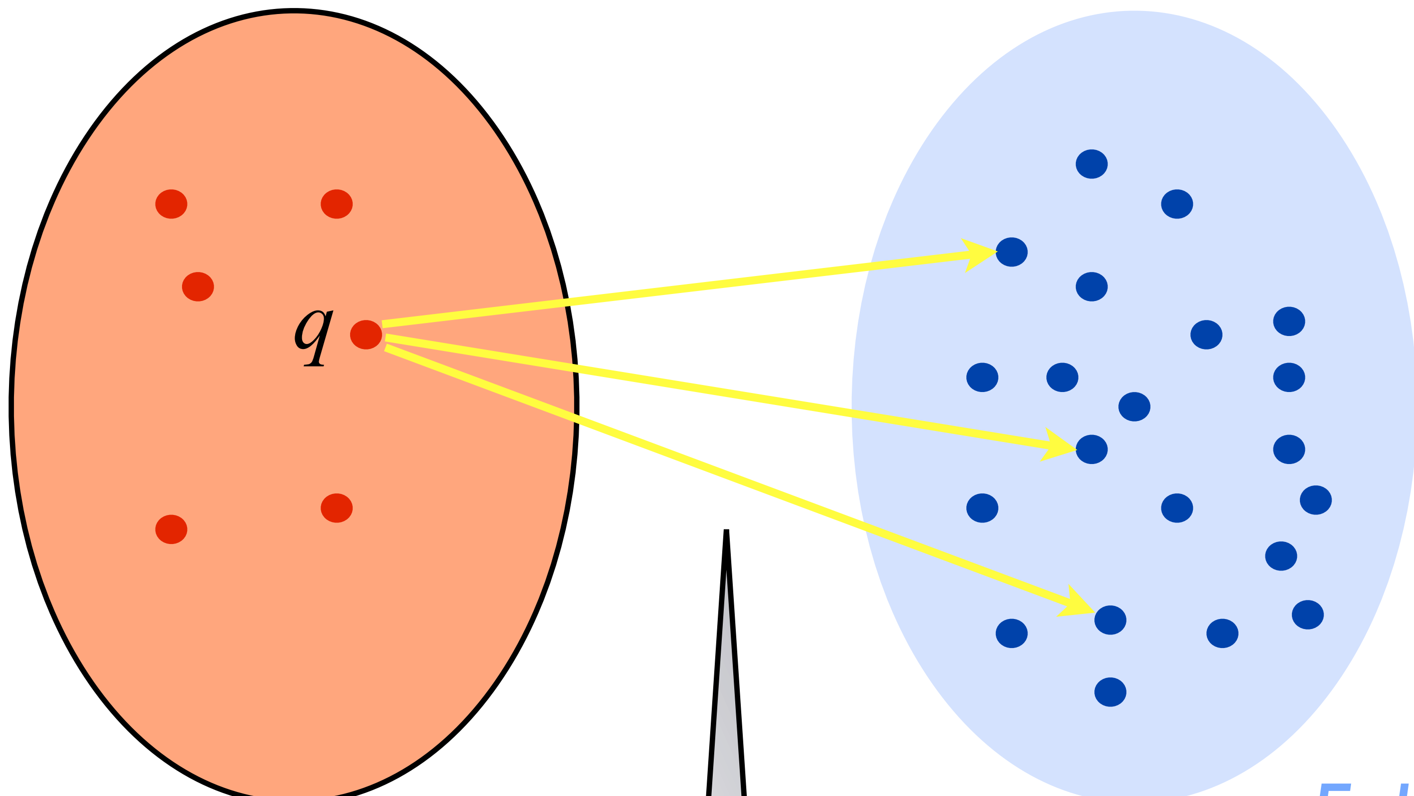
F. Scholer, H.E. Williams. **"Query association for effective retrieval"**, in Proc. of the 11th CIKM, pp. 324–331, 2002.

Query Expansion



F. Scholer, H.E. Williams. **"Query association for effective retrieval"**, in Proc. of the 11th CIKM, pp. 324–331, 2002.

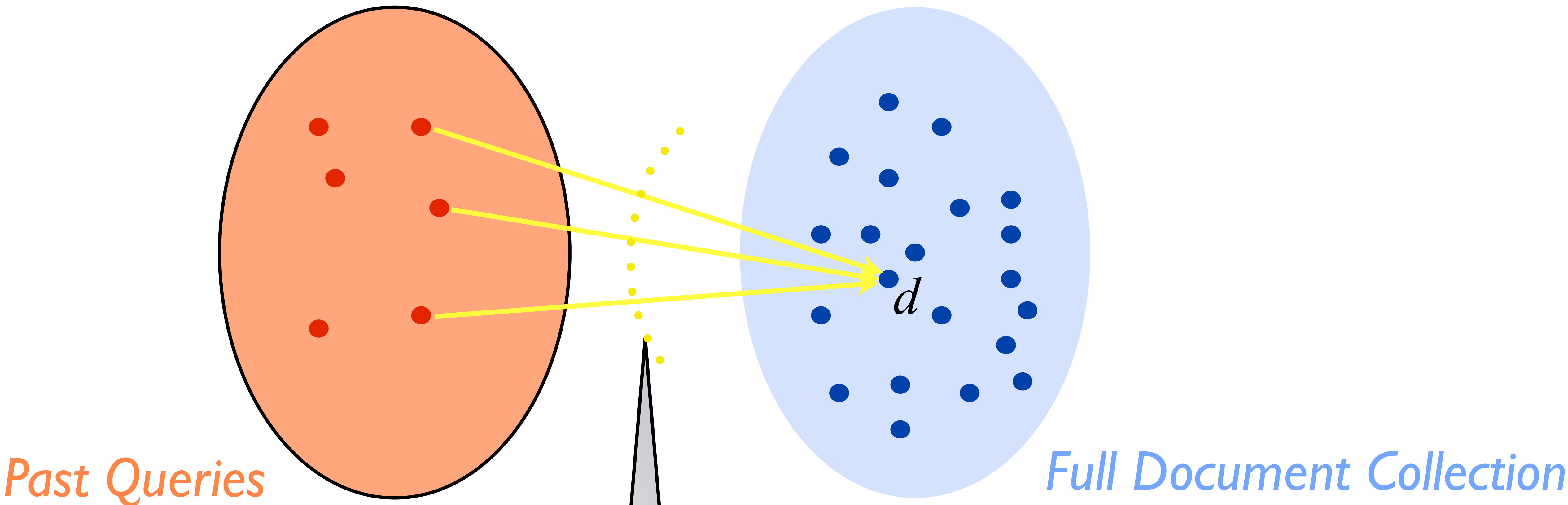
Query Expansion



Each past queries q is naturally associated with the K most relevant *documents* returned by a search engine

F. Scholer, H.E. Williams. **“Query association for effective retrieval”**, in Proc. of the 11th CIKM, pp. 324–331, 2002.

Query Expansion

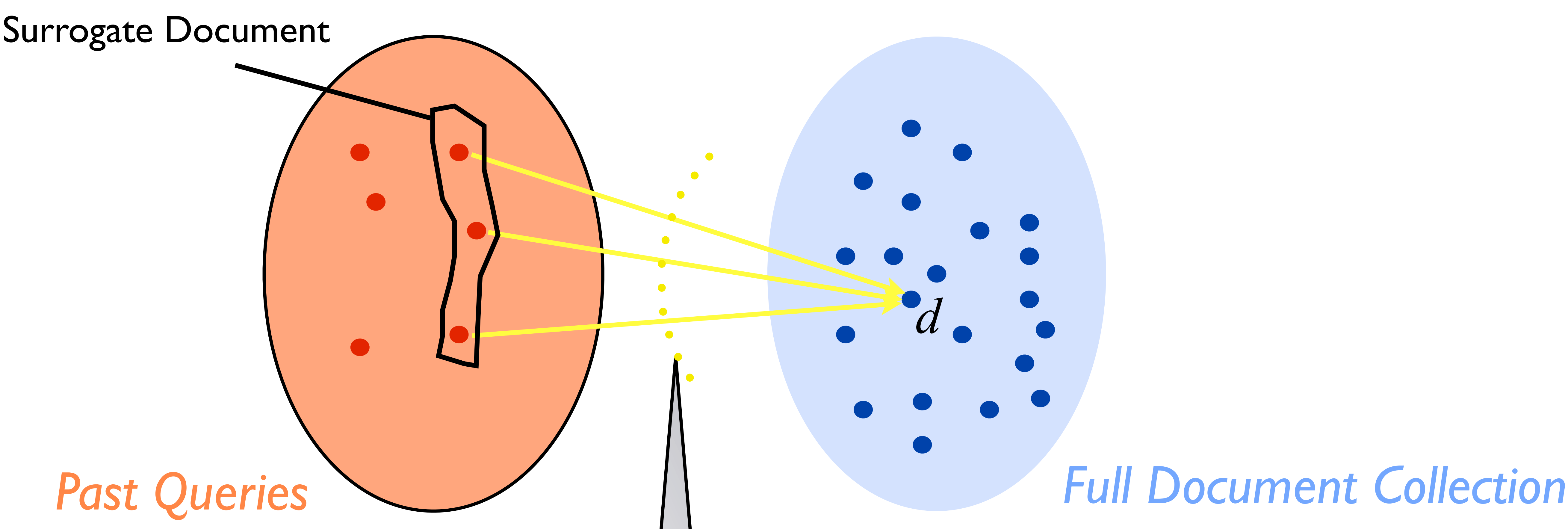


Each document d can result to be associated with many queries

Only the M closest queries are kept w.r.t. the *Okapi BM25* similarity measure

Scholer, H.E. Williams. **“Query association for effective retrieval”**, in Proc. of the 11th CIKM, pp. 324–331, 2002.
K. S. Jones, S. Walker, and S. E. Robertson, **“A probabilistic model of information retrieval: development and comparative experiments"**. Inf. Process. Manage., vol. 36, no. 6, pp. 779-808, 2000.

Query Expansion



Each document d can result to be associated with many queries

Only the M closest queries are kept w.r.t. the *Okapi BM25* similarity measure

Scholer, H.E. Williams. **"Query association for effective retrieval"**, in Proc. of the 11th CIKM, pp. 324–331, 2002.
K. S. Jones, S. Walker, and S. E. Robertson, **"A probabilistic model of information retrieval: development and comparative experiments"**. Inf. Process. Manage., vol. 36, no. 6, pp. 779-808, 2000.

Query Expansion

- Why may *surrogate documents* be a viable source of terms for expanding queries?
  - The fact that the *queries* are associated with the *document* means that, in some sense, the query terms have topical relationships with each other.
  - It may be better than expanding directly from *documents*, because the terms contained in the associated *surrogate documents* have already been chosen by users as descriptors of topics
  - It may be better than expanding directly from *queries*, because the *surrogate document* has many more terms than an individual query

Query Expansion

- Indeed, *Billerbeck et al.* do not limit themselves to exploit the query associations modeled by the bipartite graph
- In general, a *query expansion mechanism* is made up of the following steps:
    1. For a newly submitted query q, a set T of top ranked "documents" is built
    2. On the basis of T, extract and rank a list L of candidate terms
    3. Select from L the top most scoring terms and use them to expand q

B. Billerbeck, F. Scholer, H. E. Williams, and J. Zobel, **"Query expansion using associated queries"**, in Proc. of the 12th CIKM, pp. 2-9, ACM Press, 2003.

Query Expansion

- Regarding steps 1 and 2, they can be performed on either
  - the space of the Documents (FULL), or
  - the associated space of the Surrogate Documents (ASSOC)
- Four combinations are possible:
  - FULL-FULL FULL-ASSOC
    ASSOC-FULL ASSOC-ASSOC

B. Billerbeck, F. Scholer, H. E. Williams, and J. Zobel, **"Query expansion using associated queries"**, in Proc. of the 12th CIKM, pp. 2-9, ACM Press, 2003.

Query Expansion

- FULL-FULL
  - standard method, with both steps 1 and 2 on the full text Document collections

- FULL-ASSOC
  - step 1 on the space of the Documents,
  - then go to the space of the past queries (Surrogate Documents) following the associations of the bipartite graph
  - step 2 on the associated Surrogate Documents

B. Billerbeck, F. Scholer, H. E. Williams, and J. Zobel, **"Query expansion using associated queries"**, in Proc. of the 12th CIKM, pp. 2-9, ACM Press, 2003.

Query Expansion

- ASSOC-FULL
  - step 1 on the Surrogate Documents
  - then go to the space of the full Documents following the associations of the bipartite graph
  - step 2 on the full Documents

- ASSOC-ASSOC
  - both steps 1 and 2 on the Surrogate Documents

B. Billerbeck, F. Scholer, H. E. Williams, and J. Zobel, **"Query expansion using associated queries"**, in Proc. of the 12th CIKM, pp. 2-9, ACM Press, 2003.

Query Expansion

- The ASSOC-ASSOC scheme resulted 18%–20% better in P@10, P@20, P@30 than FULL-FULL expansion
- ASSOC-ASSOC was also 26%–29% better than the baseline no-expansion case
- As an example, the authors considered the query "earthquakes" (TREC query 513)
  - the average precision was
    - 0.1706 (ASSOC-ASSOC)
    - 0.1341 (no expansion)
    - 0.1162 (FULL-FULL)

B. Billerbeck, F. Scholer, H. E. Williams, and J. Zobel, **"Query expansion using associated queries"**, in Proc. of the 12th CIKM, pp. 2-9, ACM Press, 2003.

Query Expansion

- ASSOC-ASSOC
  - the expanded query is large and appears to contain only useful terms:

    earthquakes earthquake recent nevada seismograph tectonic faults perpetual 1812 kobe magnitude california volcanic activity plates past motion seismological

- FULL-FULL
  - the expanded query is more narrow

    earthquakes tectonics earthquake geology geological

B. Billerbeck, F. Scholer, H. E. Williams, and J. Zobel, **"Query expansion using associated queries"**, in Proc. of the 12th CIKM, pp. 2-9, ACM Press, 2003.

Query suggestion

- Exploit information on past users' queries
- Propose to a user a list of queries related to the one (or the ones, considering past queries in the same session) submitted
- Query suggestion vs. expansion
  - users can select the best similar query to refine their search, instead of having the query uncontrollably stuffed with a lot of terms

Query suggestion

- A naïve approach, as stated by *Zaïane and Strilets*, does not work
  - Query similarity simply based on sharing terms
  - The query "Salvatore Orlando" would be considered, to some extent, similar to "Florida Orlando", since they share term "Orlando"
- In literature there are several proposals
  - queries suggested from those appearing frequently in query sessions
  - use clustering to devise similar queries on the basis of cluster membership
  - use click-through data information to devise query similarity

O. R. Zaïane and A. Strilets, **"Finding similar queries to satisfy searches based on query traces"** in OOIS Workshops, pp. 207-216, 2002.

Query suggestion

- Exploiting query sessions
  - if a lot of previous users, when issuing the query q_1 also issue query q_2 afterwards, query q_2 is suggested for query q_1
  - *Fonseca et al.* exploited association rule mining to generate query suggestions according to the above idea

B. M. Fonseca, P. B. Golgher, E. S. de Moura, and N. Ziviani, **"Using association rules to discover search engines related queries"** in LA-WEB '03, p. 66, IEEE Computer Society, 2003.

Query suggestion

- The method used by *Fonseca et al.* is a straightforward application of association rules
  - the input data set D is composed of transactions, each corresponding to an unordered user session, where items are queries q_i
- In general, a rule extracted has the form A⇒B,

  where A and B are disjoint sets of queries
  - To reduce the computational cost, only rules where both A and B are singletons are indeed extracted:

$$q_i \Rightarrow q_j, \quad \text{where } q_i \neq q_j$$

B. M. Fonseca, P. B. Golgher, E. S. de Moura, and N. Ziviani, **"Using association rules to discover search engines related queries"** in LA-WEB '03, p. 66, IEEE Computer Society, 2003.

Query suggestion

- For each incoming query q_i
  - all the rules extracted and *sorted by confidence level*

    $q_i \Rightarrow q_1,\ q_i \Rightarrow q_2, q_i \Rightarrow q_3, \ldots., q_i \Rightarrow q_m$

  - the queries suggested are the top 5 ranked ones
- Experiments conducted using a query log of 2,312,586 queries, coming from a real Brazilian search engine
  - Low Minimum absolute support = 3 to mine the sets of frequent queries
  - This means that, given an extracted rule $q_i \Rightarrow q_j$, the unordered pair (q_i, q_j) appeared in at least 3 user sessions

B. M. Fonseca, P. B. Golgher, E. S. de Moura, and N. Ziviani, **"Using association rules to discover search engines related queries"** in LA-WEB '03, p. 66, IEEE Computer Society, 2003.

Query suggestion

- How did *Fonseca et al.* evaluate the quality of suggestions?
  - The method was based on a survey among a small group of people
  - They asked whether the top 5 queries were relevant or not
  - With the 95 *most frequently submitted queries*, the system is able to achieve a precision of 90.5%
    - However, this nice behavior happens for frequent queries only, which are largely supported in the query sessions
  - The precision drops by increasing the number of suggested queries

B. M. Fonseca, P. B. Golgher, E. S. de Moura, and N. Ziviani, **"Using association rules to discover search engines related queries"** in LA-WEB '03, p. 66, IEEE Computer Society, 2003.

Query suggestion

- *Baeza-Yates et al.* use clustering and exploits a two-tier system
  - An offline component builds clusters of past queries, using query text along with the text of clicked URLs.
  - An online component recommends queries on the basis of the input one

Baeza-Yates, C. Hurtado, and M. Mendoza, **“Query Recommendation Using Query Logs in Search Engines’**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.

Query suggestion

- Offline component:
  - the clustering algorithm operates over queries enriched by a selection of terms extracted from the documents pointed by the user clicked URLs.
  - Clusters computed by using an implementation of the k-means algorithm contained in the CLUTO software package
  - Similarity between queries computed according to a vector-space approach
    - Vectors $\vec{q}$ of *n* dimensions, one for each term

R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.

*http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview

Query suggestion

- Offline component:
  - q_i is the i-th component of the vector $\overrightarrow{q}$ associated with the term t_i of the vocabulary (all different words are considered)

$$q_i = \sum_{u \in URLs} \frac{\text{Clicks}\,(q,u) \times \text{Tf}\,(t_i,u)}{\max_t \text{Tf}\,(t,u)}$$

R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.

Query suggestion

- Offline component:
  - q_i is the i-th component of the vector $\overrightarrow{q}$ associated with the term t_i of the vocabulary (all different words are considered)

Percentage of clicks that URL u receives when answered in response to query q

$$q_i = \sum_{u \in URLs} \frac{\text{Clicks}\,(q,u) \times \text{Tf}\,(t_i,u)}{\max_t \text{Tf}\,(t,u)}$$

R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.

Query suggestion

- Offline component:
  - q_i is the i-th component of the vector $\overrightarrow{q}$ associated with the term t_i of the vocabulary (all different words are considered)

Percentage of clicks that URL u receives when answered in response to query q

Number of occurrences of the term in the document pointed to URL u

$$q_i = \sum_{u \in URLs} \frac{\mathrm{Clicks}\,(q,u) \times \mathrm{Tf}\,(t_i,u)}{\max_t \mathrm{Tf}\,(t,u)}$$

R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.

Query suggestion

- Offline component:
  - q_i is the i-th component of the vector $\overrightarrow{q}$ associated with the term t_i of the vocabulary (all different words are considered)

Percentage of clicks that URL u receives when answered in response to query q

Number of occurrences of the term in the document pointed to URL u

Sum over all the clicked URL u for query q

$$q_i = \sum_{u \in URLs} \frac{\text{Clicks}\,(q, u) \times \text{Tf}\,(t_i, u)}{\max_t \text{Tf}\,(t, u)}$$

R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.

Query suggestion

- Offline component:
  - k-means clustering algorithm
    - Partitions objects into k disjoint clusters (k is a parameter)
    - Center-based: Each object in a cluster is closer to its own center than all the other k-1 centers
    - Iterative
      - Start from casual k centers
      - At each iteration, assign points to the closest center, and then recompute the centers as the means of the current cluster points
      - The algorithm stops at a local minimum

Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.
P.-N. Tan, M. Steinbach, V. Kumar. **"Introduction to Data Mining"**. Pearson Addison-Wesley.

Query suggestion

- Offline component:
  - k-means minimizes (local minimum) the SSE (Sum of Squared Errors)

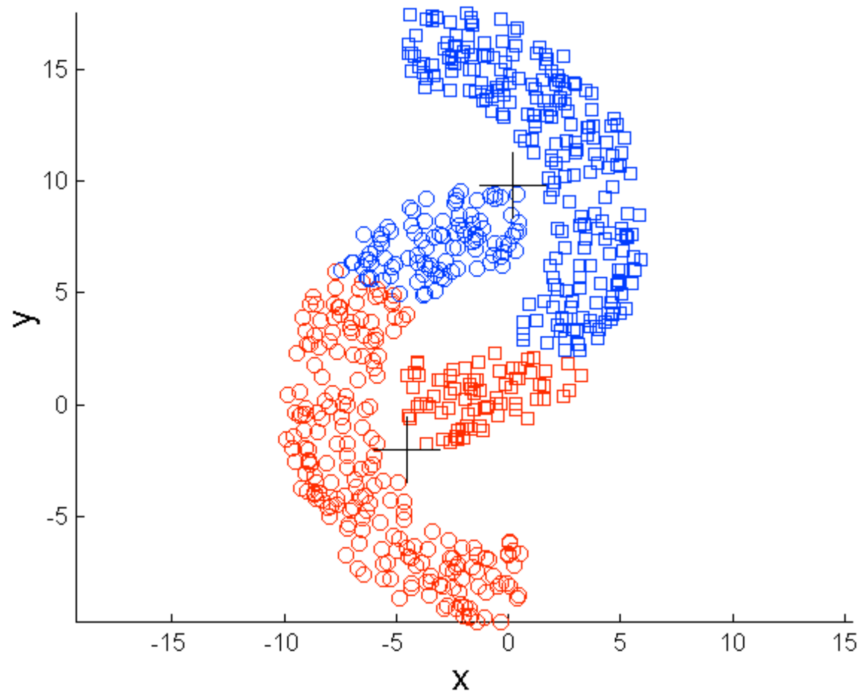
$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p,\ m_i)^2$$

  - The errors are the distances of each point from its own mean m_i (the centroid of center C_i)
  - Two clustering results can be evaluated in terms of SSE

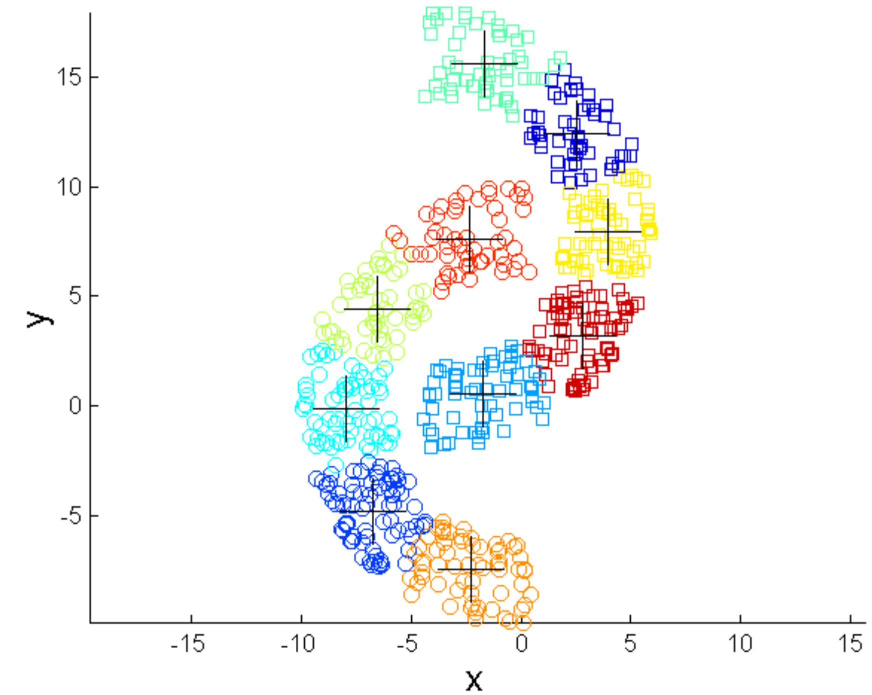
R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.
P.-N. Tan, M. Steinbach, V. Kumar. **"Introduction to Data Mining"**. Pearson Addison-Wesley.

Query suggestion

- Offline component:
  - A way to reduce *SSE* is to increase the number *k* of clusters
  - This work also when the natural clusters are not globular



k=2



k=11

R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.
P.-N. Tan, M. Steinbach, V. Kumar. **"Introduction to Data Mining"**. Pearson Addison-Wesley.

Query suggestion

- Offline component:
  - *Baeza-Yates et al.* executed k-means with different values of k
  - The SSE becomes even smaller by increasing k
  - They selected $k = 600$, for which the average error is 0.6

R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.

Query suggestion

- Online component:

(I) for an input query the most similar cluster is selected

- each cluster has a natural representative, i.e. its centroid

(II) ranking of the queries of the cluster, according to:

- attractiveness of query answer, i.e. the fraction of the documents returned by the query that captured the attention of users (clicked documents)
- similarity w.r.t. the input query (the same distance used for clustering)
- popularity of query, i.e. the frequency of the occurrences of queries

R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.

Query suggestion

- Experiments:
  - The query log (and the relative collection) comes from the *TodoCL* search engine
    - 6,042 unique queries along with associated click-throughs
    - 22,190 registered clicks spread over 18,527 different URLs
  - The algorithm was evaluated on ten different queries by a user study.
  - Presenting query suggestions ranked by attractiveness of queries yielded to more precise and high quality suggestions

R. Baeza-Yates, C. Hurtado, and M. Mendoza, **"Query Recommendation Using Query Logs in Search Engines'**, pp. 588-596. Vol. 3268/2004 of LNCS, Springer, 2004.

Query suggestion

- *Jones et al.* proposed a model for generating queries to be suggested based the concept of *query rewriting*
- A query is rewritten into a new one by means of query or phrase substitutions
    - e.g., the query *"leopard installation"* can be rewritten into *"mac os x 10.5 installation"*
- They were motivated by sponsored search, in which enormous numbers of queries must be matched to a much smaller corpus of advertiser listings

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

- Many ways in which the new suggested query q_j could be related to the original query q_i
- Same meaning
  - spelling change
  - synonym substitution (for example colloquial versus medical terminology)
- Change in meaning
  - Generalization (loss of specificity of original meaning)
  - Specification (increase of specificity relative to original meaning)
  - Related terms

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

- The data used comes from logs of user web accesses.
- A candidate reformulation is a *pair of successive queries* issued by a *single user* on a *single day:*

$$\mathrm{candidateQueryPairs}(user_i, day_j) = \{< q_1, q_2 >: (q_1 \neq q_2) \wedge \exists t : query_t(user_i, q_1) \wedge query_{t+1}(user_i, q_2)\}$$

- Using a phrase identification method, queries are segmented into phrases (where even single words can be phrases)
- The query pair: (britney spears) (mp3s) → (britney spears) (lyrics) gives both
  - an instance of a whole query pair
  - the pair: (mp3s) → (lyrics)

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

- *Jones et al.* used the *log likelihood ratio* (LLR) score in order to
    - distinguish related query and phrase pairs from candidate pairs that are unrelated.
- A high value for LLR suggests that there is a strong dependence between terms in *q1* and terms in *q2*

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

- *Jones et al.* used the *log likelihood ratio* (LLR) score in order to
    - distinguish related query and phrase pairs from candidate pairs that are unrelated.
- A high value for LLR suggests that there is a strong dependence between terms in $q1$ and terms in $q2$

dog →	dogs	9185	(pluralization)
dog →	cat	5942	(both instances of 'pet')
dog →	dog breeds	5567	(generalization)
dog →	dog pictures	5292	(more specific)
dog →	80	2420	(random junk or noise)
dog →	pets	1719	(generalization – hypernym)
dog →	puppy	1553	(specification – hyponym)
dog →	dog picture	1416	(more specific)
dog →	animals	1363	(generalization – hypernym)
dog →	pet	920	(generalization – hypernym)

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

- *Jones et al.* used the *log likelihood ratio* (LLR) score in order to
  - distinguish related query and phrase pairs from candidate pairs that are unrelated.
- A high value for LLR suggests that there is a strong dependence between terms in $q1$ and terms in $q2$

Suggested substitutions for term (or query) "dog"

Note the LLR ranks computed on user query rewriting sessions

dog	$\rightarrow$ dogs	9185	(pluralization)
dog	$\rightarrow$ cat	5942	(both instances of 'pet')
dog	$\rightarrow$ dog breeds	5567	(generalization)
dog	$\rightarrow$ dog pictures	5292	(more specific)
dog	$\rightarrow$ 80	2420	(random junk or noise)
dog	$\rightarrow$ pets	1719	(generalization – hypernym)
dog	$\rightarrow$ puppy	1553	(specification – hyponym)
dog	$\rightarrow$ dog picture	1416	(more specific)
dog	$\rightarrow$ animals	1363	(generalization – hypernym)
dog	$\rightarrow$ pet	920	(generalization – hypernym)

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

- Experimental method
  - Rank them by using LLR
  - Select a sample of the top-ranked suggestions
  - Manually classify each pair of <query, suggestion> on a scale 1-4 (see the following slide)
  - This produces a supervised knowledge base that could be exploited for learn a model

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

- Supervised knowledge base: labelled dataset

Classes of Suggestion Relevance

Class	Score	Examples		
Precise rewriting	1	automotive insurance	↦	automobile insurance
		corvette car	↦	chevrolet corvette
		apple music player	↦	apple ipod
		apple music player	↦	ipod
		cat cancer	↦	feline cancer
		help with math homework	↦	math homework help
Approximate rewriting	2	apple music player	↦	ipod shuffle
		personal computer	↦	compaq computer
		hybrid car	↦	toyota prius
		aeron chair	↦	office furniture
Possible rewriting	3	onkyo speaker system	↦	yamaha speaker system
		eye-glasses	↦	contact lenses
		orlando bloom	↦	johnny depp
		cow	↦	pig
		ibm thinkpad	↦	laptop bag
Clear mismatch	4	jaguar xj6	↦	os x jaguar
		time magazine	↦	time and date magazine

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

- Once built the labeled dataset, a learning algorithm can be trained
- *Jones et al.* trained a binary classifier

  $g(q1, q2) \rightarrow \{Positive, Negative\}$

- In order to reduce the class labels from $1 \div 4$ to $\{Positive, Negative\}$ the used the following grouping of ranks
  - **Broad**
    - *Positive=scores 1+2+3* *Negative=score 4*
  - **Specific**
    - *Positive=scores 1+2* *Negative=scores 3+4*

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

- Moreover, using the original integer ranks 1,2,3,4 the authors also exploited a linear regression predictive method
- Both the binary classifier and linear regression predictor were trained on the basis of several features extracted from the pairs *<q1, q2>*
- Example of features:
  - Word edit-distance
  - Character edit-distance

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query suggestion

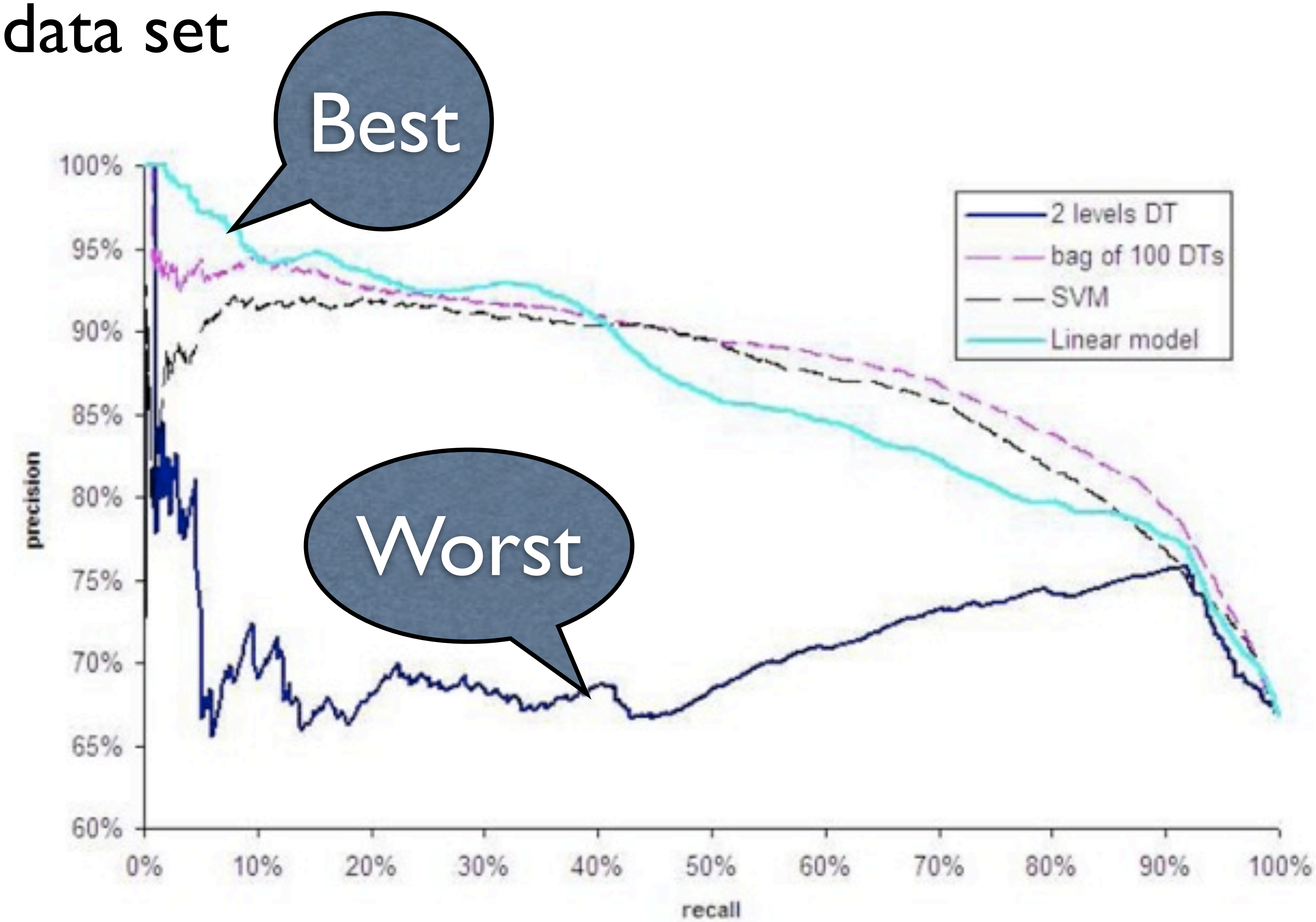
- Some results of Recall and Precision of the classifiers
- Results obtained by 100 random 90%-10% train-test splits on the labeled data set

DT: Decision tree with only two levels

SVN: Support Vector Machine

Bags of 100 DTs

Linear regression prediction



R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query personalization

- Personalization consists in presenting different ranked results for the same issued query, depending on
  - different searcher tastes
  - different contexts (places or times)
- For examples, a mathematician and an economist who issue the same query "game theory"
  - a mathematician would return many results on theory of games and theoretical studies
  - an economist would be rather interested in applications of game theory real-world economy problems

R. Jones, B. Rey, O. Madani, and W. Greiner, "**Generating query substitutions**" in WWW '06, pp. 387-396, ACM Press, 2006.

Query personalization

- One possible method to achieve Personalization is
    - "re-ranking" search results according to a specific user's profile, built automatically by exploiting knowledge mined from query logs
- We start from a negative results
    - *Teevan et al.* demonstrate that for queries which showed less variations among individuals, re-ranking results according to a personalization function may be insufficient (or even dangerous)

J. Teevan, S. T. Dumais, and E. Horvitz, **"Beyond the commons: Investigating the value of personalizing web search"**, in Proc. of Workshop on New Technologies for Personalized Inf. Access (PIA '05), 2005.

Query personalization

- *Liu et al.* categorize users and queries with a set of relevant categories
  - Return the top 3 categories for each user query
  - The categorization function is automatically computed on the basis of the retrieval history of each user
- The set of different categories are the same as the ones used by the search engine to classify web pages
  - thus such user-based categorization is used to personalize results, thus focusing on the most relevant results for each user
- The two main concepts used are
  - *User Search History*
  - *User Profile* (automatically generated)

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

- *User Search History*
  - Query "Apple"
  - Category *Food&Cooking*
  - Clicked results *page1.html* and *page2.html*

- *Users Profile*
  - Users Profile stores the set of categories hit by the corresponding user
  - Each category is associated with a sort of description: a set of weighted keywords
- For each user, Search History and User Profile are stored as
  - a set of three matrices *DT*, *DC*, and *M*

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

- *User Search History*
  - m: clicked documents or issued queries
  - n: distinct terms appearing in clicked documents or queries

Doc/Term	leopard	medow	grass	screen	tv
D1	1	0	0	0	0
D2	0.58	0.58	0	0	0
D3	1	0.7	0.5	0	0
D4	0	0	0	1	0
D5	1	0	0	0.6	0.4

m-by-n matrix DT

$DT[i, j]$ is greater than zero if term j appears in document/query i.
The entry is filled-in by computing the normalized TF-IDF score.

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

- *User Search History*
  - m: clicked documents or issued queries
  - p: possible categories

Doc/Categ	NATURE	HI-TECH
D1	1	0
D2	1	0
D3	1	0
D4	0	1
D5	0	1

m-by-p matrix DC

$DC[i, j]$ is 1 whether documents/queries i is related to category j, 0 otherwise

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

- *User Profile*
  - n: distinct terms appearing in clicked documents or queries
  - p: possible categories

Categ/Term	leopard	medow	grass	screen	tv
NATURE	1	0.4	0.4	0	0
HI-TECH	0	0	0	1	0.4

p-by-n matrix M

Matrix M is the user profile and is learnt by the previous two matrices DT, and DC by means of a machine learning algorithm.
Each row is a vector representing a category in the term-space.
Both categories and documents are represented in the same vector space and similarities between them can be computed.

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

Query "leopard"

Query "screen"

Doc/Term	leopard	medow	grass	screen	tv
D1	1	0	0	0	0
D2	0.58	0.58	0	0	0
D3	1	0.7	0.5	0	0
D4	0	0	0	1	0
D5	1	0	0	0.6	0.4

These rows store representative terms of the clicked documents (weighed by their TF-IDF) scores

Query "leopard"

Query "screen"

Doc/Categ	NATURE	HI-TECH
D1	1	0
D2	1	0
D3	1	0
D4	0	1
D5	0	1

Clicked documents

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

- The process of generating/learning the profile matrix M can be viewed as a multi-class text categorization task
- Algorithms to learn profiles
  - Linear Least Squares Fit (LLST)
  - Rocchio-based Algorithm
  - K-Nearest Neighbor (kNN)
  - Adaptive Learning
- kNN does not need to build matrix M

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

- Training/Test Data sets manually prepared by 7 users
  - queries submitted to Google. For each query, the user identified the set of related categories and relevant documents

Statistics	User 1	User 2	User 3	User 4	User 5	User 6	User 7
# of interest catetories	10	8	8	8	10	8	9
# of search records (queries)	37	50	61	26	33	29	29
avg # of related search records to one category	3.7	6.3	7.6	3.25	3.3	3.63	3.2
# of relevant documents	236	178	298	101	134	98	115
avg # of categories in one search record	1.1	1	1	1	1	1	1
# of distinct terms	7012	5550	6421	4547	4584	4538	4553

- Evaluation based one the 10-fold cross-validation strategy
  - 10 tests, each time choosing a partition testing, and the other 9 training
  - for each of the 10 tests, 4 matrixes were prepared:

DT_{train} DC_{train} DT_{test} DC_{test}

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

- Performance metric
  - *Liu et al.* are interested in measuring the accuracy of query classification on the *top 3 categories* returned for each user

$$\text{Accuracy} = \frac{1}{n} \sum_{c_j \in \text{topK}} \frac{1}{1 + \text{rank}_{c_i} - \text{ideal\_rank}_{c_i}}$$

n is the number of related categories to the query

topK are the K category vectors (3 in these experiments) having the highest cosine similarity measure with the query

$rank_{ci}$ is the rank of category ci, i.e. an integer ranging from 1 to K (3), as computed by $sim(q; c_j)$,

$ideal\_rank_{ci}$ is the rank assigned by the user

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

- Performance metric
  - *Liu et al.* are interested in measuring the accuracy of query classification on the top 3 categories returned for each user

$$\text{Accuracy} = \frac{1}{n} \sum_{c_j \in \text{topK}} \frac{1}{1 + \text{rank}_{c_i} - \text{ideal\_rank}_{c_i}}$$

Accuracy = 1 when the returned ranks match the ideal ones, and $n=K$

n is the number of related categories to the query

topK are the K category vectors (3 in these experiments) having the highest cosine similarity measure with the query

$rank_{ci}$ is the rank of category ci, i.e. an integer ranging from 1 to K (3), as computed by $sim(q; c_j)$,

$ideal\_rank_{ci}$ is the rank assigned by the user

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

Query personalization

- Results

Method	pLLSF	LLSF	bRocchio	kNN
Avg Accuracy	0.8236	0.7843	0.8224	0.8207

- If *small data sets* concerning *user search history* are available, the accuracy of methods using the user search history is small
- Adaptive learning methods (the user profile is modified by the new search records) should be preferable, due to the extremely high variability of queries in search engines
- Not all the methods above are suitable for becoming adaptive

F. Liu, C. Yu, and W. Meng, **"Personalized web search by mapping user queries to categories"** in 11th CIKM '02, pp. 558-565, ACM Press, 2002.

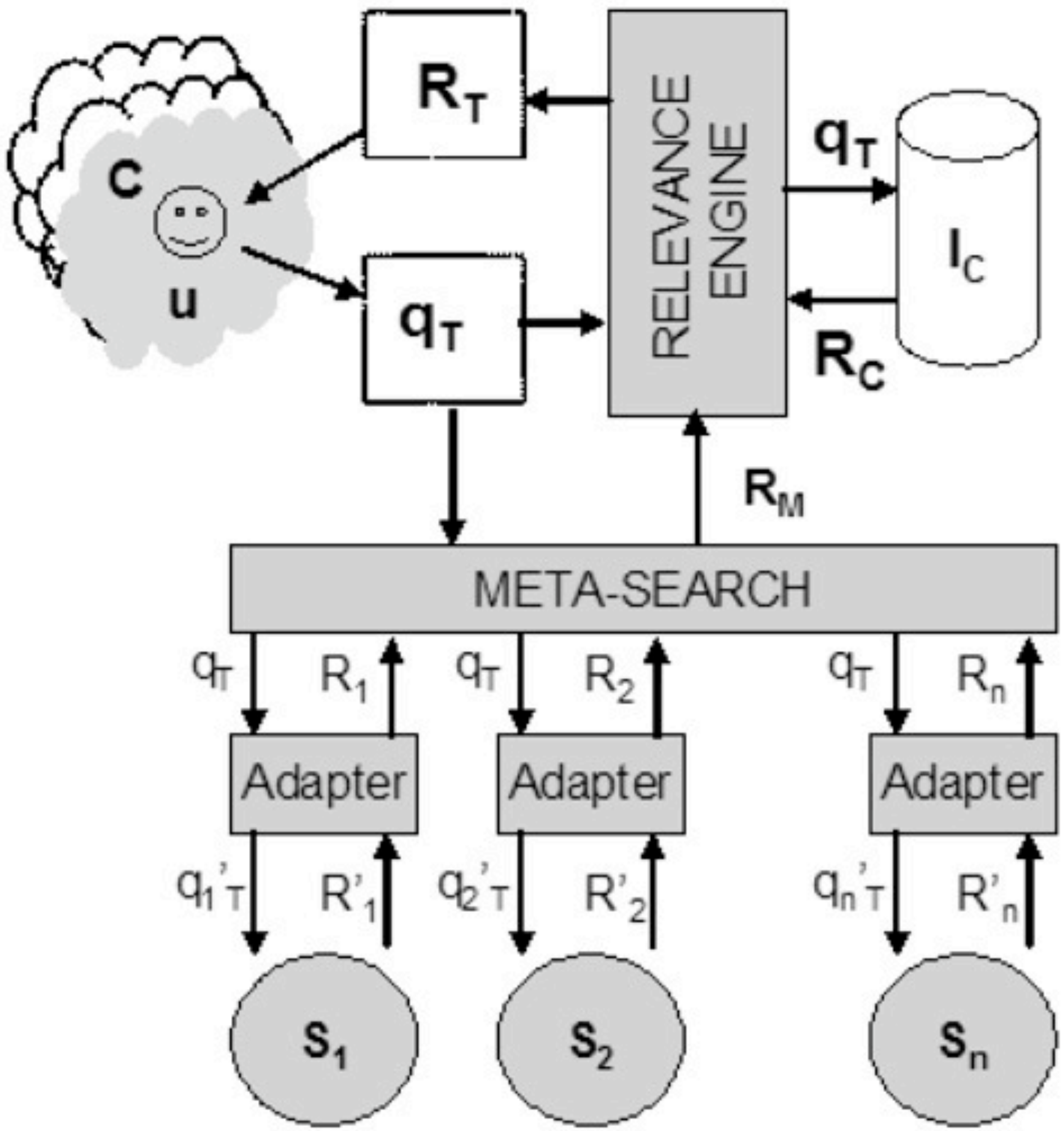
Query personalization

- *Boydell and Smith* use snippets of clicked results
  - They argued that results (in a result list) are selected because the user recognizes in their snippets certain combinations of terms that are related to their information needs
- They propose to build a community-based snippet index that reflects the evolving interests of a group of searchers
- The index is used for (community-based) personalization through re-ranking of the search results
  - The index is built at the proxy-side
  - No usage information is stored at the server-side
  - Harmless with respect to issues of users' privacy

O. Boydell and B. Smyth, **"Capturing community search expertise for personalized web search using snippet-indexes"**, in CIKM '06, pp. 277-286, ACM, 2006

Query personalization

Collaborative Web Search (CWS)



- A user u in some community C
- The results of an initial meta-search, R_M, are revised with reference to the community's snippet index I_C
- A new result-list, R_C, is returned. This list is adapted to community preferences.
- R_M and R_C are combined and returned to the user as R_T

O. Boydell and B. Smyth, **"Capturing community search expertise for personalized web search using snippet-indexes"**, in CIKM '06, pp. 277-286, ACM, 2006

Query personalization

- A common method explooited by other CWS systems:
  - find a set of related queries $q_1, \ldots, q_k$ such that these queries share some minimal overlapping terms within q_T
  - the main issue of this method is that sometimes two related queries do not contain any common terms
    - *e.g. "Captain Kirk" and "Starship Enterprise";*

Query personalization

- In the CWS by *Boydell and Smyth*, each past queries is indexed along with the surrogate clicked documents (snippets)
- Main advantage:
  - A result *r* that was previously selected for query Q1=*"Captain Kirk"*, can potentially be returned in response to query Q2=*"Starship Enterprise"*
  - If the terms in Q1 occurred in the snippet of a result previously selected in response to Q2

Query personalization

- Evaluation by a live-user trial of the system (snippet-based indexing on top of a Google plus Yahoo meta-search engine)
    - About 60 employees of a local software firm, for a period of 2 weeks recording a total of 430 search sessions.
    - Results was recorded both at the end of the first week, and at the end of the entire period

Metric	Week 1	Week 2
Total sessions	246	184
Overall Success Rate	41%	60%

- Many users found the re-ranking of search result useful (Overall Success Rate)
- This rate raised in the second week denoting that a longer training period improves the quality of personalization

O. Boydell and B. Smyth, **“Capturing community search expertise for personalized web search using snippet-indexes"**, in CIKM '06, pp. 277-286, ACM, 2006

Query personalization

- *Dou et al.* carried out a large-scale evaluation of personalized search strategies
- The framework is made up of four parts:

(1) Query results retrieval

(2) Personalization

(3) Ranked lists combination

(4) Evaluation of personalization effectiveness.

Z. Dou, R. Song, and J. Wen, **"A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization

(1) Query results retrieval

- return the top 50 search results, obtained from the MSN search engine for the test query q

(2) Personalization

- given the list U returned by the search engine, rank its elements according to the personalization score (from the original ranking τ_1 to the personalized one τ_2)
- personalization is analyzed under a *person-level re-ranking strategy* (by considering the history of a single user to carry out personalization), or under a *group-level re-ranking strategy* (by focusing on queries and results of a community of people).

(3) Ranked lists combination

- uses the Borda fusion algorithm to merge τ_1 and τ_2 into the final ranked list τ

Z. Dou, R. Song, and J. Wen, **"A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization

- *Person-level re-ranking strategies*
  - P-Click: *Personal Click*
    - Click count on result p by user u on query q
  - L-Profile: *Long-term - Personal Profile*
    - Two vectors associated with the user (profile) and the page, defined on the basis 67 pre-defined topic categories obtained by the KDD-Cup-2005
  - S-Profile: *Short-term - Personal Profile*

    Only the pages that were the most recently seen by user u
  - LS-Profile: *Long & Short-term - Personal Profile*
    - Fuse the long-term personalized score and the short-term personalized score using a simple linear combination

Z. Dou, R. Song, and J. Wen, **"A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization

- *Group-level re-ranking strategies*
  - The group-based personalization is tested with a kNN classifier approach
    - The personalization is based on the k users having the closest preferences with the current user.

Z. Dou, R. Song, and J. Wen, **“A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization

- Evaluation
  - The study differs deeply from the previous ones since it does not exploit any *live-user trials*, but instead an evaluation function based on query log sessions
  - The MSN search engine along with a query log coming from the same engine are used as testing framework
    - Query logs collecting 12 days of queries submitted in August 2006 was used
  - *T*hey use information about past clicks done by users to evaluate the relevance of the personalized ranking
    - In particular, evaluation is done through the use of two measurements: Rank Scoring [D. H. John et al.] and Average Rank [Qiu et al.]

Z. Dou, R. Song, and J. Wen, **"A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

D. H. John S. Breese and C. Kadie. **"Empirical analysis of predictive algorithms for collaborative filtering"**. In Proc. of UAI '98, pages 43–52, 1998

F. Qiu and J. Cho, **"Automatic identification of user interest for personalized search"**, in WWW '06, pp. 727-736, ACM, 2006.

Query personalization

- Evaluation: *basic statistics of the dataset* (MSN log)
  - All 10,000 users have search activities in the training set, because users in the test set are sampled from the training days
  - More than 80% distinct queries are only issued once in a 12-day period, and about 90% distinct queries string are issued only by one user.
  - The 3% most popular distinct queries are issued by more than 47% users

Item	ALL	Training	Test
#days	12	11	1
#users	10,000	10,000	1,792
#queries	55,937	51,334	4,639
#distinct queries	34,203	31,777	3,465
#Clicks	93,566	85,642	7,924
#Clicks/#queries	1.6727	1.6683	1.7081
#sessions	49,839	45,981	3,865

Z. Dou, R. Song, and J. Wen, **“A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization

- The baseline method is WEB (search engine without personalization)
- Column ***all*** correspond to the entire query log
- Column ***not-optimal*** corresponds to the queries whose top result was not the one selected by users, i.e. the queries on which the search engine performed poorly
- Click-based methods (P-Click and G-Click) always outperform the baseline
- However, the cumulative results are not exciting:

method	all		not-optimal	
	Rank Similarity	Average Rank	Rank Similarity	Average Rank
WEB	69.4669	3.9240	47.2623	7.7879
P-Click	**70.4350**	**3.7338**	**49.0051**	**7.3380**
L-Profile	66.7378	4.5466	45.8485	8.3861
S-Profile	66.7822	4.4244	45.1679	8.3222
LS-Profile	68.5958	4.1322	46.6518	8.0445
G-Click	70.4168	3.7361	48.9728	7.3433

Z. Dou, R. Song, and J. Wen, **"A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization

- *Dou et al.* evaluated the variance in results clicked for a query
- They showed that personalization is effective whenever this variance is high
  - This means that there are many topics associated with a single result returned for a query

$$\text{ClickEntropy}(q) = \sum_{p \in \mathcal{P}(q)} -P(p|q) \log_2 P(p|q)$$

- $ClickEntropy(q) = 0$ iff $P(p|q)=1$
  - Therefore, the minimum entropy is obtained when clicks are always on the same page. Personalization, in this case, is of little (or no) utility.

Z. Dou, R. Song, and J. Wen, **"A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization

- *Dou et al.* evaluated the variance in results clicked for a query
- They showed that personalization is effective whenever this variance is high
  - This means that there are many topics associated with a single result returned for a query

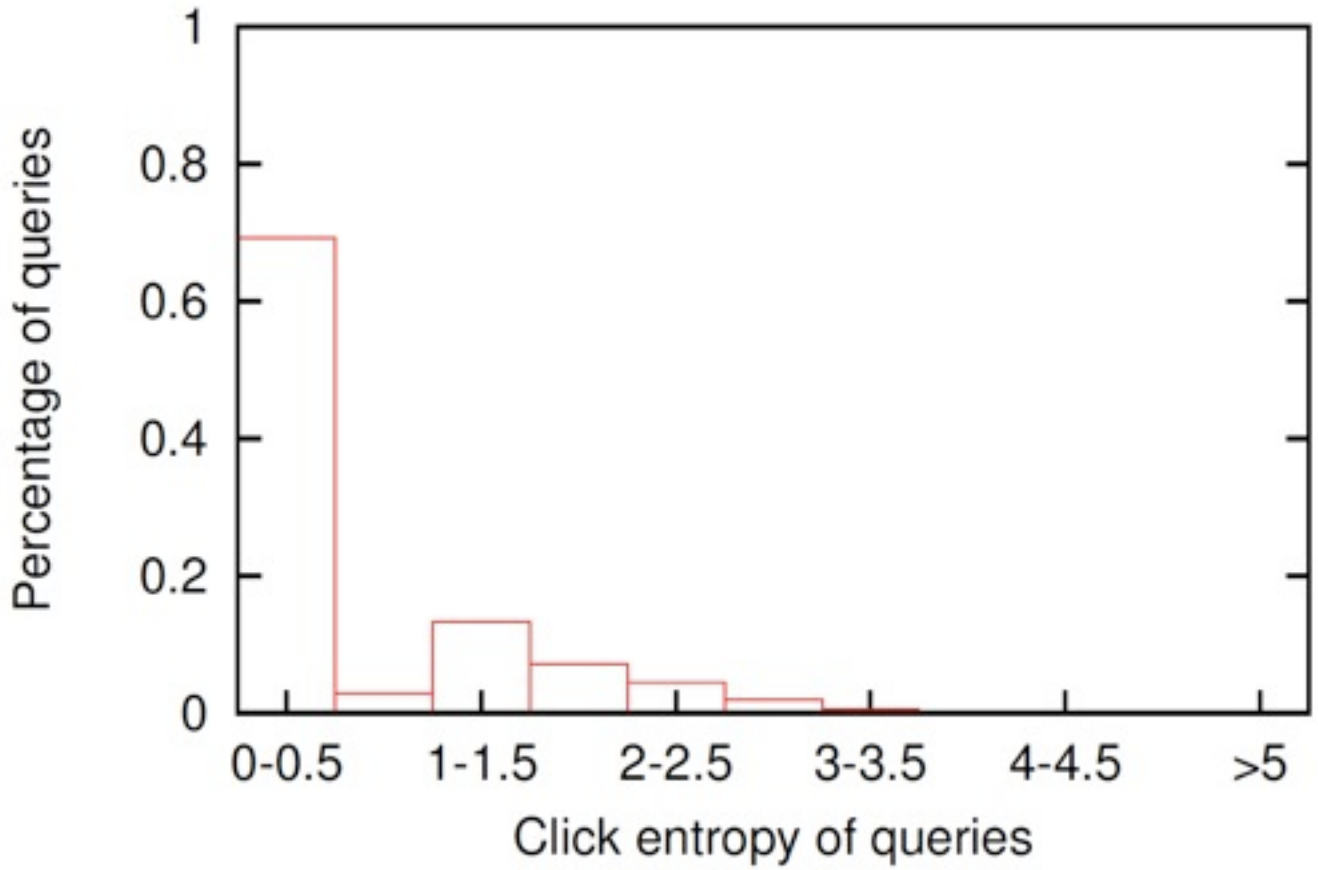
$$\text{ClickEntropy}(q) = \sum_{p \in \mathcal{P}(q)} -P(p|q) \log_2 P(p|q)$$

$$P(p|q) = \frac{|\text{Clicks}(q, p, \bullet)|}{|\text{Clicks}(q, \bullet, \bullet)|}$$

- *ClickEntropy(q) = 0* iff *P(p|q)=1*
  - Therefore, the minimum entropy is obtained when clicks are always on the same page. Personalization, in this case, is of little (or no) utility.

Z. Dou, R. Song, and J. Wen, **"A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

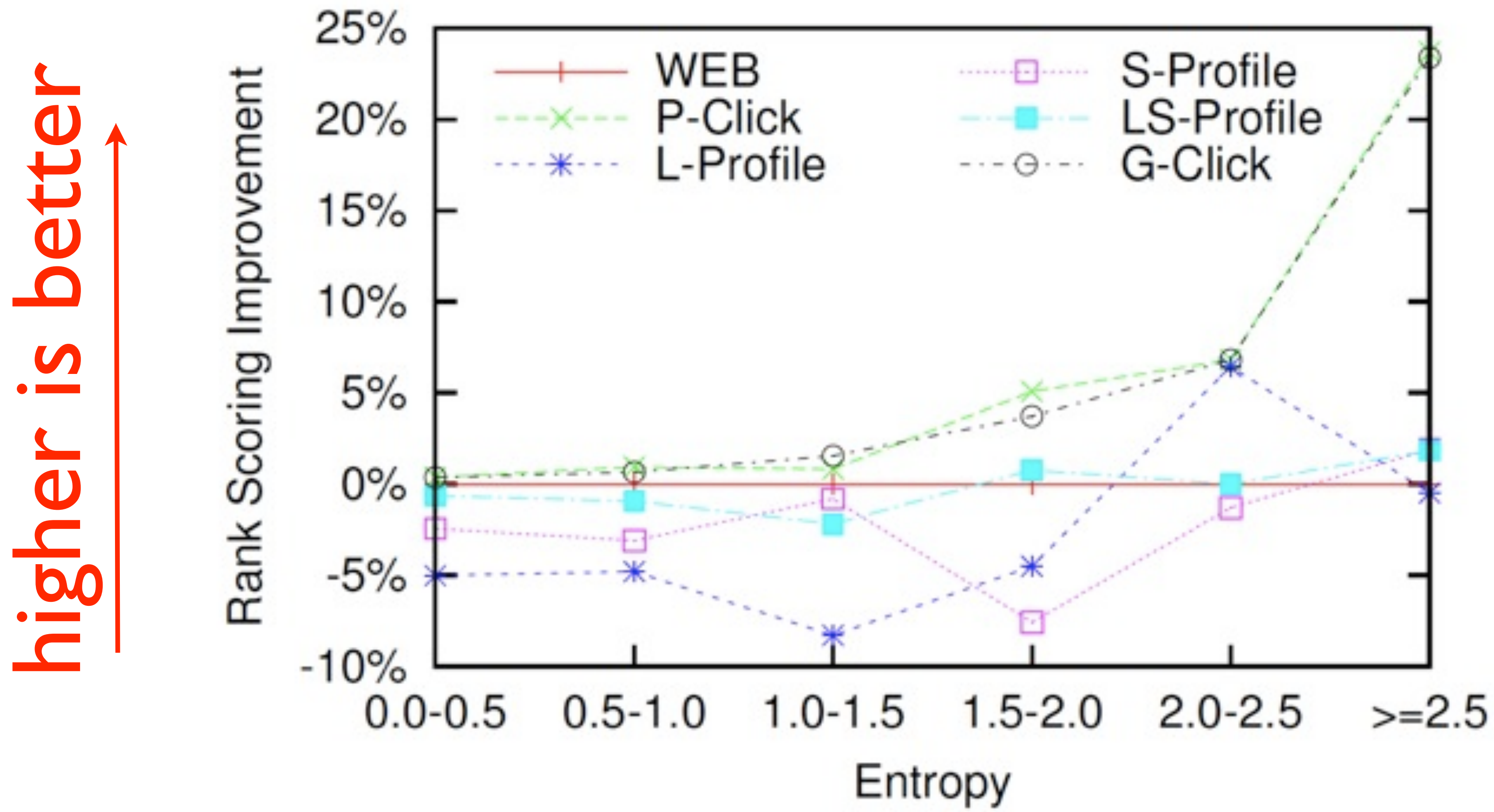
Query personalization



- MSN query log: about 70% of the queries with very low entropy (0-0.5)
  - clicks, in this case, were almost all referred to the same page
- In terms of personalization, this means that in, almost, 70% of the cases personalization does not help

Z. Dou, R. Song, and J. Wen, **“A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization



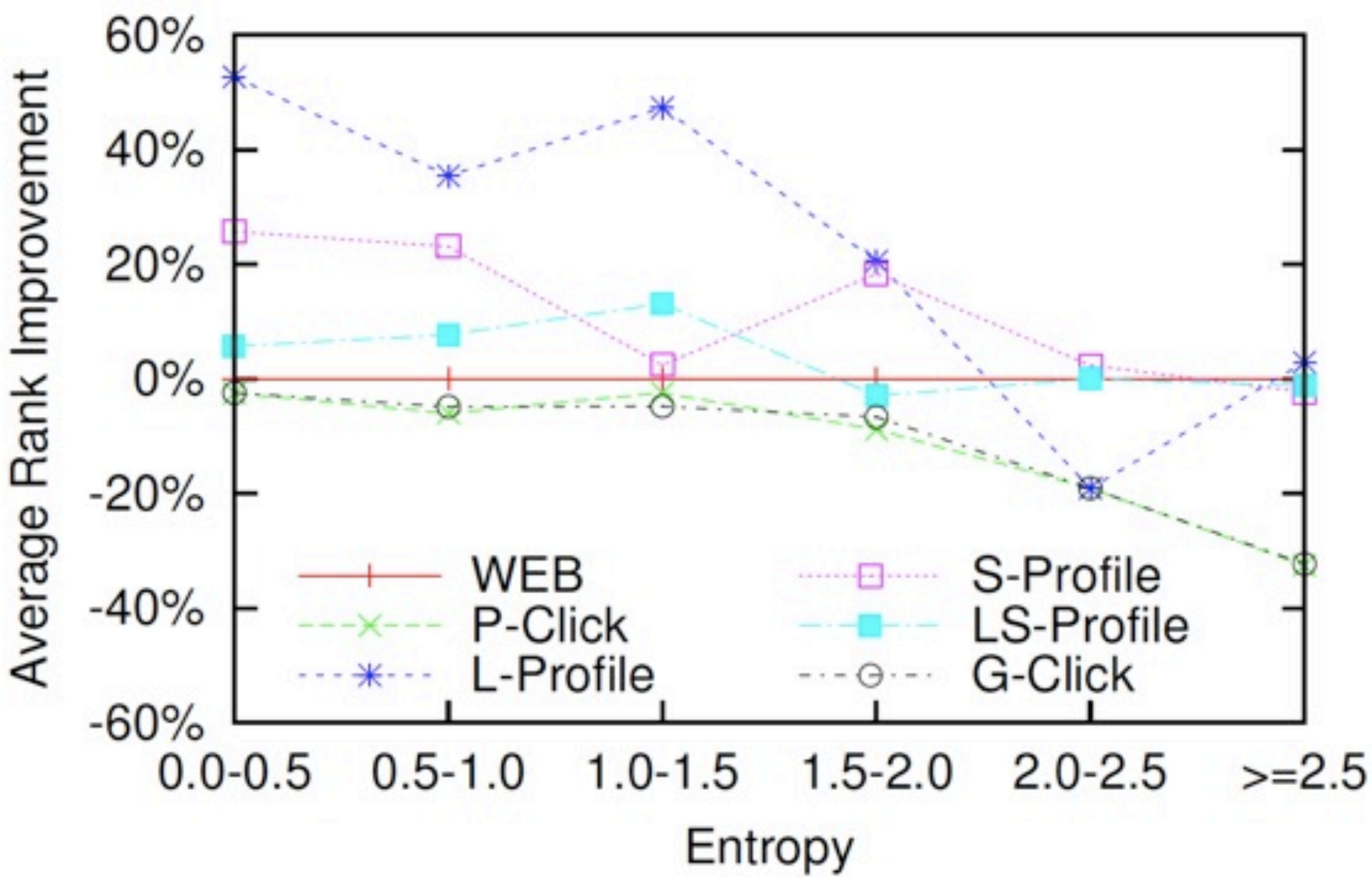
- Both Ranking scoring, and Average Rank perform better at higher entropy levels
    - P-Click and G-Click resulted the best ones

Roughly speaking, this means that whenever accuracy improvement is needed (on high variance query results) personalization is of great help

Z. Dou, R. Song, and J. Wen, **“A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization

lower is better



- Both Ranking scoring, and Average Rank perform better at higher entropy levels
  - P-Click and G-Click resulted the best ones
- Roughly speaking, this means that whenever accuracy improvement is needed (on high variance query results) personalization is of great help

Z. Dou, R. Song, and J. Wen, **"A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Query personalization

- The click-based methods sensibly outperformed the profile-based ones
  - This seems to be in contrast with the results shown in literature so far
- *Dou et al.* [68] state that this might have been due to a "rough implementation" of their system.
- Actually, a deeper analysis have shown that profile based strategies, especially the L-Profile, suffer of the inability to adapt to changes in users' information needs

Z. Dou, R. Song, and J. Wen, **"A large-scale evaluation and analysis of personalized search strategies"**, in WWW2007, pp. 572-581, 2007

Learning to Rank

- Unlike personalized ranking, the aim of "learning to rank" techniques is
  - to compute a global model (i.e. a function that is independent of the specific user) to assign relevance scores to each result page
- Basically, it works as follows
  - first select the best features to be used to identify the importance of a page
  - then train a machine learning algorithm (a classifier/predictor) using these features on a ranked subset (i.e., the training set corpus) of the web pages in order to learn a model/function
- In this tutorial, we shall not consider the "learning to rank" methods that do not make use of query logs

T. Joachims, H. Li, T.-Y. Liu, and C. Zhai, **"Learning to rank for information retrieval (lr4ir 2007)"**, SIGIR

Learning to Rank

- In traditional IR experiments, ranking precision has been measured with the help of a popular benchmark
  - The TREC collection, and the human-based relevance judgements
- The ranking precision of a web search engine, instead, is very difficult to evaluate.
  - Basically, in shortage of humans devoted to evaluate the quality of results for queries, click-through information must be used to infer relevance information
  - If a document receives a click it is relevant for the query it has answered.

Learning to Rank

- Click-through information can thus be used to infer relevance information: *if a document receives a click it is relevant for the query it has answered*
- If f is a ranking function, we can define its performance as the average rank of the clicked results (lower is better)

$$\mathrm{Perf}(f) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{|D_i|}\sum_{j=1}^{|D_i|}\mathrm{rank}(f, Q_i, D_{ij})$$

- Example
  - for query q_1 the user clicks on the 1st, 2nd, and 4th results
  - for query q_2 the user clicks on the 2nd and 4th results

$$\mathrm{Perf}(f) = \tfrac{1}{2}\left(\tfrac{7}{3} + \tfrac{6}{2}\right) = 2.67$$

Learning to Rank

- Click-through information can thus be used to infer relevance information: *if a document receives a click it is relevant for the query it has answered*
- If f is a ranking function, we can define its performance as the average rank of the clicked results (lower is better)

$$\mathrm{Perf}(f) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{|D_i|}\sum_{j=1}^{|D_i|}\mathrm{rank}(f, Q_i, D_{ij})$$

Queries: $Q_1,\ldots,Q_{|Q|}$

- Example
  - for query q_1 the user clicks on the 1st, 2nd, and 4th results
  - for query q_2 the user clicks on the 2nd and 4th results

$$\mathrm{Perf}(f) = \frac{1}{2}\left(\frac{7}{3}+\frac{6}{2}\right) = 2.67$$

Learning to Rank

- Click-through information can thus be used to infer relevance information: *if a document receives a click it is relevant for the query it has answered*
- If f is a ranking function, we can define its performance as the average rank of the clicked results (lower is better)

$$\mathrm{Perf}(f) = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{|D_i|}\sum_{j=1}^{|D_i|}\mathrm{rank}(f, Q_i, D_{ij})$$

D_{ij} : j-th document clicked in answer to query Q_i

Queries: $Q_1,\ldots,Q_{|Q|}$

- Example
  - for query q_1 the user clicks on the 1st, 2nd, and 4th results
  - for query q_2 the user clicks on the 2nd and 4th results

$$\mathrm{Perf}(f) = \frac{1}{2}\left(\frac{7}{3}+\frac{6}{2}\right) = 2.67$$

Learning to Rank

- *Joachims et al.* observed that a click on a result is not an unbiased estimator for its importance
  - The fact that users click on the first result more than on the others seem to be related with a trust feeling with the search engine ranking
  - The search engine ranking influences the user, so that the click-through data does not suffice to conclude that it's an implicit feedback
- Is it possible to find a set of query log features that could give an unbiased estimate of user's perceived relevance for a web page?

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, **"Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search"** ,ACM Trans. Inf. Syst., vol. 25, no. 2, p. 7, 2007

Learning to Rank

- *Joachims et al.* observed that users scan a result page from top to bottom, and thus
  - *if a user clicks on the i-th result of a query, s/he considered it more important than the previous ones*
- Starting from the previous key observation, *Joachims et al.* proposes a series of strategies to extract *implicit relevance feedback from click-through data*
- Running example
  - Let q be a query returning the ordered pages p_1 to p_7
  - The asterisked symbols represent the clicked pages:

$$p_1^*, p_2^*, p_3, p_4^*, p_5, p_6, p_7^*$$

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, **"Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search"** , ACM Trans. Inf. Syst., vol. 25, no. 2, p. 7, 2007

Learning to Rank

- One of the proposed strategies: *Click > Skip Above*
  - For each clicked-on page p_i, extract the *preference examples* (features), denoted as
    $$rel(p_i) > rel(p_j), \quad i > j,$$
    where p_j was not clicked-on
  - *rel(.)* is the function measuring the relevance of a page
  - Examples of features extracted from
    $$p_1^*, p_2^*, p_3, p_4^*, p_5, p_6, p_7^*$$

$rel(p4) > rel(p3)$, $rel(p7) > rel(p5)$,
$rel(p7) > rel(p3)$, $rel(p7) > rel(p6)$

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, **"Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search"** , ACM Trans. Inf. Syst., vol. 25, no. 2, p. 7, 2007

Learning to Rank

- Other proposed strategies:
    - *Last Click > Skip Above*
    - *Click > Earlier Click*
    - *Click > Skip Previous*
    - *Click > No-Click Next*

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, **"Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search"** , ACM Trans. Inf. Syst., vol. 25, no. 2, p. 7, 2007

Learning to Rank

- Evaluation
  - How accurate is this implicit feedback compared to the explicit feedback?
  - The authors compared the pairwise preferences generated from the clicks to the explicit relevance judgments (a user study has been used)
  - The Table shows the percentage of times the preferences generated from clicks agree with the direction of a strict preference of judge

Strategy	Features per Query	Normal (%)	Swapped (%)
Inter-Judge Agreement	N/A	89.5	N/A
Click > Skip Above	1.37	88.0 ± 9.5	79.6 ± 8.9
Last Click > Skip Above	1.18	89.7 ± 9.8	77.9 ± 9.9
Click > Earlier Click	0.20	75.0 ± 25.8	36.8 ± 22.9
Click > Skip Previous	0.37	88.9 ± 24.1	80.0 ± 18.00
Click > No Click Next	0.68	75.6 ± 14.1	66.7 ± 13.1

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, **“Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search"** , ACM Trans. Inf. Syst., vol. 25, no. 2, p. 7, 2007

Learning to Rank

- Query chains
  - Users usually do not issue just a single query: whenever they look for a specific information, they tend to issue more than a single query
  - Query Chains can be exploited to infer implicit relevance feedback on document clicks in sequences of user queries
- Running example
  - A query chain of 4 queries
  - Thus, 4 result sets, some of them clicked-on by the user (asterisked)

$$q_1: \; p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}$$
$$q_2: \; p^*_{21}, p_{22}, p^*_{23}, p_{24}, p^*_{25}, p_{26}, p_{27}$$
$$q_3: \; p_{31}, p^*_{32}, p_{33}, p_{34}, p_{35}, p_{26}, p_{37}$$
$$q_4: \; p^*_{41}, p_{42}, p_{43}, p_{44}, p_{45}, p_{36}, p_{47}$$

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, **“Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search"** , ACM Trans. Inf. Syst., vol. 25, no. 2, p. 7, 2007

Learning to Rank

- One example of the proposed strategy *Click > Skip Earlier QC*
  - Extension of "Click > Skip Above" to multiple result sets
  - A preference is generated between two pages from different result sets within the same query chain, if a page in an earlier result set was skipped and a page in a later result set was instead clicked
  - Examples of features extracted from:

$$\cancel{p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}}$$
$$p^*_{21}, p_{22}, p^*_{23}, p_{24}, p^*_{25}, \cancel{p_{26}, p_{27}}$$
$$p_{31}, p^*_{32}, \cancel{p_{33}, p_{34}, p_{35}, p_{26}, p_{37}}$$
$$p^*_{41}, \cancel{p_{42}, p_{43}, p_{44}, p_{45}, p_{36}, p_{47}}$$

  - *rel(p32) > rel(p22),* *rel(p32) > rel(p24),*
    rel(p41) > rel(p22), *rel(p41) > rel(p24),*
    rel(p41) > rel(p31)

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, **"Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search"** ,ACM Trans. Inf. Syst., vol. 25, no. 2, p. 7, 2007

Learning to Rank

Strategy	Features per Query	Normal (%)	Swapped (%)
Click > Skip Earlier QC	0.49	84.5 ± 16.4	71.7 ± 17.0
Last Click > Skip Earlier QC	0.33	77.3 ± 20.6	80.8 ± 20.2
Click > Click Earlier QC	0.30	61.9 ± 23.5	51.2 ± 17.1
Click > TopOne NoClickEarlier QC	0.35	86.4 ± 21.2	77.3 ± 15.1
Click > TopTwo NoClickEarlier QC	0.70	88.9 ± 12.9	80.0 ± 10.1
TopOne > TopOne Earlier QC	0.84	65.3 ± 15.2	68.2 ± 12.8

- As in the non-QC methods, the Table shows the accuracy of the methods proposed concerning the Query Chains
  - comparing the pairwise preferences generated from the clicks to the explicit relevance judgments
- Strategy *Click > TopTwo NoClickEarlier QC* produces the best results

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, **"Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search"** , ACM Trans. Inf. Syst., vol. 25, no. 2, p. 7, 2007

Learning to Rank

- For a query q and a document collection $D=\{d1,\dots, dm\}$, the optimal retrieval system aims at returning a ranking r^* that orders the documents in D according to their relevance to the query
- An IR system returns an ordering $r_{f(q)}$ that is obtained by sorting documents in D according to scores computed by a function f over the query q, i.e. $f(q)$
- Formally, both r^*, and $r_{f(q)}$ are weak ordering binary relations over $D \times D$
  - A relation $r \subseteq D \times D$ is defined as $r = \{ (d_i, d_j) \text{ s.t. } d_i <_r d_j \}$
- To optimize $f(q)$ in order to produce a ranking as close as possible to the optimal one r^*, we need to define the similarity between the two orderings: r^* and $r_{f(q)}$

T. Joachims, **"Optimizing search engines using clickthrough data"**, in KDD '02, pp. 133-142, ACM Press, 2002.

Learning to Rank

- To define the similarity between the two orderings: r^* and $r_{f(q)}$
  - One of the most used metric to measure similarity between two ranked lists is the *Kendall's distance metric* τ.
  - It counts the number *P* of concordant pairs, and *Q* of discordant pairs on r^* and $r_{f(q)}$
  - If $|D|=m$: $\tau(r_a, r_b) = \frac{P-Q}{P+Q} = 1 - \frac{2Q}{\binom{m}{2}}$
  - Maximizing $\tau(r^*, r_{f(q)})$ is equivalent to minimize the average rank of relevant documents

T. Joachims, **"Optimizing search engines using clickthrough data"**, in KDD '02, pp. 133-142, ACM Press, 2002.
M. Kendall, **"Rank Correlation Methods."** Hafner, 1955

Learning to Rank

- To define the similarity between the two orderings: r^* and $r_{f(q)}$
  - One of the most used metric to measure similarity between two ranked lists is the *Kendall's distance metric* τ.
  - It counts the number *P* of concordant pairs, and *Q* of discordant pairs on r^* and $r_{f(q)}$
  - If $|D|=m$: $\tau(r_a, r_b) = \frac{P-Q}{P+Q} = 1 - \frac{2Q}{\binom{m}{2}}$

    $\tau=1$ if complete concordance, $\tau<1$ otherwise
  - Maximizing $\tau(r^*, r_{f(q)})$ is equivalent to minimize the average rank of relevant documents

T. Joachims, **"Optimizing search engines using clickthrough data"**, in KDD '02, pp. 133-142, ACM Press, 2002.
M. Kendall, **"Rank Correlation Methods."** Hafner, 1955

Learning to Rank

- An example of *Kendall's distance metric* τ.

$$\tau(r_a, r_b) = \frac{P - Q}{P + Q} = 1 - \frac{2Q}{\binom{m}{2}}$$

- Consider the two rankings:

$$d_1 <_{r_a} d_2 <_{r_a} d_3 <_{r_a} d_4 <_{r_a} d_5$$

$$d_3 <_{r_b} d_2 <_{r_b} d_1 <_{r_b} d_4 <_{r_b} d_5$$

  - The number *Q* of discordant pairs is 3:
    $\{d_2, d_3\}$, $\{d_1, d_2\}$, $\{d_1, d_3\}$
    while all remaining P=7 pairs are concordant
  - Therefore: $\tau(r_a, r_b) = 0.4$

T. Joachims, **"Optimizing search engines using clickthrough data"**, in KDD '02, pp. 133-142, ACM Press, 2002.
M. Kendall, **"Rank Correlation Methods."** Hafner, 1955

Learning to Rank

- A simplifies learning algorithm in information retrieval could exploit a binary classification
  - Only classes "relevant" and "non-relevant"
- Due to strong majority of "non-relevant" documents, good predictive accuracy can be achieved by always classifying pages as "non-relevant"
- *Joachims* proposed the *RankSVM* algorithm, that takes an empirical risk minimization approach
  - Given an independently and identically distributed training sample S of size n containing queries q_i with their target rankings r_i*

$$(q_1, r_1^*), (q_2, r_2^*), \ldots, (q_n, r_n^*)$$

T. Joachims, **"Optimizing search engines using clickthrough data"**, in KDD '02, pp. 133-142, ACM Press, 2002.

Learning to Rank

- The learner $\mathcal{L}$ will select a ranking function f from a family of ranking functions F that maximizes the empirical r_i* of the training sample

$$\tau_S(\mathrm{f}) = \frac{1}{n}\sum_{i=1}^{n}\tau(\mathrm{r}_{\mathrm{f}(\mathrm{q}_i)}, \mathrm{r}_i^*)$$

- This setup is analogous to classification by minimizing training error
  - the target is not a class label, but a binary ordering relation

T. Joachims, **"Optimizing search engines using clickthrough data"**, in KDD '02, pp. 133-142, ACM Press, 2002.

Learning to Rank

- Evaluation carried out with a user study
  - Made on training data from the Cornell University Library's search engine
  - 32% of people preferred the *rankSVM* trained over QC (Query Chain)
  - 20% of people preferring the *non-rankSVM* version of ranking
  - 48% of people remained indifferent

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay, **“Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search"** , ACM Trans. Inf. Syst., vol. 25, no. 2, p. 7, 2007
T. Joachims, **“Optimizing search engines using clickthrough data”**, in KDD '02, pp. 133-142, ACM Press, 2002.

Learning to Rank

- Many other approaches in the literature:
  - *RankNet*, for instance, proposed by *Burges et al.*, is said to be used by the Microsoft's Live search engine, and adopts a neural network approach
  - Several other approaches have been proposed during these last years: *RankBoost*, *GBRank*, *LambdaRank*, *NetRank*

C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, **"Learning to rank using gradient descent"**, in ICML '05, pp. 89-96, ACM, 2005

Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer, "An effcient boosting algorithm for combining preferences," J. Mach. Learn. Res., vol. 4, pp. 933-969, 2003.

Z. Zheng, K. Chen, G. Sun, and H. Zha, **"A regression framework for learning ranking functions using relative relevance judgments"** in SIGIR '07, pp. 287-294, ACM, 2007.

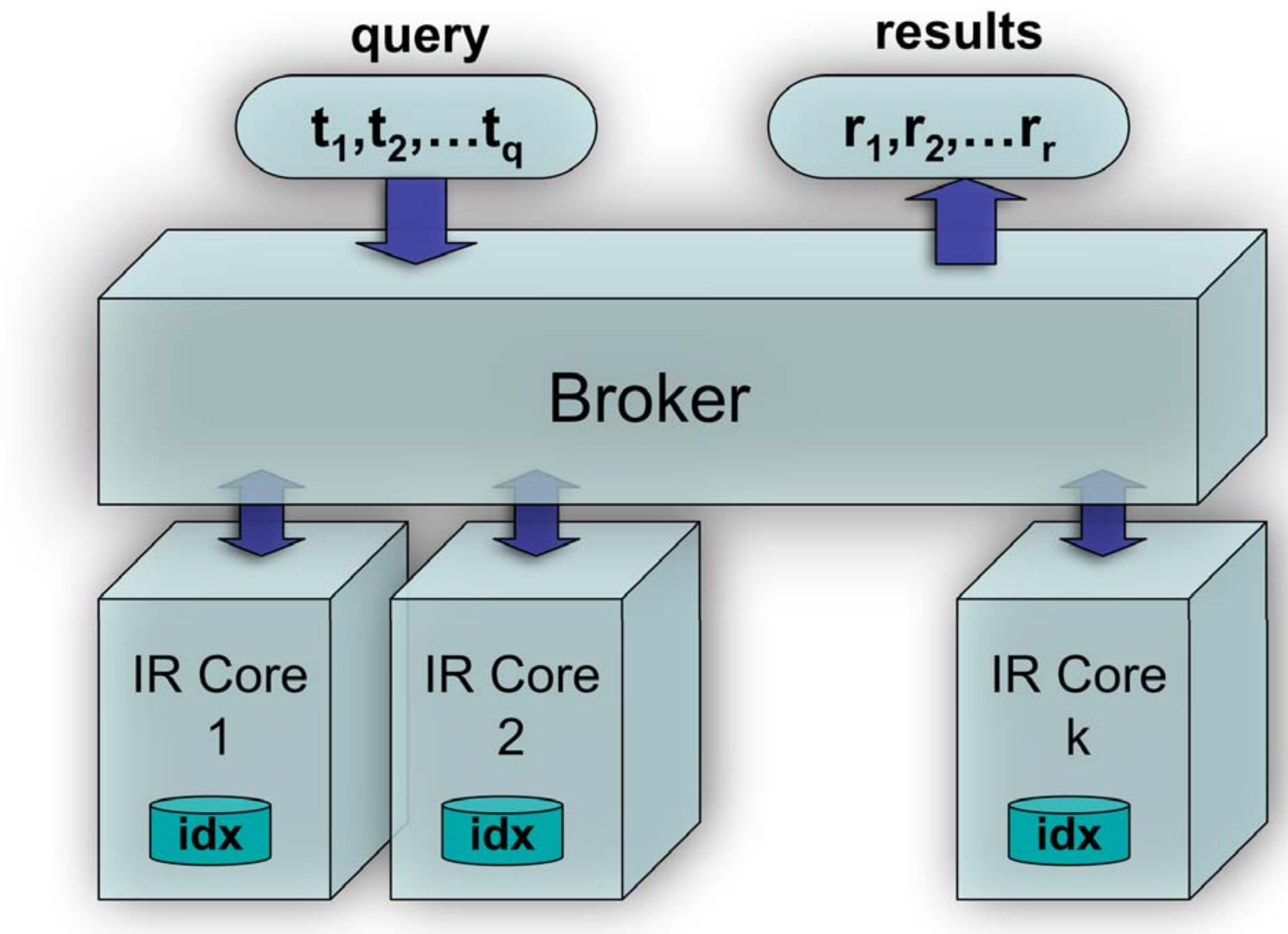
C. J. C. Burges, R. Ragno, and Q.V. Le, **"Learning to rank with nonsmooth cost functions"**, in NIPS, pp. 193-200, MIT Press, 2006.

A. Agarwal and S. Chakrabarti, **"Learning random walks to rank nodes in graphs"**, in ICML '07, pp. 9-16, ACM, 2007.

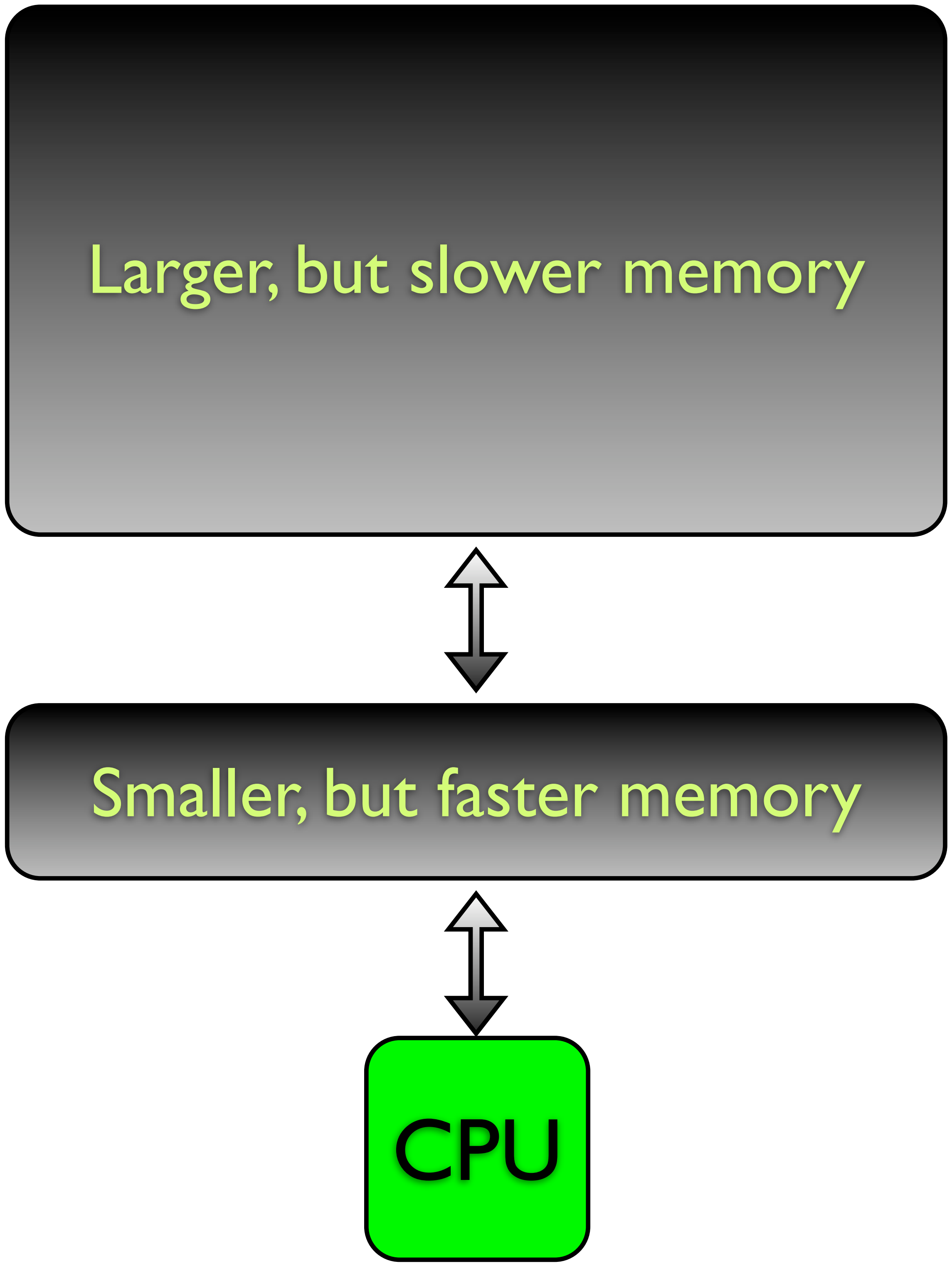
Tutorial Outline

- Query Logs
- Data Mining Techniques for QL Mining
- Enhancing Effectiveness of Search Systems
- Enhancing Efficiency of Search Systems
  - Caching
  - Index Partitioning and Querying in Distributed IR Systems

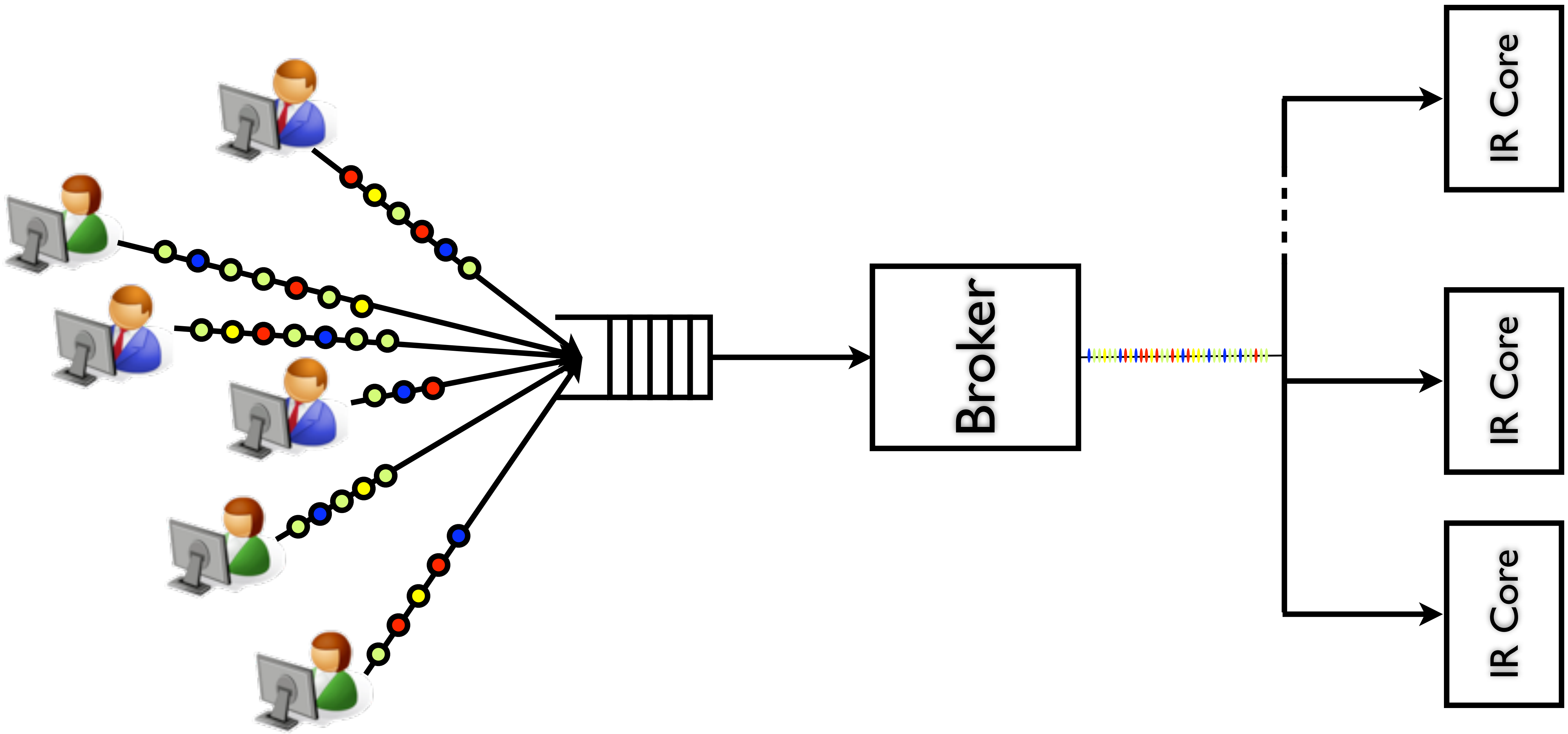
Sketching a Distributed Search Engine



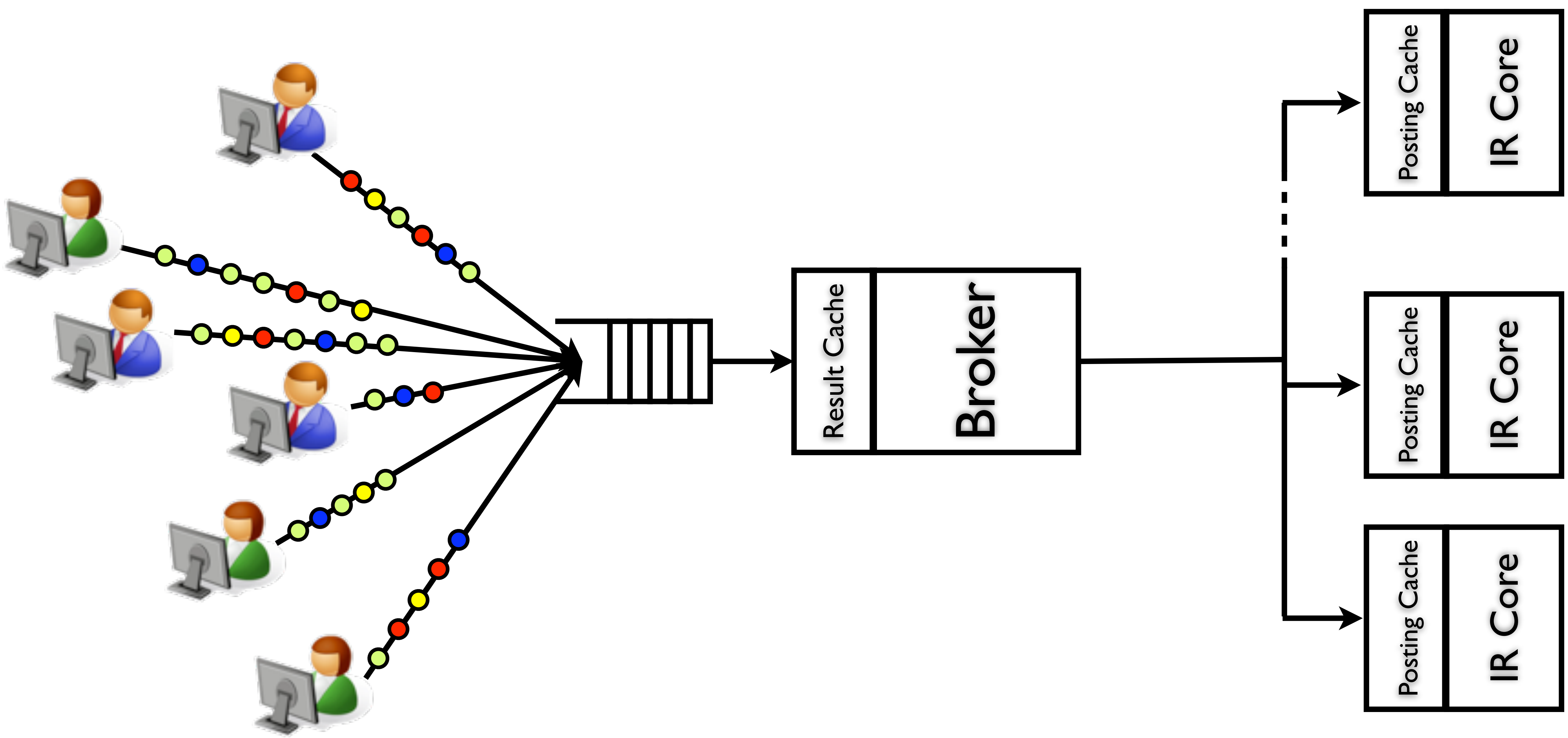
Caching in General



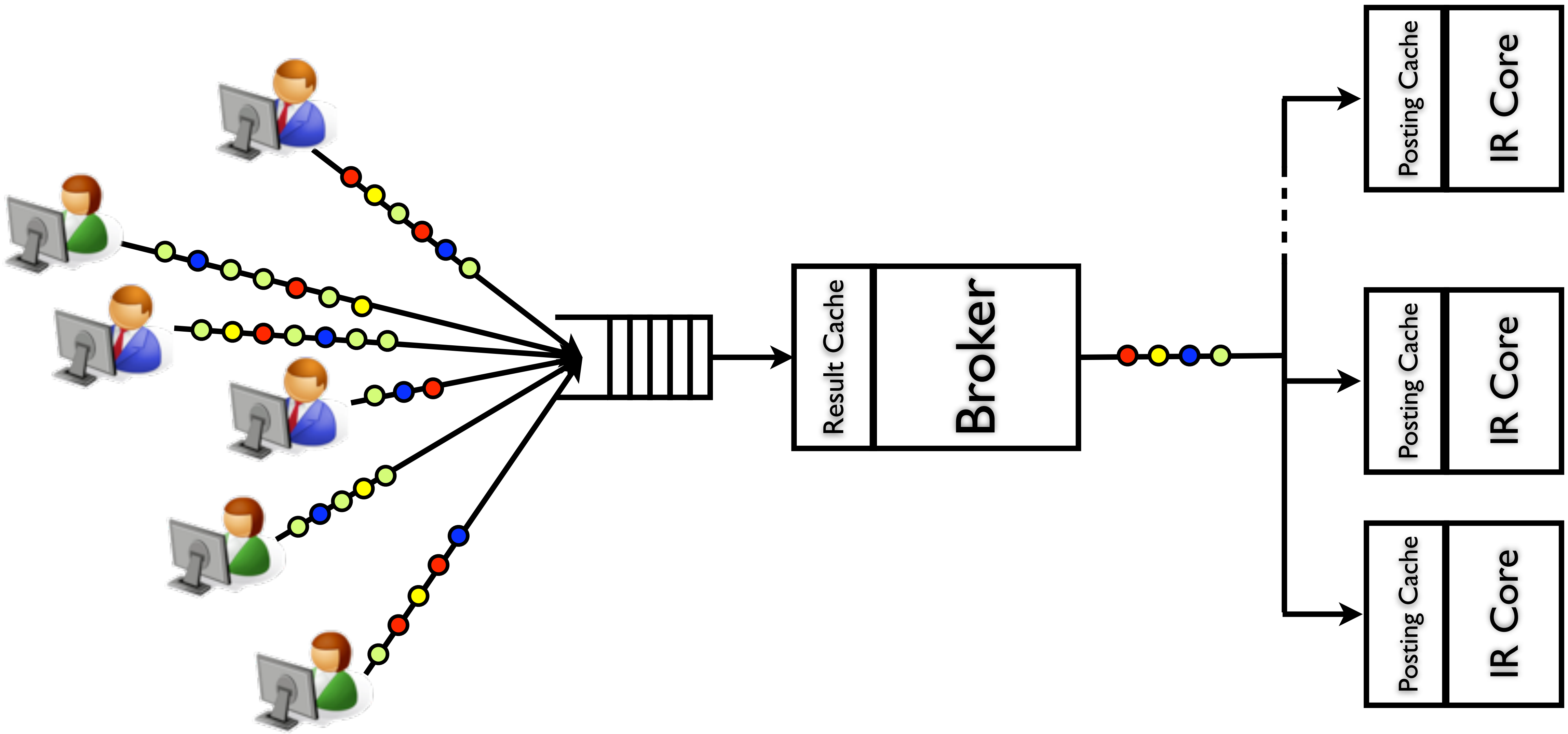
W/O Caching



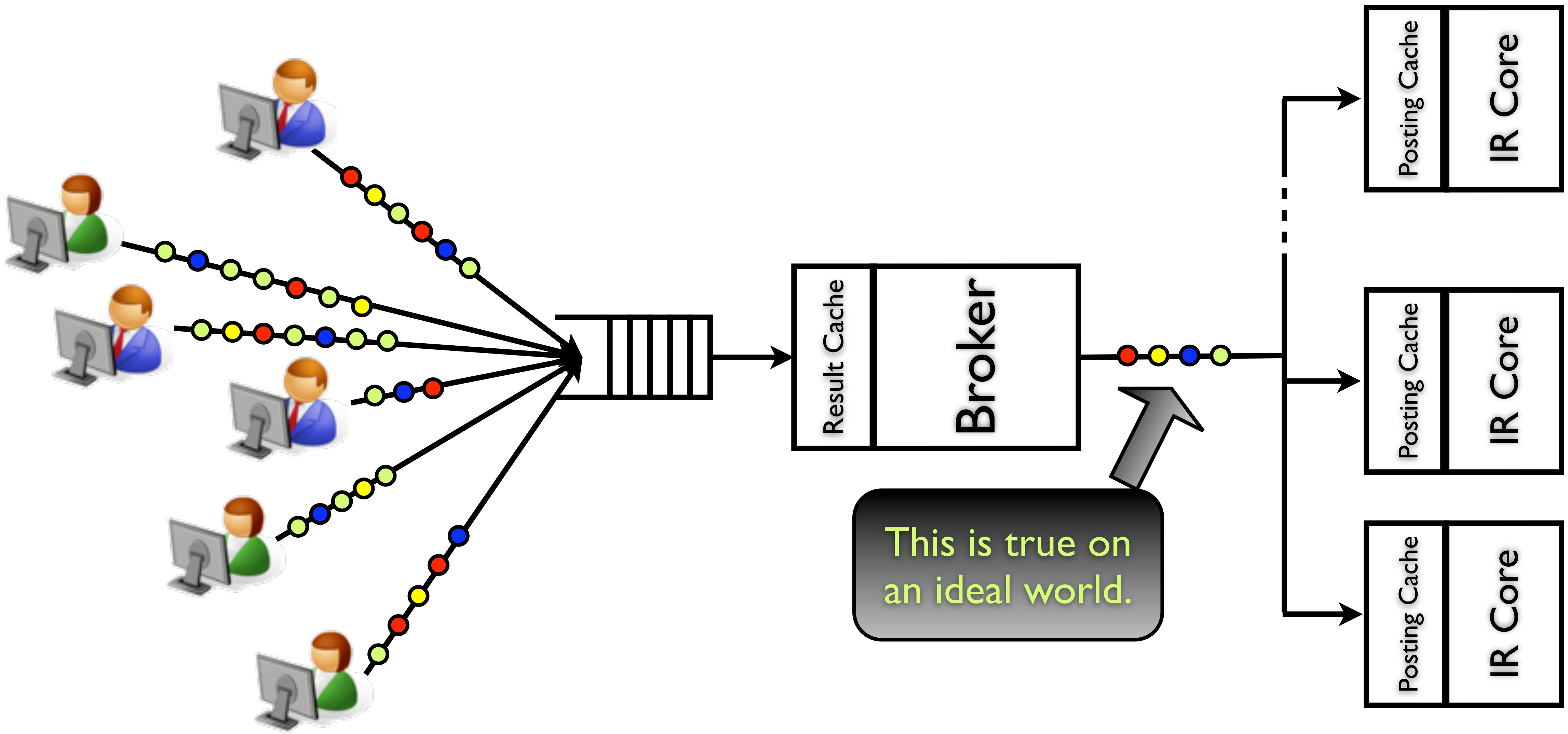
With Caching



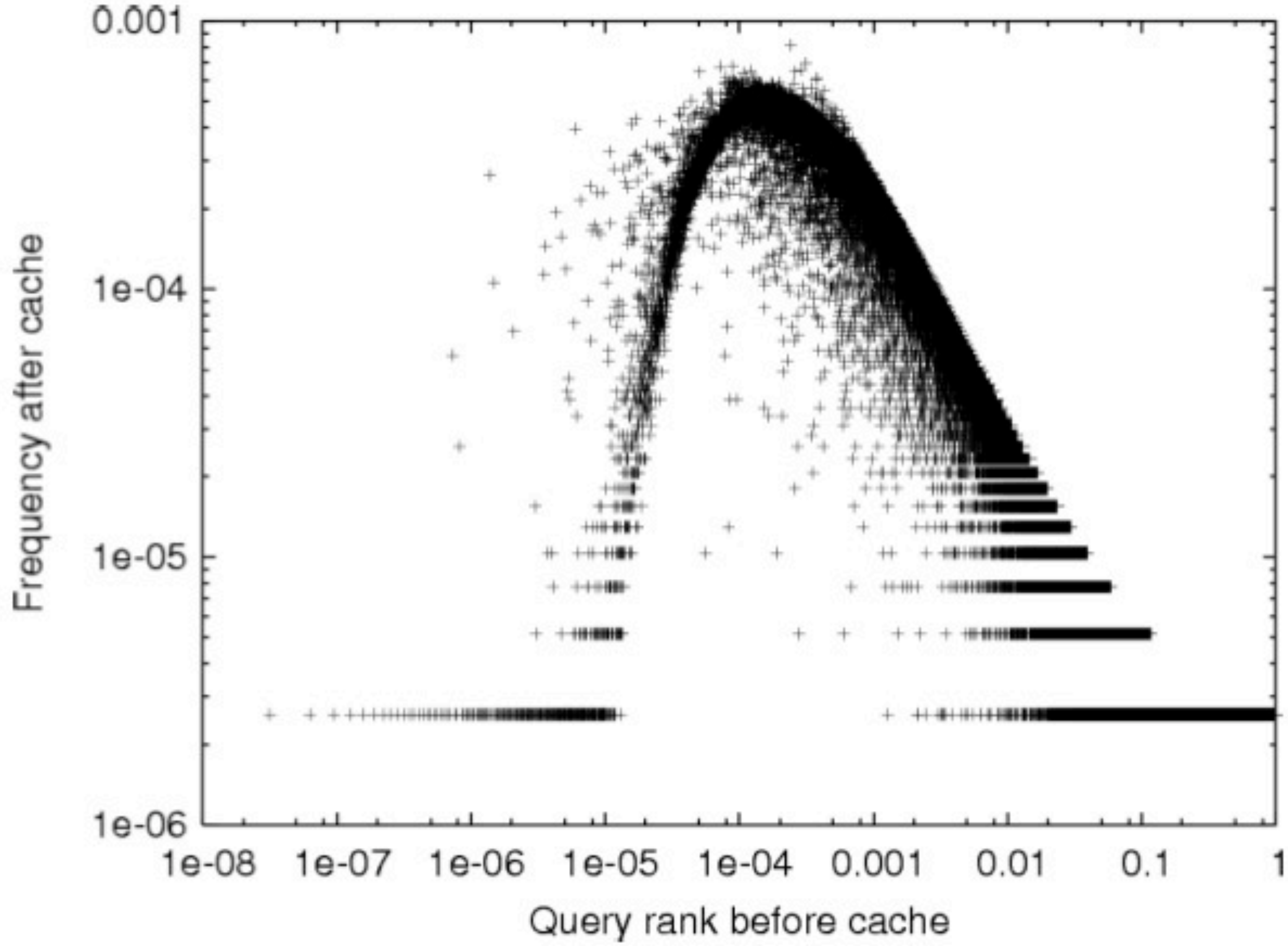
With Caching



With Caching

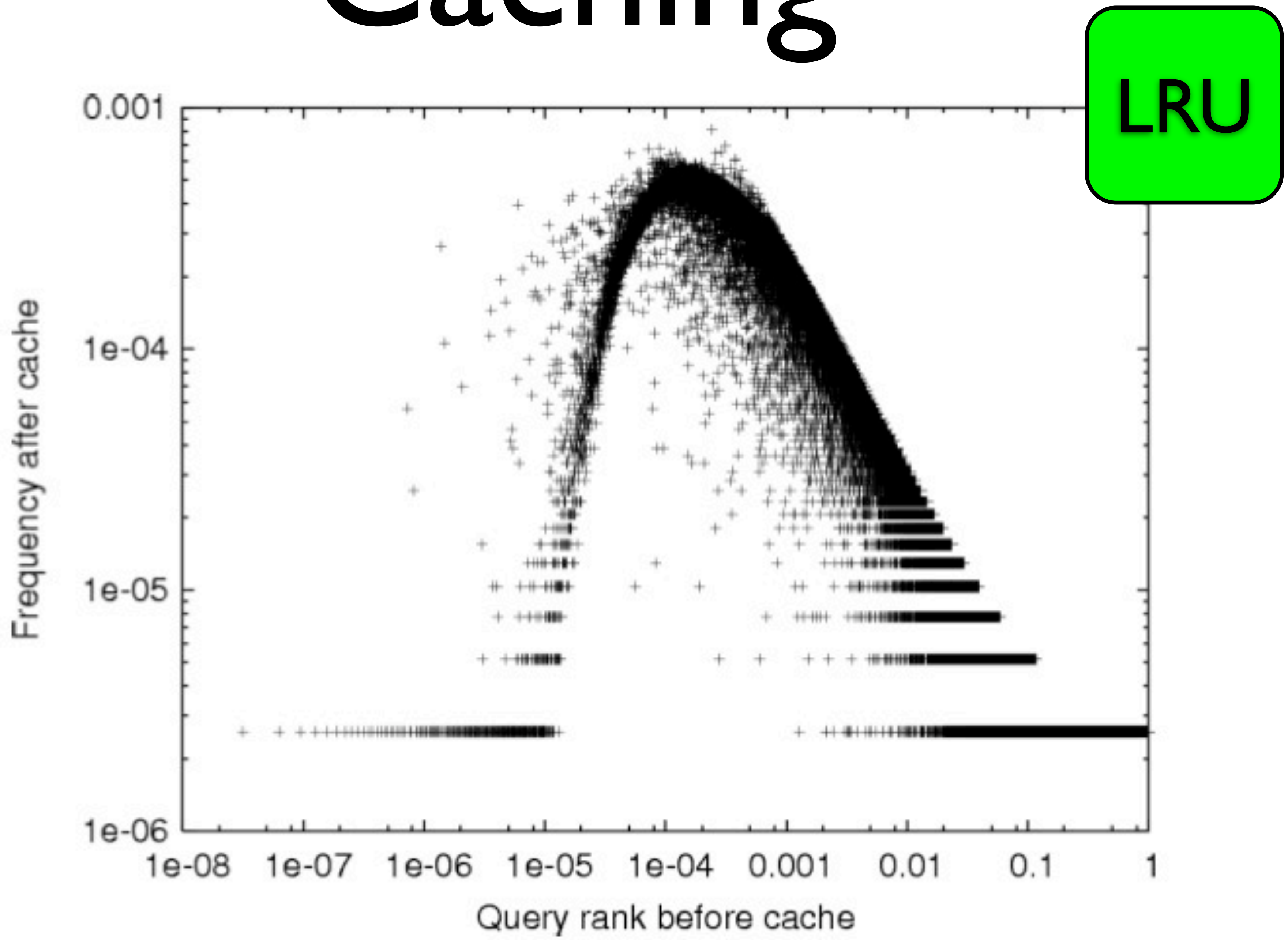


Filtering Effect of Caching



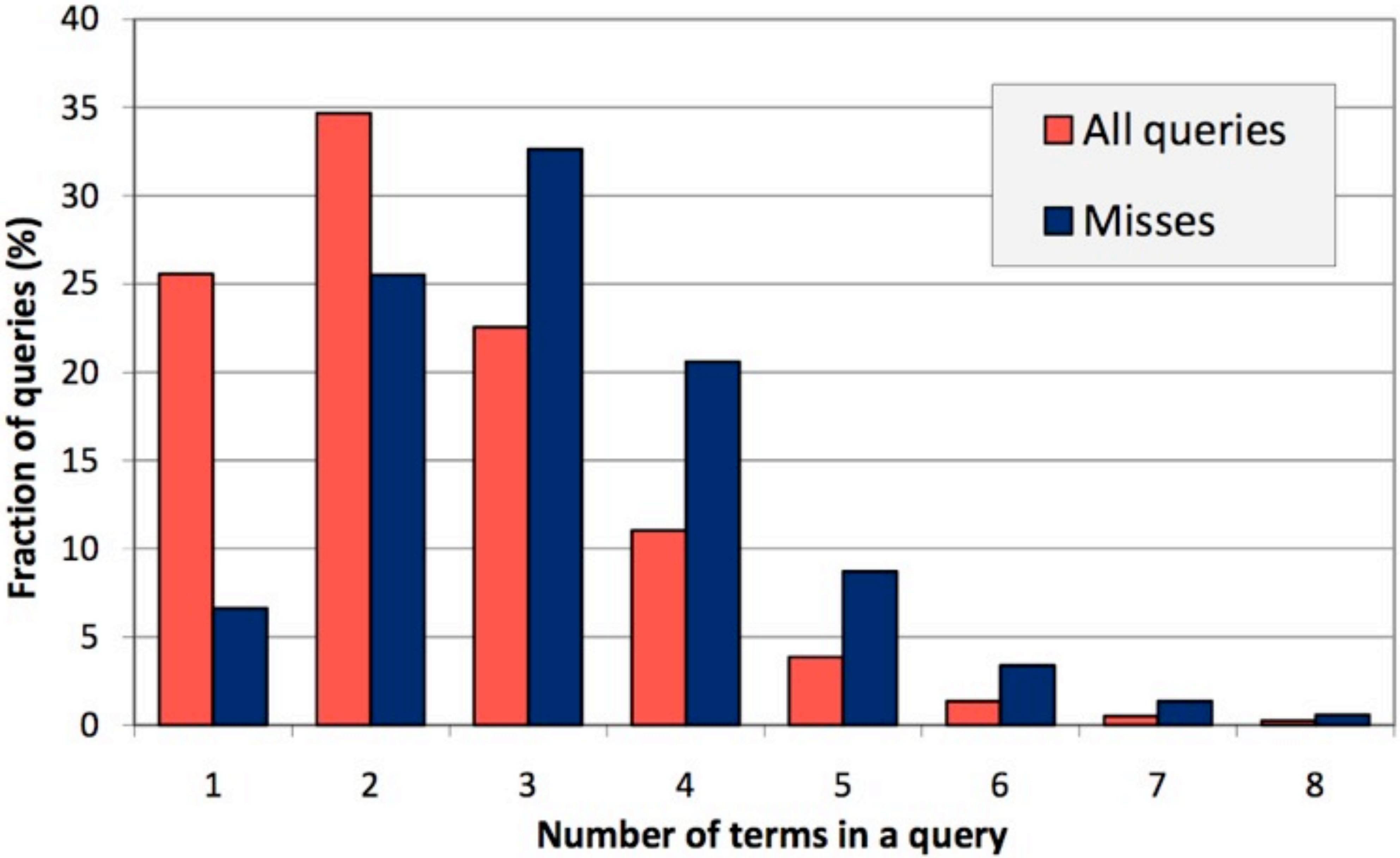
R. Baeza-Yates, A. Gionis, F. P. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri, "**Design trade-offs for search engine caching**," ACM Trans. Web, vol. 2, no. 4, pp. 1–28, 2008.

Filtering Effect of Caching



R. Baeza-Yates, A. Gionis, F. P. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri, "**Design trade-offs for search engine caching**," ACM Trans. Web, vol. 2, no. 4, pp. 1–28, 2008.

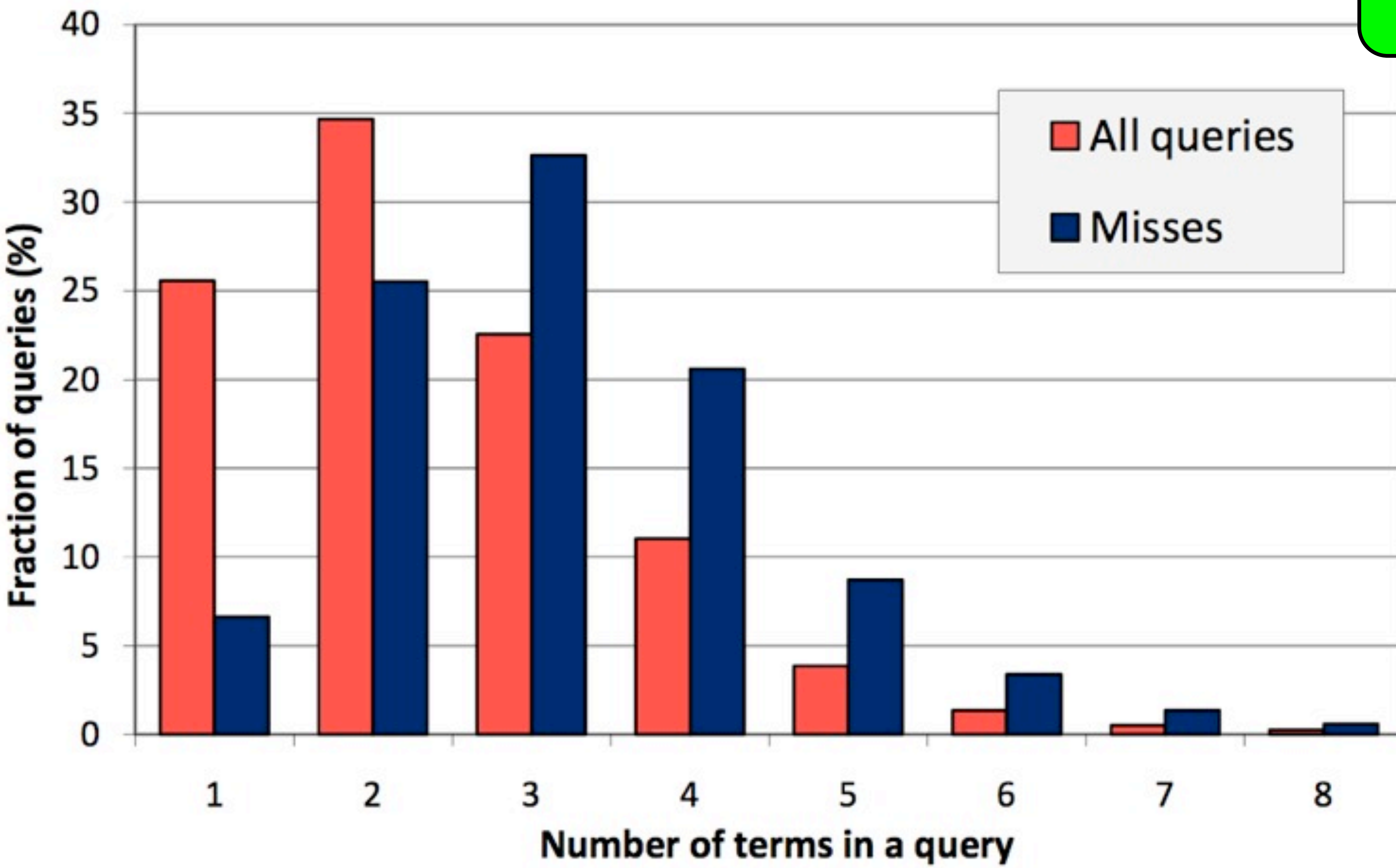
Filtering Effect of Caching



Skobeltsyn, G., Junqueira, F., Plachouras, V., and Baeza-Yates, R., "**ResIn: a combination of results caching and index pruning for high-performance web search engines**," SIGIR 2008, pp 131-138.

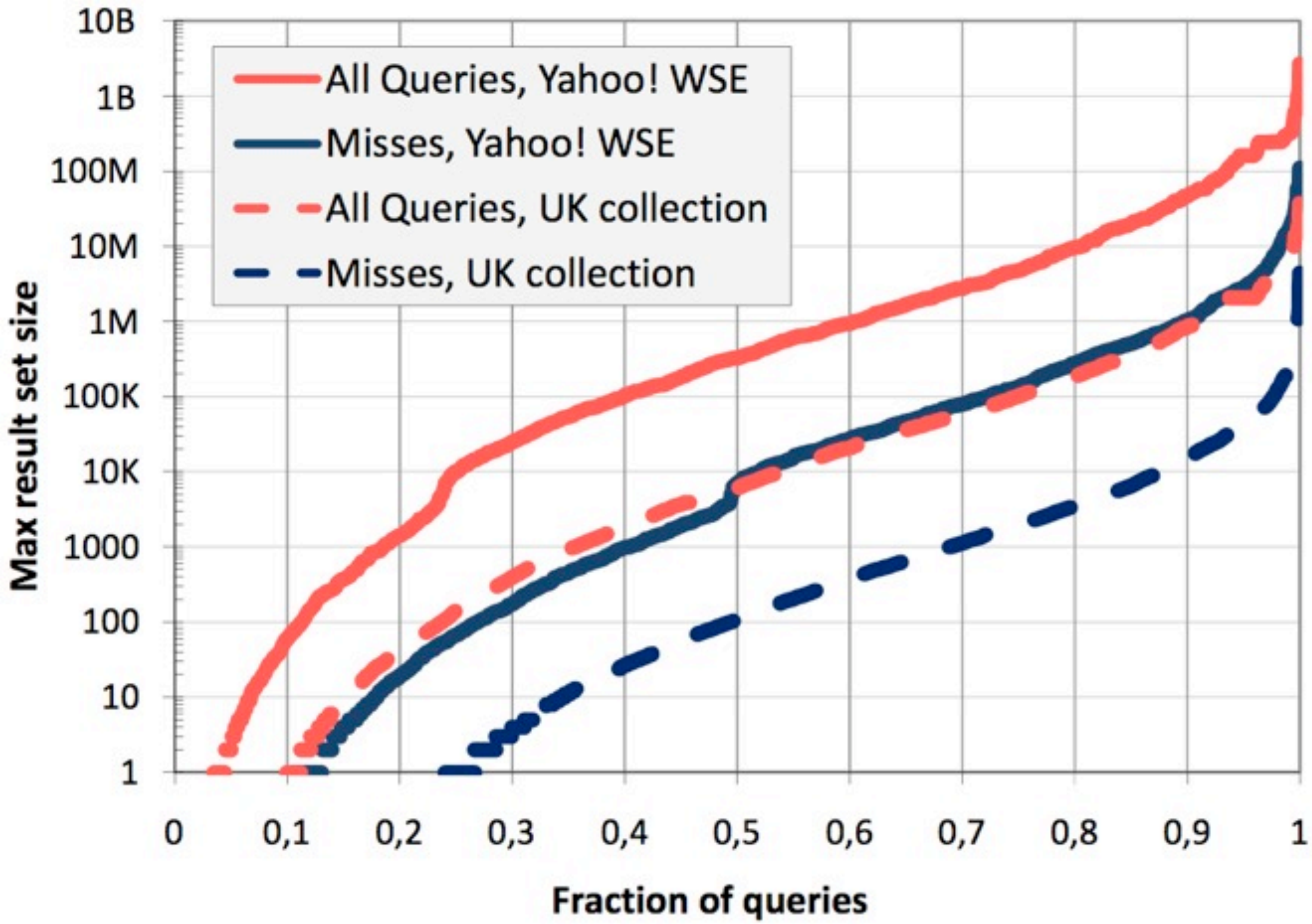
Filtering Effect of Caching

LRU



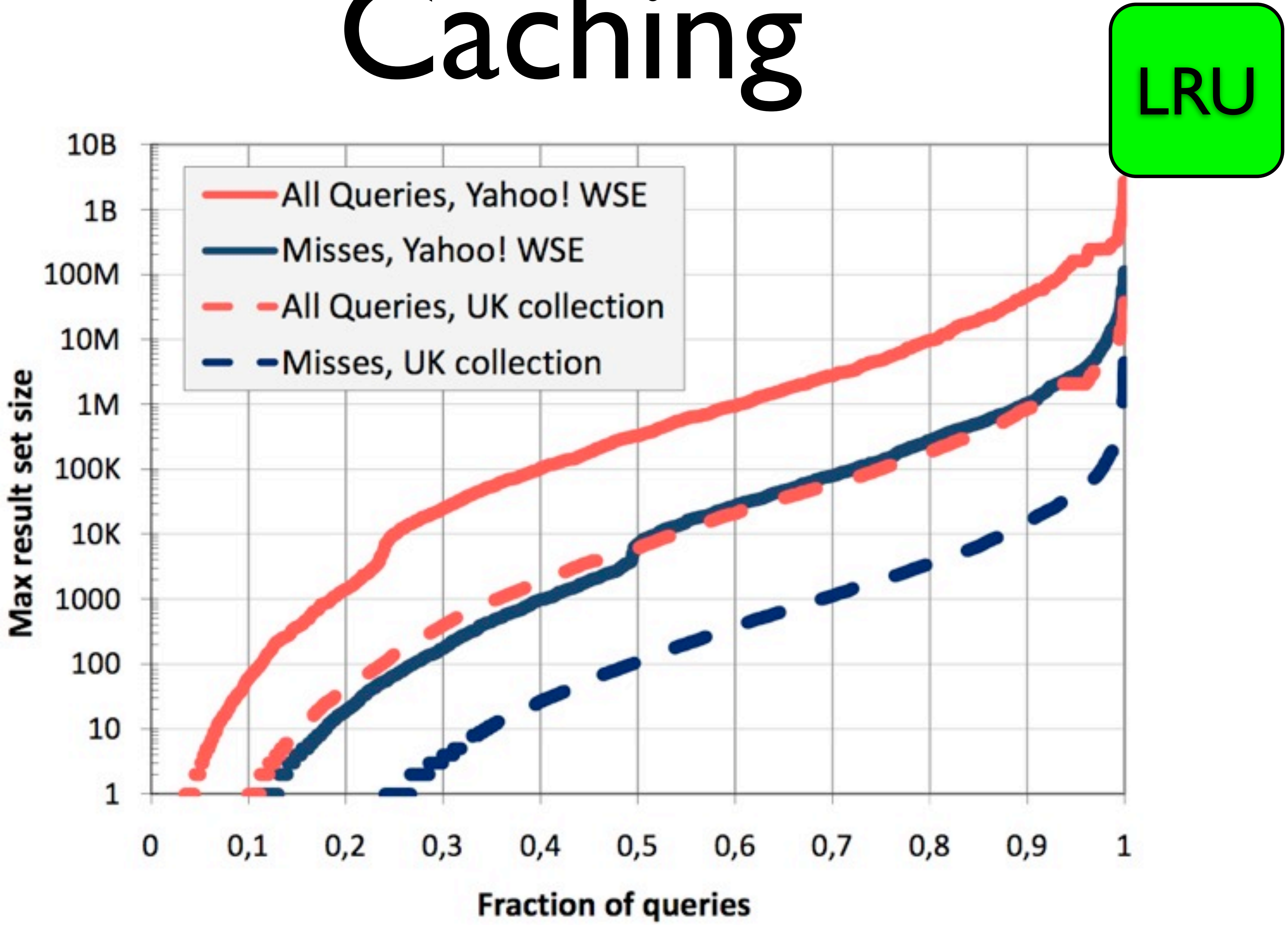
Skobeltsyn, G., Junqueira, F., Plachouras, V., and Baeza-Yates, R., "**ResIn: a combination of results caching and index pruning for high-performance web search engines**," SIGIR 2008, pp 131-138.

Filtering Effect of Caching



Skobeltsyn, G., Junqueira, F., Plachouras, V., and Baeza-Yates, R., "**ResIn: a combination of results caching and index pruning for high-performance web search engines**," SIGIR 2008, pp 131-138.

Filtering Effect of Caching



Skobeltsyn, G., Junqueira, F., Plachouras, V., and Baeza-Yates, R., "**ResIn: a combination of results caching and index pruning for high-performance web search engines**," SIGIR 2008, pp 131-138.

Caching Performance Evaluation

- **Hit-Ratio**: i.e. how many times the cache is useful
- **Query Throughput**: i.e. the number of queries the cache can serve in a second

- But... what really impacts on caching performance?

“Things” to Cache in Search Engines

- Results
  - in answer to a user query
- Posting Lists
  - e.g. for the query “new york” cache the posting lists for term *new* and for term *york*
- Partial queries
  - cache subqueries, e.g. for “new york times” cache only “new york”

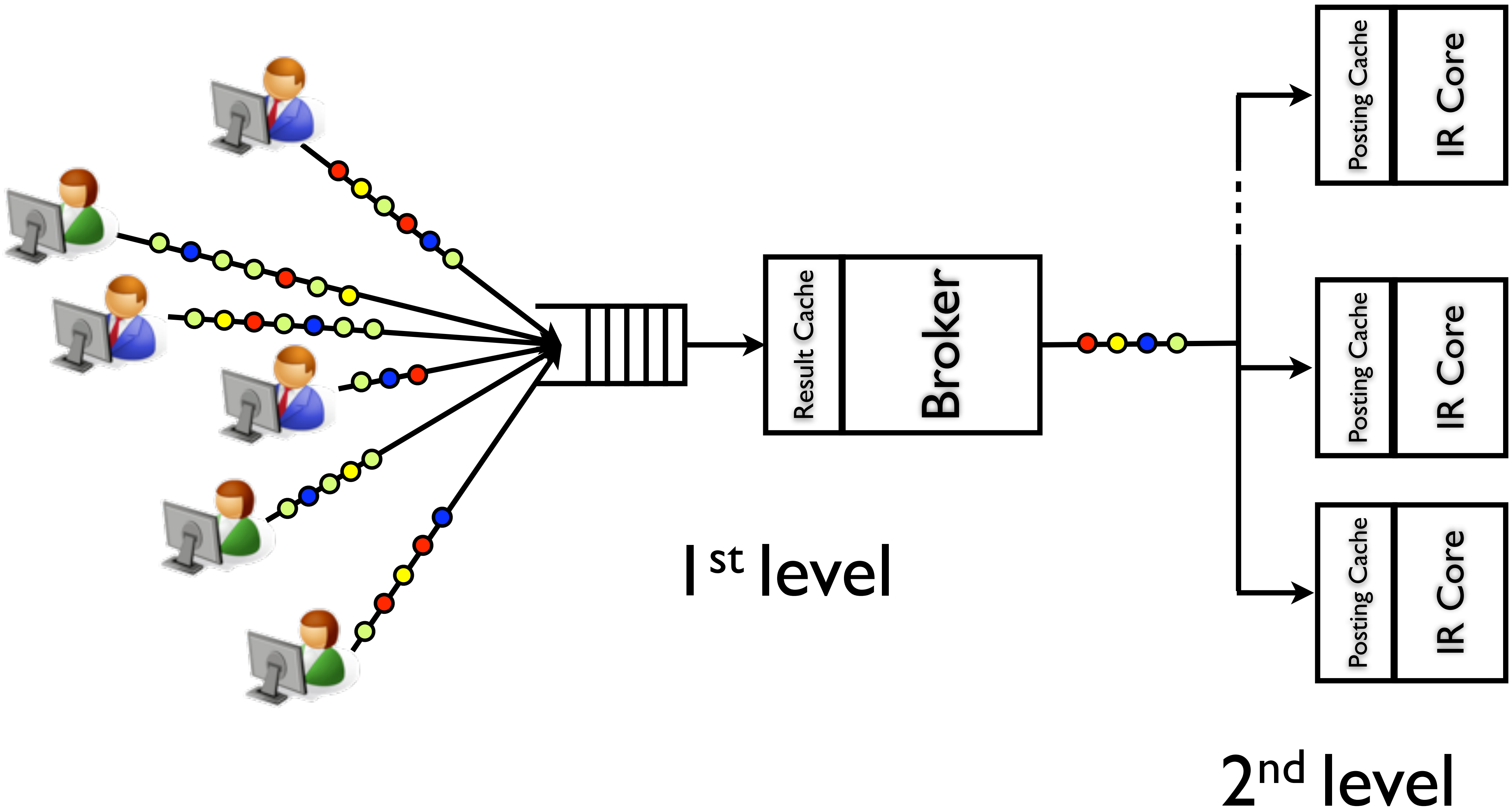
Caching for Search Engines Workloads

- Caching Architectures:
  - Two-Level Caching
  - Three-Level Caching
- Caching Policies
  - PDC
  - SDC
  - AC

Two-Level Caching

- Firstly studied in:
    - Saraiva, P. C., Silva de Moura, E., Ziviani, N., Meira, W., Fonseca, R., and Riberio-Neto, B. 2001. **Rank-preserving two-level caching for scalable search engines**. In Proceedings of ACM SIGIR '01. ACM, New York, NY, 51-58.
- Further analyzed in:
    - Baeza-Yates, R., Gionis, A., Junqueira, F. P., Murdock, V., Plachouras, V., and Silvestri, F. 2008. **Design trade-offs for search engine caching**. ACM Trans. Web 2, 4 (Oct. 2008), 1-28.

Two-Level Caching



Three-level Caching

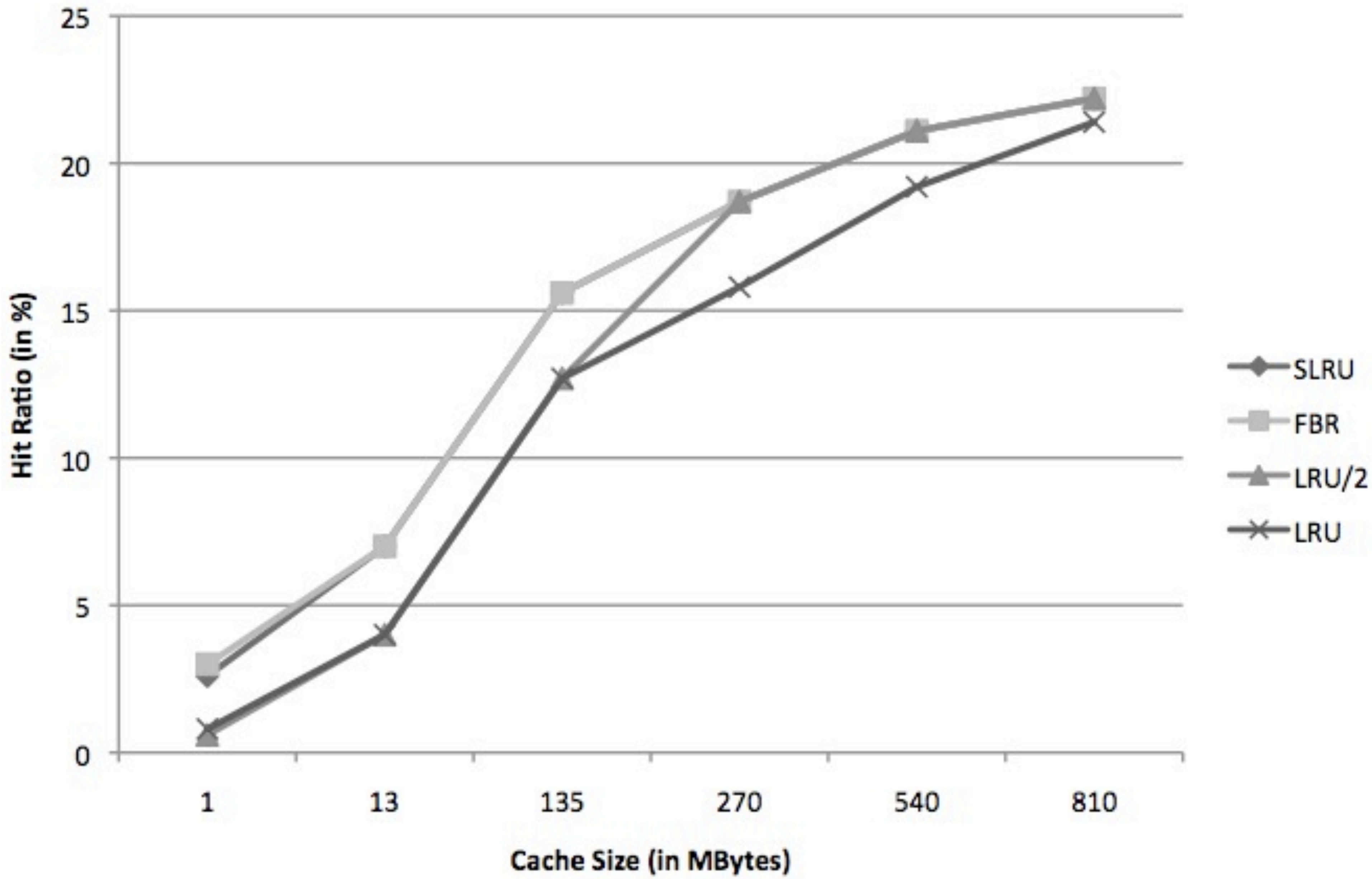
- Adds one level between results and posting lists cache.
- e.g., stores frequently occurring pairs of terms.
  - Long, X. and Suel, T. 2005. **Three-level caching for efficient query processing in large Web search engines**. In Proceedings of the 14th international Conference WWW '05. 257-266.
  - Skobeltsyn, G., Junqueira, F., Plachouras, V., and Baeza-Yates, R. 2008. **ResIn: a combination of results caching and index pruning for high-performance web search engines**. In Proceedings of the 31st Annual international ACM SIGIR '08. 131-138.

Traditional Replacement Policies

- LRU
- LFU
- SLRU
- ...

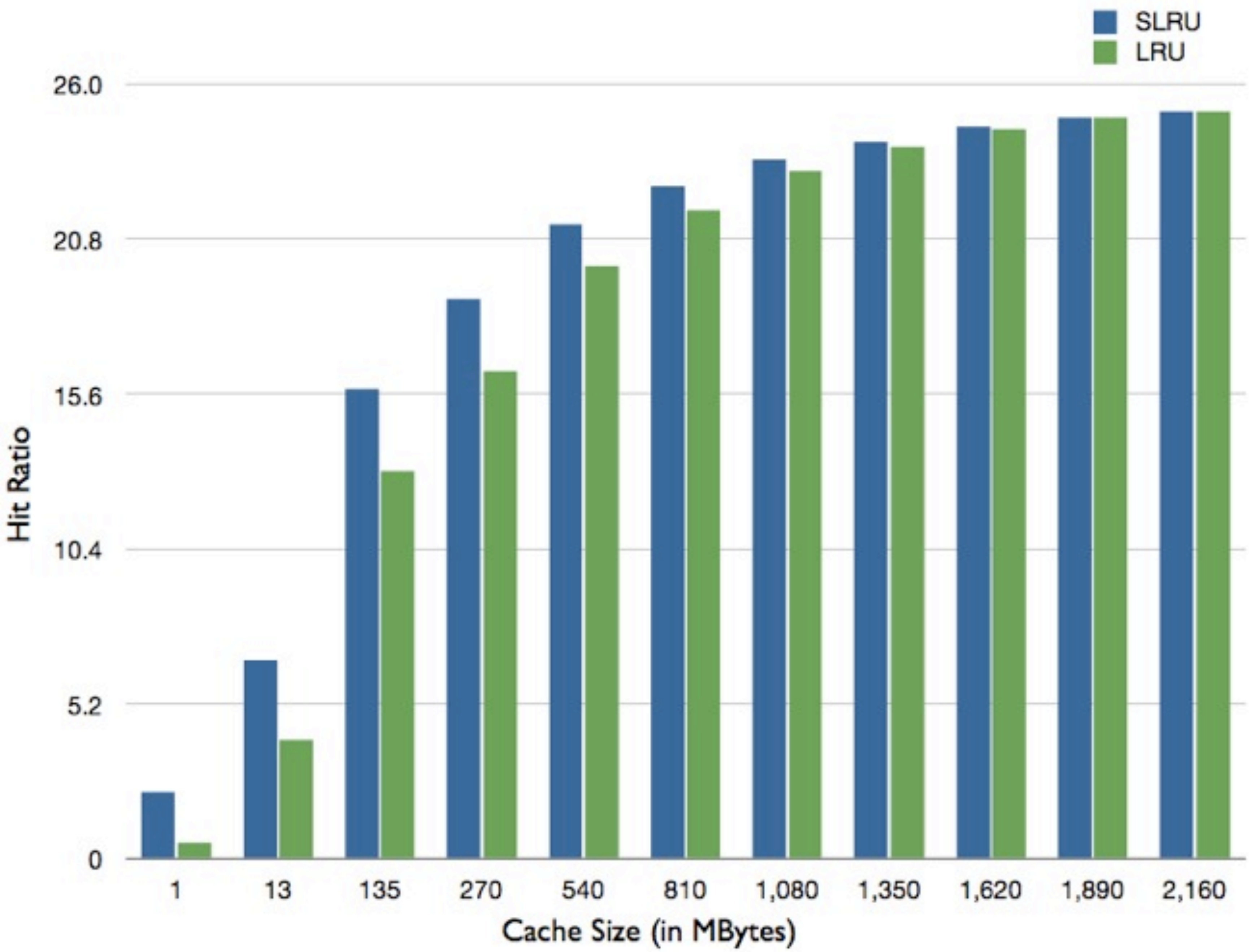
Evangelos P. Markatos: **On caching search engine query results**. Computer Communications 24(2): 137-143 (2001)

Hit Ratios on Excite



Fabrizio Silvestri: **Mining Query Logs: Turning Search Usage Data into Knowledge**.
Foundations and Trends in Information Retrieval. (To Appear).

SLRU vs. LRU on Excite

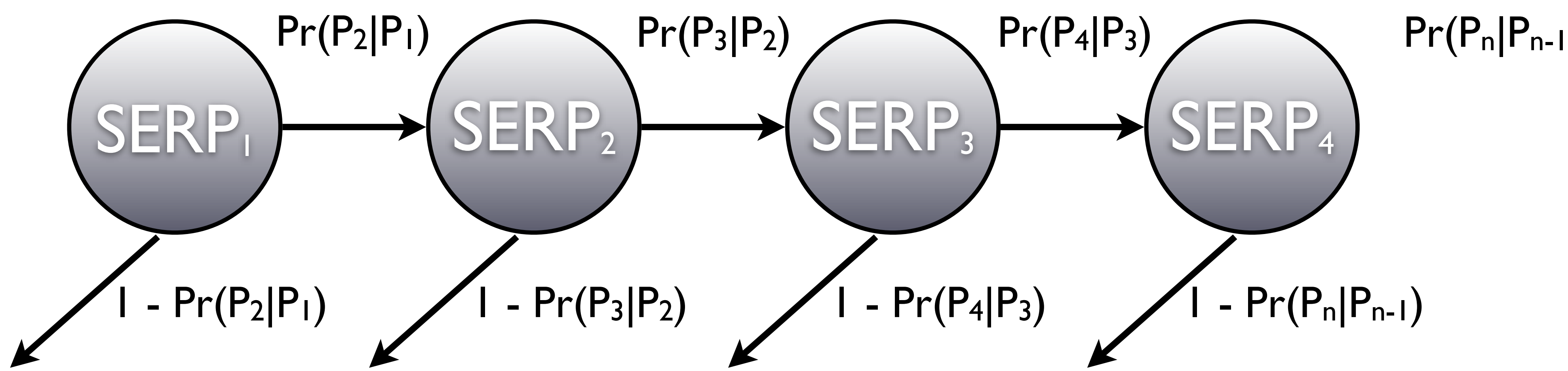


Evangelos P. Markatos: **On caching search engine query results**. Computer Communications 24(2): 137-143 (2001)

Search Engine Tailored Policies

- PDC
  - Probability Driven Caching
- SDC
  - Static Dynamic Caching
- AC
  - Admission Control

PDC



- IDEA: design a policy tailored over users' behavior on search pages
- With high probability users do not go beyond the first page of results
- For some query users browse many result pages.

Lempel, R. and Moran, S. 2003. **Predictive caching and prefetching of query results in search engines.** In Proceedings of WWW '03. 19-28.

PDC Priorities

- Priorities are assigned using an approximation of the Markovian SERP request model
- Each SERP different from the first one has a priority computed on historical queries (query log)
    - PDC caches pages that has follow-up queries more likely to be submitted. Why?

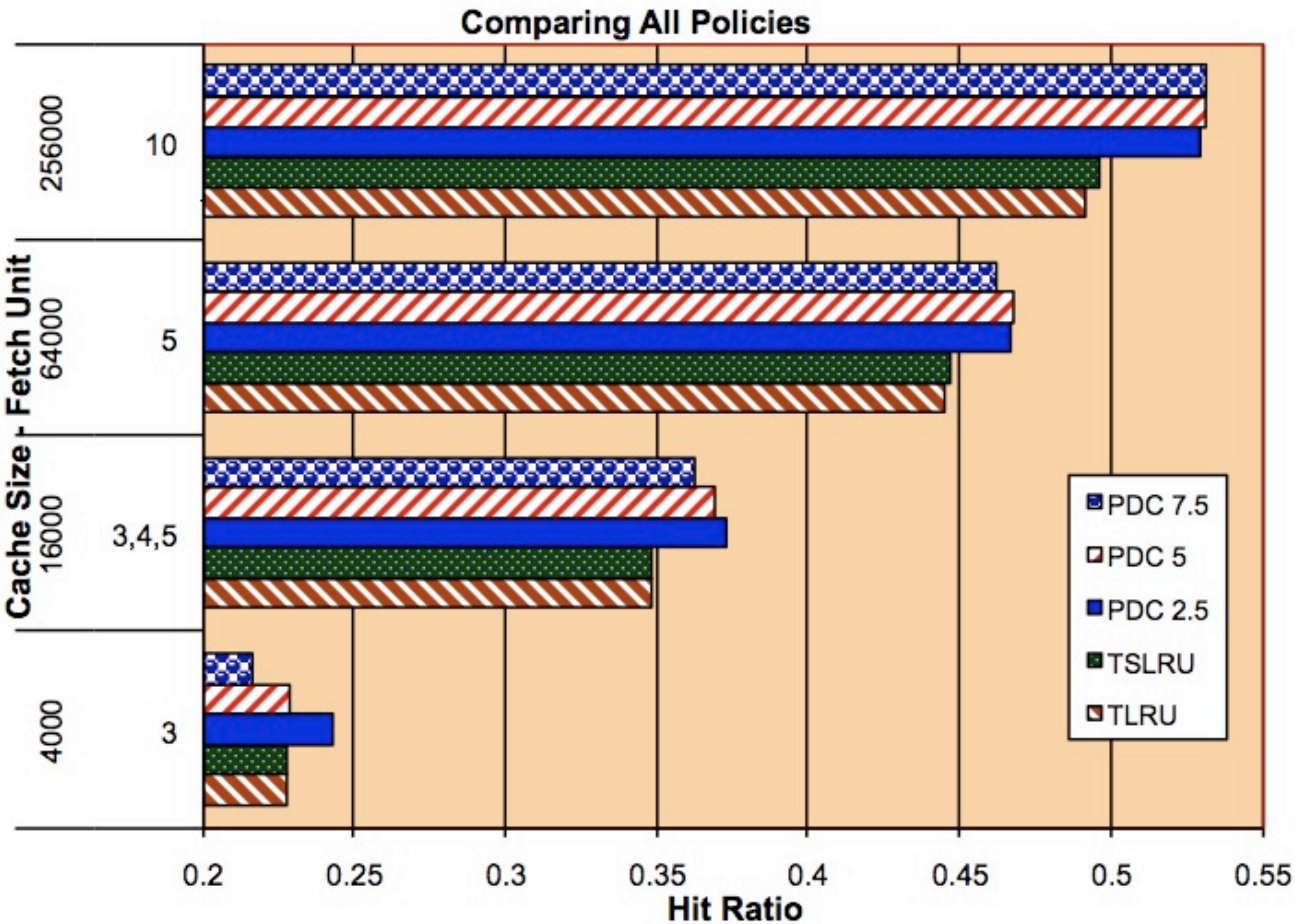
Lempel, R. and Moran, S. 2003. **Predictive caching and prefetching of query results in search engines.** In Proceedings of WWW '03. 19-28.

PDC and Prefetching

- in PDC results are organized according to "*Fetch Units*"
- When SERP i is requested for a query Q, the cache is first looked up
- If i is not cached, SERPs i, i + 1, ..., i + f are requested to the back-end
  - That is we prefetch f SERPs.
  - The fetch unit is of size f.

Lempel, R. and Moran, S. 2003. **Predictive caching and prefetching of query results in search engines.** In Proceedings of WWW '03. 19-28.

PDC Results



Lempel, R. and Moran, S. 2003. **Predictive caching and prefetching of query results in search engines.** In Proceedings of WWW '03. 19-28.

PDC's Main Lessons Learned

Lempel, R. and Moran, S. 2003. **Predictive caching and prefetching of query results in search engines.** In Proceedings of WWW '03. 19-28.

PDC's Main Lessons Learned

- Hit ratio benefits a lot from the use of historical data

Lempel, R. and Moran, S. 2003. **Predictive caching and prefetching of query results in search engines.** In Proceedings of WWW '03. 19-28.

PDC's Main Lessons Learned

- Hit ratio benefits a lot from the use of historical data
- Prefetching helps a lot!

Lempel, R. and Moran, S. 2003. **Predictive caching and prefetching of query results in search engines.** In Proceedings of WWW '03. 19-28.

PDC's Main Lessons Learned

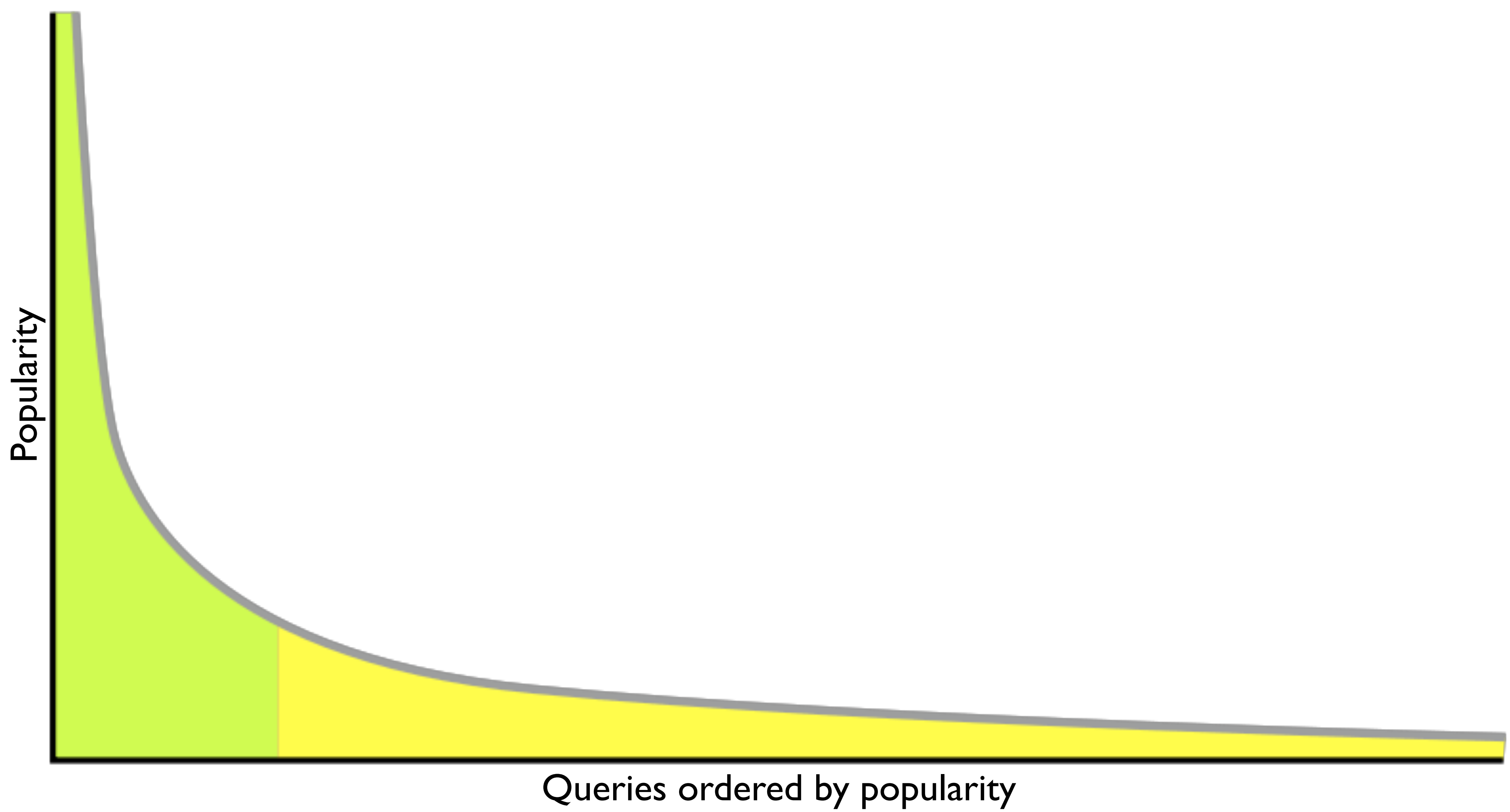
- Hit ratio benefits a lot from the use of historical data
- Prefetching helps a lot!
- Differently from previous caching policies, PDC not necessarily caches every submitted queries!!!

Lempel, R. and Moran, S. 2003. **Predictive caching and prefetching of query results in search engines.** In Proceedings of WWW '03. 19-28.

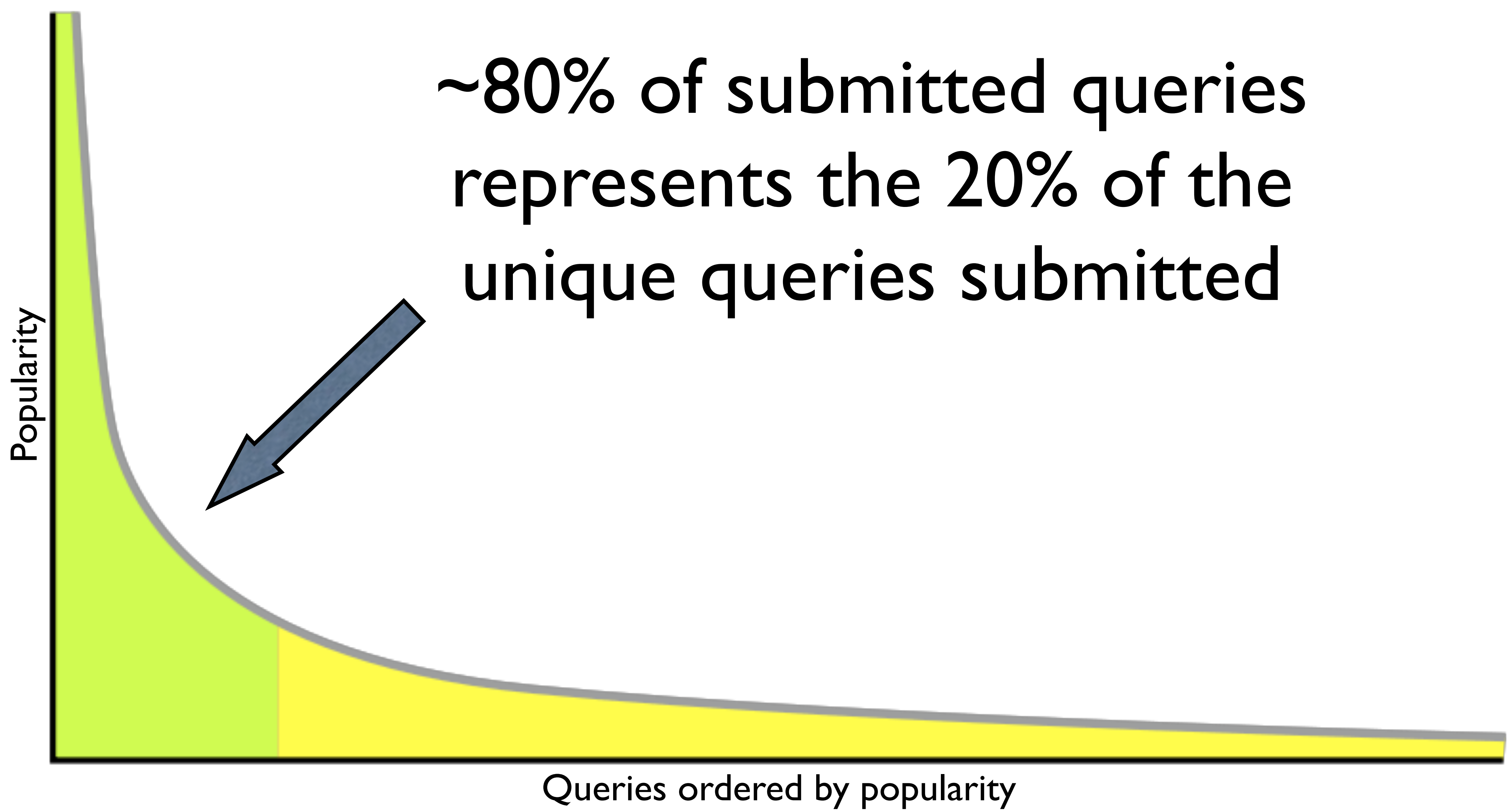
Overcoming PDC Complexity

- PDC uses query logs to estimate the likelihood of follow-up queries.
- Why not using query logs to estimate likelihood of resubmitting a query.
- Catching the head of the long tail distribution we might obtain high hit ratios

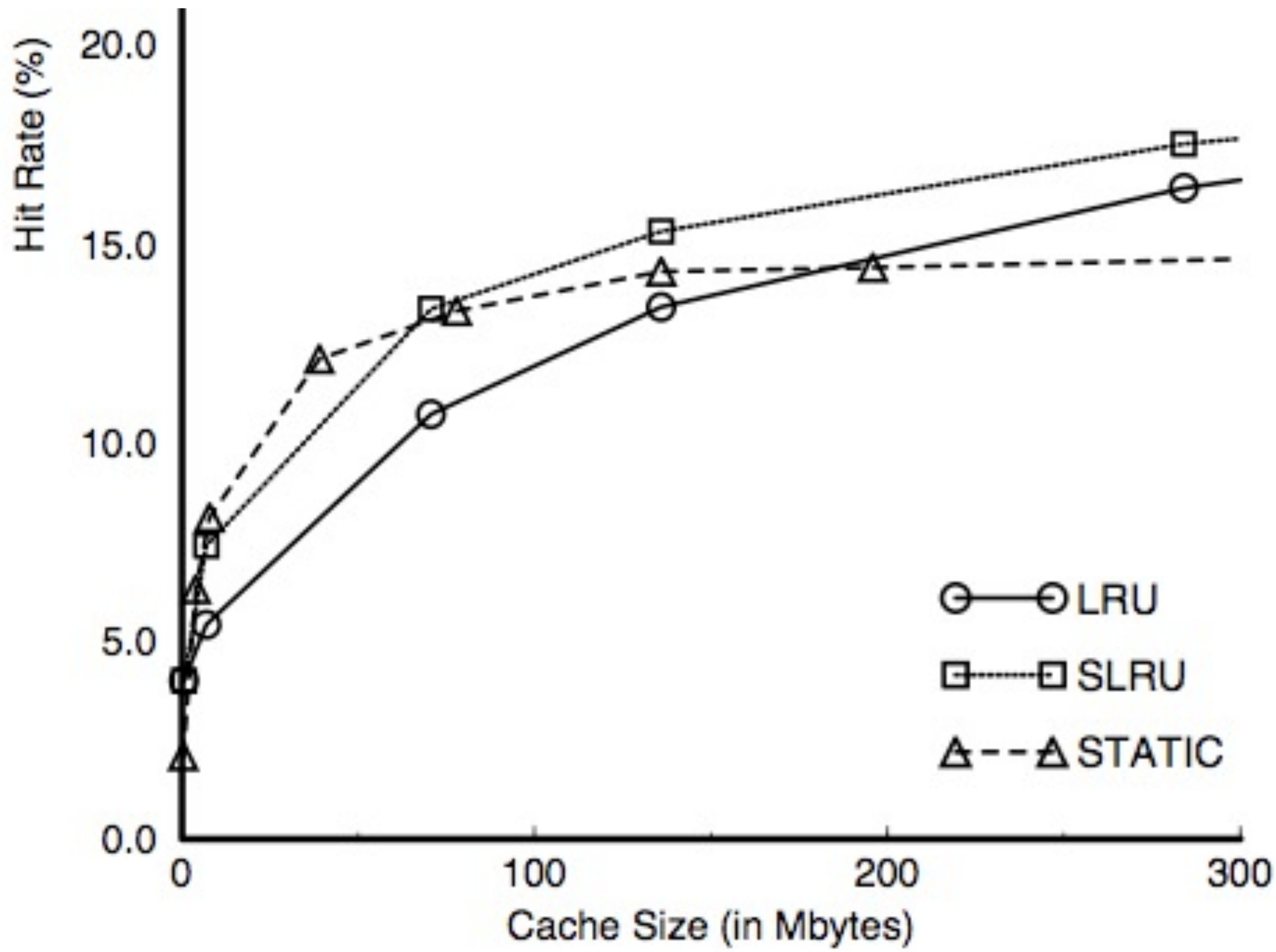
That is...



That is...


~80% of submitted queries
represents the 20% of the
unique queries submitted
Popularity
Queries ordered by popularity

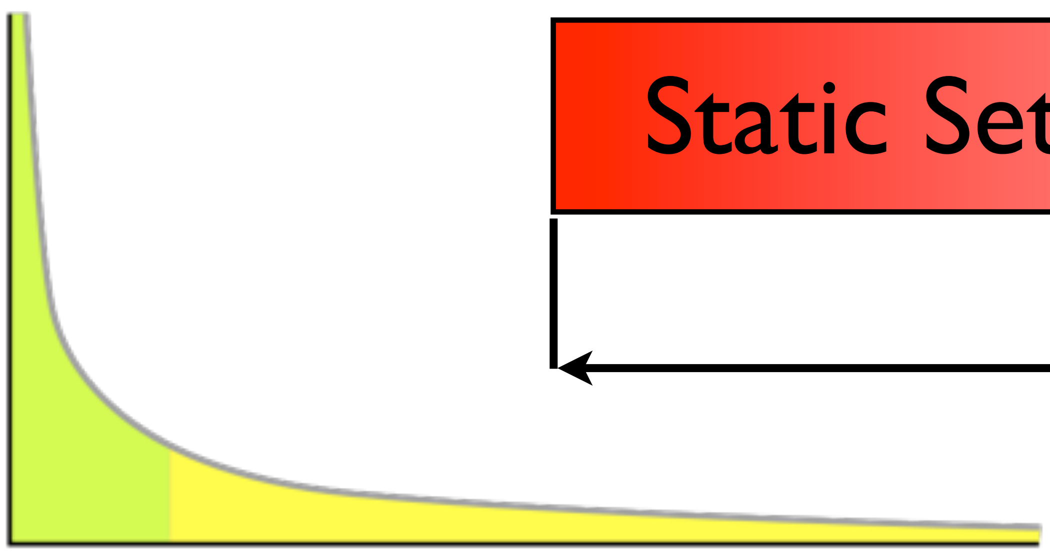
But...

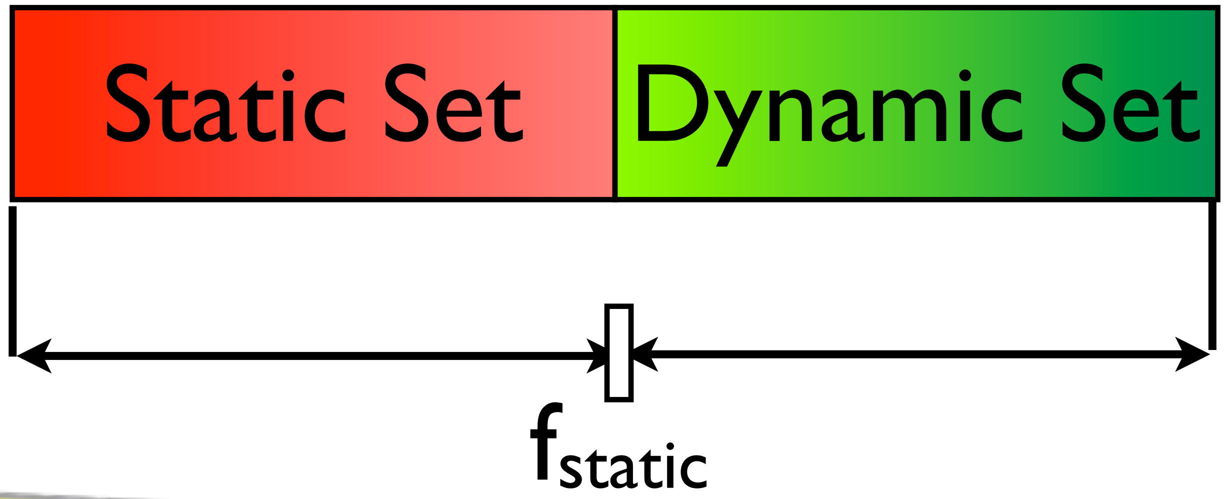


Evangelos P. Markatos: **On caching search engine query results**. Computer Communications 24(2): 137-143 (2001)

Static Dynamic Caching

- SDC (Static Dynamic Caching) adds to the classical static caching schema a dynamically managed section.
- The idea:

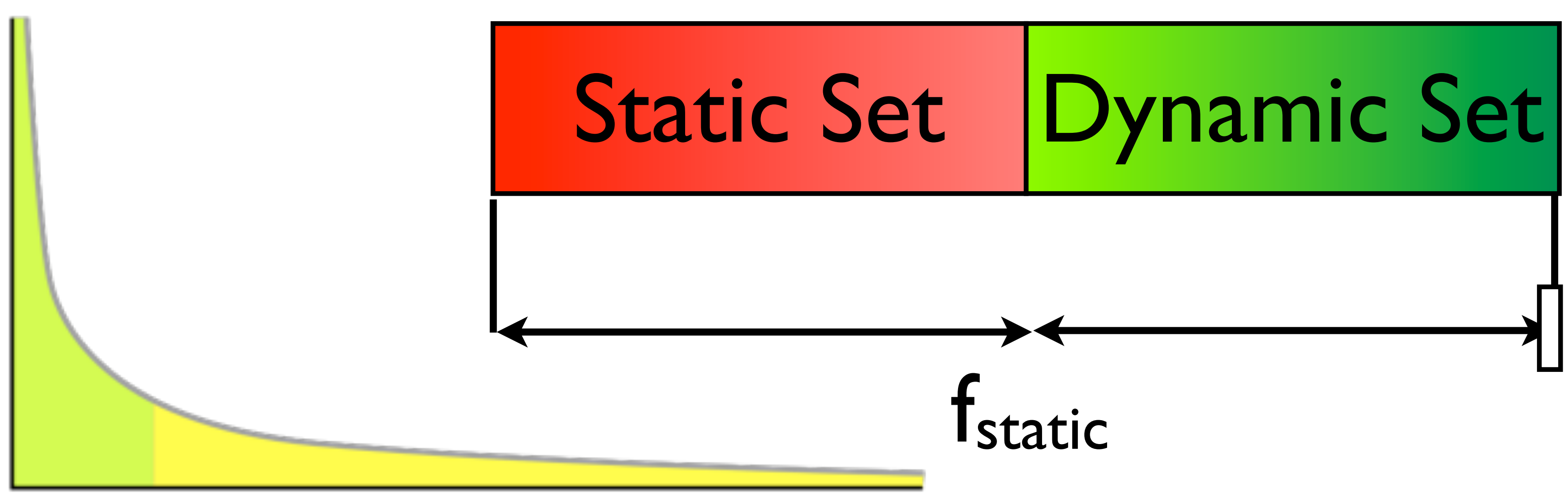




T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

Static Dynamic Caching

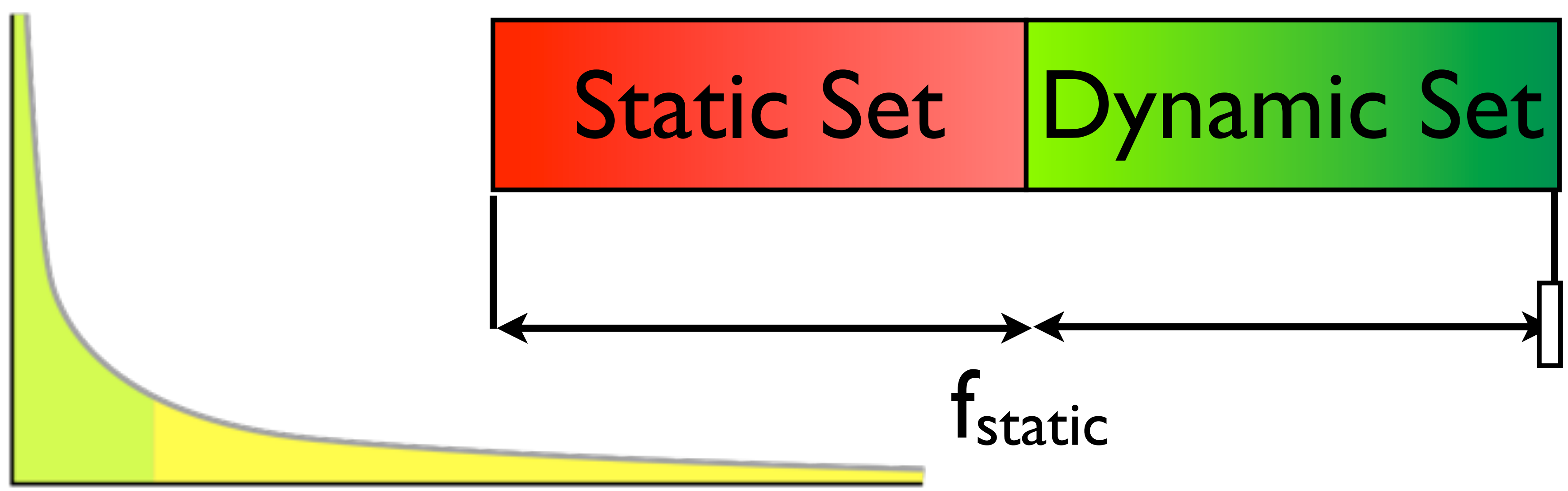
- SDC (Static Dynamic Caching) adds to the classical static caching schema a dynamically managed section.
- The idea:



T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

Static Dynamic Caching

- SDC (Static Dynamic Caching) adds to the classical static caching schema a dynamically managed section.
- The idea:

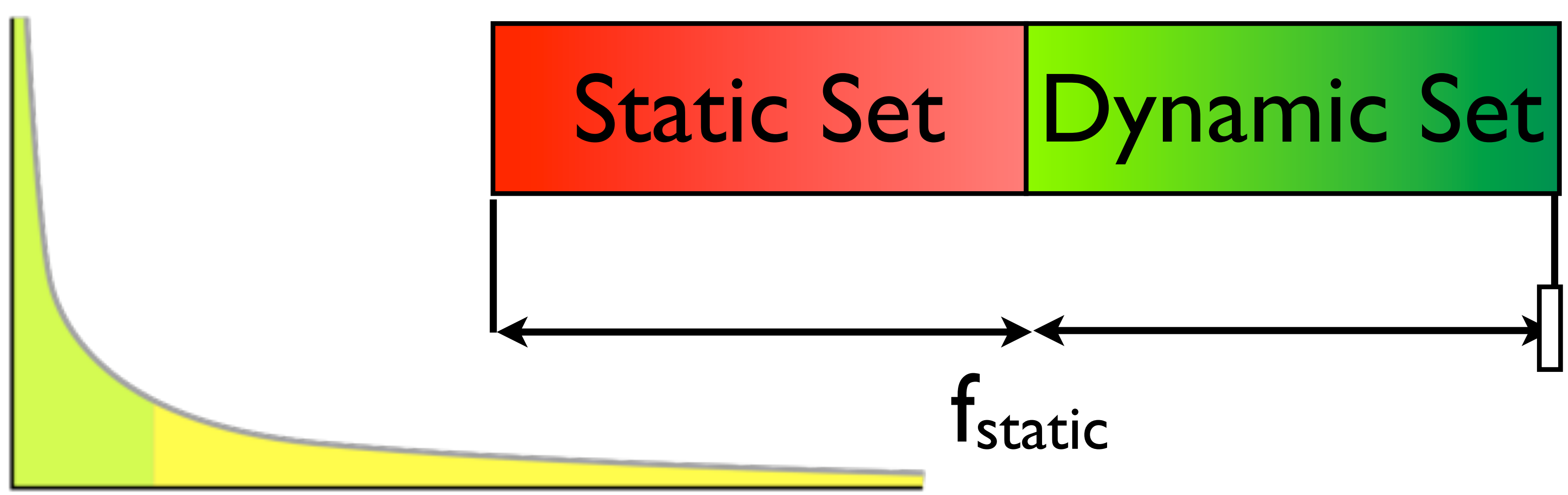


- LRU
- SLRU
- PDC
- ...

T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

Static Dynamic Caching

- SDC (Static Dynamic Caching) adds to the classical static caching schema a dynamically managed section.
- The idea:



- LRU
- SLRU
- PDC
- ...

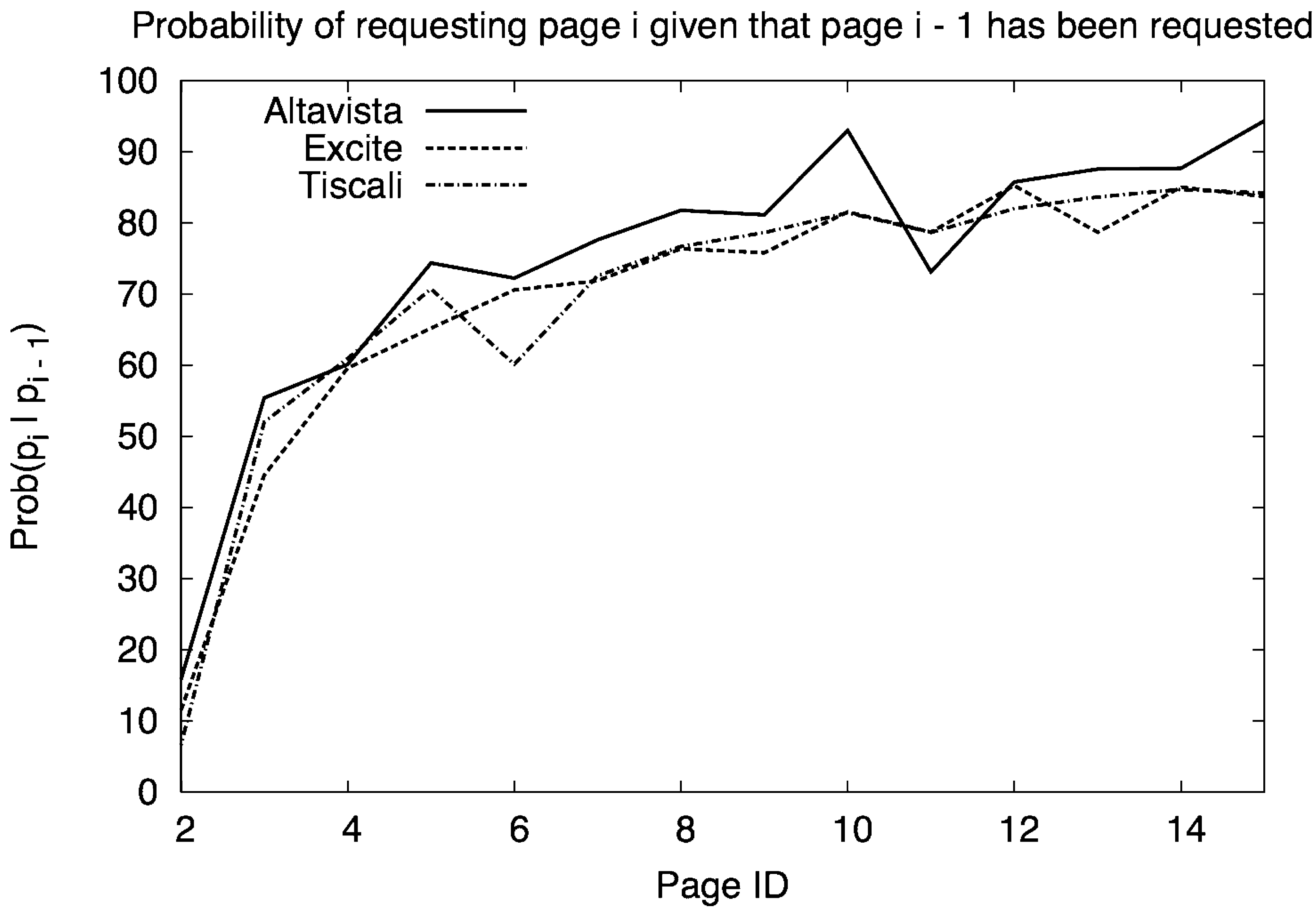
T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

SDC and Prefetching

- SDC adopts an "adaptive" prefetching technique:
  - For the first SERP do not prefetch
  - For the follow-up SERPs prefetch f pages

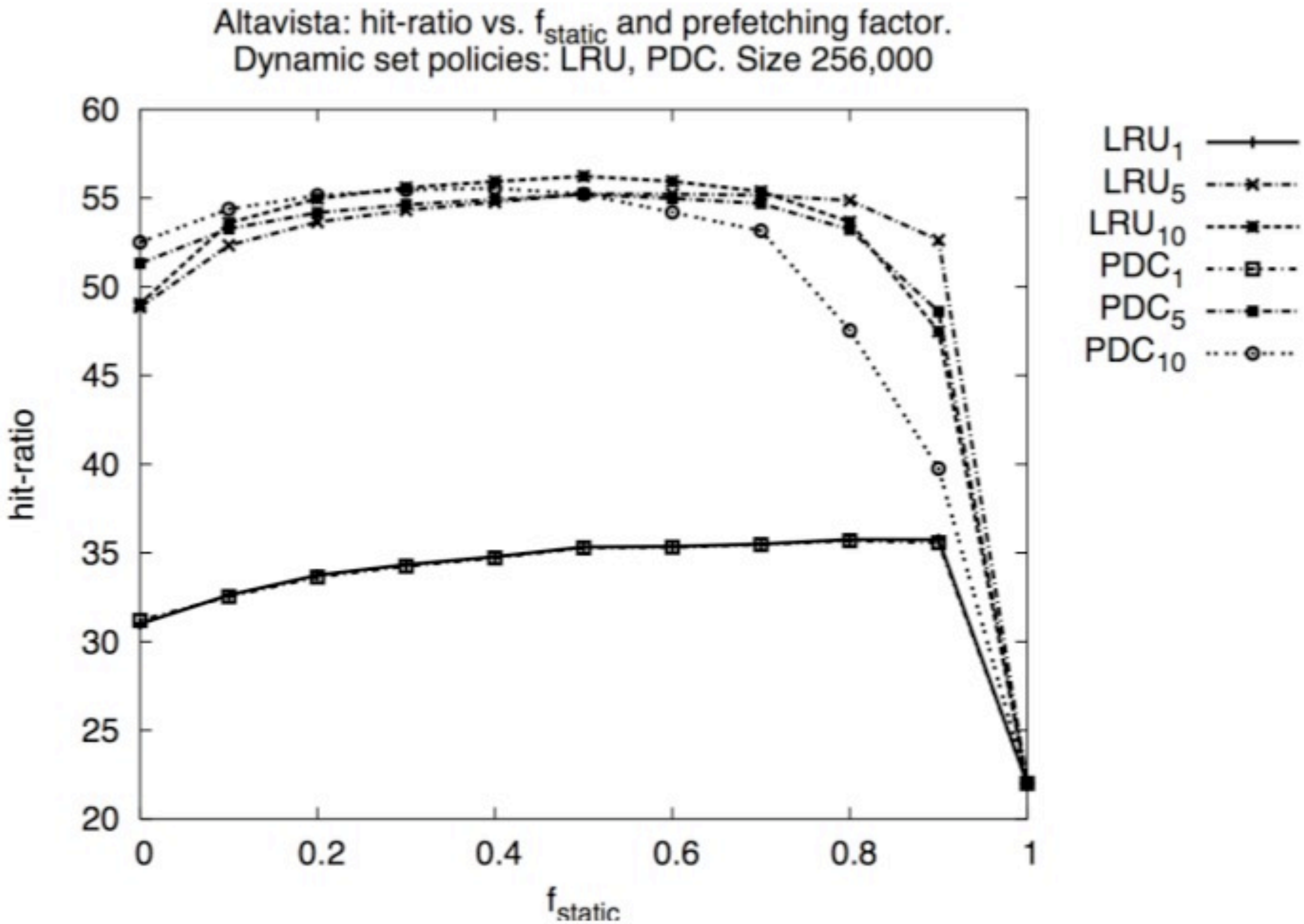
T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

SDC and Prefetching



T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

SDC Hit-Ratios



T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

SDC's Main Lessons Learned

T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

SDC's Main Lessons Learned

- Hit ratio benefits a lot from the use of historical data

T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

SDC's Main Lessons Learned

- Hit ratio benefits a lot from the use of historical data
- Prefetching helps a lot!

T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

SDC's Main Lessons Learned

- Hit ratio benefits a lot from the use of historical data
- Prefetching helps a lot!
- Static caching alone is not useful, yet...

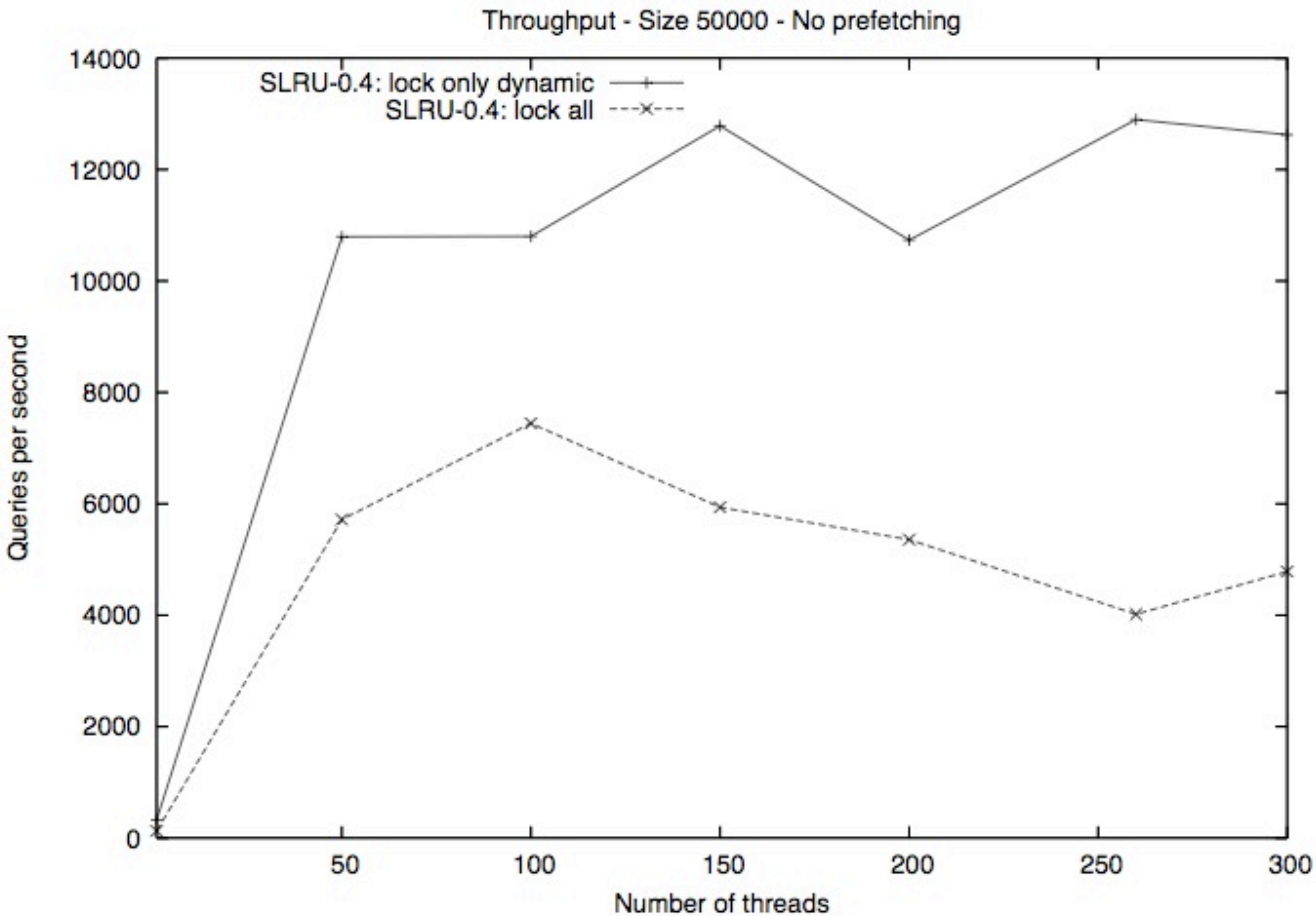
T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

SDC's Main Lessons Learned

- Hit ratio benefits a lot from the use of historical data
- Prefetching helps a lot!
- Static caching alone is not useful, yet...
  - A good combination of a static and a dynamic approach helps a lot!!!

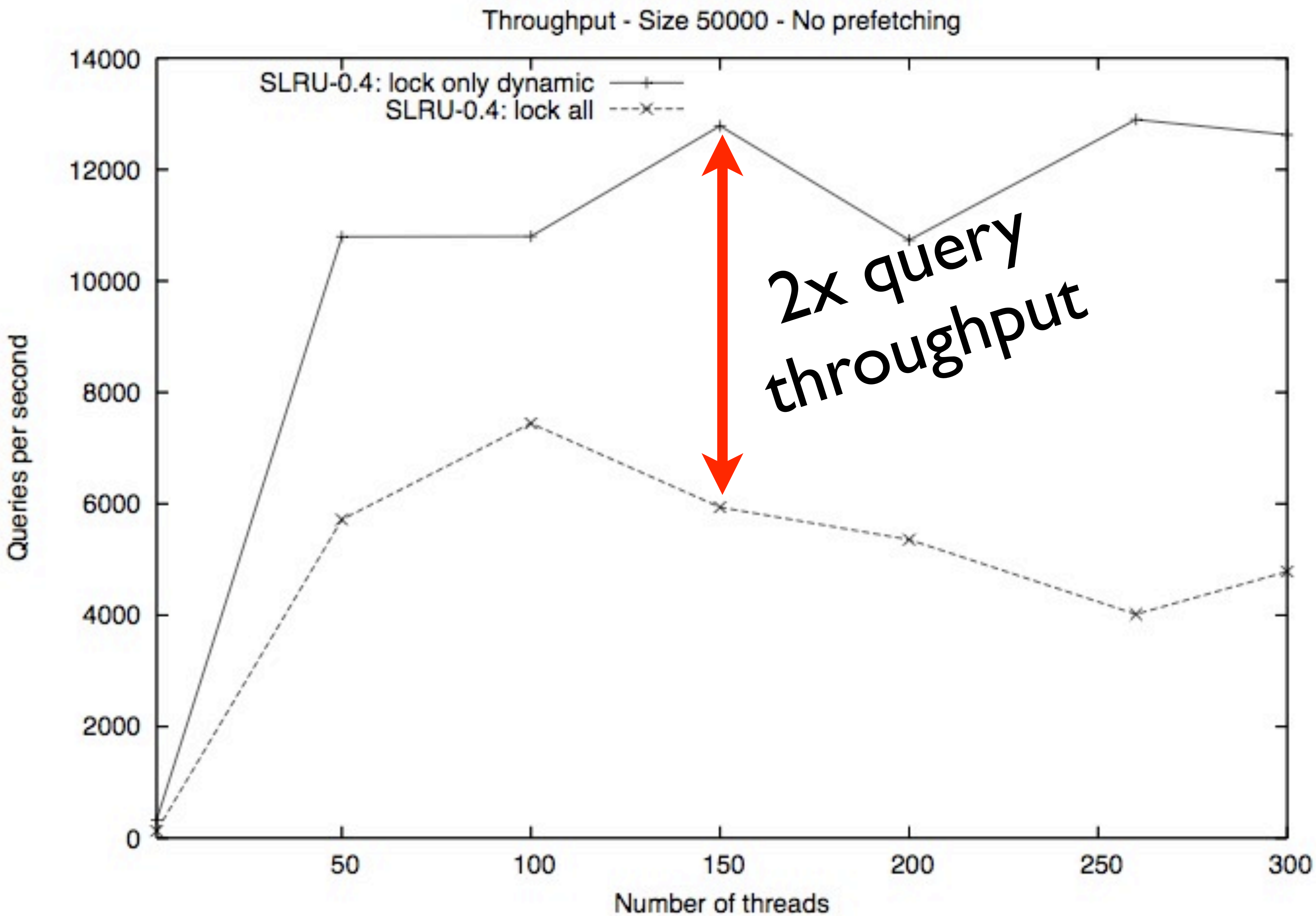
T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

That's *<u>not</u>* All Folks!



T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

That's *not* All Folks!



T. Fagni, R. Perego, F. Silvestri, and S. Orlando, "**Boosting the performance of web search engines: Caching and prefetching query results by exploiting historical usage data**," ACM Trans. Inf. Syst., vol. 24, no. 1, pp. 51–78, 2006.

Admission Control

- An interesting idea of SDC: frequent queries are cached permanently
- AC of Baeza-Yates et al. generalizes the idea by using two dynamically updated sets:
  - A Controlled Cache (**CC**)
  - An Uncontrolled Cache (**UC**)
- When a new query arrives an admission policy is applied to steer a query to the CC or to the UC.
- If the query is likely to be seen in the future move it to CC, otherwise send it to UC.

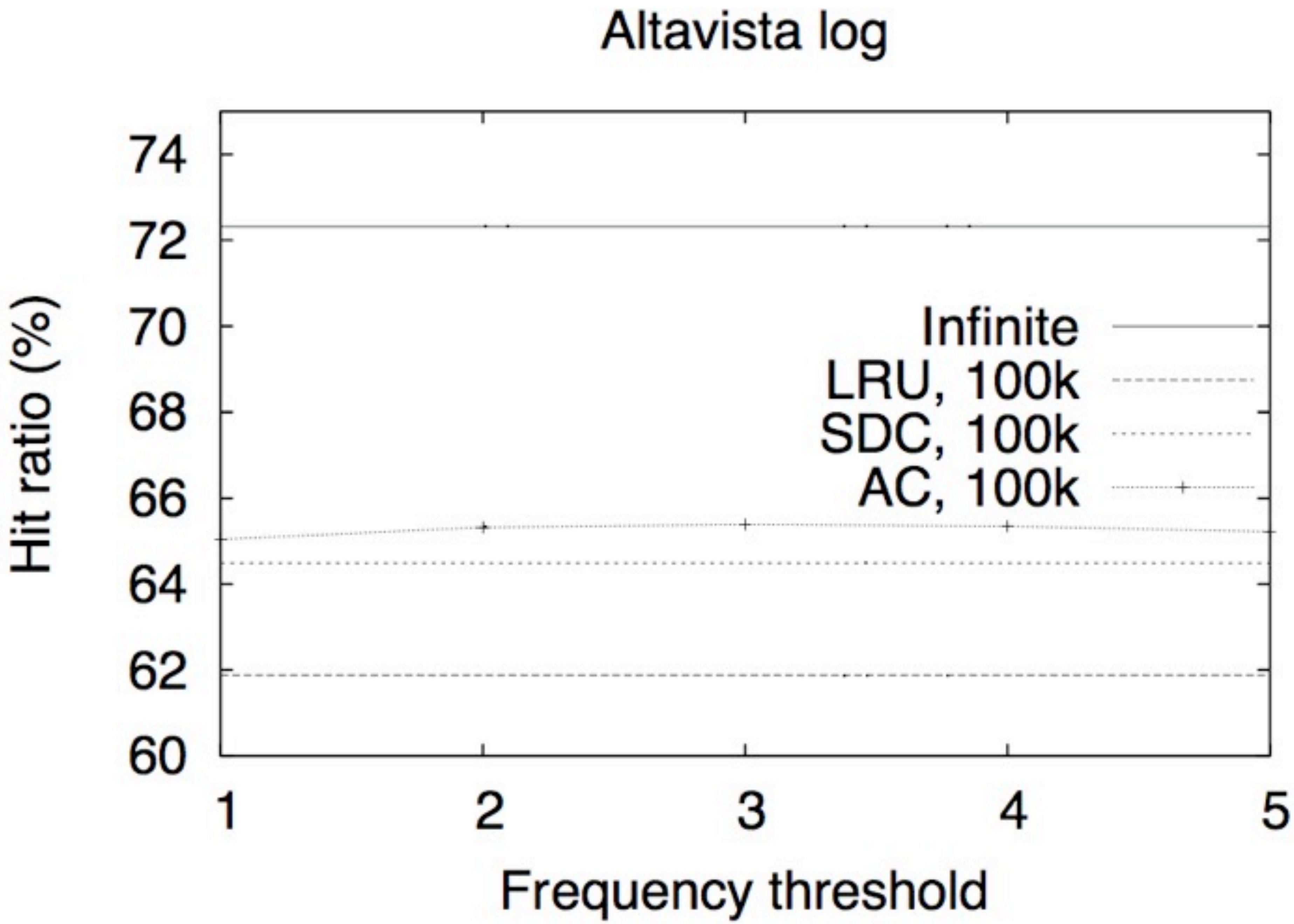
Ricardo Baeza-Yates, Flavio Junqueira, Vassilis Plachouras, and Hans Friedrich Witschel, "**Admission Policies for Caches of Search Engine Results**," SPIRE 2007, 74-85.

Admission Policy

- Makes use of features, e.g.:
  - Stateful features:
    - *PastF*: the frequency of the query in the (relatively recent) past
  - Stateless features:
    - *LenC*: the length of the query in characters
    - *LenW*: the length of the query in words

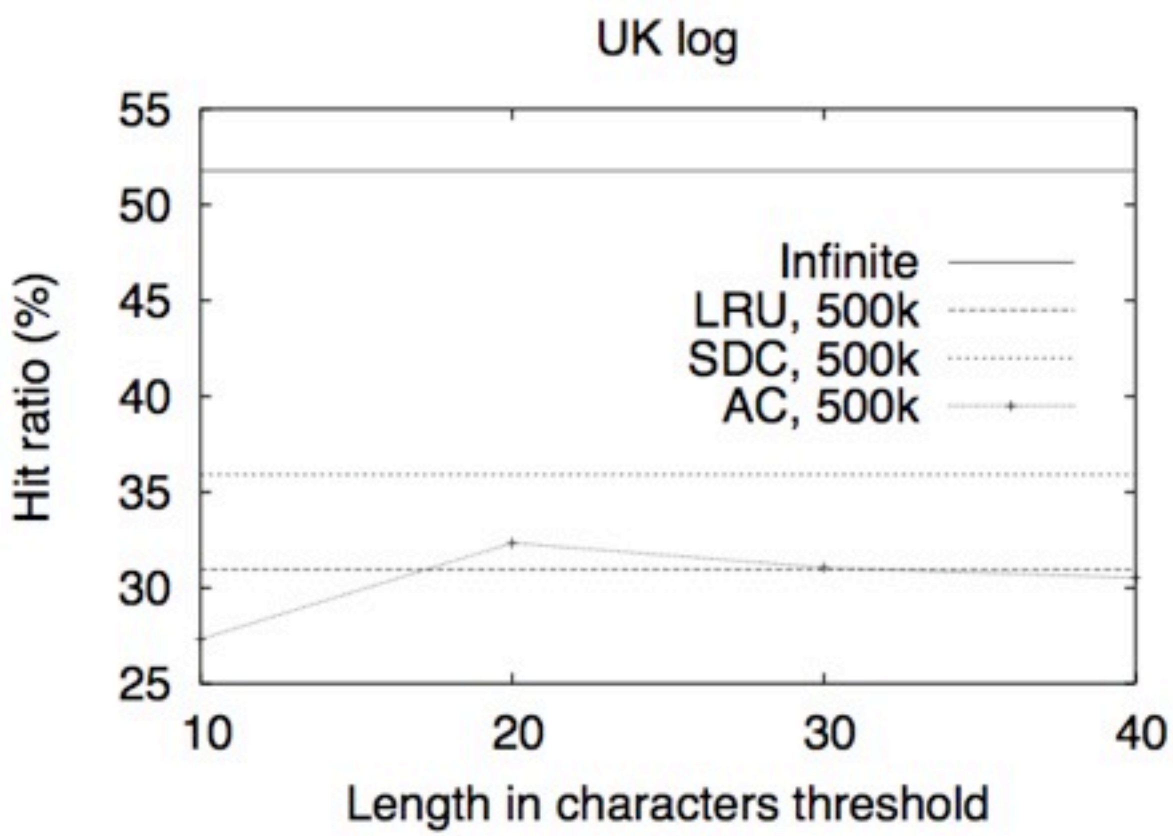
Ricardo Baeza-Yates, Flavio Junqueira, Vassilis Plachouras, and Hans Friedrich Witschel, "**Admission Policies for Caches of Search Engine Results**," SPIRE 2007, 74-85.

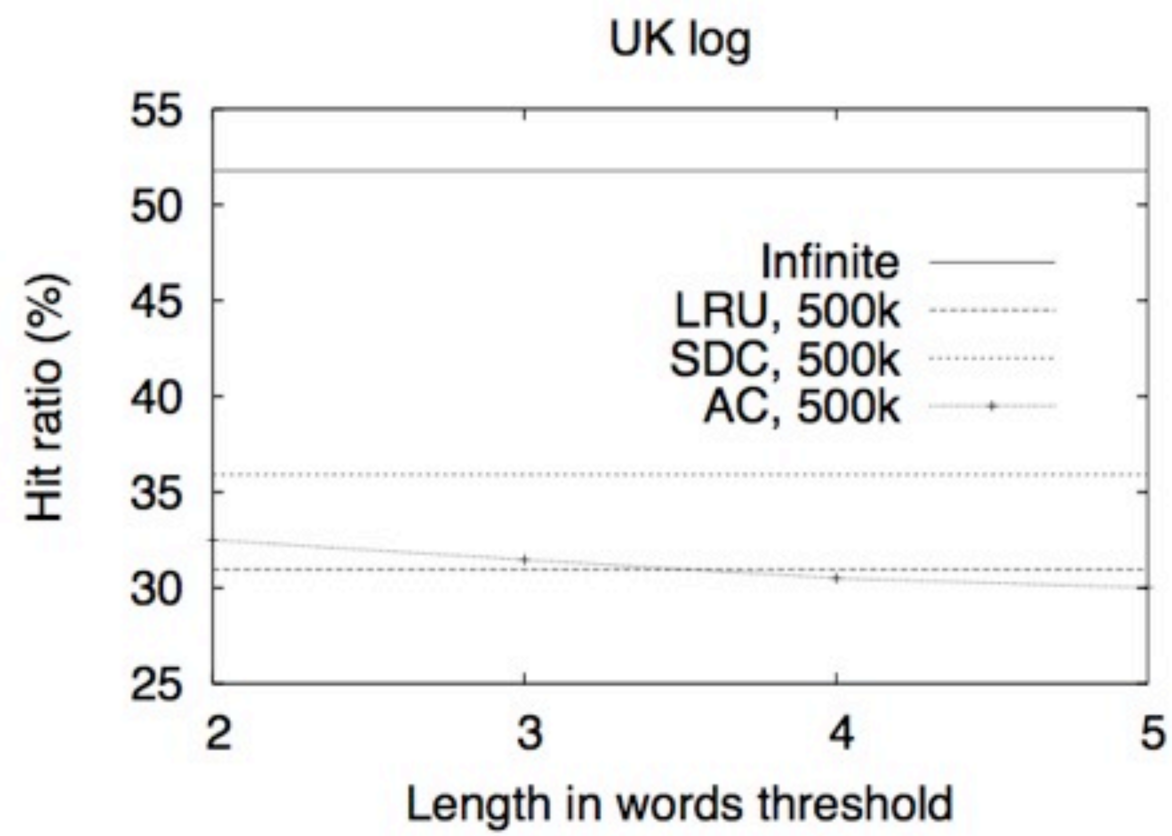
Hit-Ratio Results (Past1-5)



Ricardo Baeza-Yates, Flavio Junqueira, Vassilis Plachouras, and Hans Friedrich Witschel, "**Admission Policies for Caches of Search Engine Results**," SPIRE 2007, 74-85.

Hit-Ratio Results (LenC- LenW)



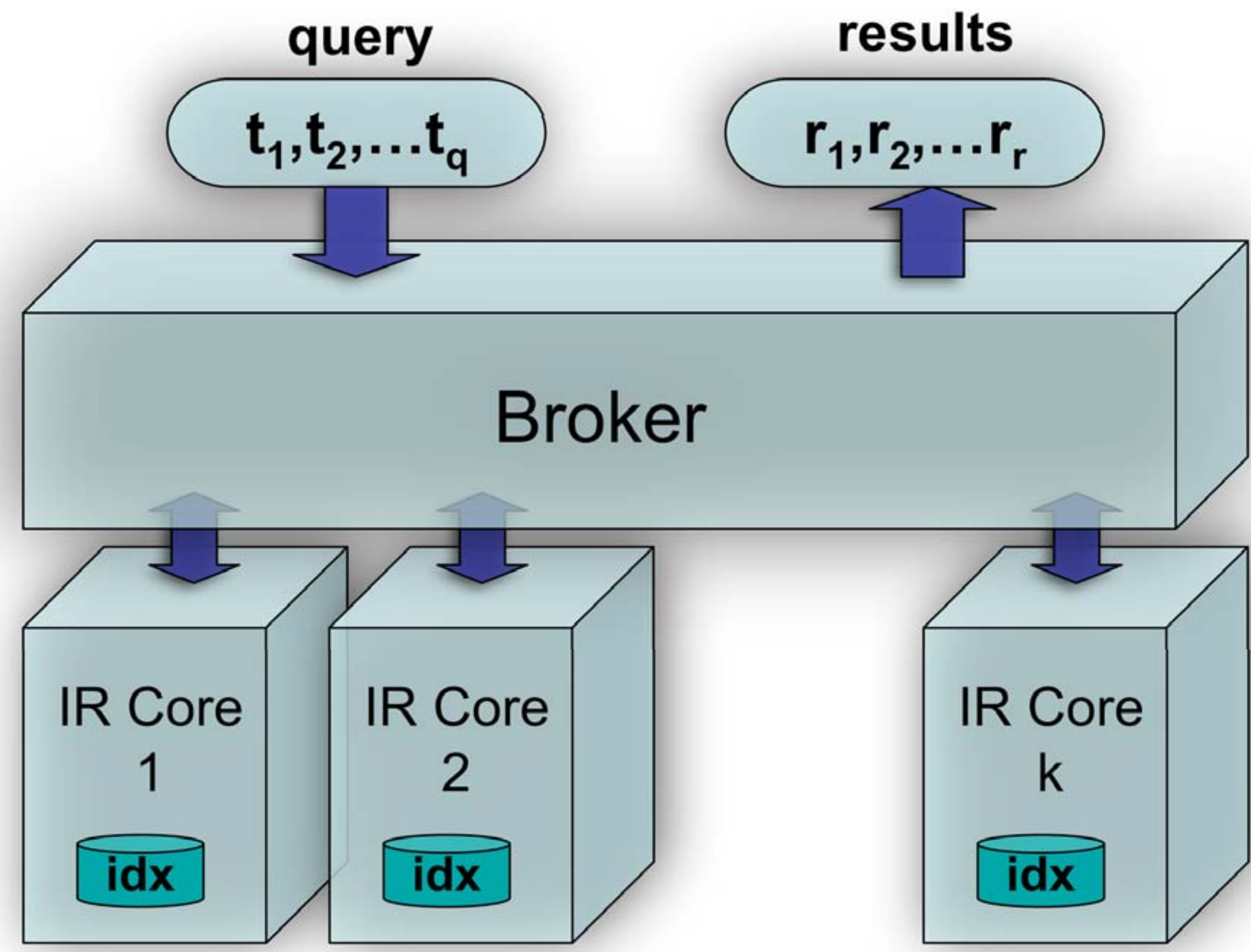


Ricardo Baeza-Yates, Flavio Junqueira, Vassilis Plachouras, and Hans Friedrich Witschel, "**Admission Policies for Caches of Search Engine Results**," SPIRE 2007, 74-85.

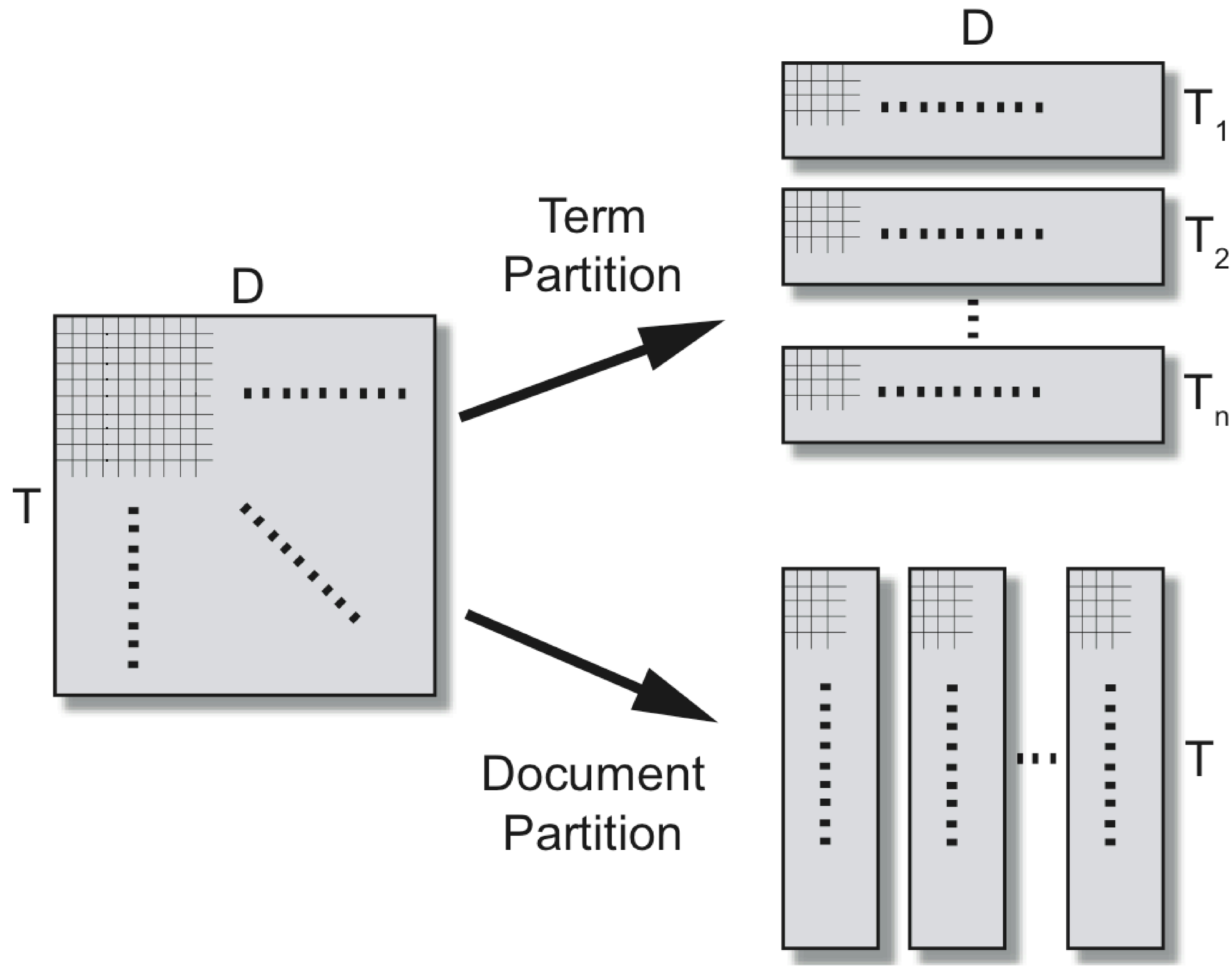
Not Only Caching

- Improve efficiency using query logs can also be done by:
  - query routing
  - data/index partitioning

Sketching a Distributed Search Engine



Index Partitioning



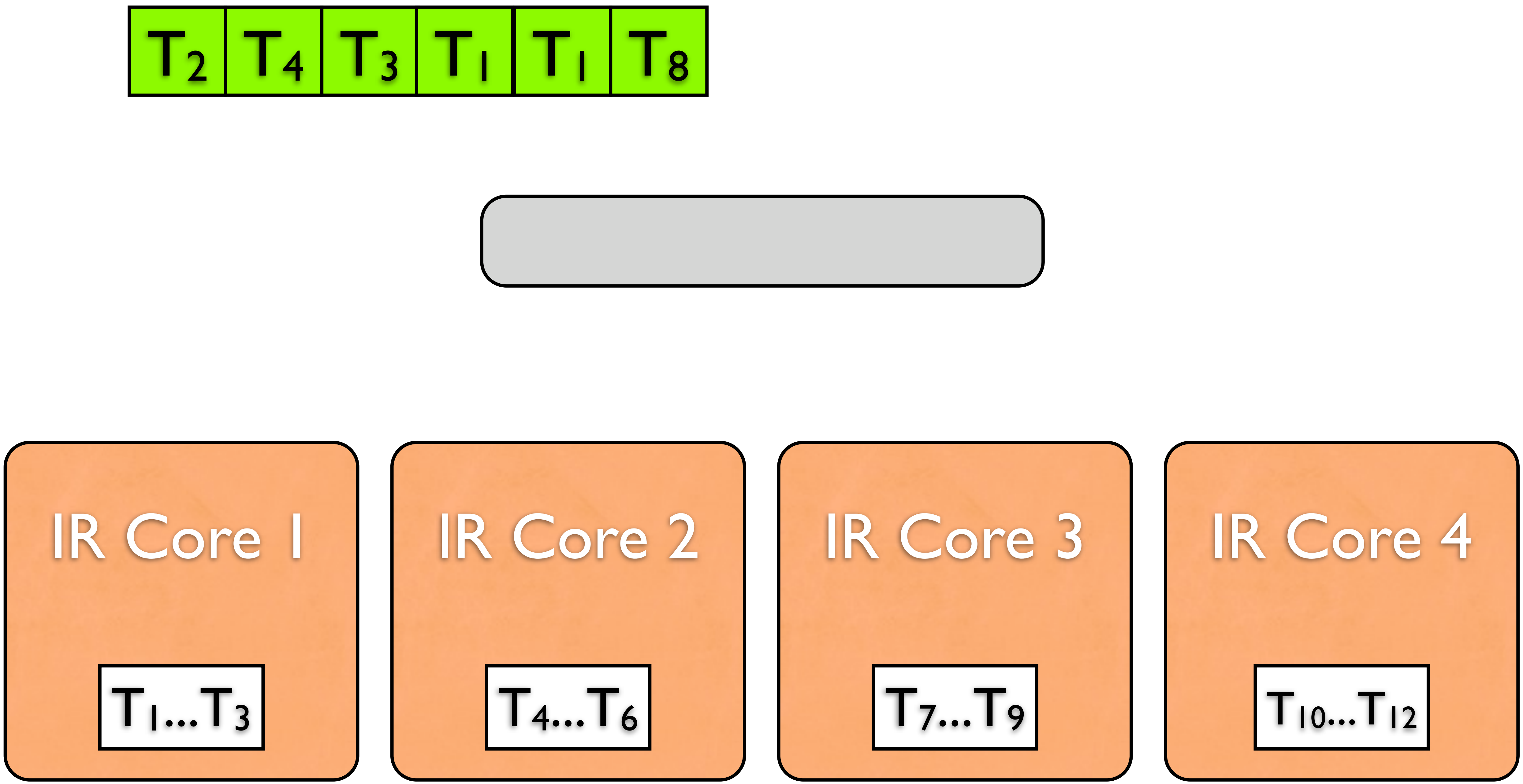
Term Partitioning Systems

- Query routing is "trivial"...
- Whenever a query $Q=(t_1,t_2, ..., t_n)$ is received route it to the servers managing those terms.
- But..
  - *not scalable* (indexing is n log n, term partitioning needs reindexing the entire collection from scratch when un update occurs)
  - *load imbalance* among IR cores

Why Studying Term Partitioned Systems?

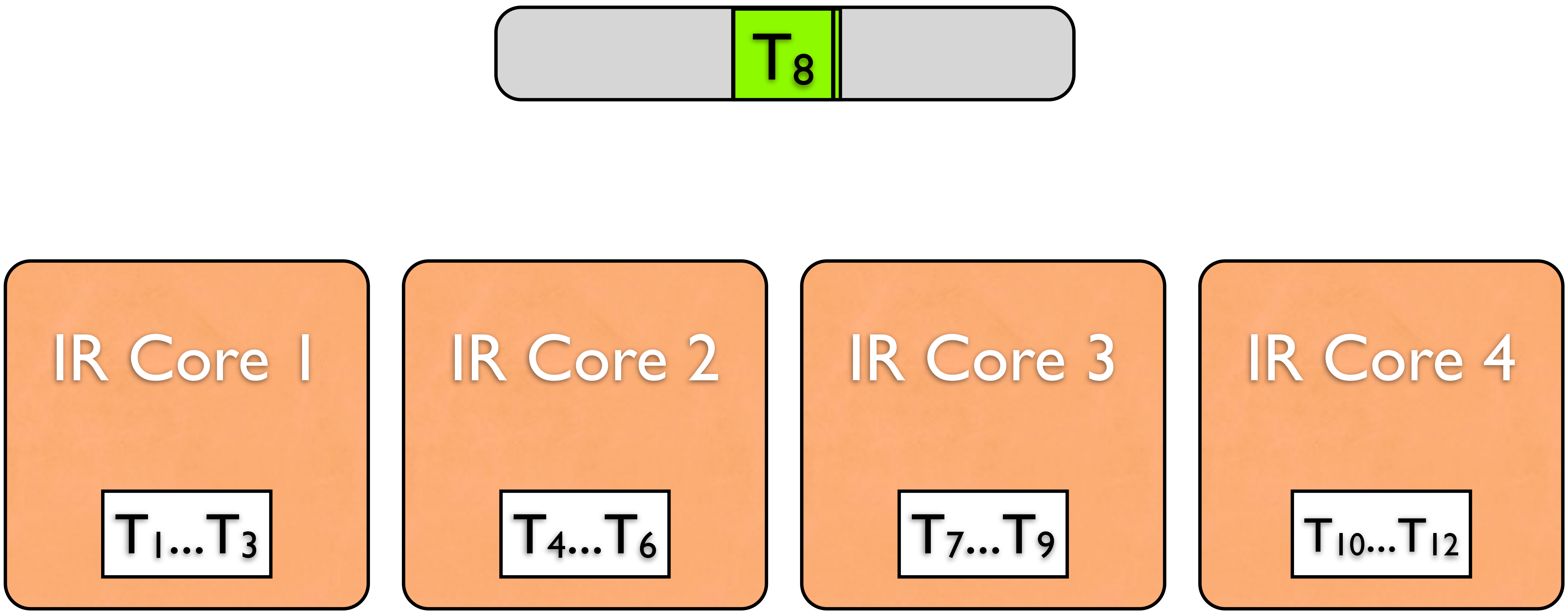
- In principle:
  - less IR Cores queried
  - less operations performed
- Briefly...
  - More available capacity!

Term Partitioning
(Random Partitioning)



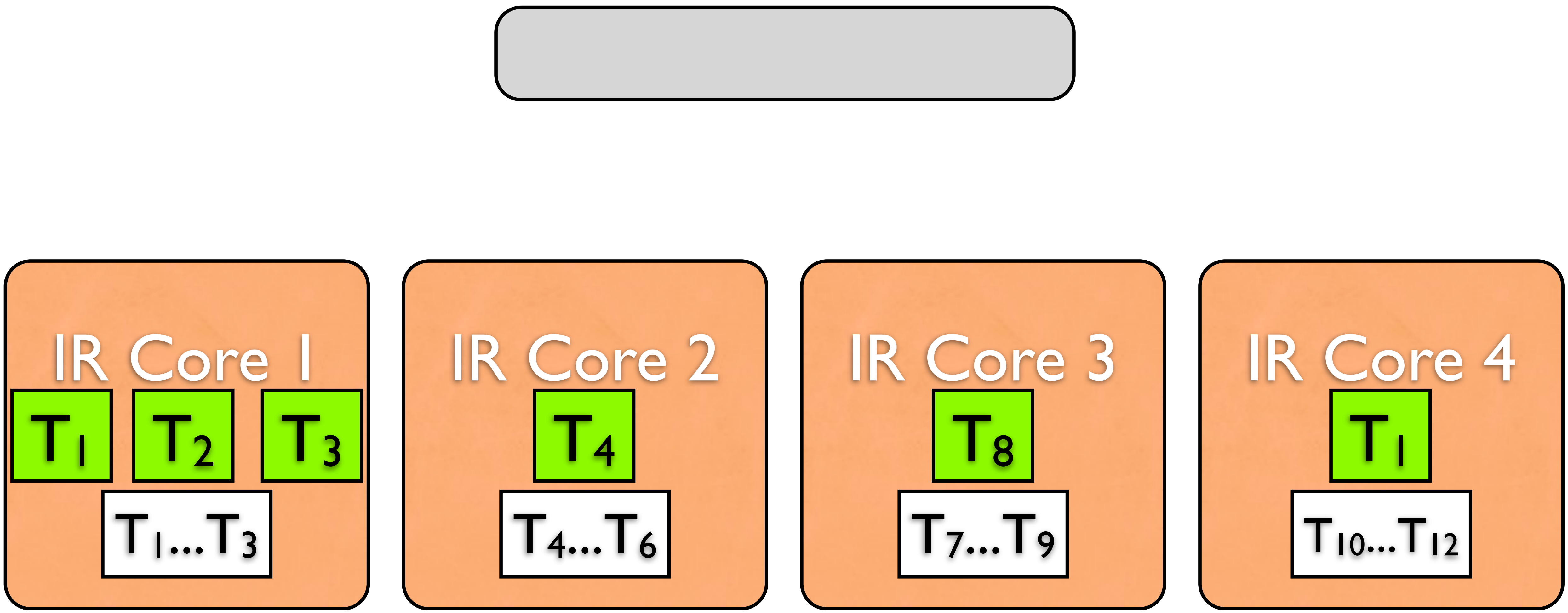
Term Partitioning

(Random Partitioning)

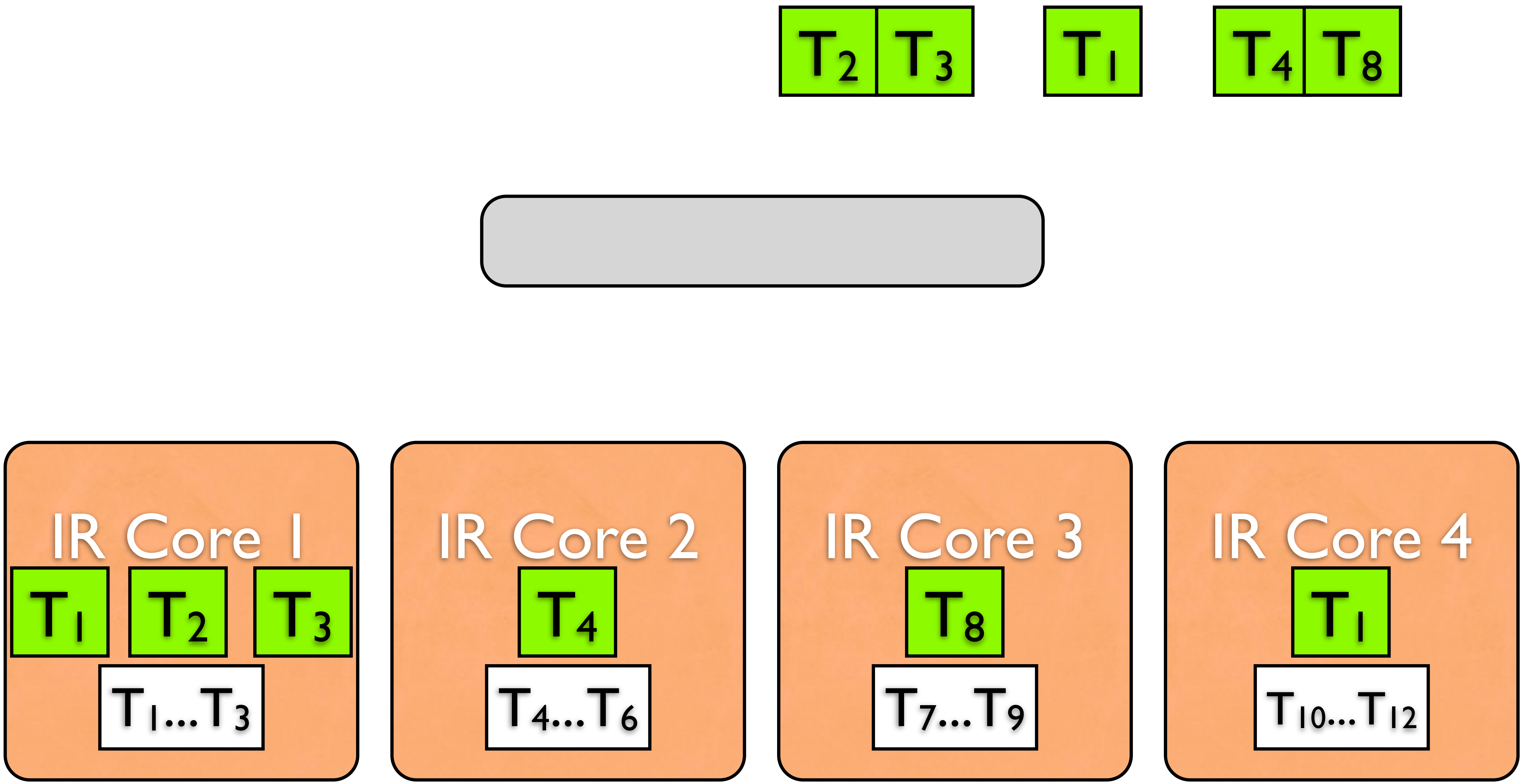


Term Partitioning

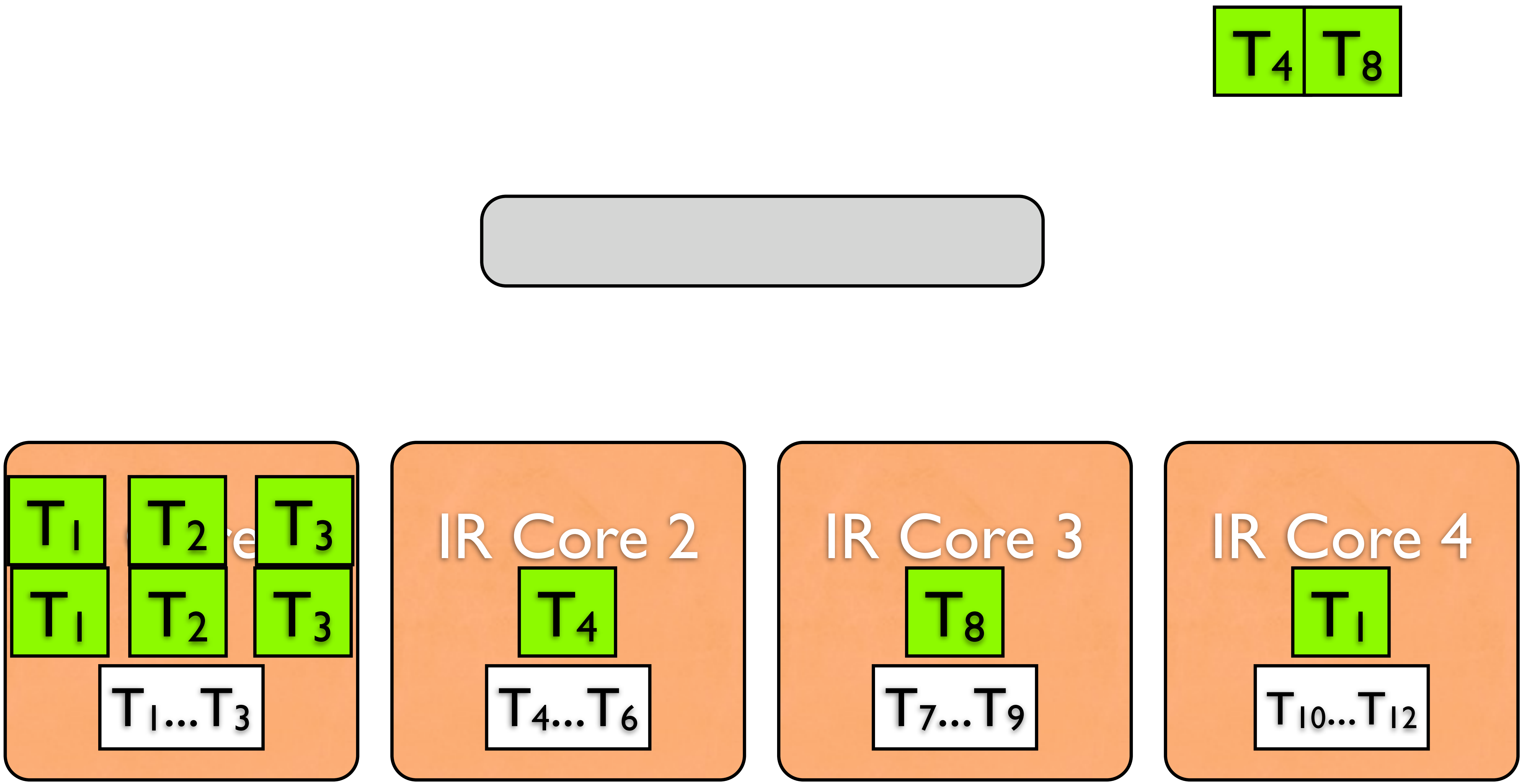
(Random Partitioning)



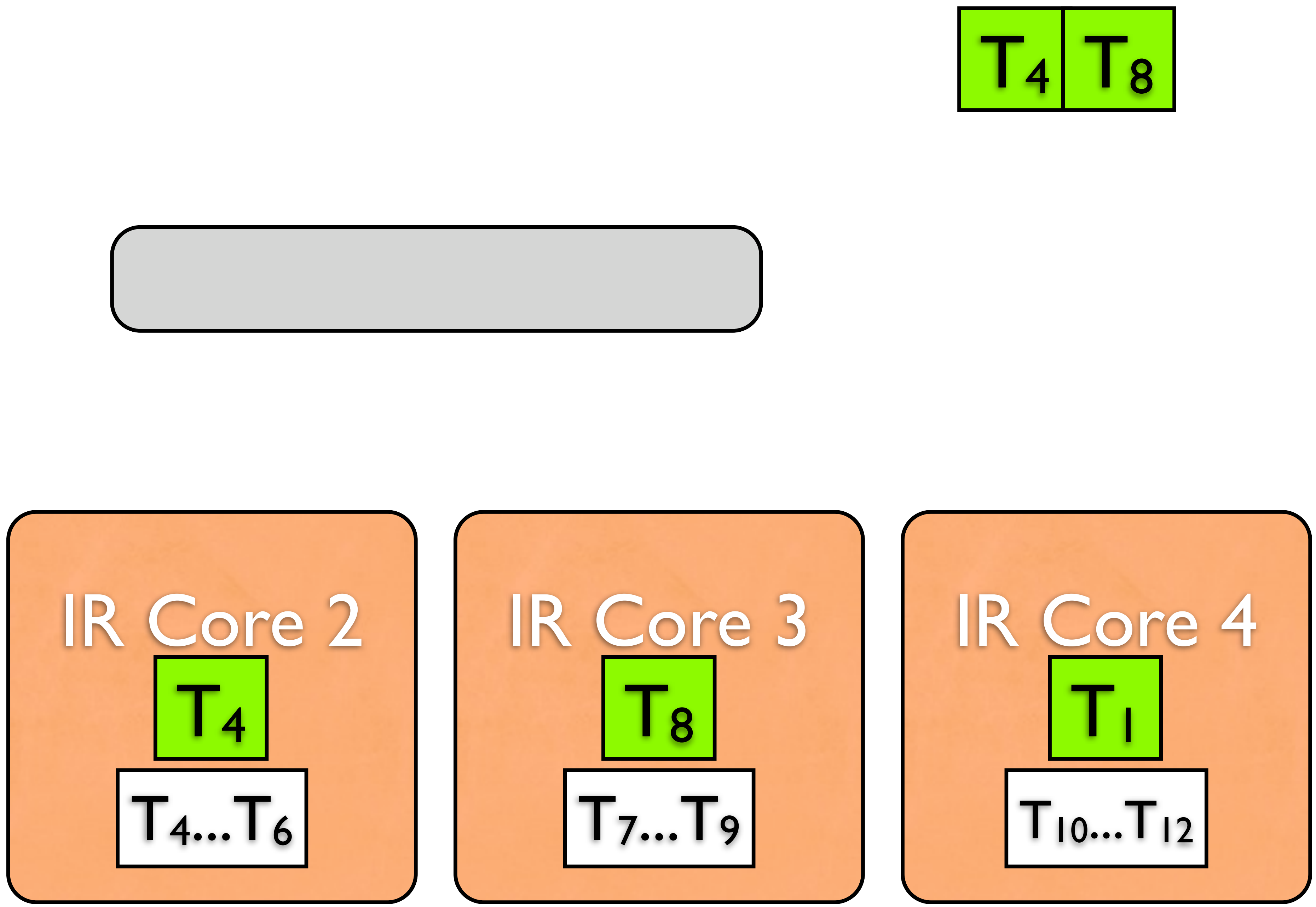
Term Partitioning
(Random Partitioning)



Term Partitioning
(Random Partitioning)

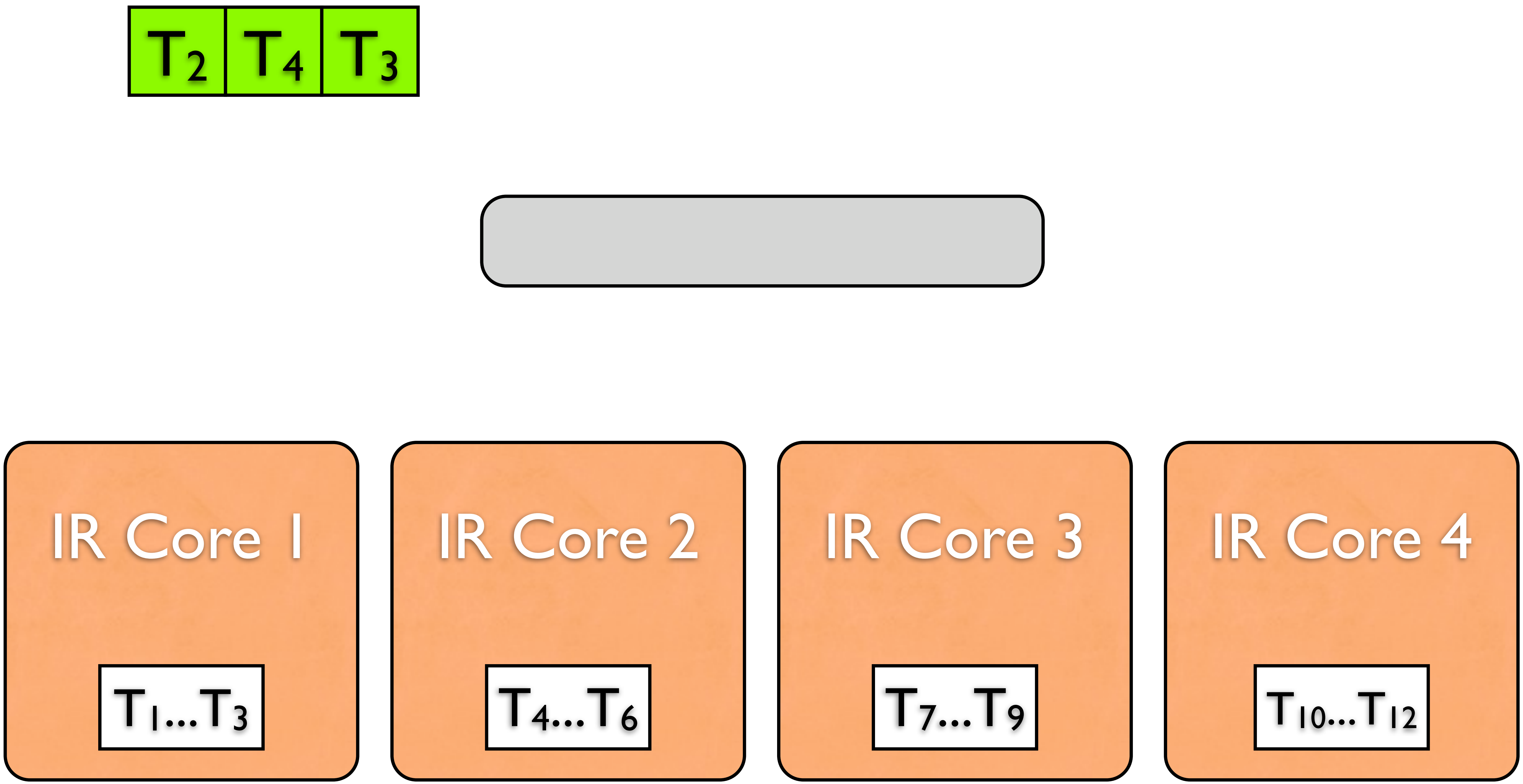


Term Partitioning
(Random Partitioning)



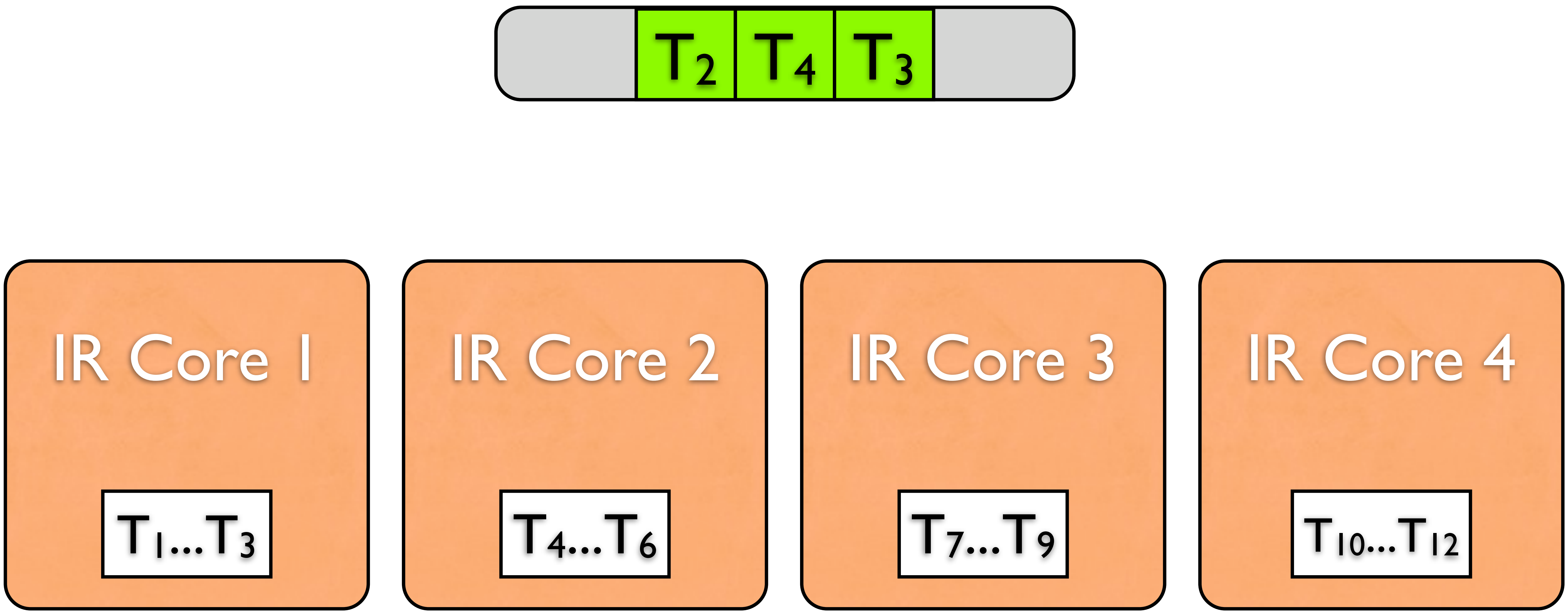
Pipelined Term Part.

(Random Partitioning)



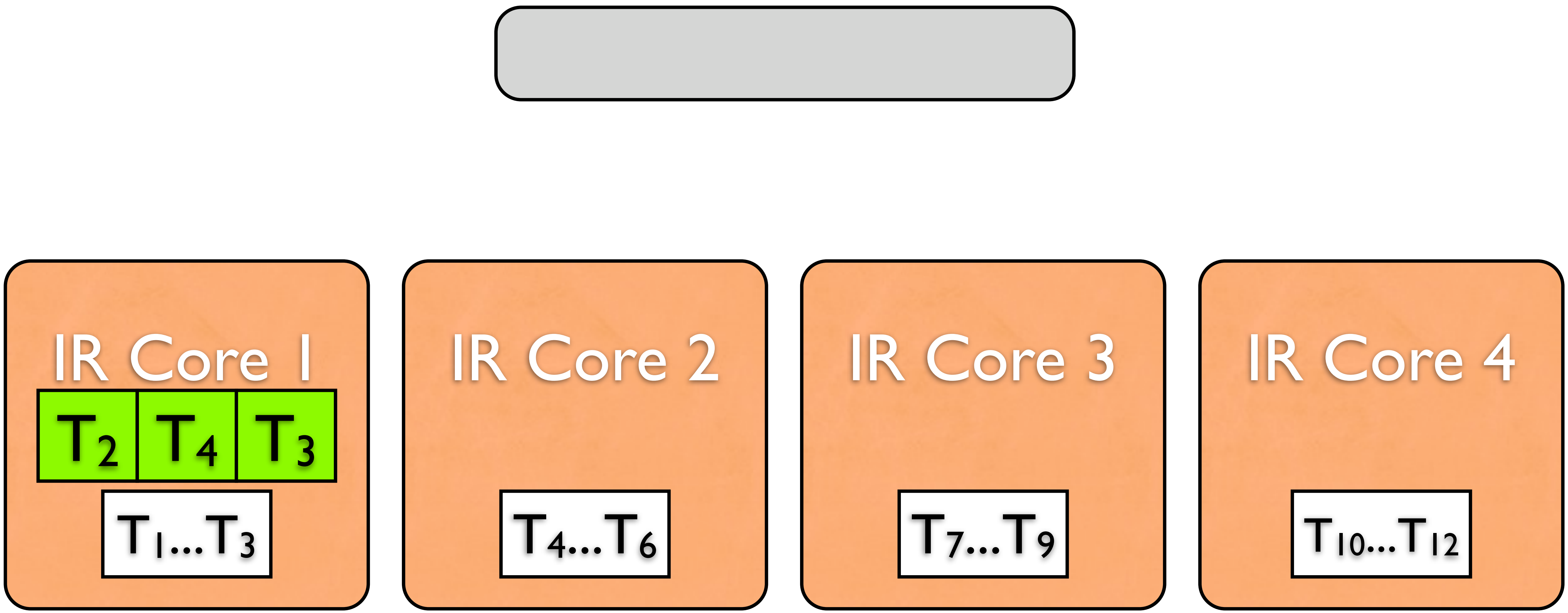
Pipelined Term Part.

(Random Partitioning)



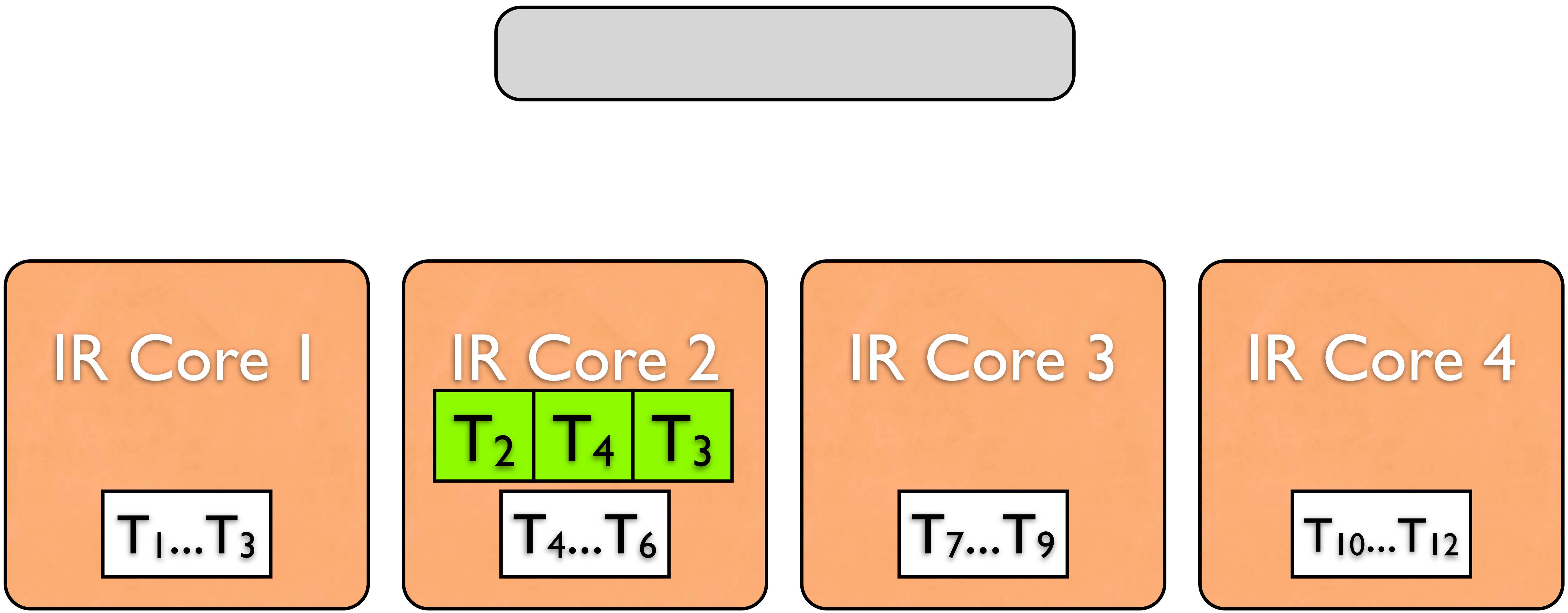
Pipelined Term Part.

(Random Partitioning)



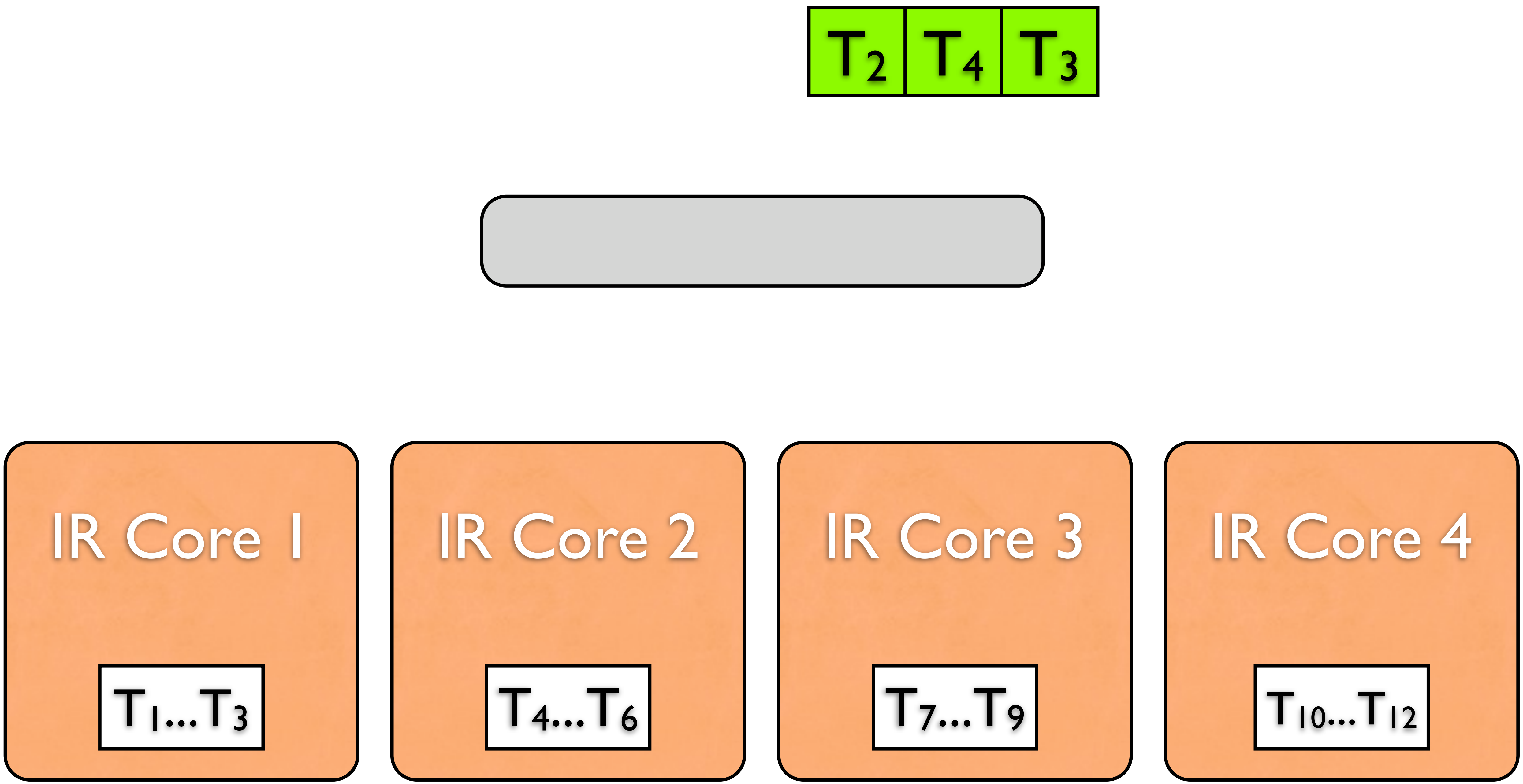
Pipelined Term Part.

(Random Partitioning)



Pipelined Term Part.

(Random Partitioning)

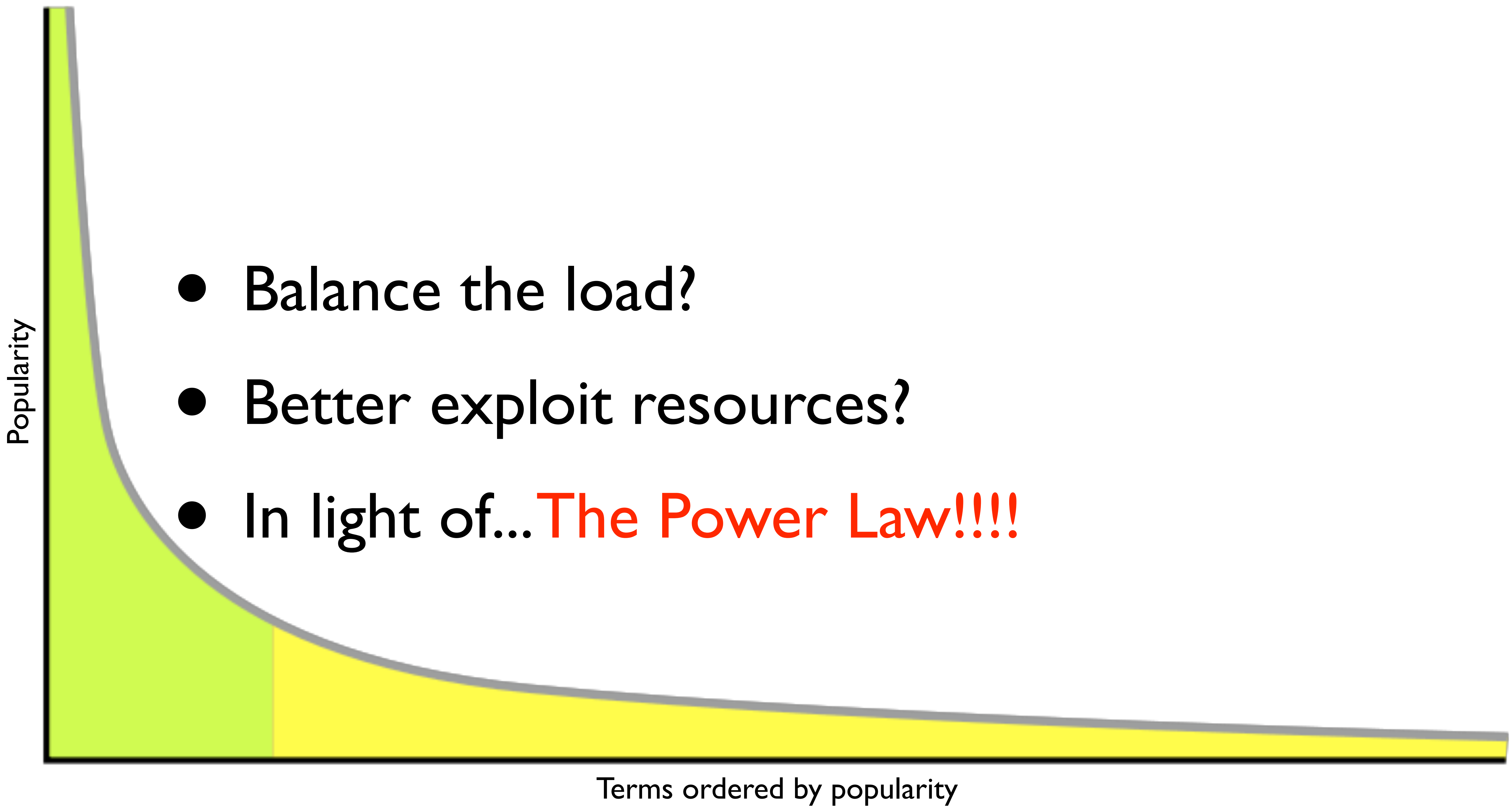


How can we...

- Balance the load?
- Better exploit resources?
- In light of...

How can we...

- Balance the load?
- Better exploit resources?
- In light of... The Power Law!!!!



Two Approaches in Literature

- Moffat, A., Webber, W., and Zobel, J. 2006. **Load balancing for term-distributed parallel retrieval**. In Proceedings of SIGIR 2006. Seattle, Washington, USA, August 06 - 11, 2006.
- Lucchese, C., Orlando, S., Perego, R., and Silvestri, F. 2007. **Mining query logs to optimize index partitioning in parallel web search engines**. In Proceedings of Infoscale 2007. Suzhou, China, June 06 - 08, 2007.

The Idea...

- If terms co-occur frequently in past queries...
  - pack them together up in the same IR Core.
- but...
  - Power law prevents load balancing...
    - Knapsack problem can help in balancing the load.
      - Fit in partitions terms with weight $L_t=Q_t x B_t$ (Q_t occurrences of t in the query log, B_t length of t's postings list)

Load Balancing Results

- On .GOV2 collection. Queries adapted from WT10G.

Strategy	Batch 2	3	4	5	6	Avg
Random	1.45	1.44	1.46	1.50	1.48	1.47
Using f_t	1.43	1.20	1.23	1.40	1.42	1.34
Past L_t	1.14	1.26	1.23	1.19	1.17	1.20
Current L_t	1.00	1.00	1.00	1.00	1.00	1.00

Moffat, A., Webber, W., and Zobel, J. 2006. **Load balancing for term-distributed parallel retrieval.** In Proceedings of SIGIR 2006. Seattle, Washington, USA, August 06 - 11, 2006.

Load Balancing Results

- On .GOV2 collection. Queries adapted from WT10G.

Not using query frequency.

Strategy	Batch					Avg
	2	3	4	5	6	
Random	1.45	1.44	1.46	1.50	1.48	1.47
Using f_t	1.43	1.20	1.23	1.40	1.42	1.34
Past L_t	1.14	1.26	1.23	1.19	1.17	1.20
Current L_t	1.00	1.00	1.00	1.00	1.00	1.00

Moffat, A., Webber, W., and Zobel, J. 2006. **Load balancing for term-distributed parallel retrieval.** In Proceedings of SIGIR 2006. Seattle, Washington, USA, August 06 - 11, 2006.

Load Balancing Results

- On .GOV2 collection. Queries adapted from WT10G.

Not using query frequency.

Strategy	Batch 2	Batch 3	Batch 4	Batch 5	Batch 6	Avg
Random	1.45	1.44	1.46	1.50	1.48	1.47
Using f_t	1.43	1.20	1.23	1.40	1.42	1.34
Past L_t	1.14	1.26	1.23	1.19	[illegible]	1.20
Current L_t	1.00	1.00	1.00	[illegible]	[illegible]	[illegible]

L_t is given by an oracle

Moffat, A., Webber, W., and Zobel, J. 2006. **Load balancing for term-distributed parallel retrieval.** In Proceedings of SIGIR 2006. Seattle, Washington, USA, August 06 - 11, 2006.

Load Balancing + Replication Results

- On .GOV2 collection. Queries adapted from WT10G.

Strategy	Batch 2	Batch 3	Batch 4	Batch 5	Batch 6	Avg
Duplicate 1	1.26	1.20	1.10	1.17	1.11	1.17
Duplicate 10	1.06	1.29	1.17	1.18	1.16	1.17
Duplicate 100	1.09	1.14	1.10	1.13	1.15	1.12
Duplicate 1000	1.08	1.09	1.07	1.19	1.09	1.10
Multi-replicate	1.05	1.12	1.09	1.16	1.12	1.11

Moffat, A., Webber, W., and Zobel, J. 2006. **Load balancing for term-distributed parallel retrieval.** In Proceedings of SIGIR 2006. Seattle, Washington, USA, August 06 - 11, 2006.

But...

- Query Throughput.
- On .GOV2 collection. Queries from a real life MSN query log

Strategy	Batch					Avg
	2	3	4	5	6	
Hashed	1.82	1.82	1.86	1.84	1.83	1.83
Duplicate 100	2.21	2.14	2.25	2.17	2.20	2.19
Doc-distributed	2.21	2.25	2.24	2.31	2.27	2.26

Moffat, A., Webber, W., and Zobel, J. 2006. **Load balancing for term-distributed parallel retrieval.** In Proceedings of SIGIR 2006. Seattle, Washington, USA, August 06 - 11, 2006.

But...

- Query Throughput.
- On .GOV2 collection. Queries from a real life MSN query log

Hashed
~
Random

Strategy	Batch 2	3	4	5	6	Avg
Hashed	1.82	1.82	1.86	1.84	1.83	1.83
Duplicate 100	2.21	2.14	2.25	2.17	2.20	2.19
Doc-distributed	2.21	2.25	2.24	2.31	2.27	2.26

Moffat, A., Webber, W., and Zobel, J. 2006. **Load balancing for term-distributed parallel retrieval.** In Proceedings of SIGIR 2006. Seattle, Washington, USA, August 06 - 11, 2006.

Adding a Dimension

- Number of IR Cores used by each query
- Basically, instead of sending lists around try to keep many query processing local to a node.
  - More load unbalance
- Trade-off: The Term-Assignment Problem

Lucchese, C., Orlando, S., Perego, R., and Silvestri, F. 2007. **Mining query logs to optimize index partitioning in parallel web search engines**. In Proceedings of Infoscale 2007. Suzhou, China, June 06 - 08, 2007.

The Term-Assignment Problem

The Term-Assignment Problem. *Given a weight* α, $0 \leq \alpha \leq 1$, *a query stream* Φ, *the Term-Assignment Problem asks for finding the partitioning* λ *which minimizes*

$$\Omega_\lambda(\Phi) = \alpha \cdot \frac{\sum_{Q \in \Phi} |H_\lambda(Q)|}{N_\omega} + (1-\alpha) \cdot \frac{\widehat{L}_\lambda(\Phi)}{N_L}$$

where N_ω *and* N_L *are normalization constants.*

Lucchese, C., Orlando, S., Perego, R., and Silvestri, F. 2007. **Mining query logs to optimize index partitioning in parallel web search engines**. In Proceedings of Infoscale 2007. Suzhou, China, June 06 - 08, 2007.

The Term-Assignment Problem

The Term-As ... $\leq \alpha \leq 1$,
a query strea ... *r finding*
the partitionir

Ω_λ (Φ ...)

where N_ω *and*

Use a Frequent Itemsets Mining Algorithm to find pairs of co-occurring terms. Then, optimize Omega not only considering single terms but also term-sets.

Lucchese, C., Orlando, S., Perego, R., and Silvestri, F. 2007. **Mining query logs to optimize index partitioning in parallel web search engines**. In Proceedings of Infoscale 2007. Suzhou, China, June 06 - 08, 2007.

Number of Queried IR Cores (Servers)

Servers	Baseline Cases random	Baseline Cases bin packing	Term Assignment $\alpha = 0.9$
$\Phi_{test} = TodoBR$			
1	28	28	50
2	31	30	20
3	17	17	14
> 3	24	25	16
$\Phi_{test} = AltaVista$			
1	29	29	41
2	39	39	38
3	21	21	16
> 3	11	11	5

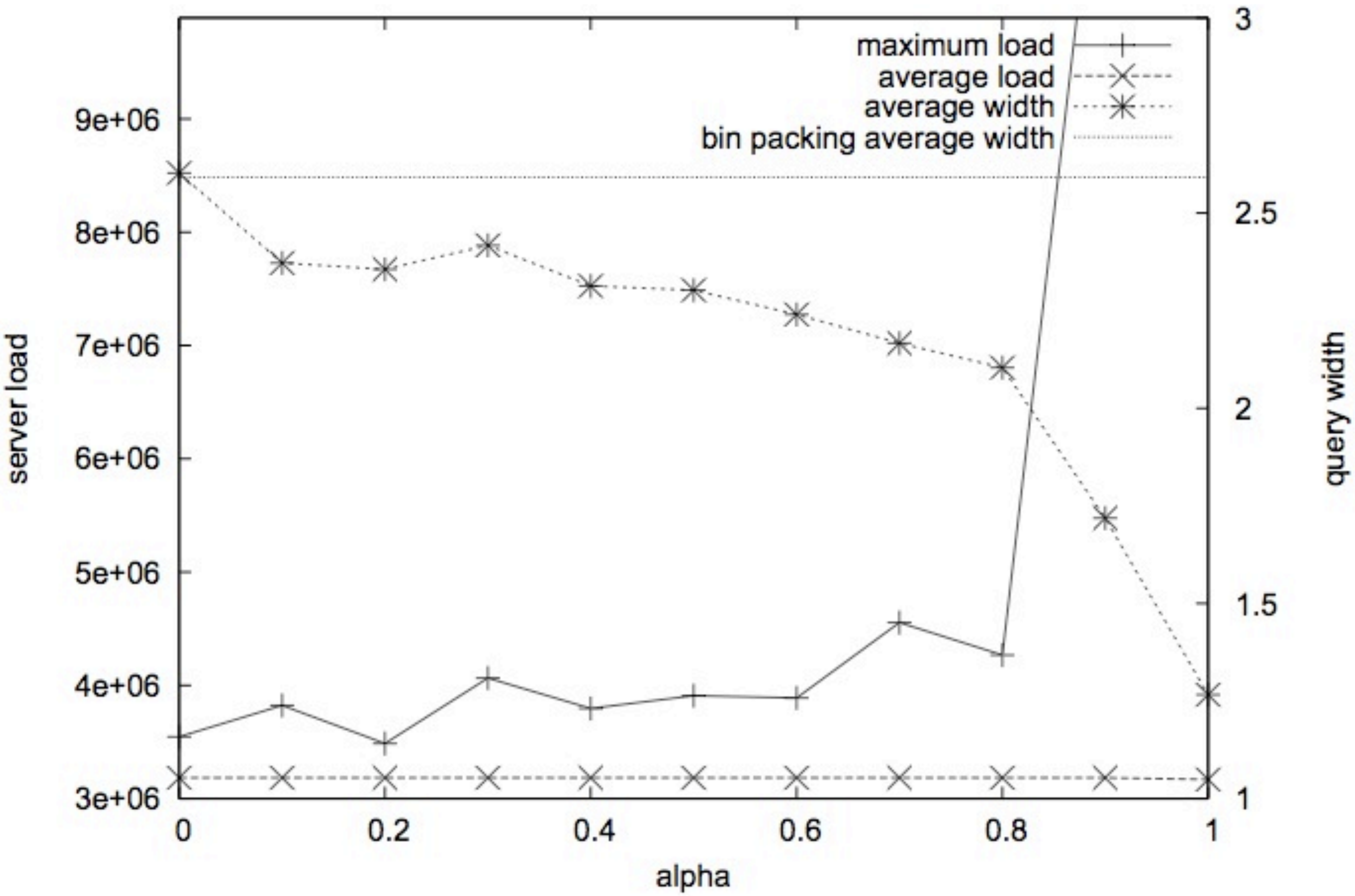
Lucchese, C., Orlando, S., Perego, R., and Silvestri, F. 2007. **Mining query logs to optimize index partitioning in parallel web search engines**. In Proceedings of Infoscale 2007. Suzhou, China, June 06 - 08, 2007.

Impact of Replications on the # of IR Cores

	Replication Factors					
	0.0001		0.0005		0.001	
Servers	*bin pack.*	*term ass.*	*bin pack.*	*term ass.*	*bin pack.*	*term ass.*
TodoBR						
1	42	54	56	62	63	67
2	31	22	22	18	19	16
3	12	10	9	8	8	8
> 3	15	14	12	11	10	9

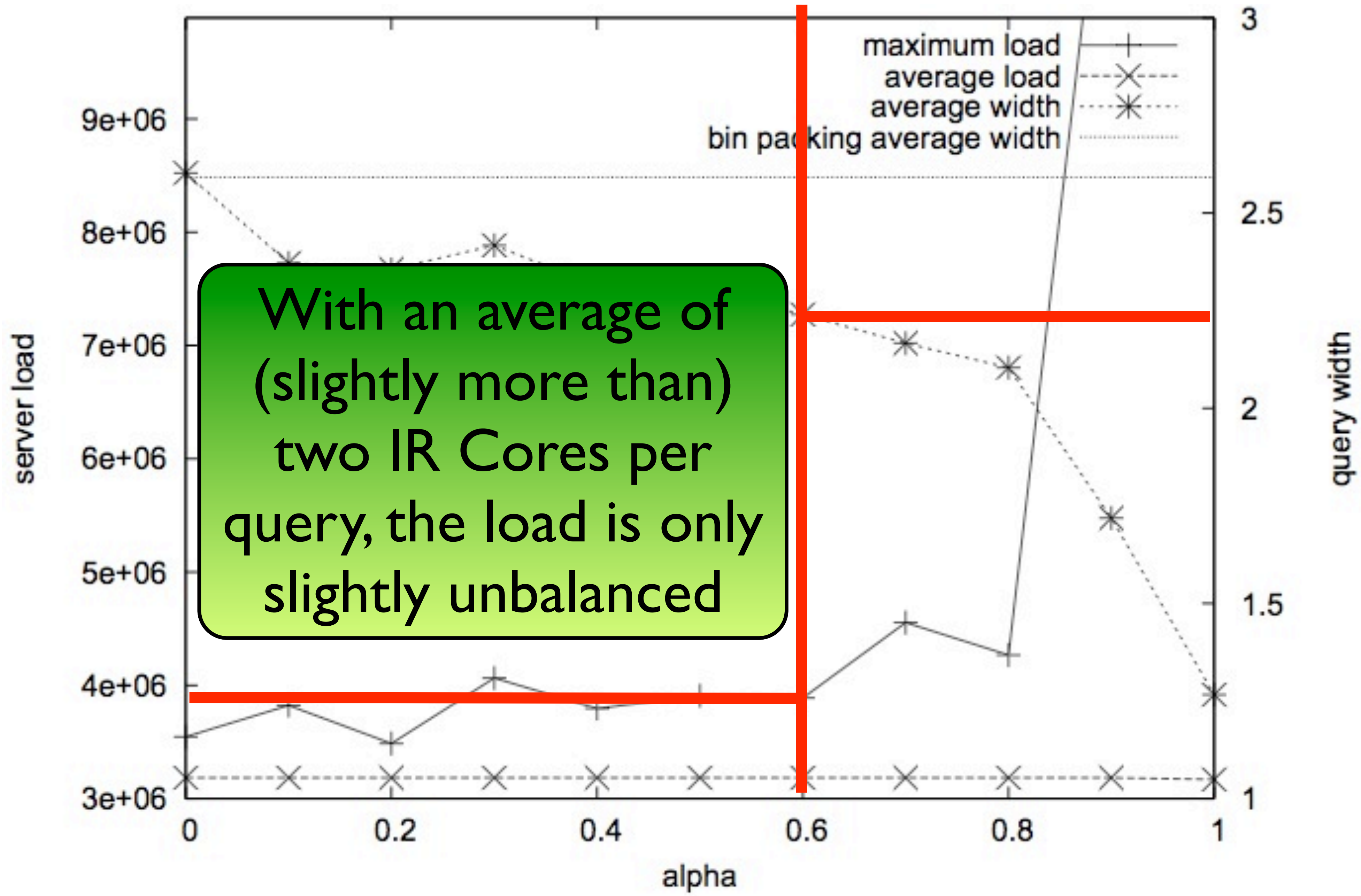
Lucchese, C., Orlando, S., Perego, R., and Silvestri, F. 2007. **Mining query logs to optimize index partitioning in parallel web search engines**. In Proceedings of Infoscale 2007. Suzhou, China, June 06 - 08, 2007.

The Overall Picture



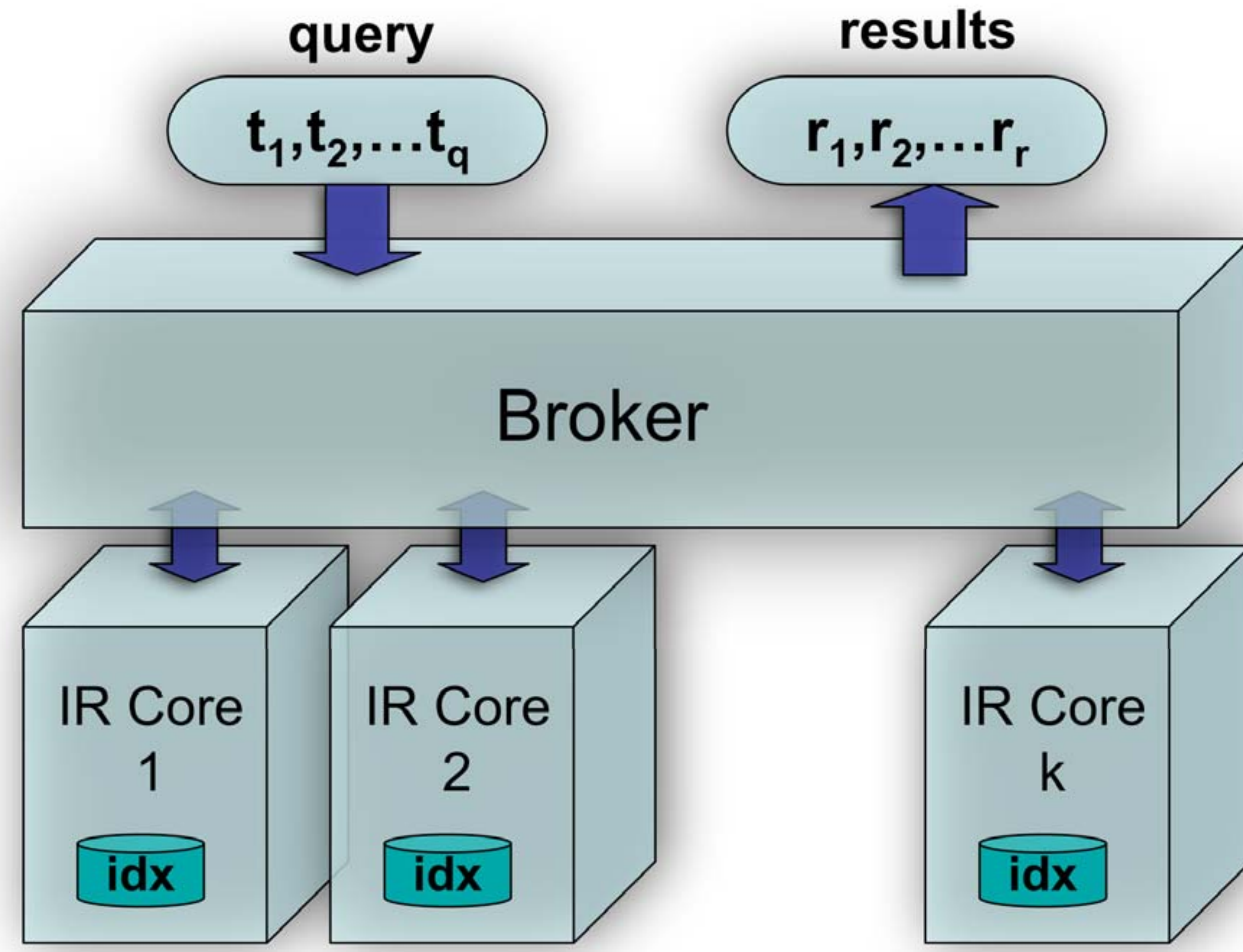
Lucchese, C., Orlando, S., Perego, R., and Silvestri, F. 2007. **Mining query logs to optimize index partitioning in parallel web search engines**. In Proceedings of Infoscale 2007. Suzhou, China, June 06 - 08, 2007.

The Overall Picture

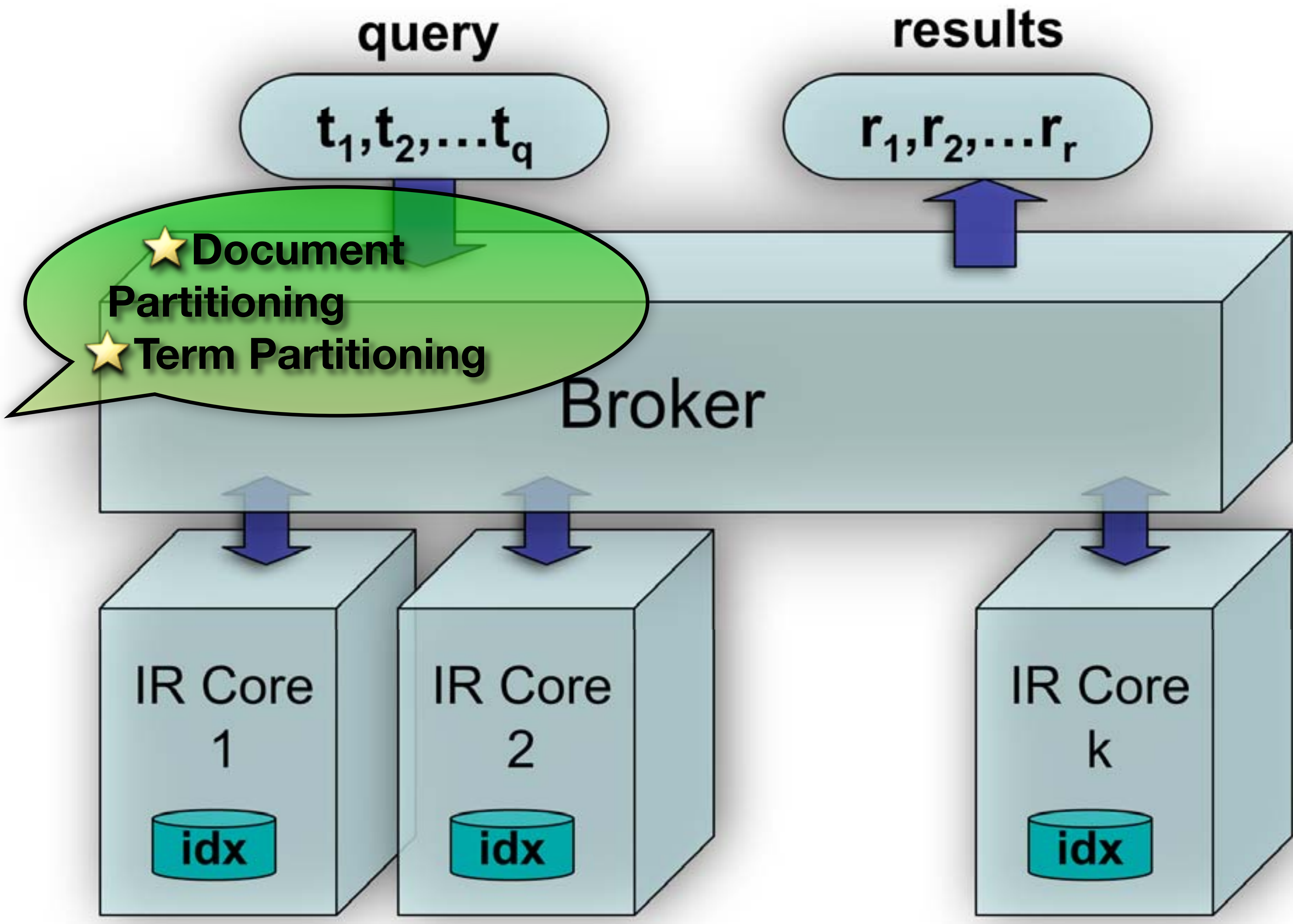


Lucchese, C., Orlando, S., Perego, R., and Silvestri, F. 2007. **Mining query logs to optimize index partitioning in parallel web search engines**. In Proceedings of Infoscale 2007. Suzhou, China, June 06 - 08, 2007.

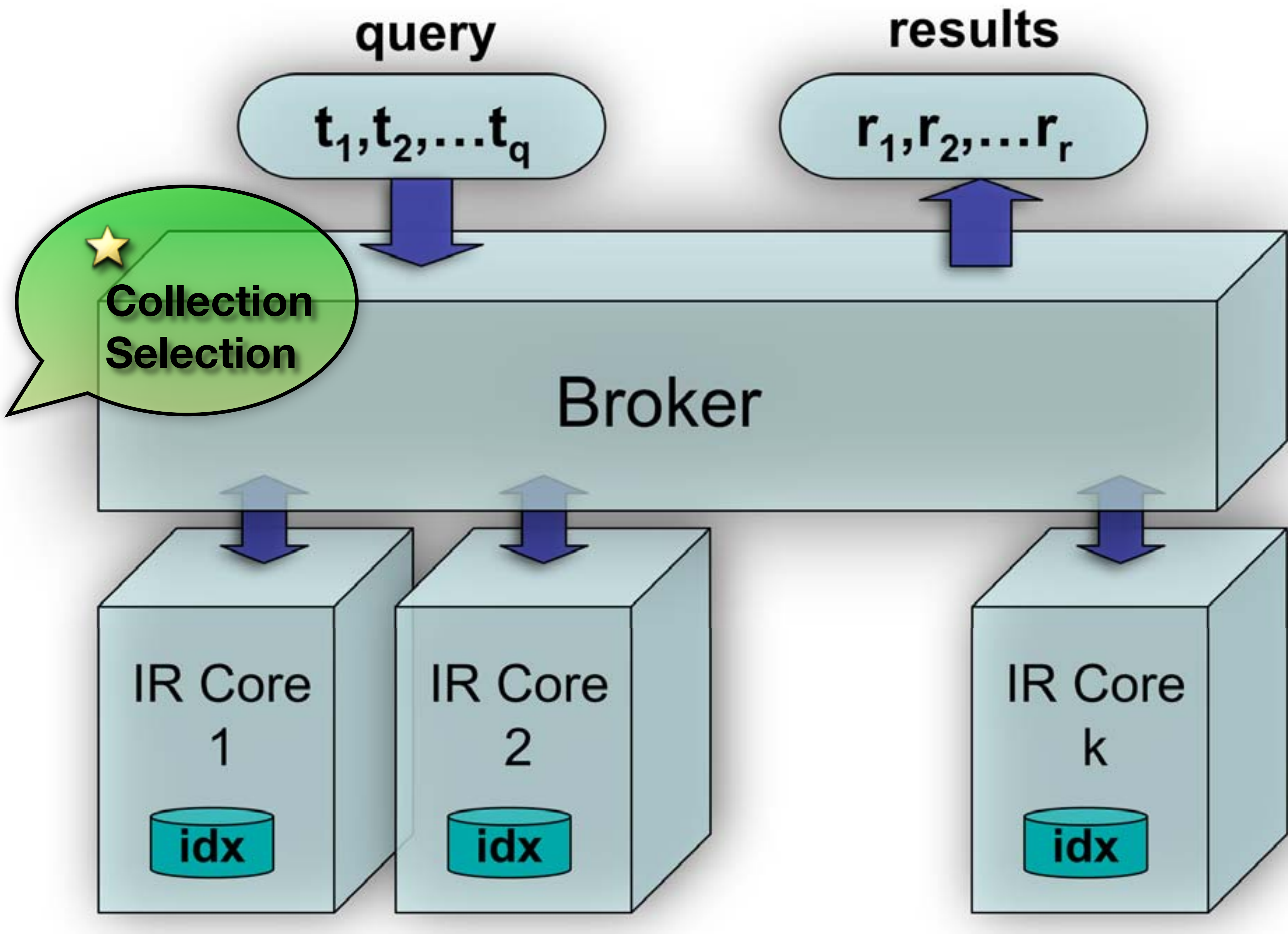
Document Partitioning



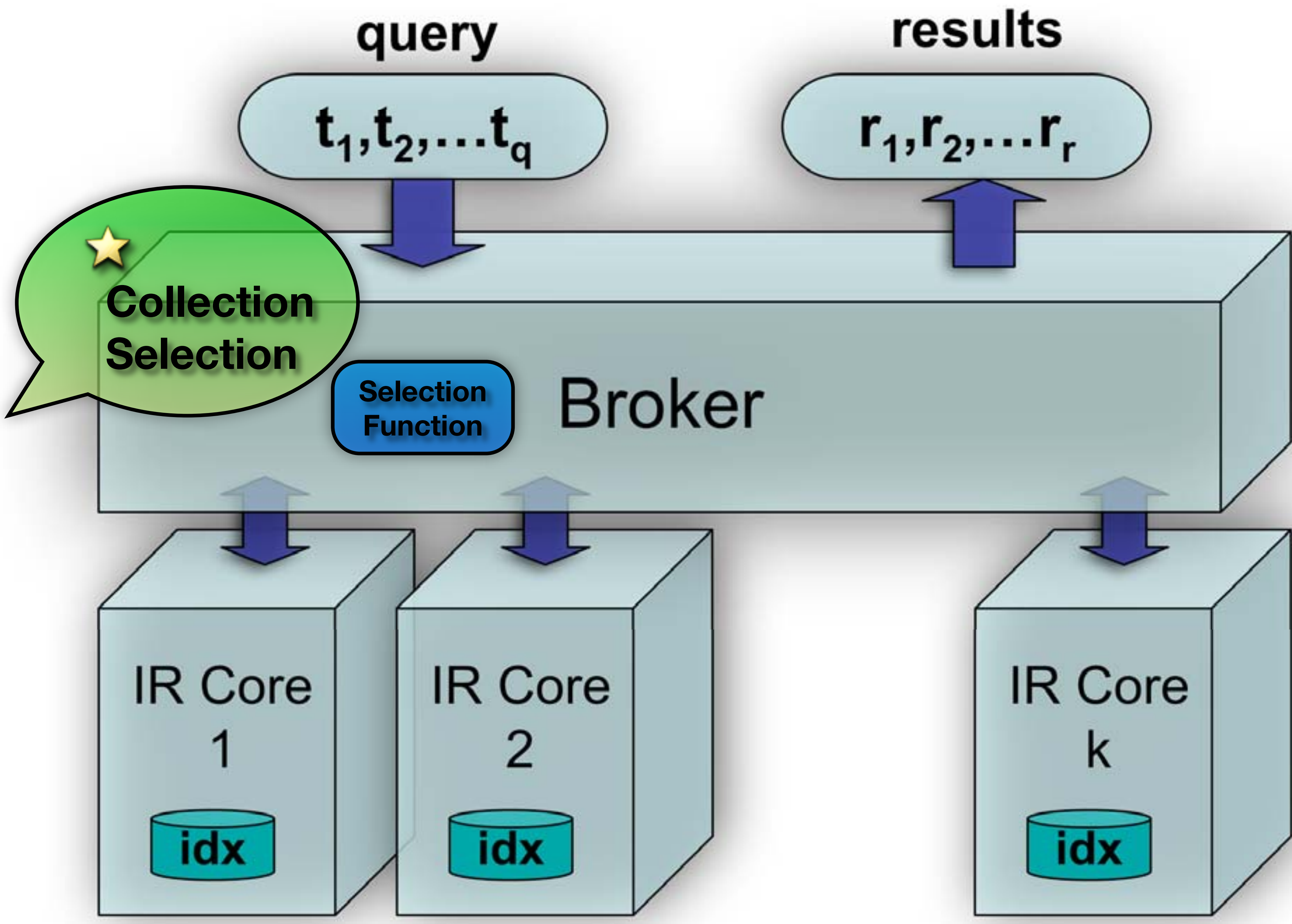
Document Partitioning



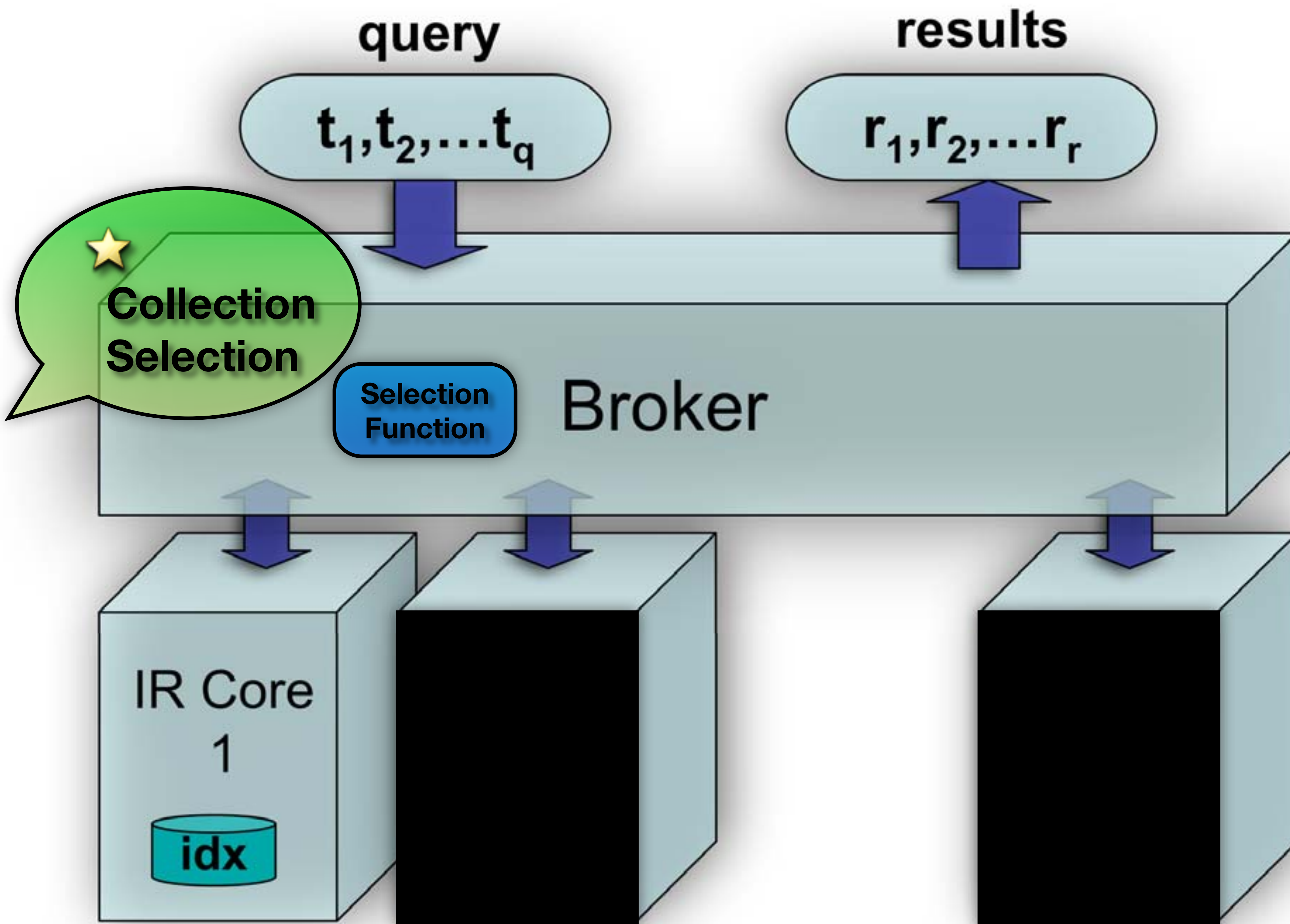
Document Partitioning



Document Partitioning



Document Partitioning



Collection Selection

- Traditionally used in Federated Distributed IR systems to reduce the number of queried servers.
- Rarely (???) used in Web Search Engine systems
  - see Google's MICRO paper on their architecture.

To Select or Not To Select?

- Pros
  - Reduced Load on IR Cores
  - Potentially Eliminates Noise due to the presence of non relevant documents w.r.t. a query
- Cons
  - Load Imbalance
  - Reduced Precision

The Curse of Reduced Precision

- The reduction in precision is an issue that have to be taken into serious consideration.
- "Luckily", precision in collection selection architectures can be enhanced by using:
  - Ad-hoc partitioning strategies
  - Collection Prioritization
  - Incremental Caching

Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Query-Vector Document Model

- It is a vector-space like model
- Documents are represented by queries they answer.
- Example.
  Query "Edvard Munch", "expressionist painters" are answered by documents d1, d2, d7, d3. Then documents d1, d2, d7, and d3 are represented by {"Munch", "Edward", "painters", "expressionist"}

Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

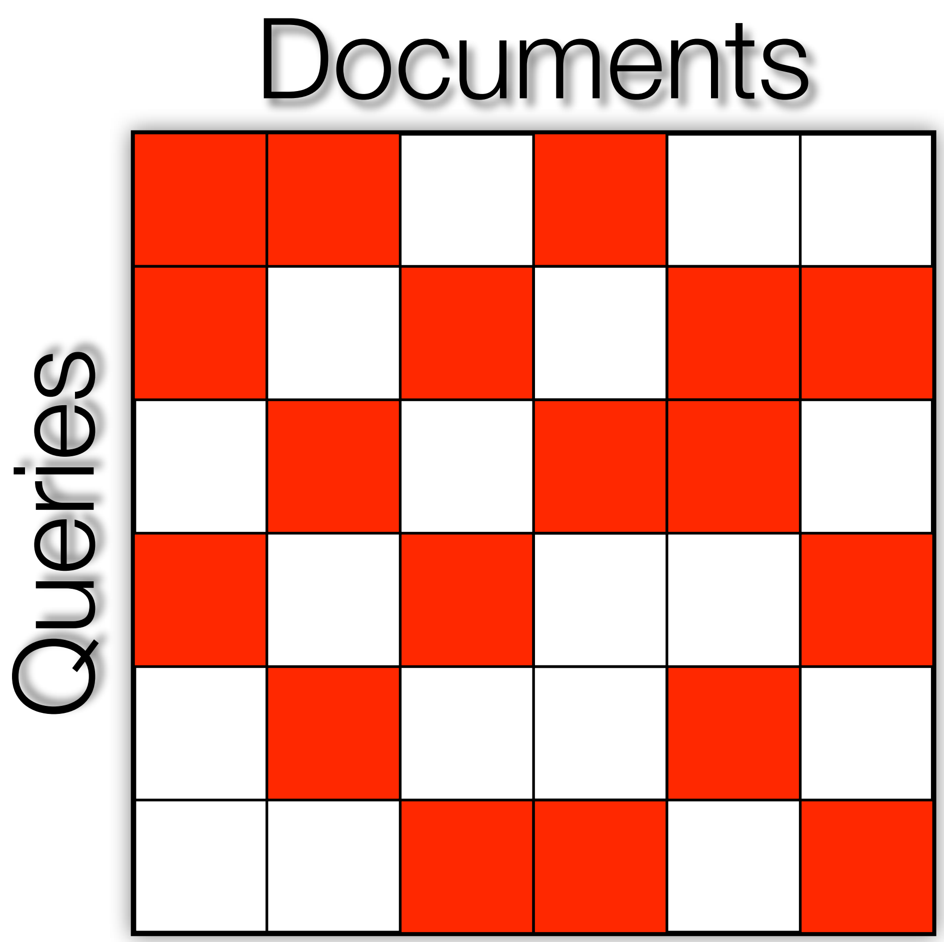
Query-Vector Document Model

- It is a vector-space like model
- Documents are represented by queries they answer.
- Example.
  Query "Edvard Munch", "
  are answered by docume
  documents d1, d2, d7, an
  {"Munch", "Edward", "painters", "expressionist"}

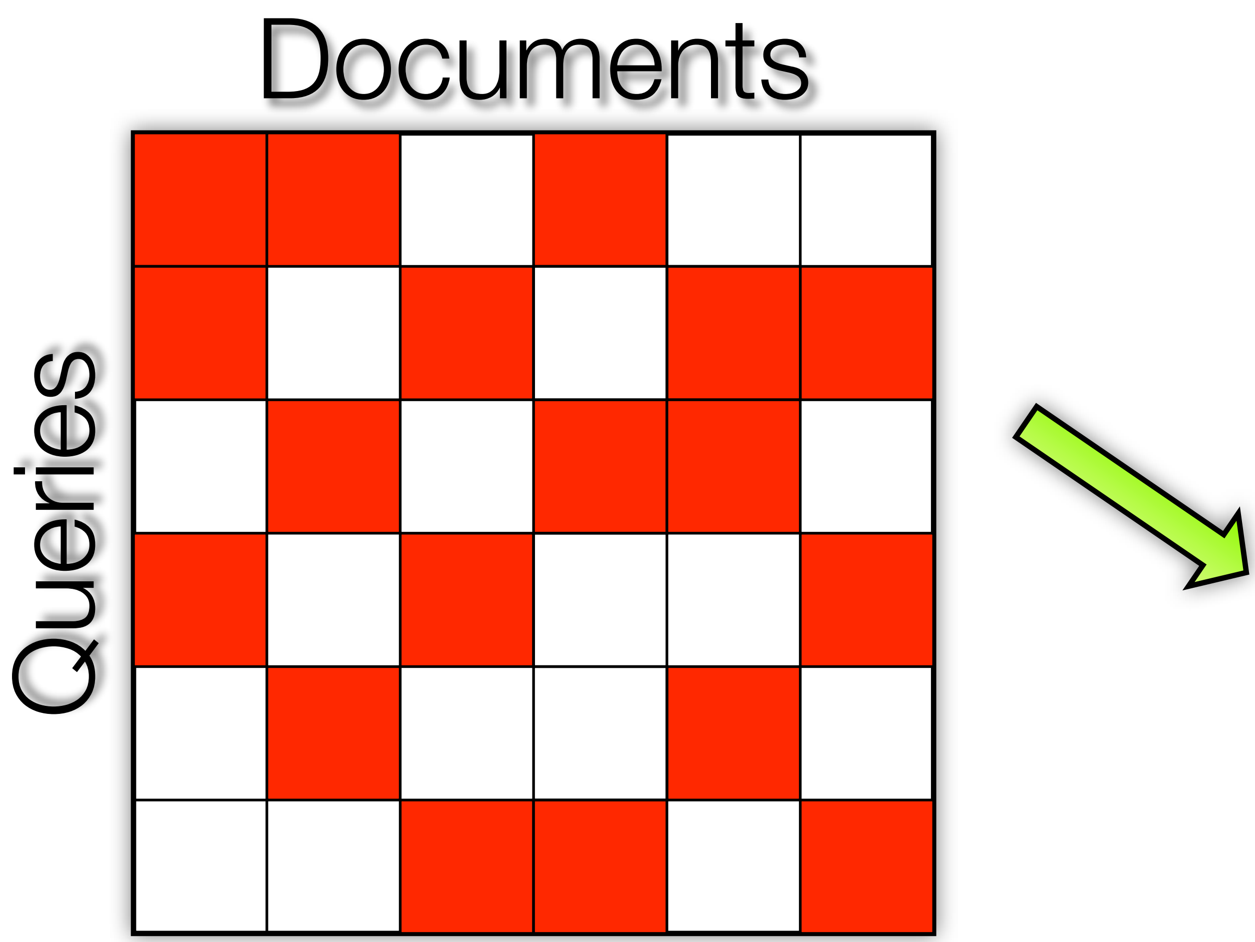
Possible Refinements:
- Consider Ranking Scores
- Consider Clicks

Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

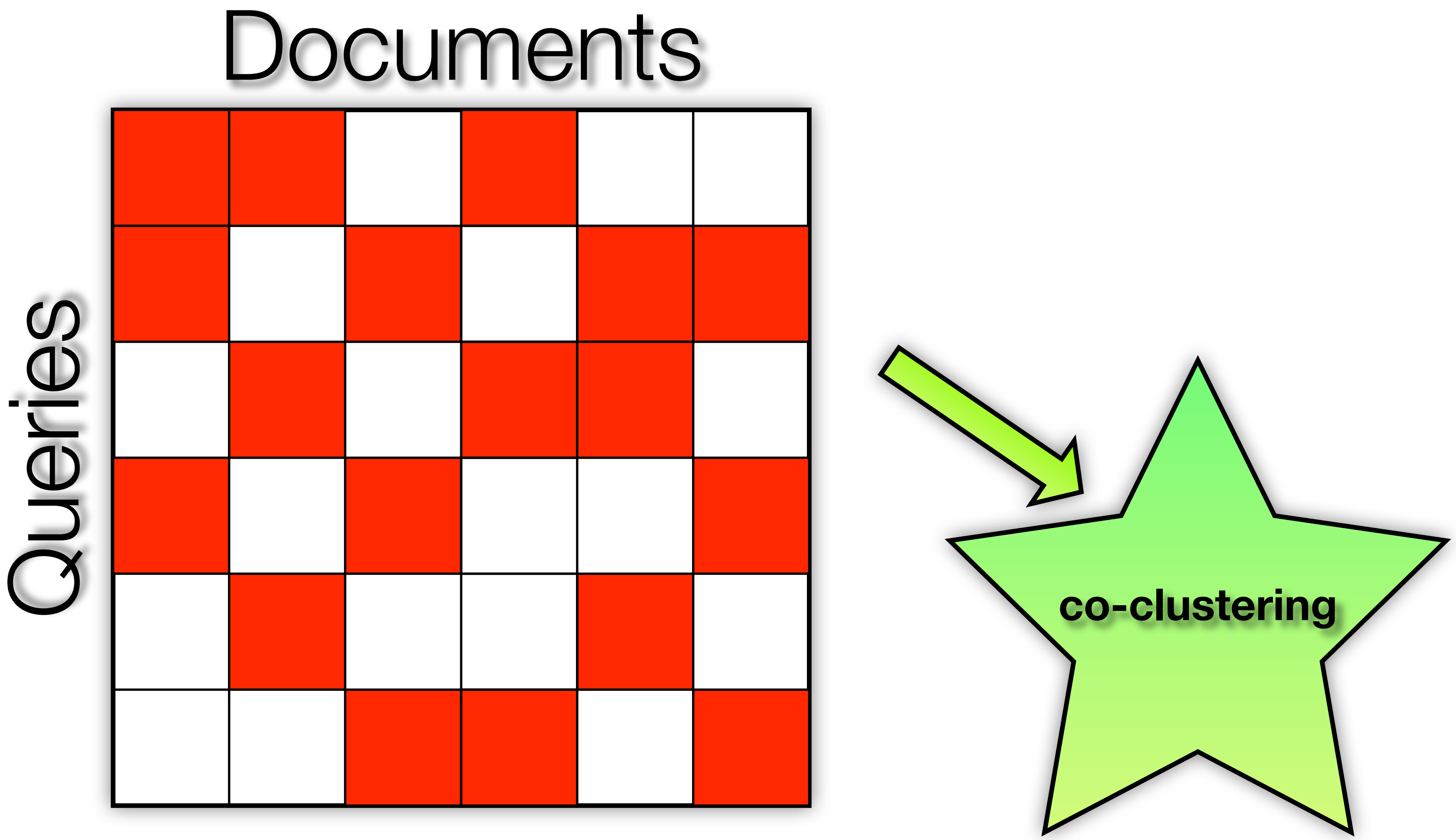
QV-Based Partitioning



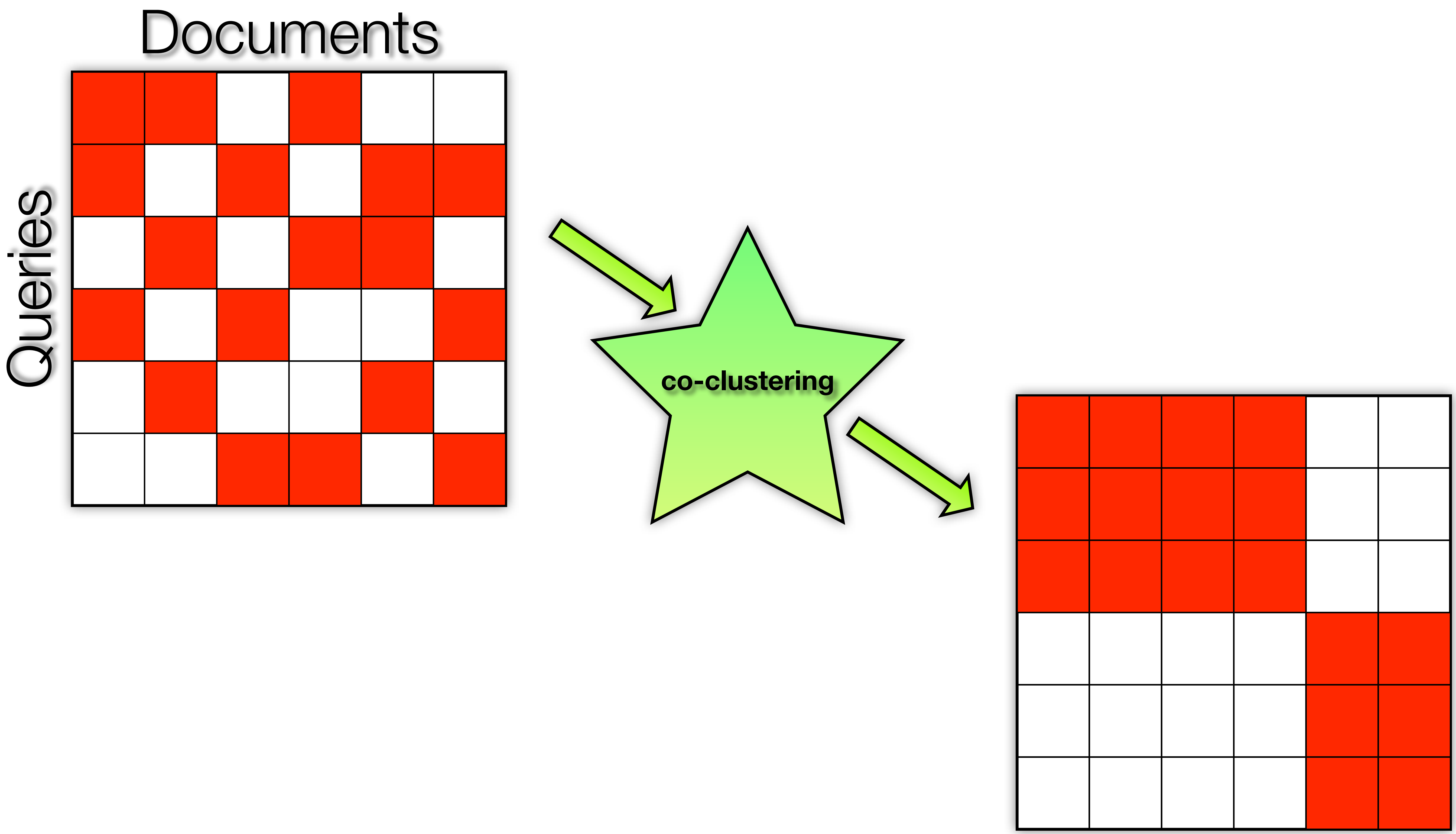
QV-Based Partitioning



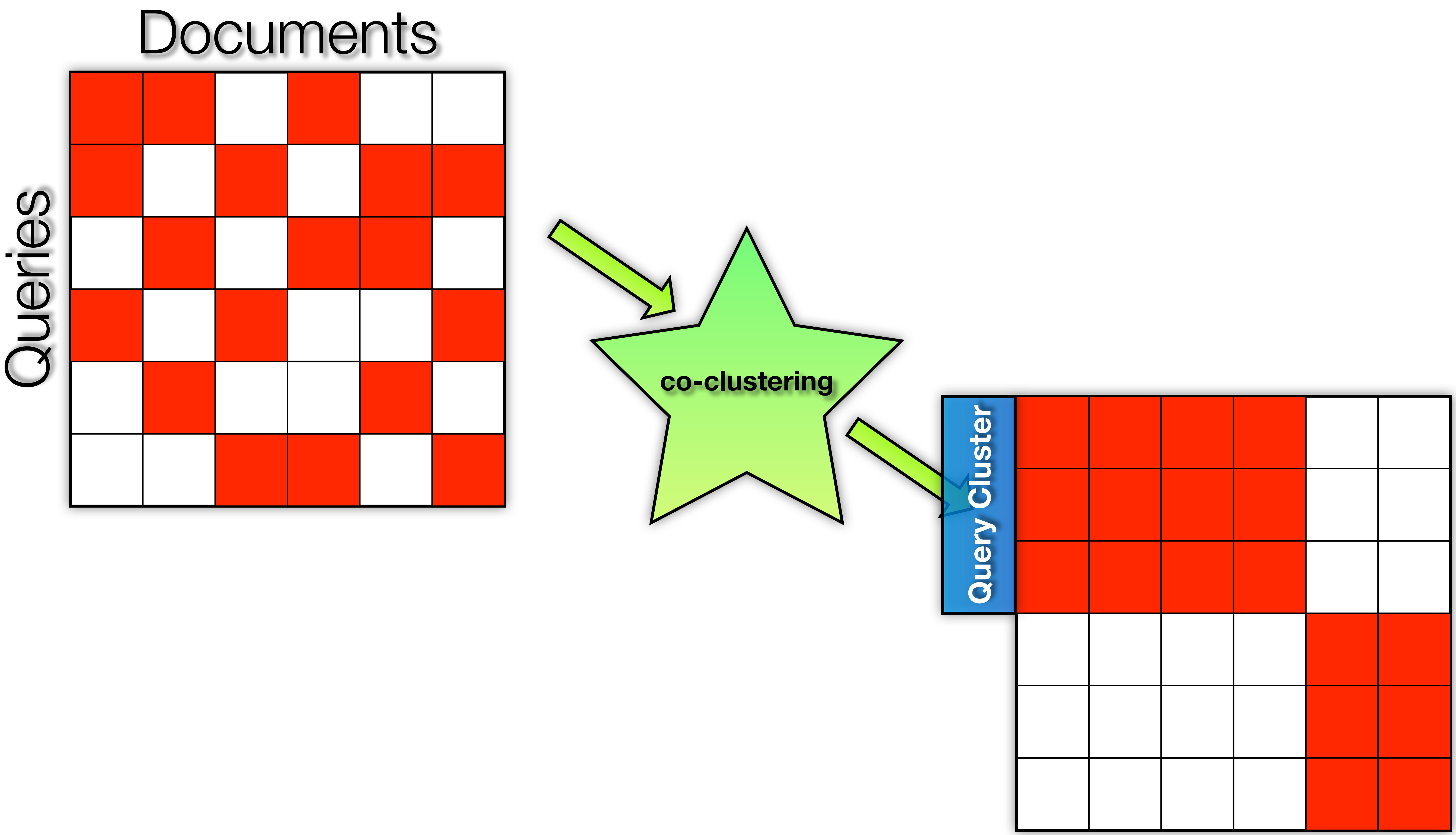
QV-Based Partitioning



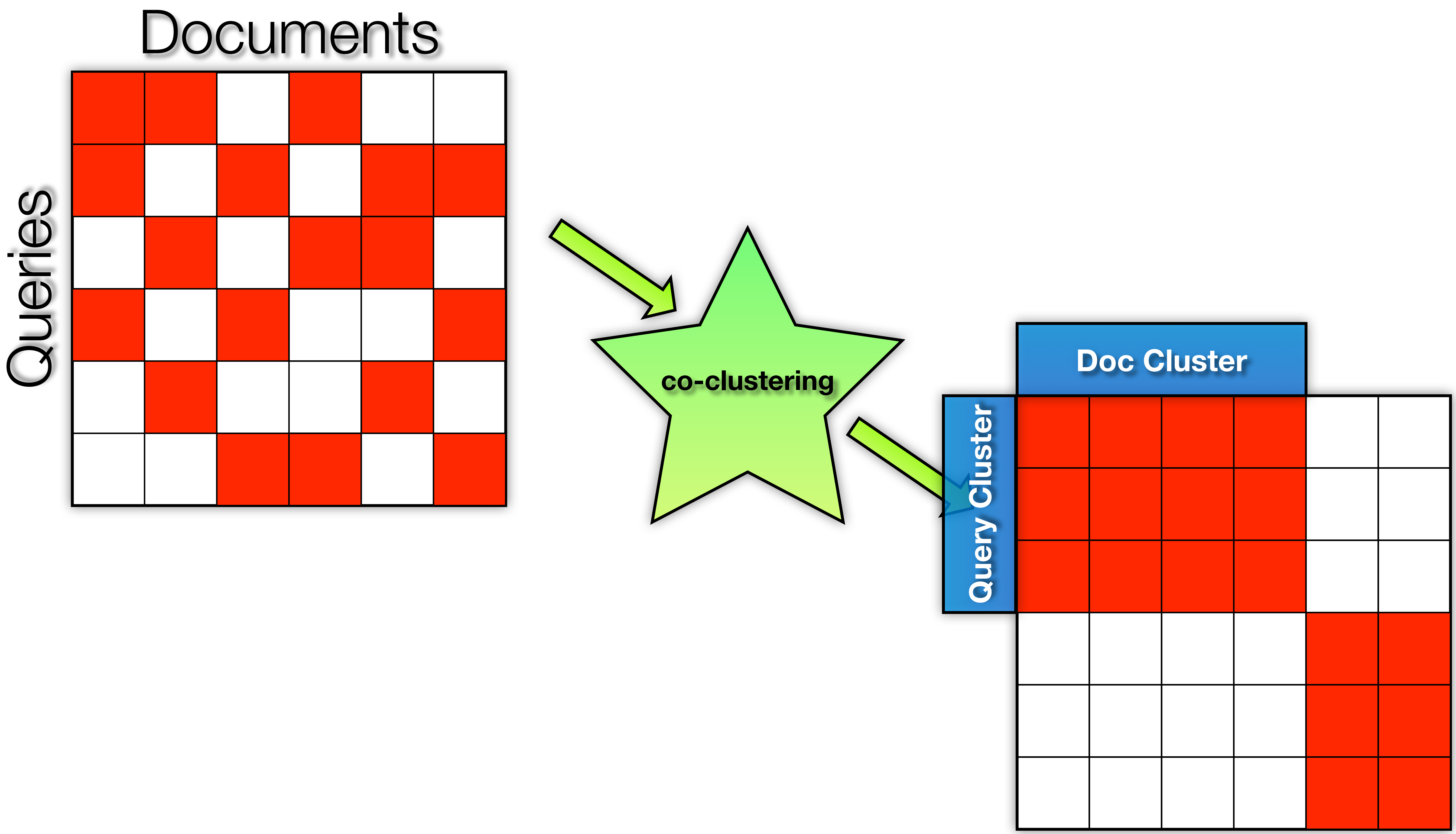
QV-Based Partitioning



QV-Based Partitioning

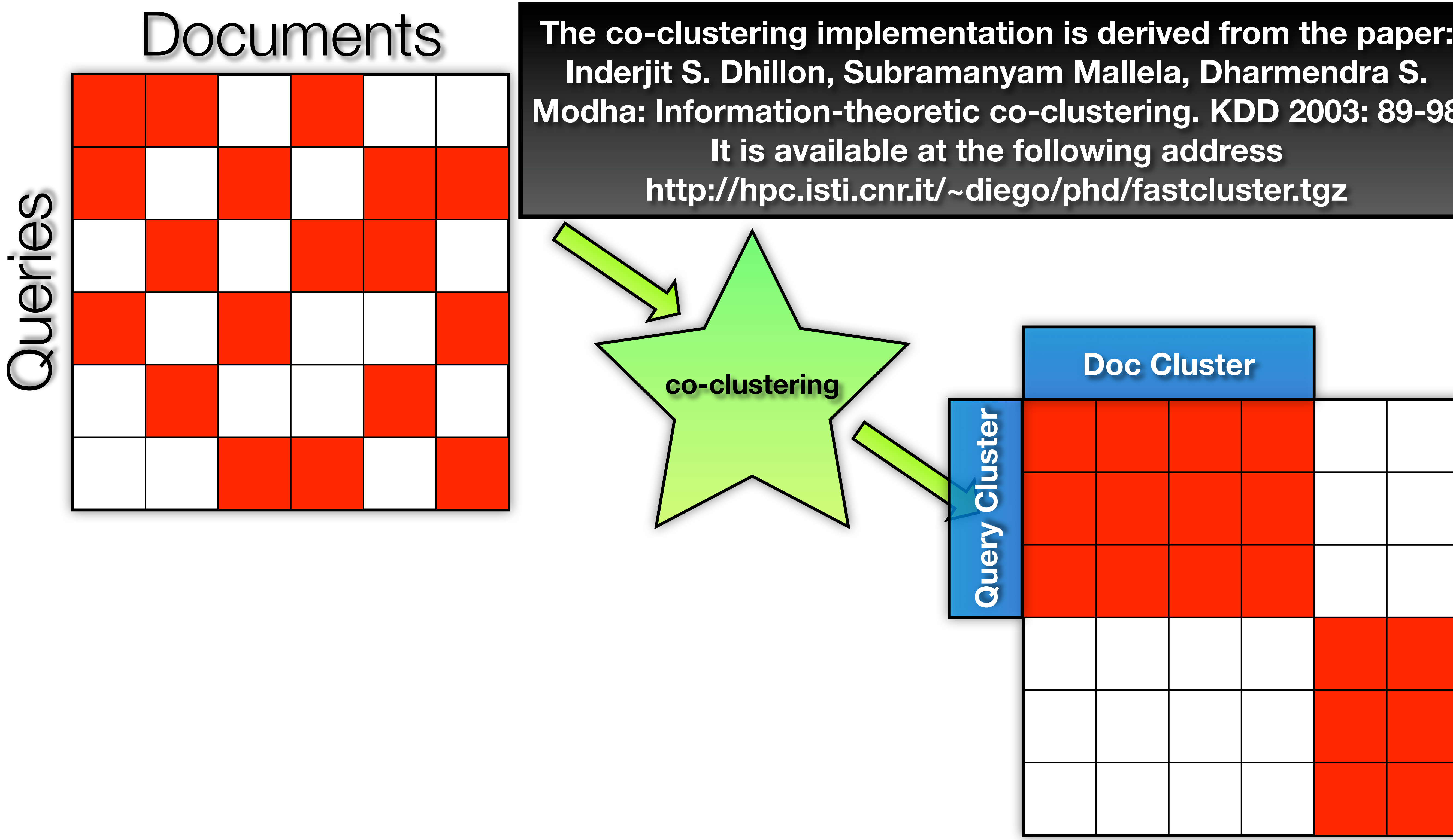


QV-Based Partitioning

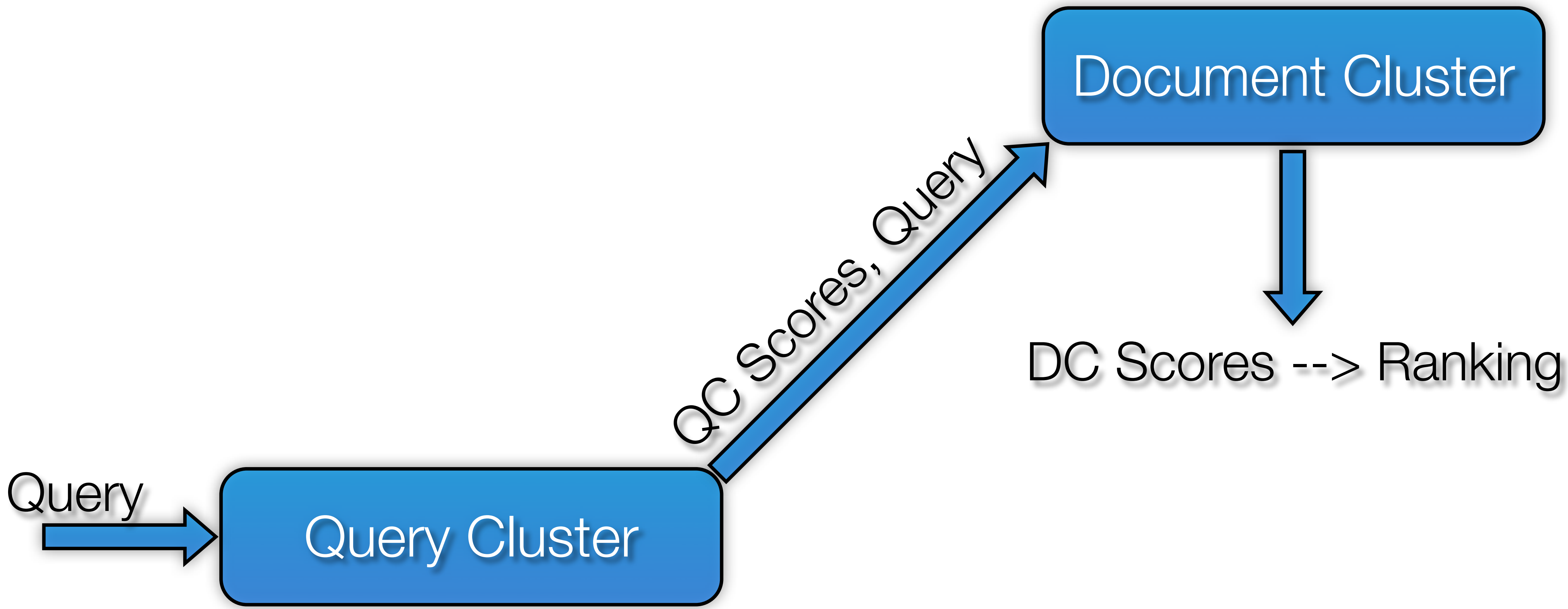


QV-Based Partitioning

The co-clustering implementation is derived from the paper: Inderjit S. Dhillon, Subramanyam Mallela, Dharmendra S. Modha: Information-theoretic co-clustering. KDD 2003: 89-98
It is available at the following address
http://hpc.isti.cnr.it/~diego/phd/fastcluster.tgz



QV-Based Collection Selection



Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Comparison with SoA- Collection Selection

- CORI
- does not make use of usage information.
- It exploits simple statistics on collections' vocabulary.
- At the Query Broker side, collection's information storing consumes quite a lot of memory.
  - QV representation for collection selection is about 19% of the size needed by CORI metadata.

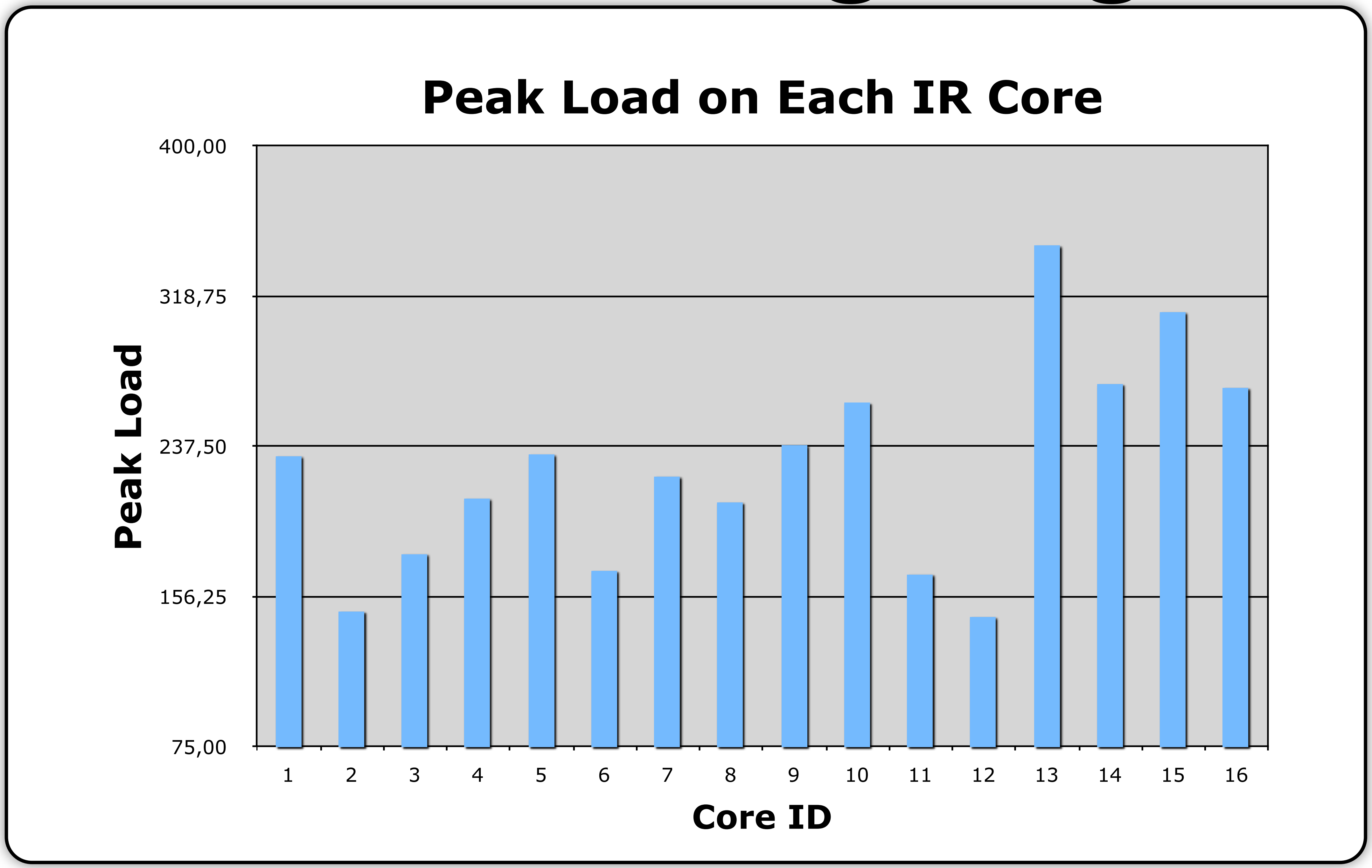
James P. Callan, Zhihong Lu, W. Bruce Croft. **Searching Distributed Collections with Inference Networks**. SIGIR 1995.

Comparison in Terms of Precise Results

Intersection (%) at 10	1	2	4	8	16	OVR
Random	5	11	25	50	93	100
QV-based	34	45	58	76	96	100
Improvement	**6.8X**	**4.1X**	**2.3X**	**1.5X**	**>1X**	**-**

Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

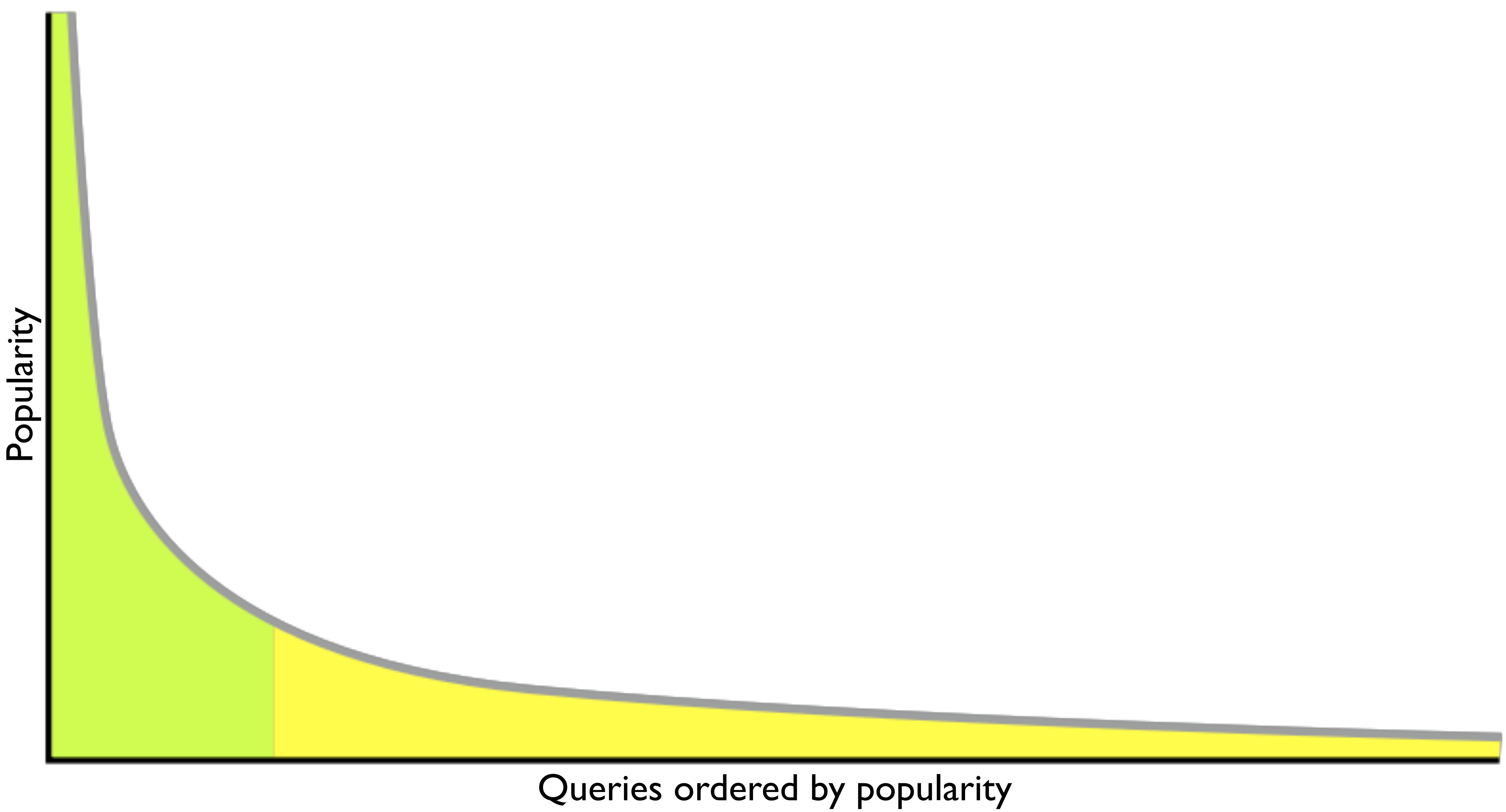
Load Balancing...Again!



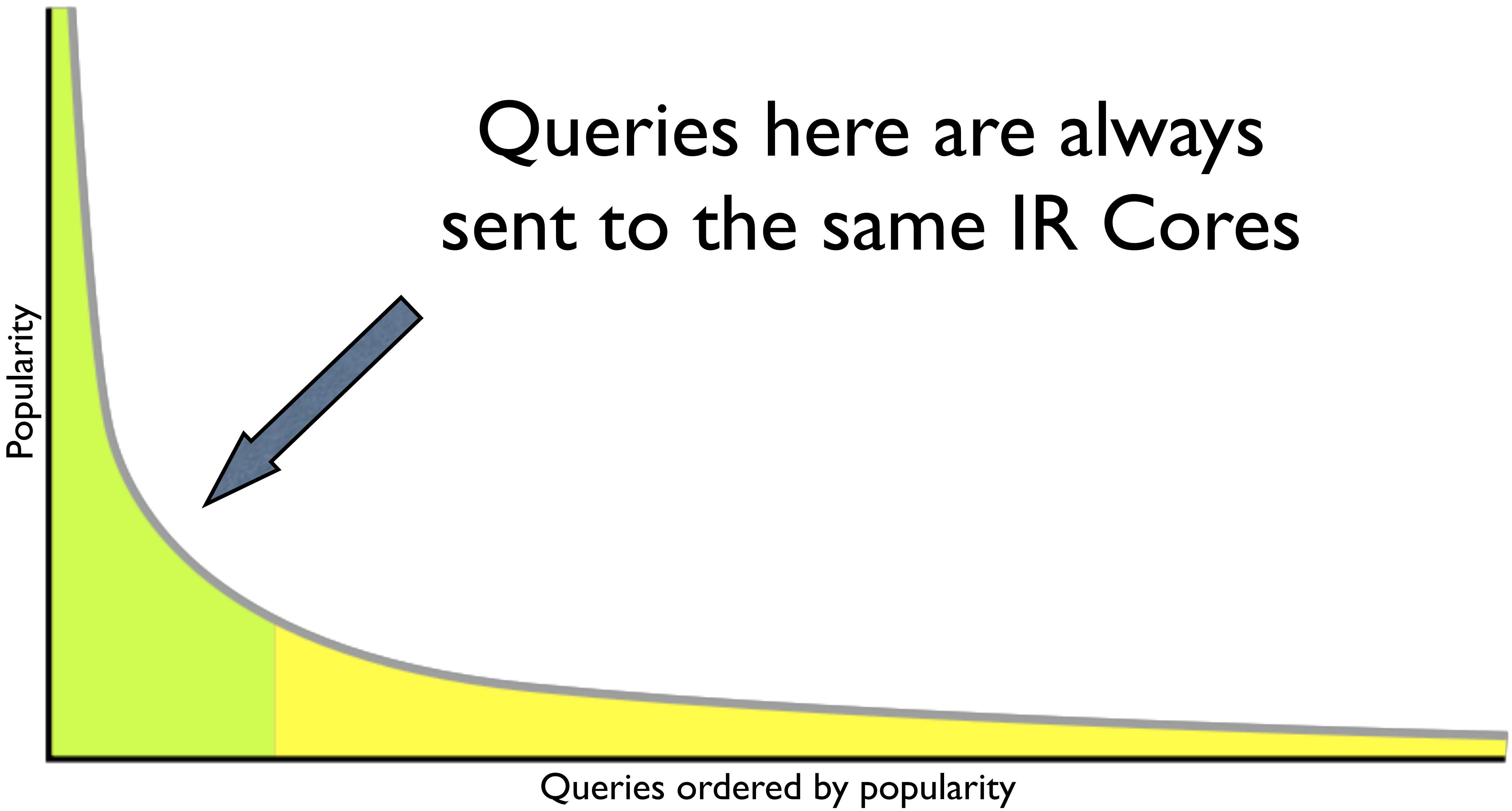
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Guess Why...

Guess Why...

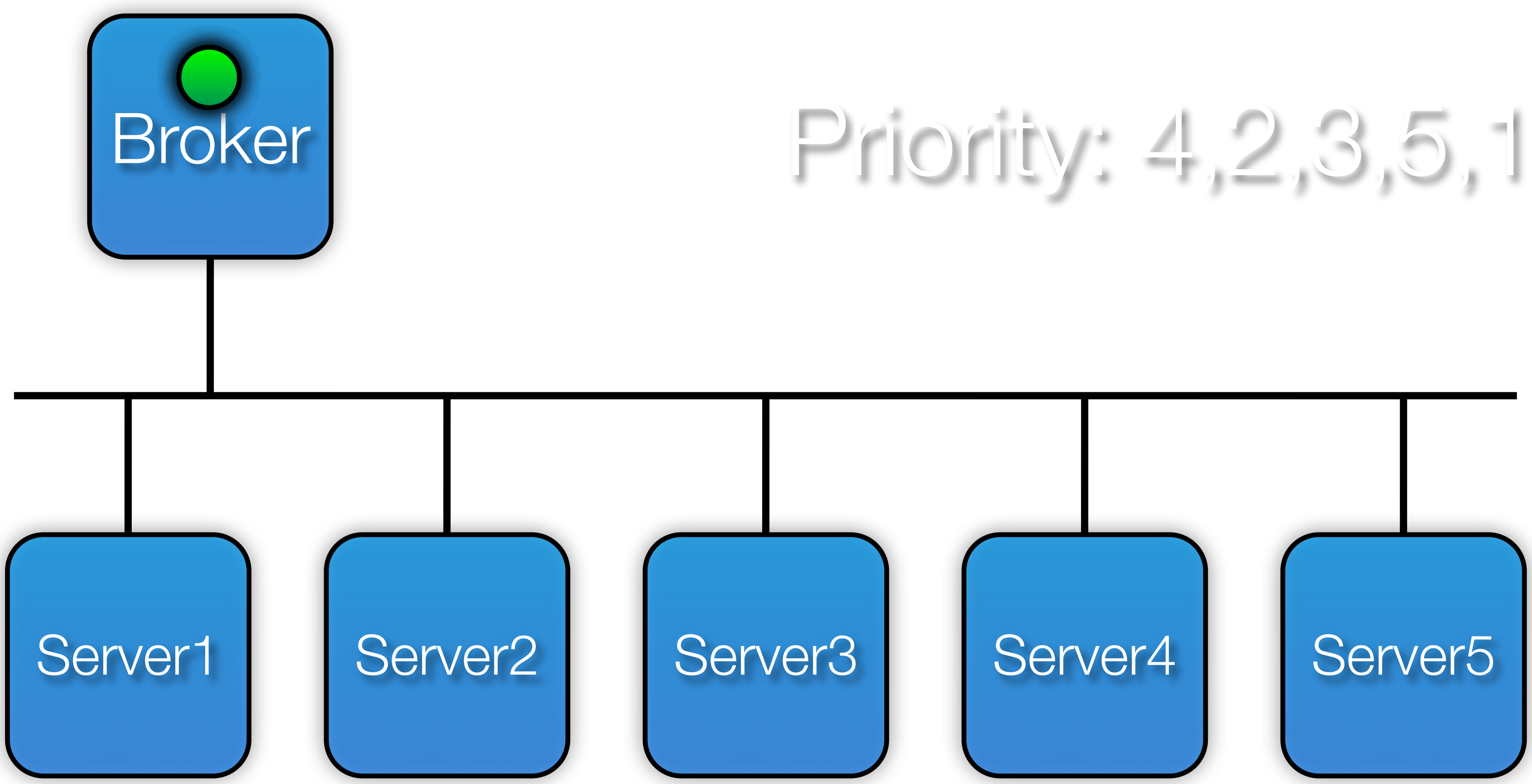


Guess Why...



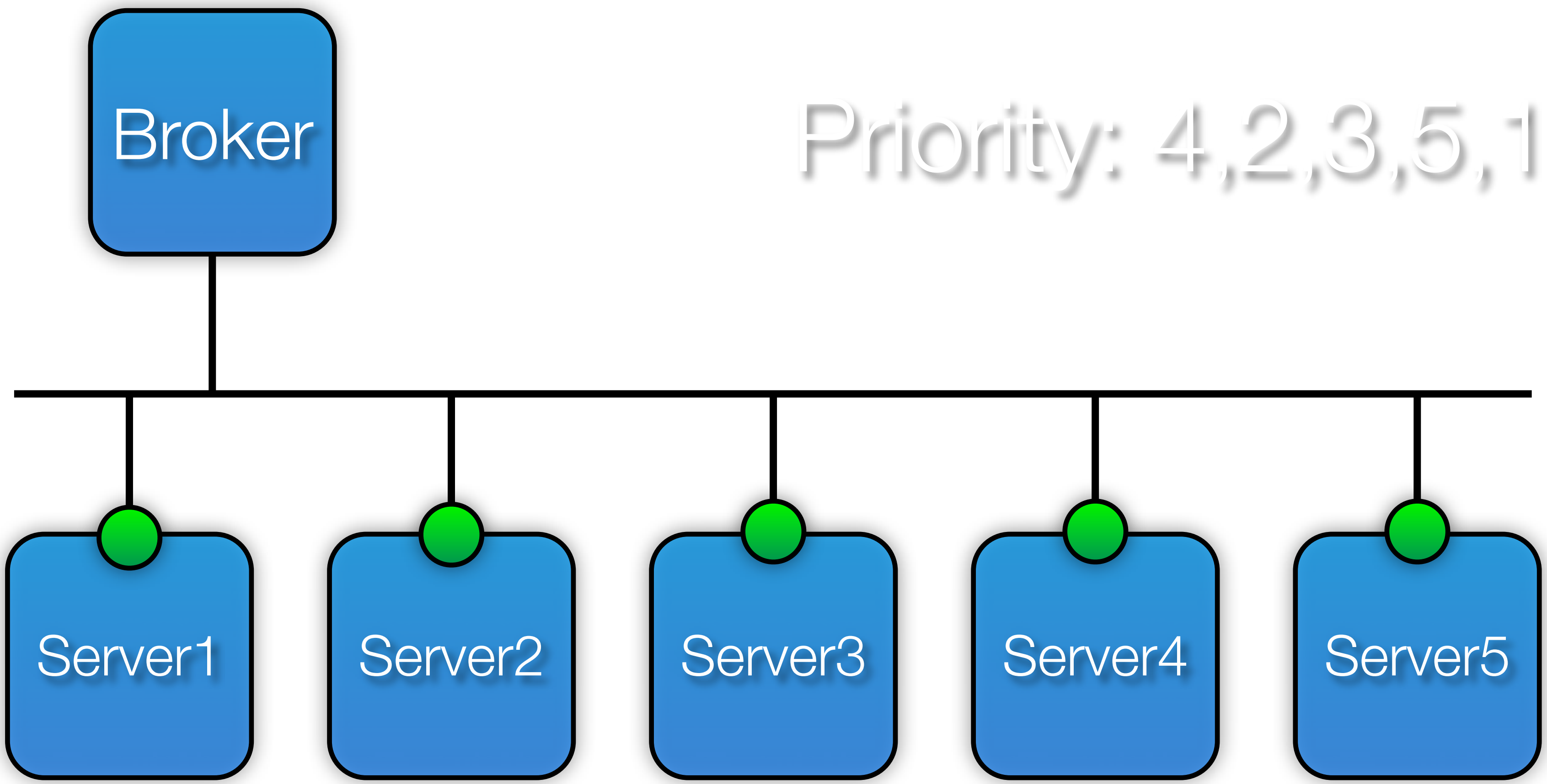
Queries here are always sent to the same IR Cores
Popularity
Queries ordered by popularity

Collection Prioritization



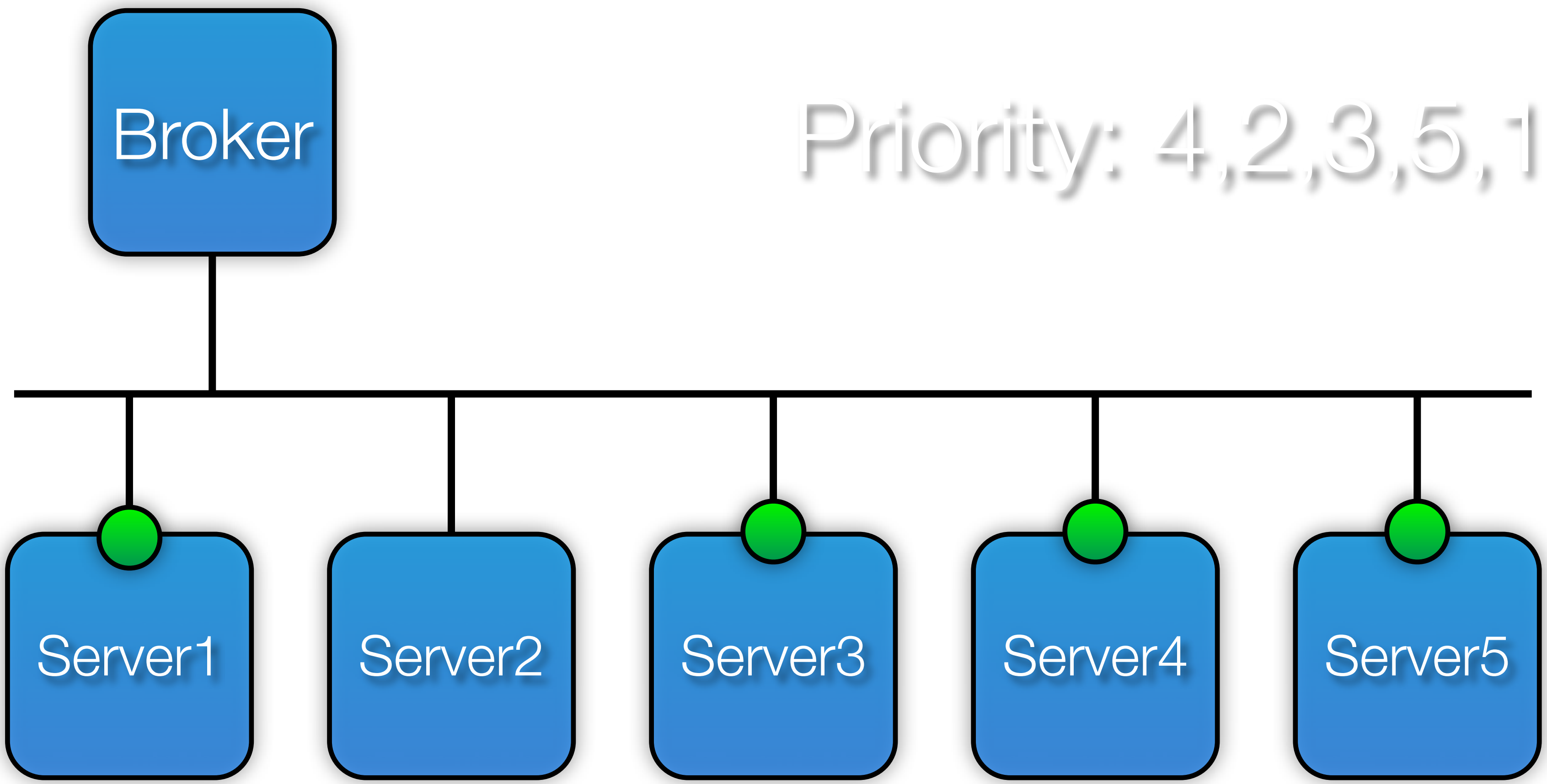
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Collection Prioritization



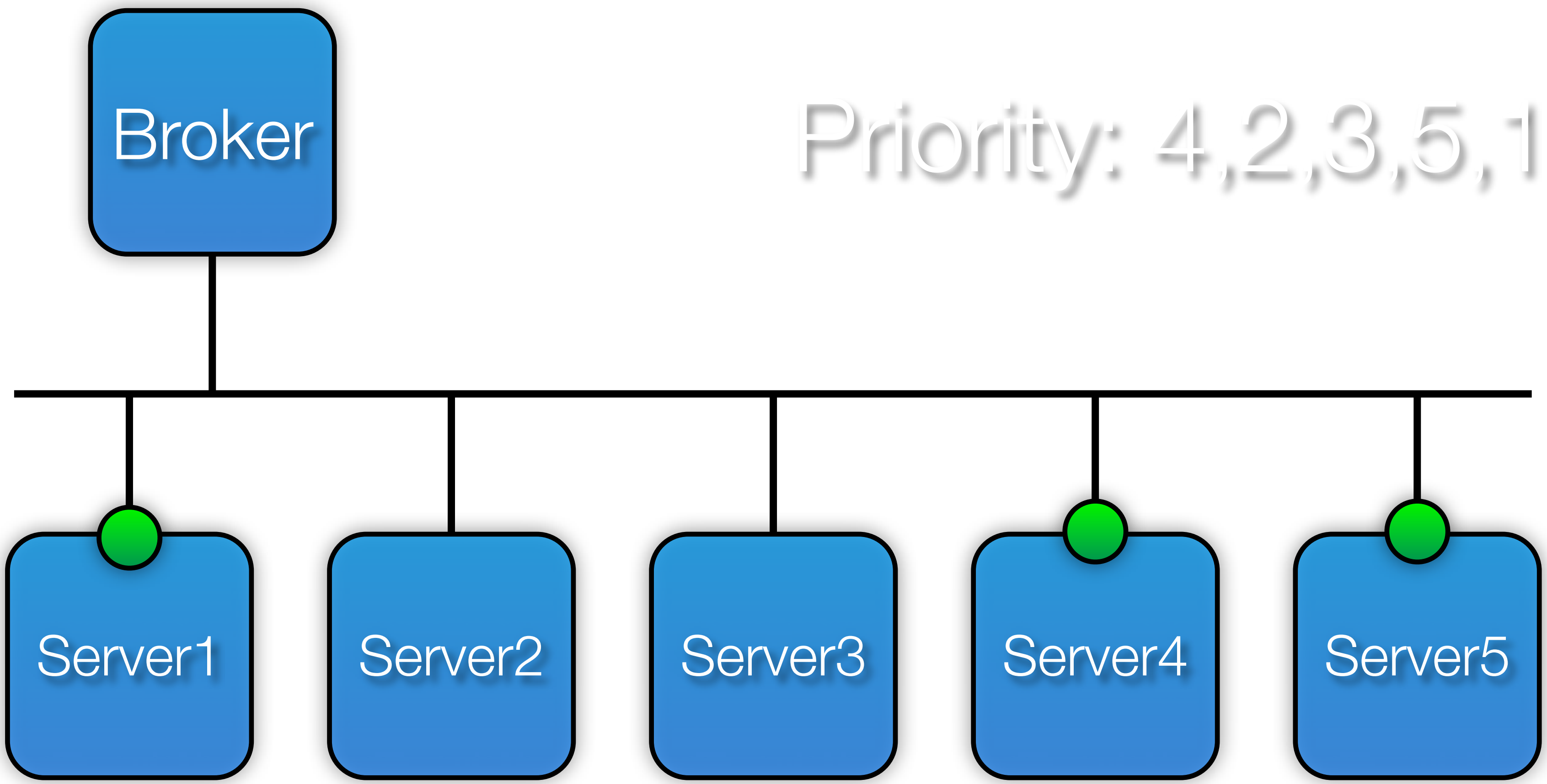
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Collection Prioritization



Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Collection Prioritization



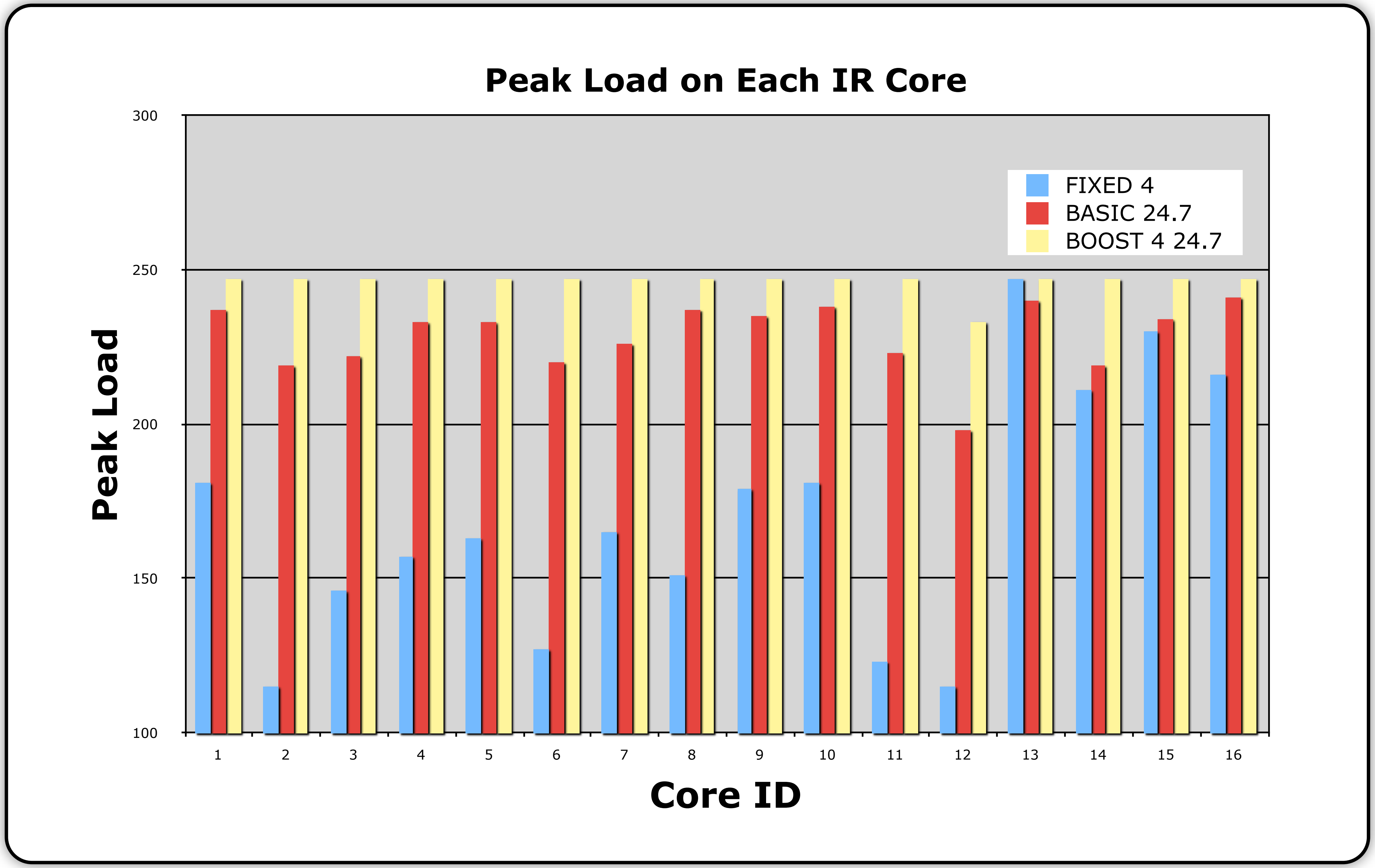
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

How IR Cores Take Decisions

- **load-driven basic<*L*>**
  - Accept queries only if the load is below *(1 - rank/NServer)* * *L*. E.g. the second best server accepts the query only if its load is below *90%* of *L*.
- **load-driven boost<*L,T*>**
  - It is the same as the basic strategy except that the first *T* servers' load threshold is always *L* and then it starts to decrease in a linear fashion as in the basic case.

Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Load... Balanced!



Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

What about Precision?

	FIXED 4	BASIC<25>	BOOST<4, 25>
5	34	56	71
10	34	68	70
20	34	68	70

Intersection (%) at

Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

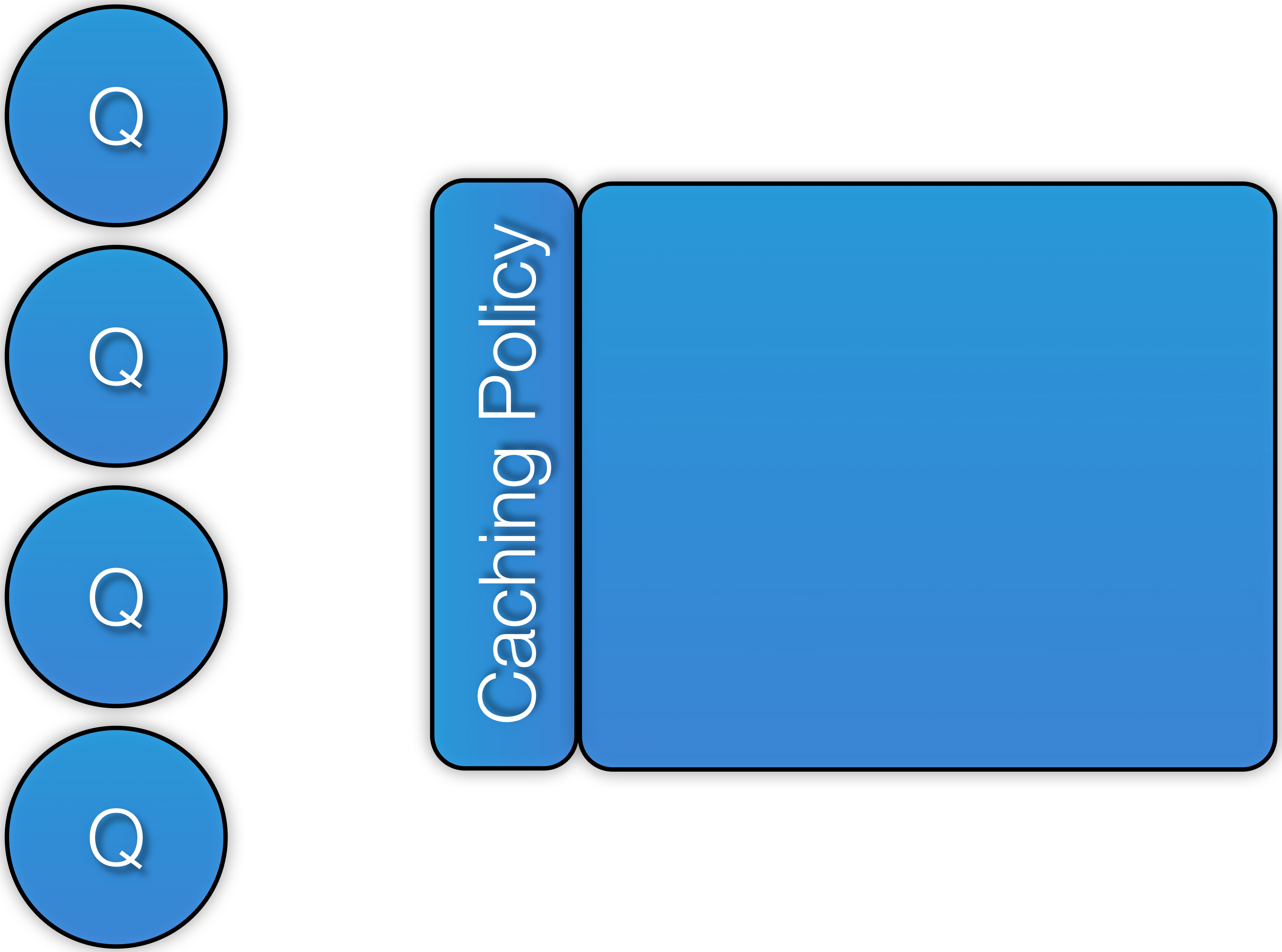
What about Precision?

	FIXED 4	BASIC<25>	BOOST<4, 25>
5	0,88	0,91	0,92
10	0,87	0,90	0,90
20	0,85	0,89	0,90

Competitive Similarity

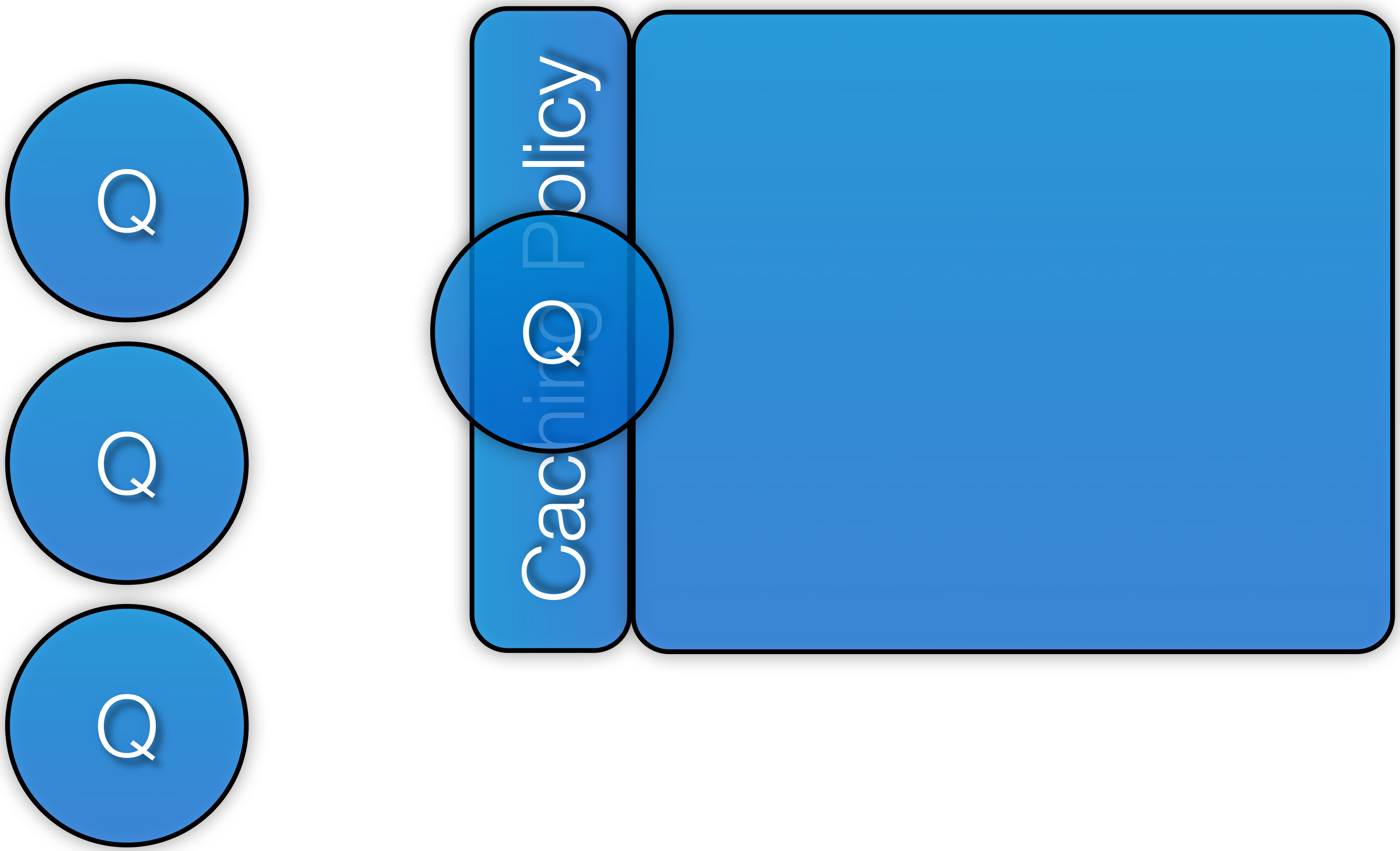
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



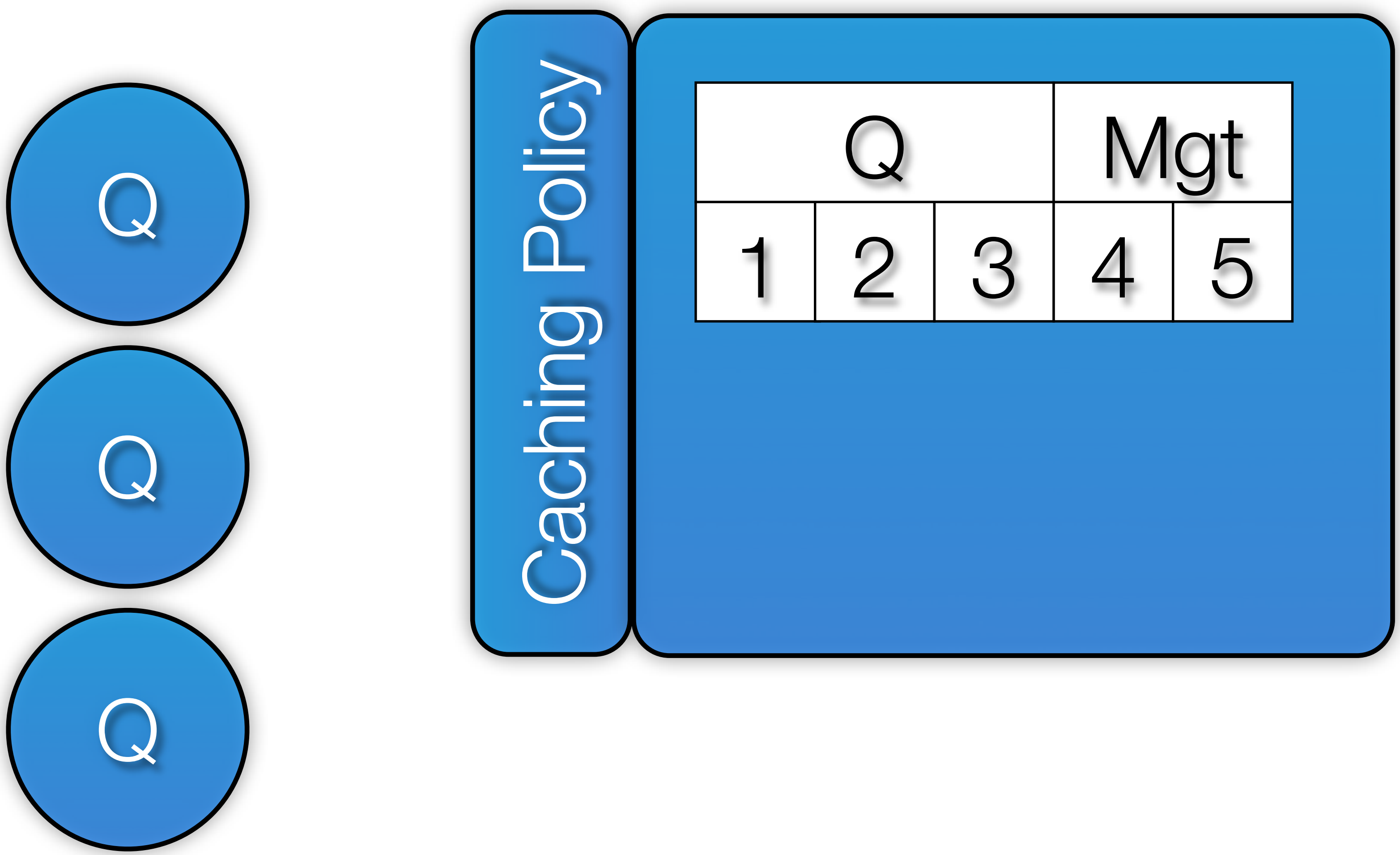
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



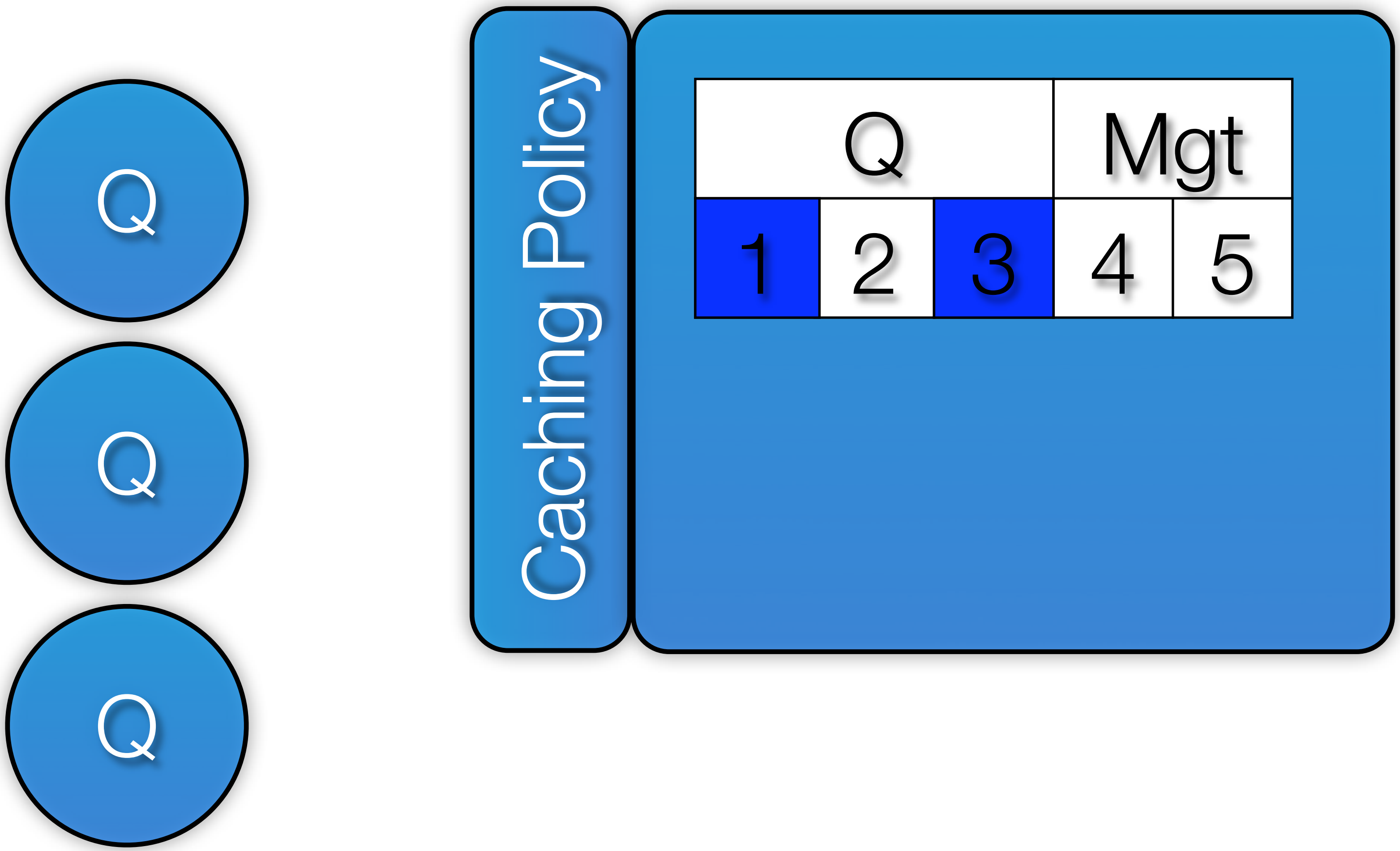
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



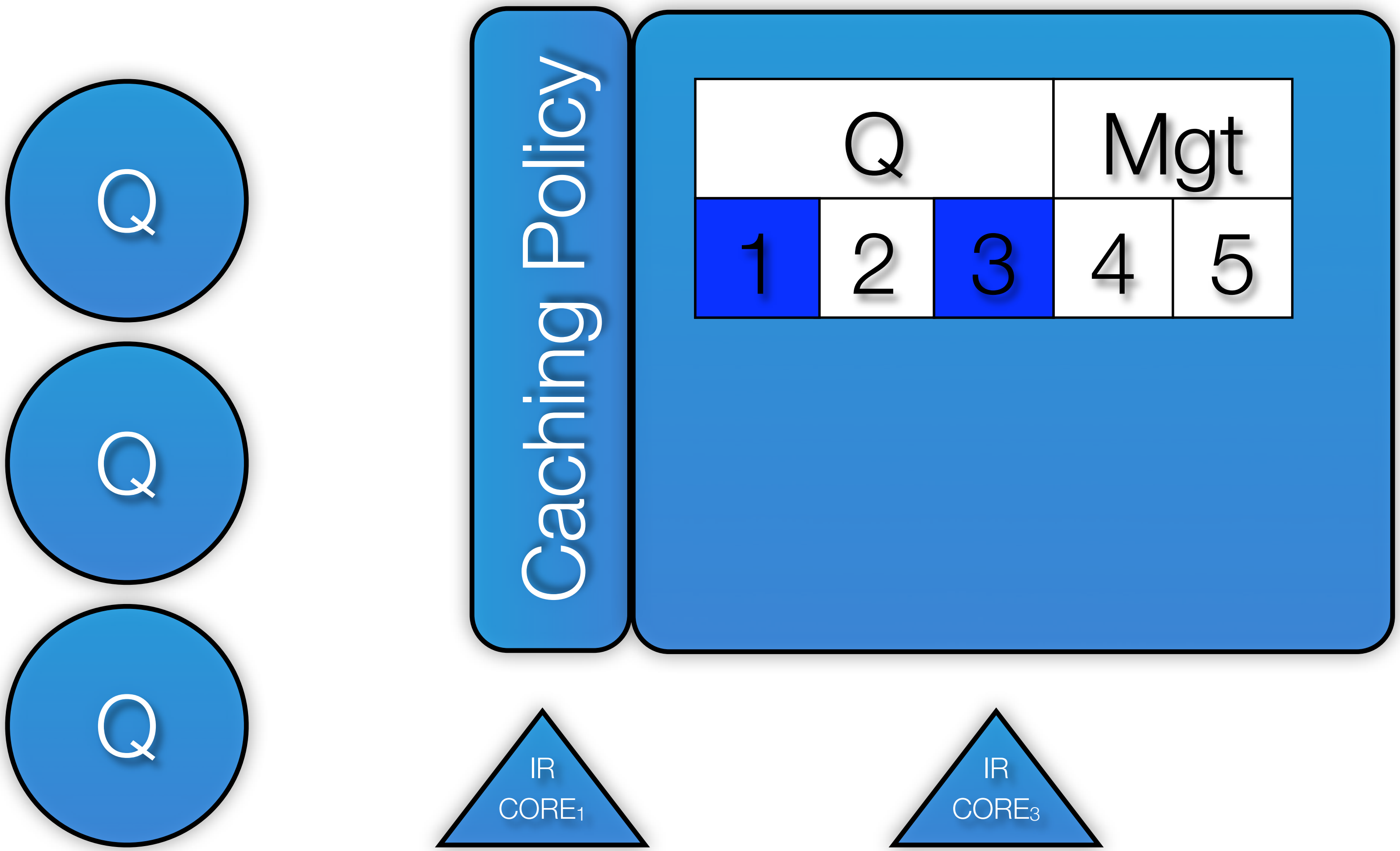
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



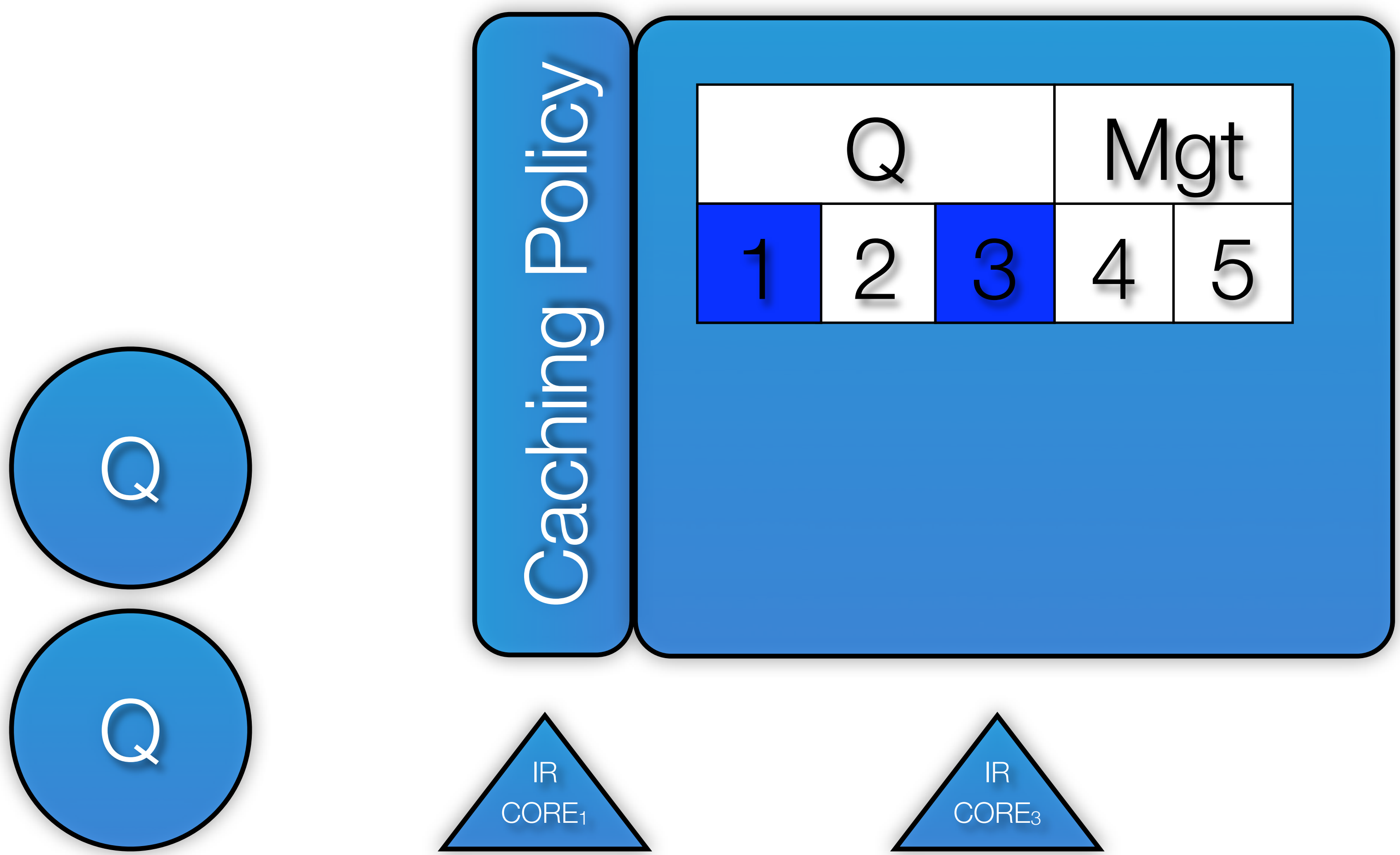
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



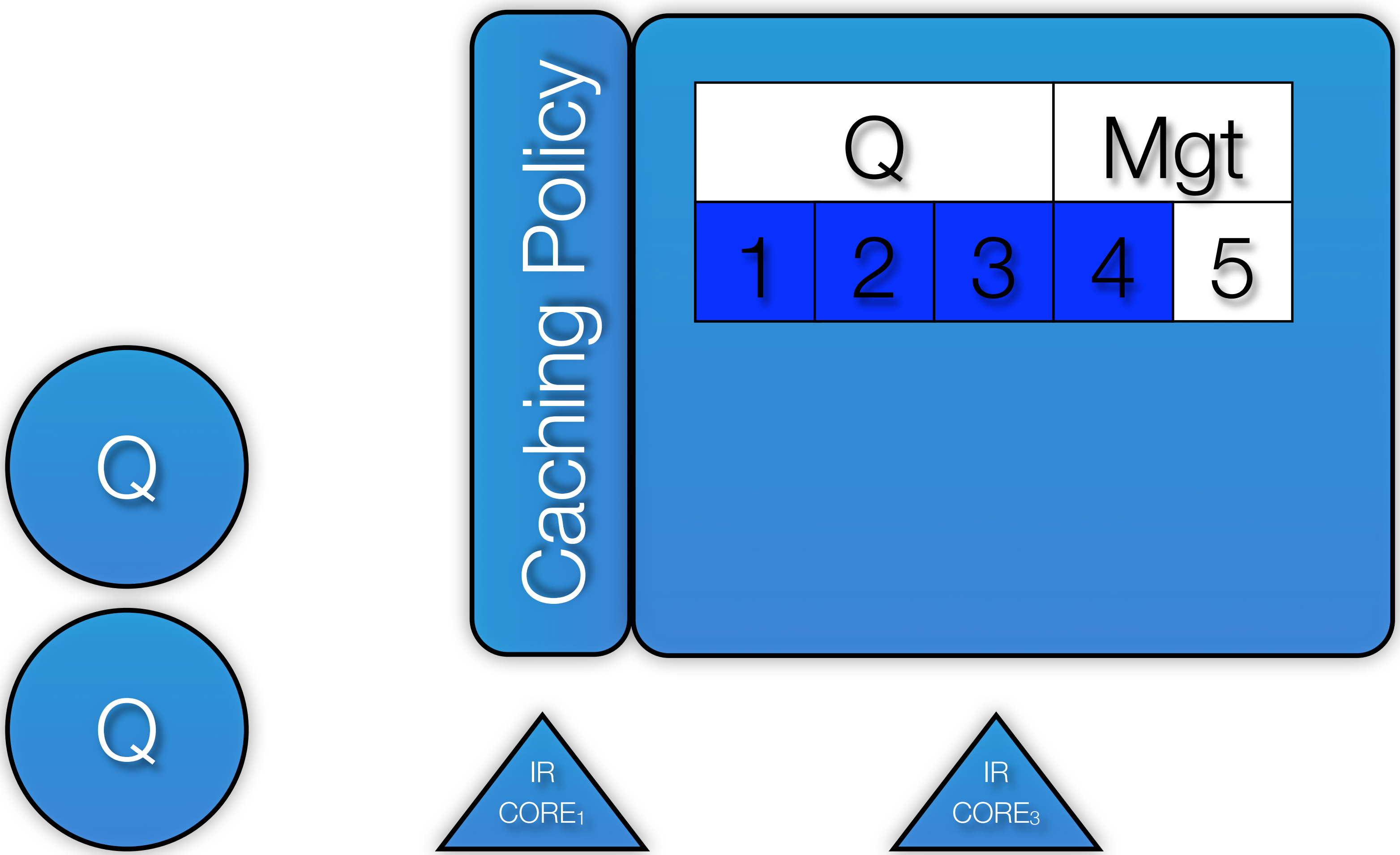
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



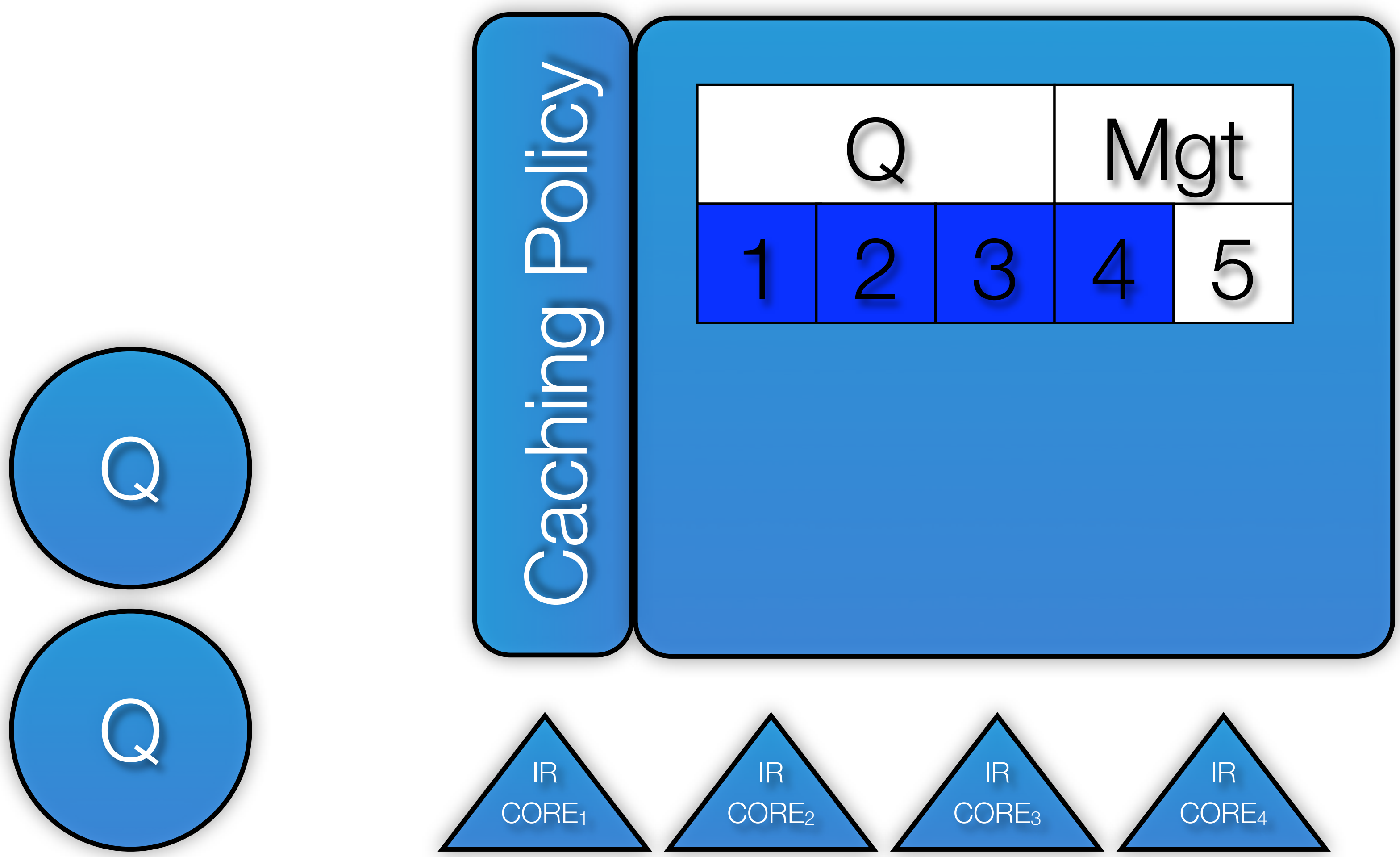
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



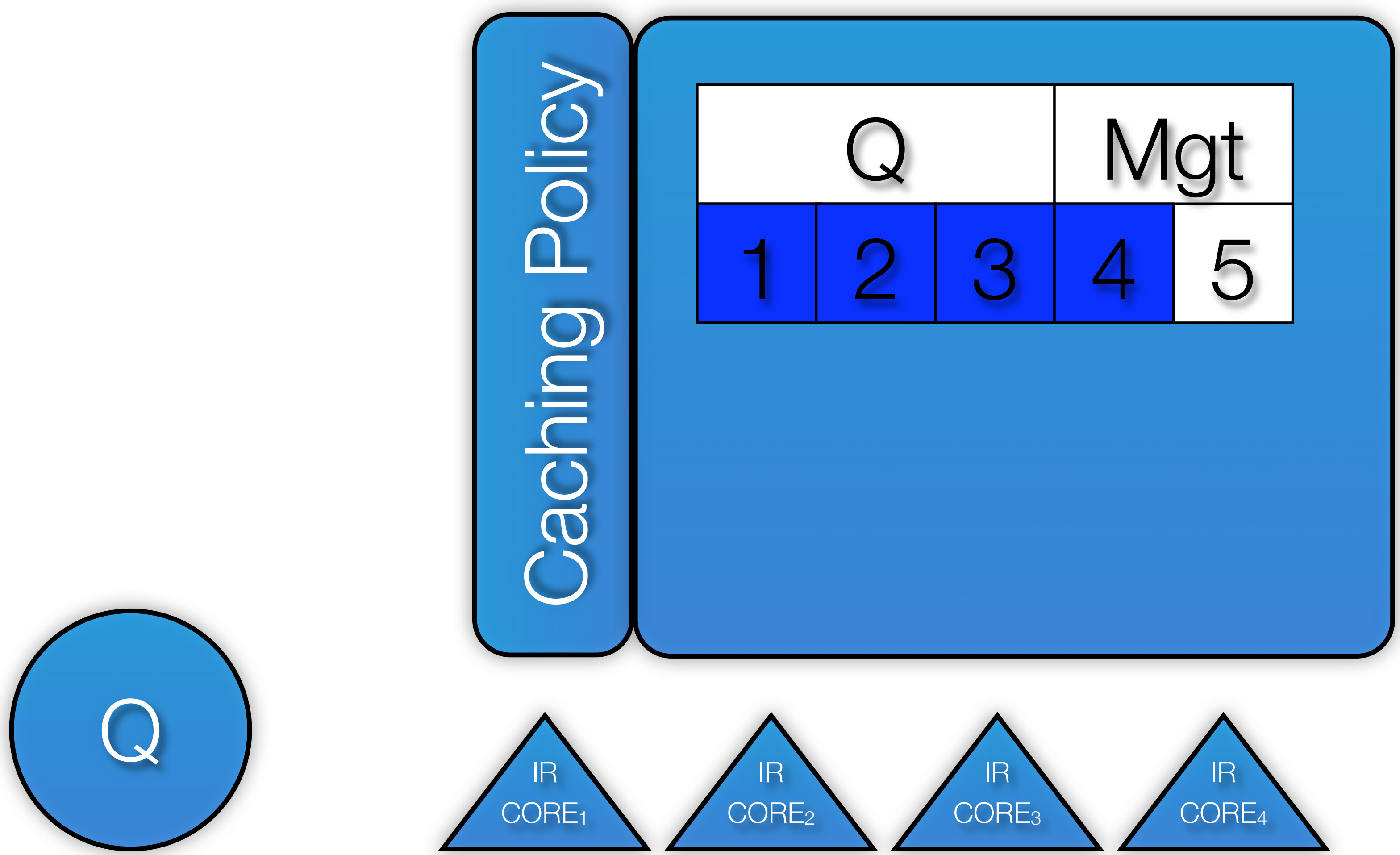
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



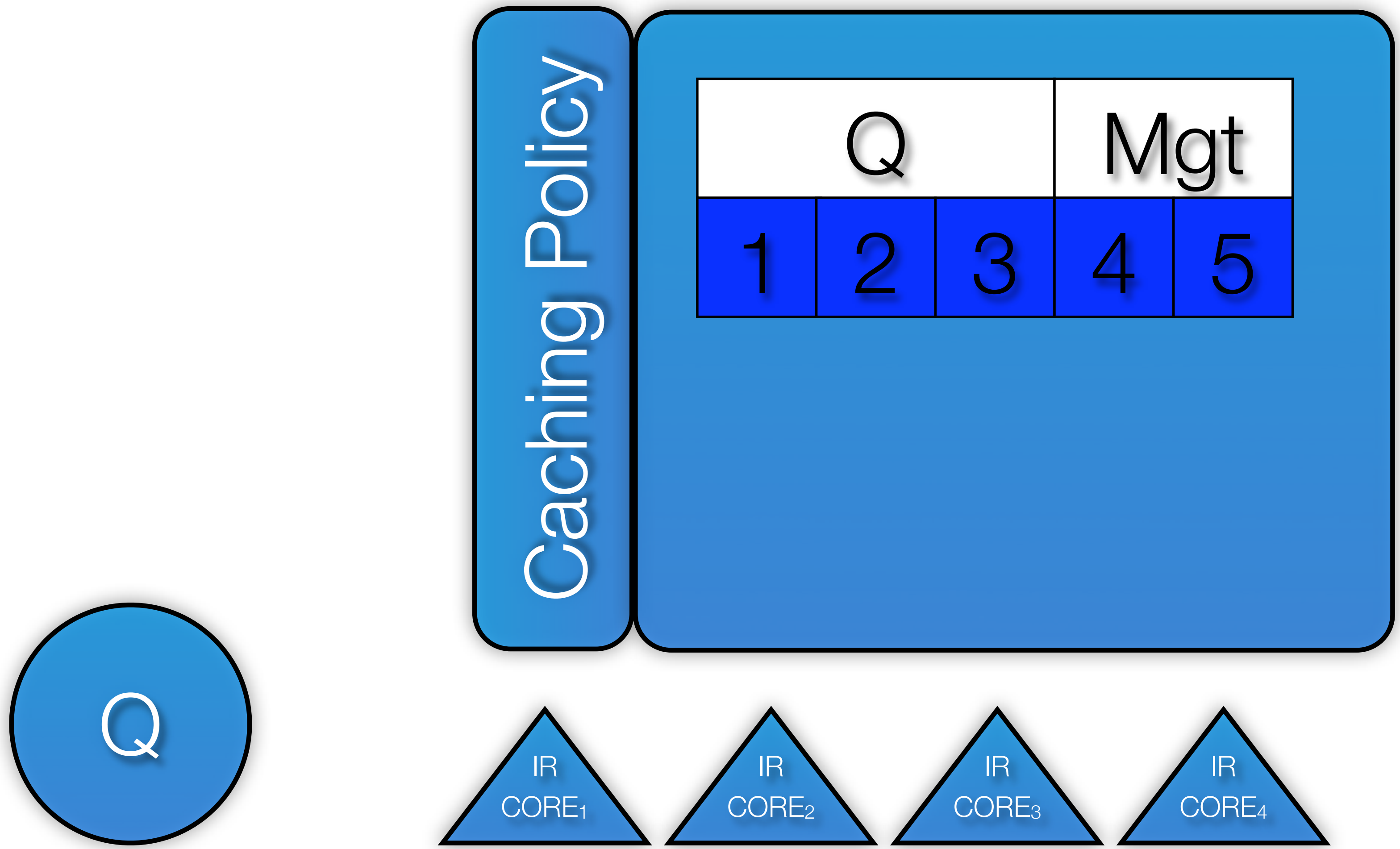
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



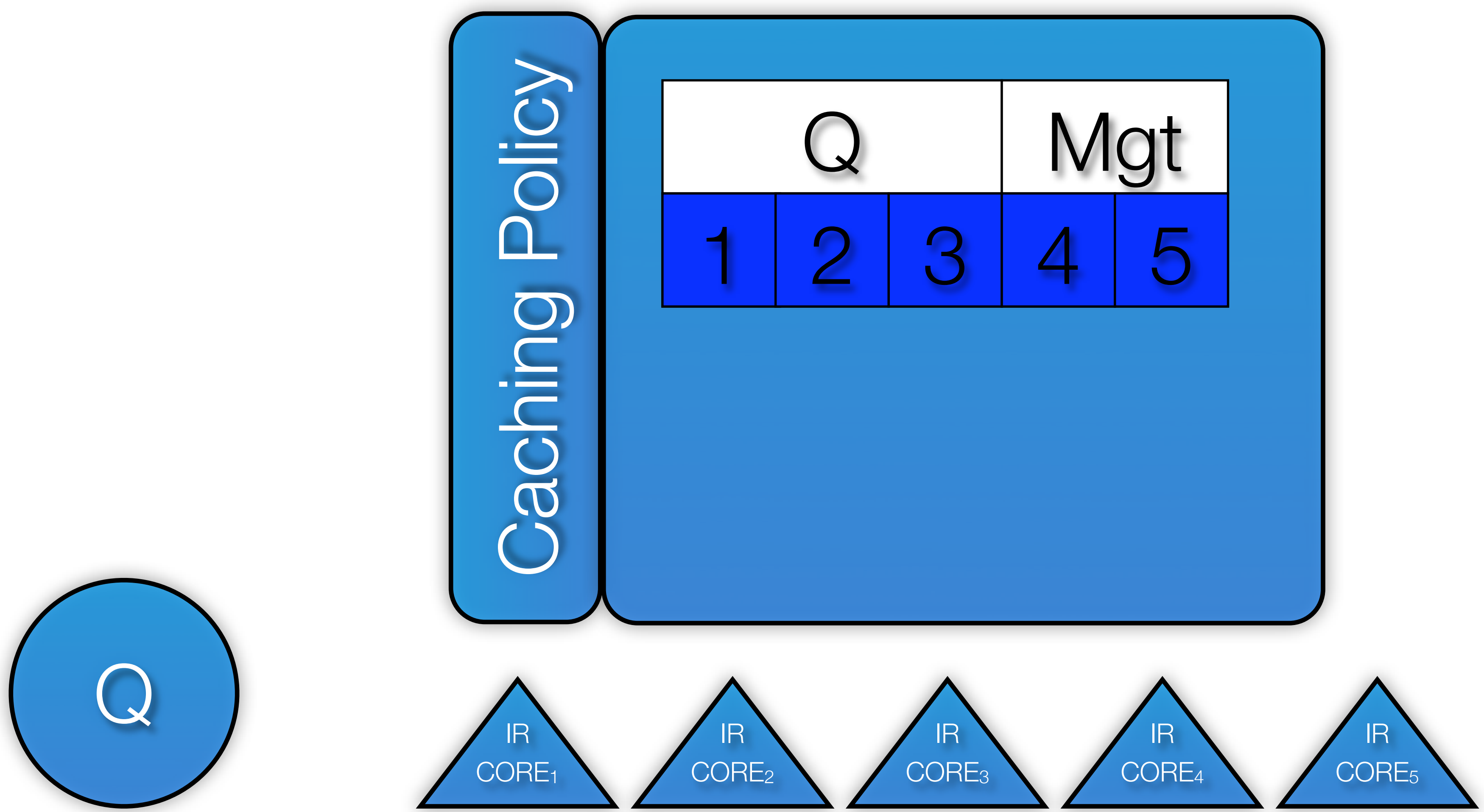
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



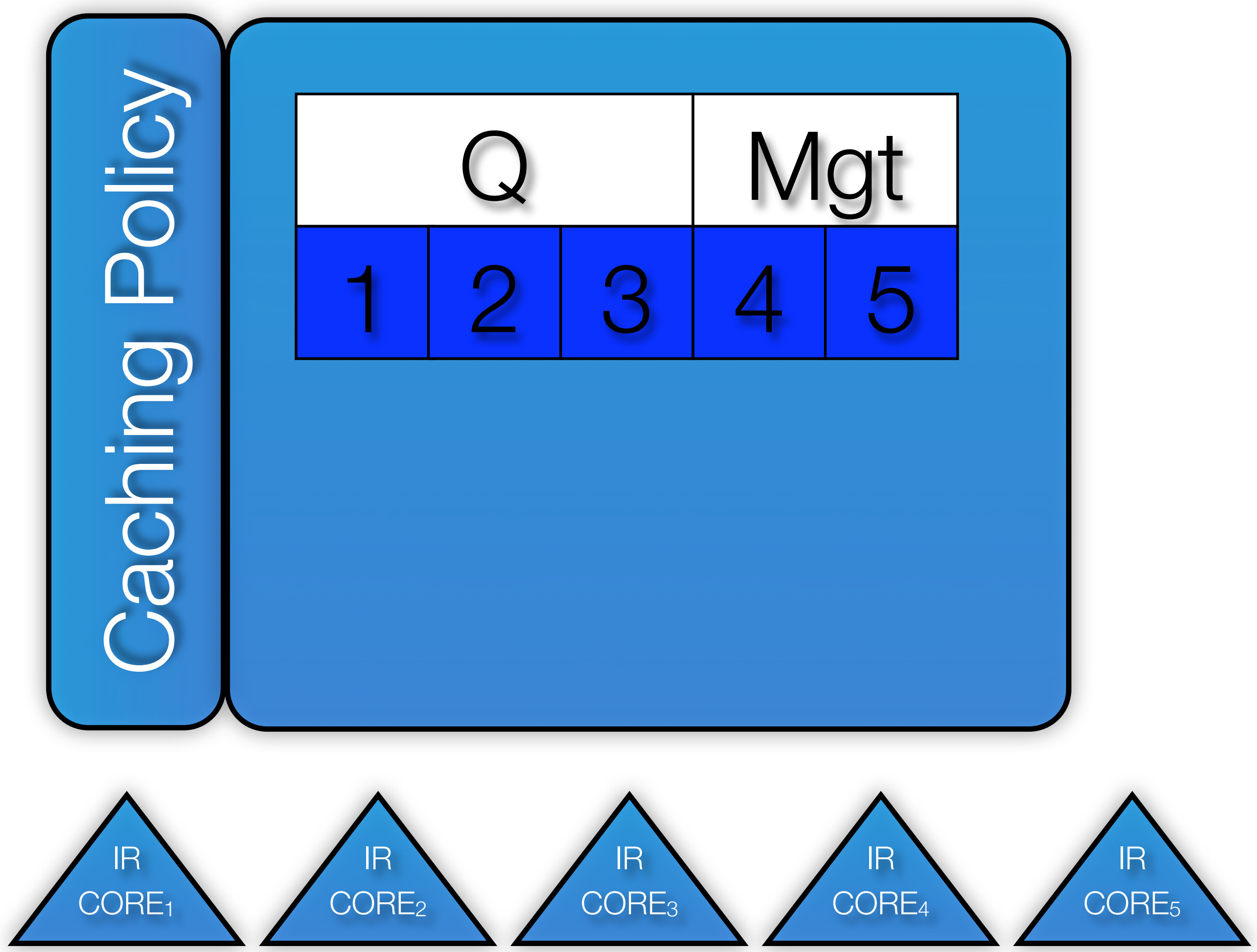
Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Incremental Caching



Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

Does it Work?

	BASIC<25>	BOOST<4, 25>	BOOST<4, 25> + INC
5	0,91	0,92	0,94
10	0,90	0,90	0,93
20	0,89	0,90	0,93

Competitive Similarity

Diego Puppin, Raffaele Perego, Fabrizio Silvestri, Ricardo Baeza-Yates. **Tuning the Capacity of Search Engines: Load-driven Routing and Incremental Caching to Reduce and Balance the Load**. To appear in ACM Transactions on Information Systems (TOIS).

The End

- Thank you for your attention!
- Hope you will get interested in the topics discussed.
- For more information:
  - Fabrizio Silvestri. **Mining Query Logs: Turning Search Usage Data into Knowledge**. Foundations and Trends in Information Retrieval. To Appear.