Query-log based techniques for optimizing WSE effectiveness

Salvatore Orlando[+], Raffaele Perego[*], Fabrizio Silvestri[*]
[*]ISTI - CNR, Pisa, Italy
[+]Università Ca' Foscari Venezia, Italy

Tutorial Outline

- Enhancing Effectiveness of Search Systems
  - Query Expansion/Suggestion/Personalization
  - Learning to Rank: Ranking SVM

Research issues (I)

- 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 by user studies involving small group of homogeneous people, e.g., metrics are tested on small human-annotated testbeds

Research issues (2)

- Privacy is nowadays a big concerns for user communities. Many of the techniques presented
  - need to store not only queries in the log, but also clicked results
  - need to store information to rebuild knowledge about user query sessions
  - need to build user profiles for personalization
- Personalization of query results is a valuable feature for increasing the effectiveness of a search engine
  - Profile-based search is computationally expensive
  - Personalization may prevent the adoption of global techniques aiming at enhancing performance (like those discussed in this tutorial)

Tutorial Outline

- Enhancing Effectiveness of Search Systems
  - Query Expansion/Suggestion/Personalization
  - Learning to Rank: Ranking SVM

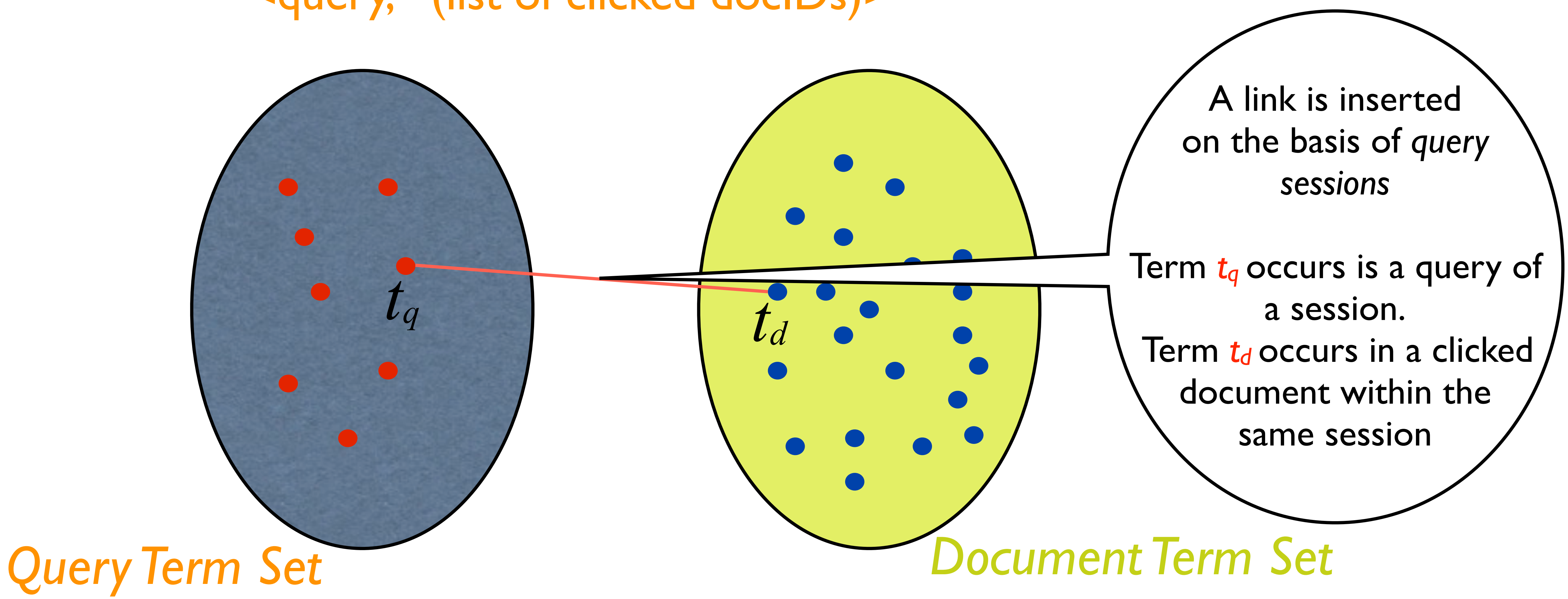
Query Expansion

- Queries are short, poorly built, and sometimes mistyped
- *Cui et al.* observed that queries and corresponding (clicked) 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.

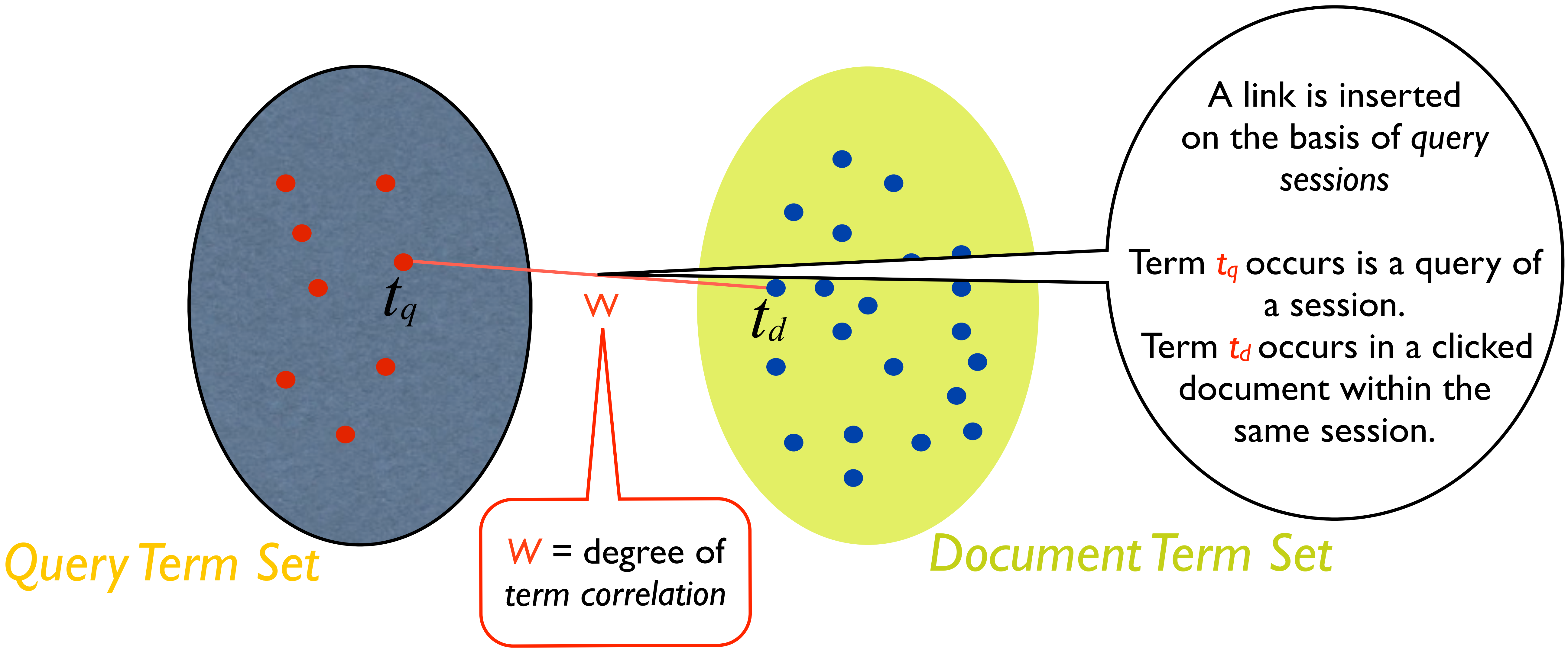
Query Expansion

- 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



- Correlation is given by the conditional probability $P(t_d \mid t_q)$
  - 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

- 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. Higher is better.
- The *top-k* ranked terms (those with the highest weights) are selected as expansion terms for query Q
  - e.g., the top terms of query 'Steve Jobs' : Apple, ipad, iphone

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

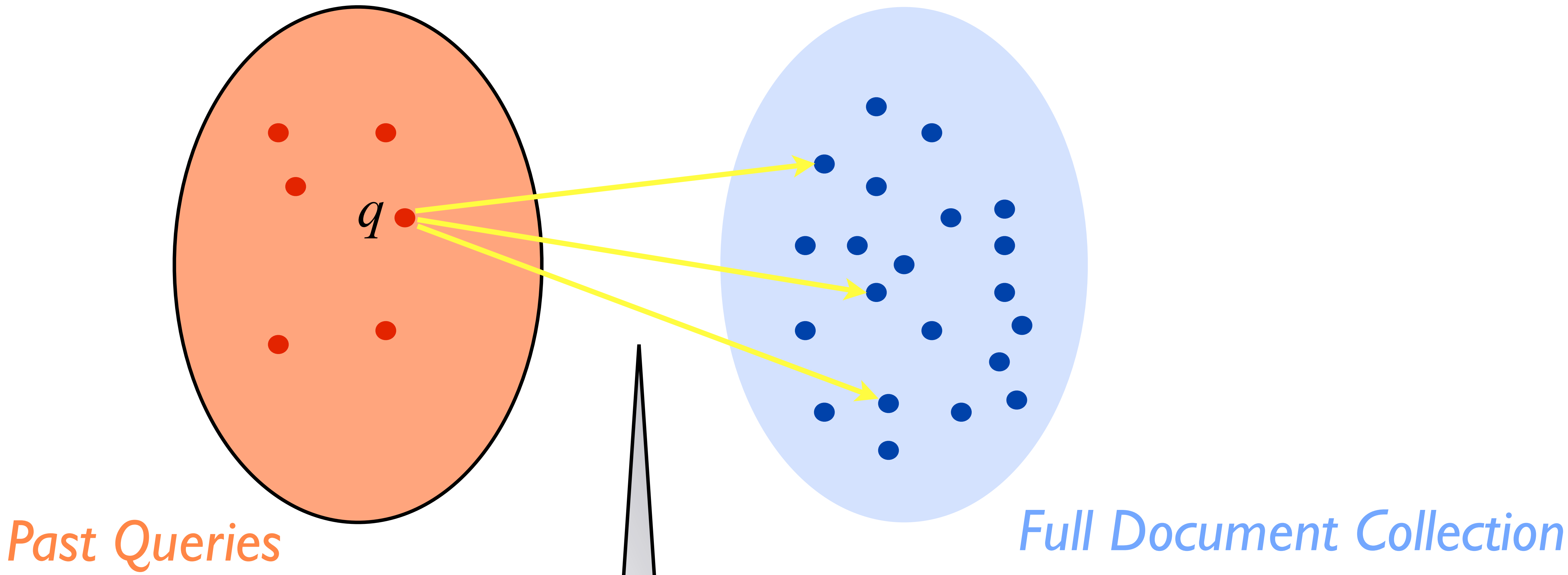
Query Expansion

- *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.

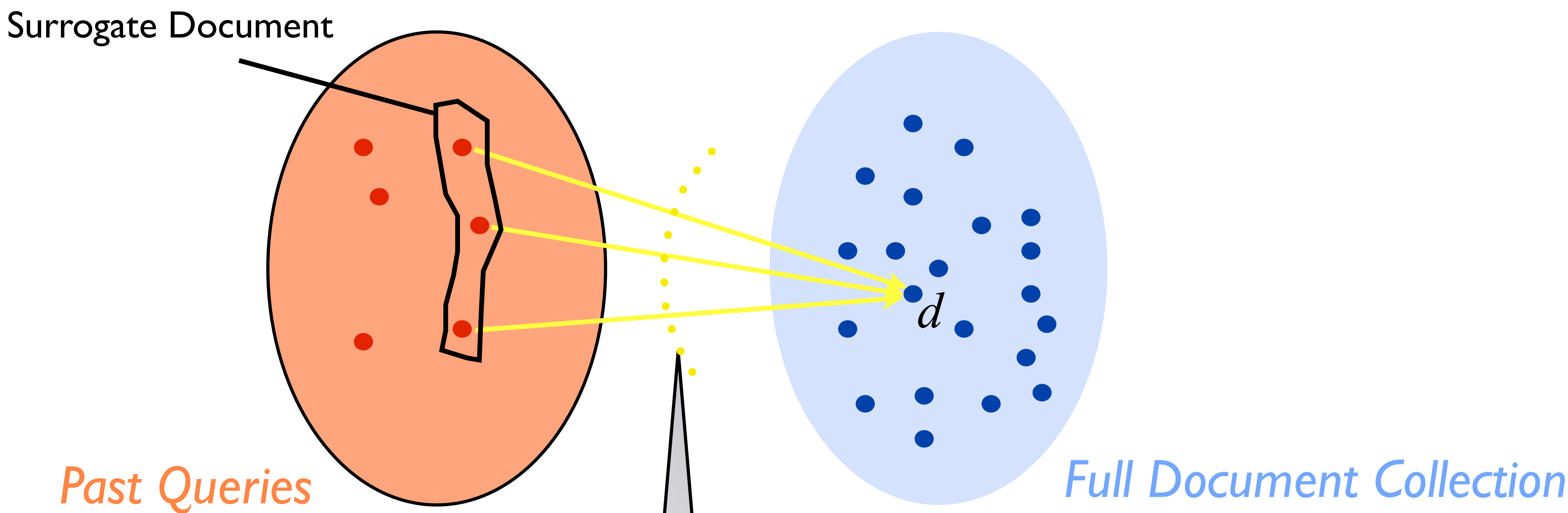
Query Expansion



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

- 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

- The *query expansion mechanism* (pseudo relevance feedback) is made up of the following steps:

  1. For a newly submitted query *q*, a set T of top ranked (full or surrogate) "documents" is built
  2. On the basis of T, extract and rank a list L of candidate terms (from the set of full or surrogate documents)
  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

- Once built the bipartite graph, the space of the surrogate documents, steps 1 and 2 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.

Tutorial Outline

- Enhancing Effectiveness of Search Systems
  - Query Expansion/Suggestion/Personalization
  - Learning to Rank: Ranking SVM

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*
    $q_i \Rightarrow q_1$, $q_i \Rightarrow q_2$, $q_i \Rightarrow q_3$, ..., $q_i \Rightarrow q_m$
  - 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)

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

- 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.

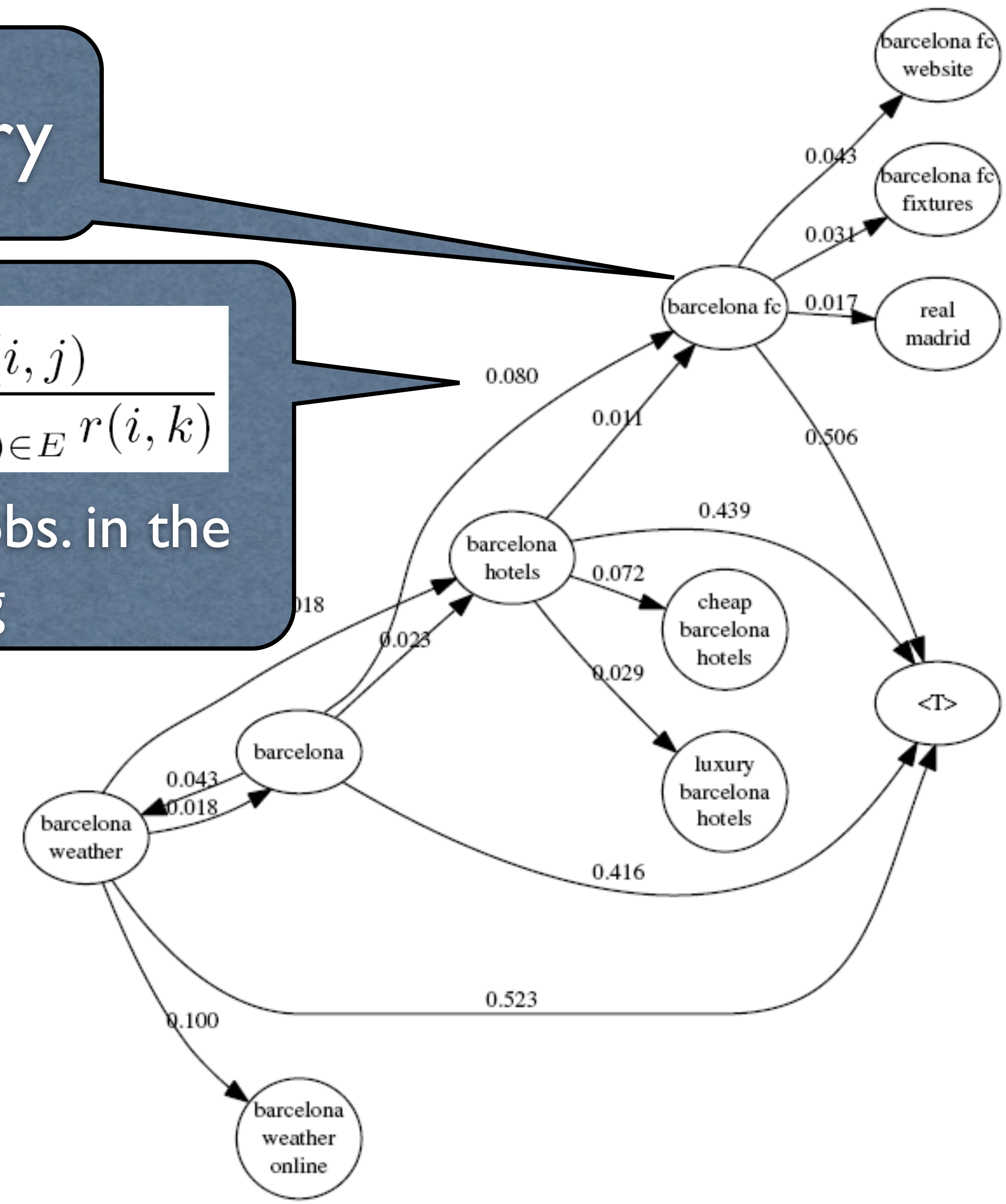
Query suggestion

Node=query

$$w(i,j) = \frac{r(i,j)}{\sum_{k:(i,k)\in E} r(i,k)}$$

r = # of the pair obs. in the query log

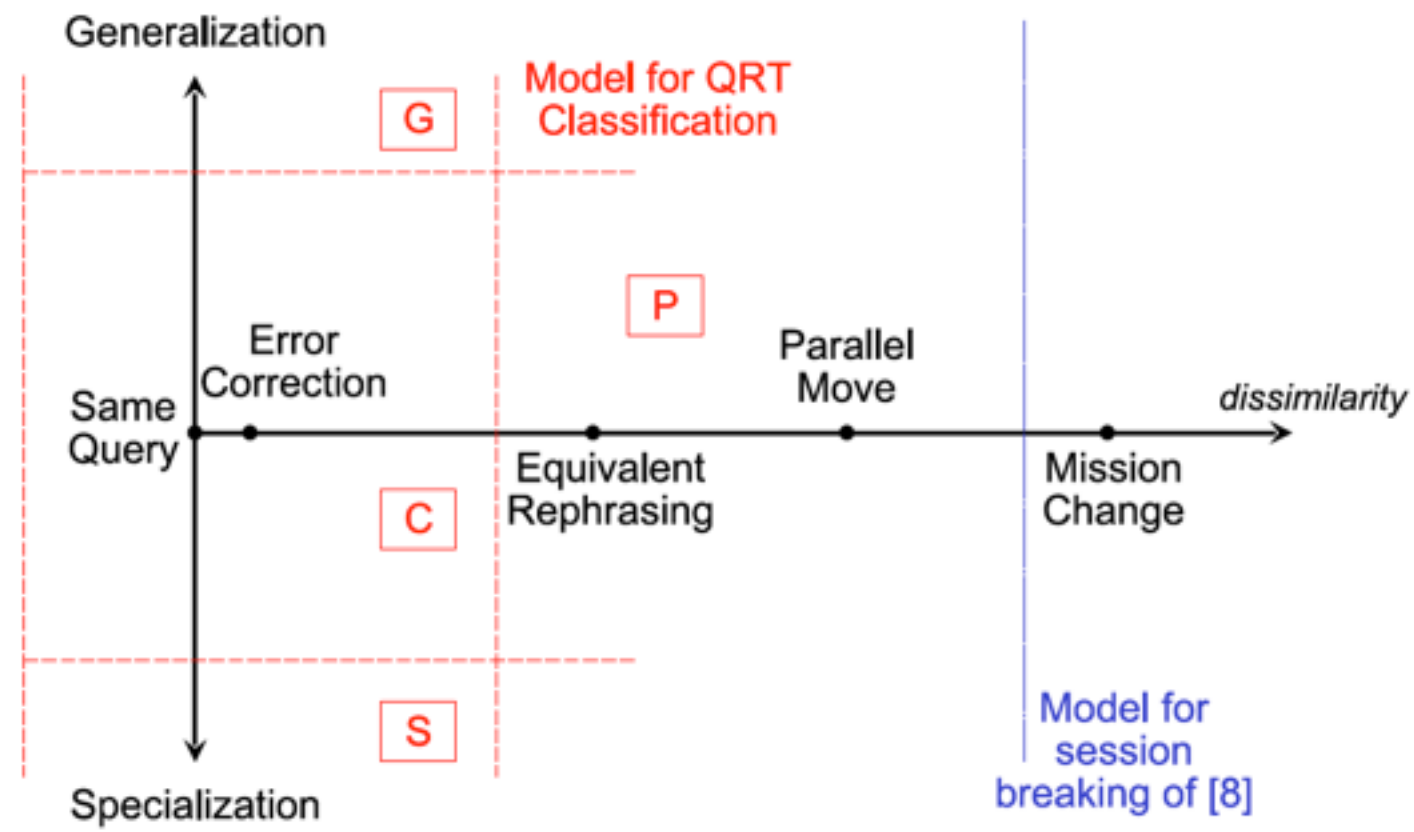
- Suggest queries by using a **query-flow graph (QFG)**
  - a directed edge means that the two linked queries are likely to be part of the same "search task"



P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **The query-flow graph: model and applications.** CIKM 2008: 609-618
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **Query suggestions using query-flow graphs.** WSCD, 2009

Query suggestion

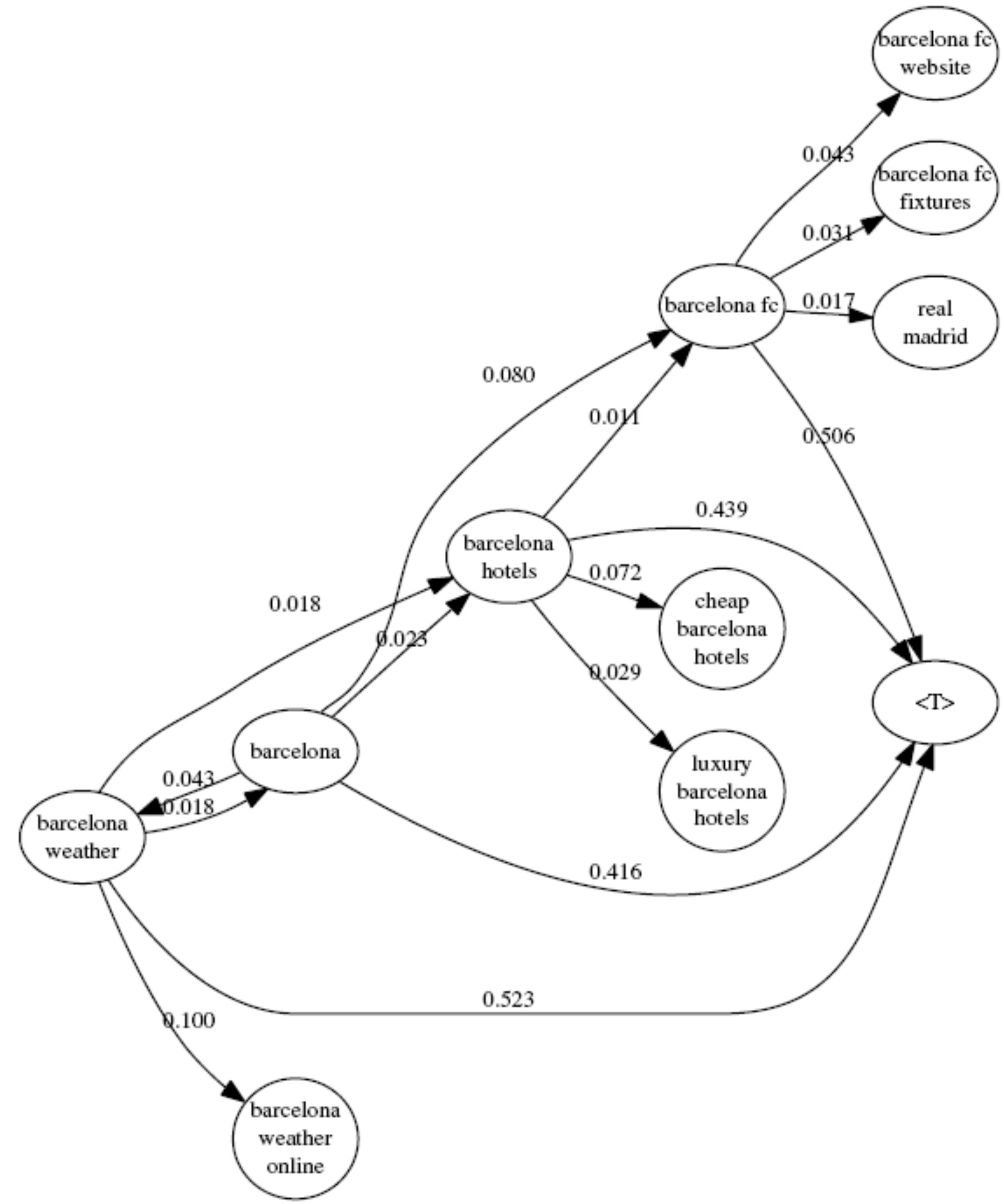
- Four query reformulation types (abbreviated QRT):
  - Generalization
  - Specialization
  - Error Correction
  - Parallel Move



P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **The query-flow graph: model and applications.** CIKM 2008: 609-618
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **Query suggestions using query-flow graphs.** WSCD, 2009
P. Boldi, F. Bonchi, C. Castillo, S. Vigna **From 'dango' to 'japanese cakes': Query Reformulation Models and Patterns,** WI09

Query suggestion

- A model for automatic classification of QRTs is learnt from a human-labeled query log
- Sliced QF graph: one for each types of edge



P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **The query-flow graph: model and applications.** CIKM 2008: 609-618
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **Query suggestions using query-flow graphs.** WSCD, 2009
P. Boldi, F. Bonchi, C. Castillo, S. Vigna **From 'dango' to 'japanese cakes': Query Reformulation Models and Patterns,** WI09

Query suggestion

- The method for query suggestion proposed by Boldi, et al. is inspired by the work by Craswell and Szummer
  - random walks on the query-click bipartite graph (queries and clicked URLs), where the edges are symmetric
- In their experiment, they show that query recommendations based on short random walks on the query-flow graph (without using users' clicks) can match in precision, and often improve, recommendations based on query-click graphs

N. Craswell and M. Szummer, **"Random walks on the click graph"** in SIGIR 2007.
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **Query suggestions using query-flow graphs.** WSCD, 2009

Query suggestion

- Recommendation Random Walk Algorithms
  - Different Query-flow subgraphs
  - Baseline: query-click bipartite graph by Crasswell and Szummer
- Test Corpus
  - 114 input queries in MSN Live having frequency between 700 and 1500

N. Craswell and M. Szummer, **"Random walks on the click graph"** in SIGIR 2007.
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **Query suggestions using query-flow graphs.** WSCD, 2009

Query suggestion

- Query-flow subgraphs
  - Queryflow-S
  - Queryflow-SP
  - Queryflow-SC
  - Queryflow-SCP
  - Queryflow-GSPC

- **G**eneralization
- **S**pecialization,
- **C**orrection
- **P**arallel Move

N. Craswell and M. Szummer, **“Random walks on the click graph”** in SIGIR 2007.
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **Query suggestions using query-flow graphs.** WSCD, 2009

Query suggestion

- Top 5 recommendations per query per system
  - 5 assessors rated a recommendation as *Useful, Somewhat useful, Not useful*
  - **Usefulness**: "Related to the same intent and provides information not available in original query"

Table 1: Example assessments for query "cnn news"

Useful	Somewhat useful	Not useful
cnn world news	abc7chicagonews	CNN
msnbc news	nba scores	cnn.com
fox news	cnnfyi	verizon netmail

- Specialization transitions seems to produce the most useful recommendations
  - probability that a recommendation issued by the system is useful or somewhat useful (about 55%)

N. Craswell and M. Szummer, **"Random walks on the click graph"** in SIGIR 2007.
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **Query suggestions using query-flow graphs.** WSCD, 2009

Query suggestion

- Recently Szpektor, et al. argued that the long-tail distribution of query submitted to a SE implies that a large fraction of queries are rare
- Most query assistance services perform poorly or are not even triggered on long-tail queries

I. Szpektor, A. Gionis, Y. Maarek, **"Improving recommendation for long-tail queries via templates"** in WWW 2011.

Query suggestion

- They introduce the concept of query template
  - a query-flow graph can be built by considering transitions between query template rather than individual queries
- Example:
  - rare query “Montezuma surf” ➡ recognize that Montezuma is a <city>
  - a template rule ‘<city> surf → <city> beach’ has been observed
  - we suggest “Montezuma beach”

I. Szpektor, A. Gionis, Y. Maarek, **“Improving recommendation for long-tail queries via templates”** in WWW 2011.
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **The query-flow graph: model and applications.** CIKM 2008: 609-618

Query suggestion

- First, tokenize queries
  - all the possible *tokenizations* of "chocolate cookie recipe" are: (chocolate)(cookie)(recipe), (chocolate)(cookie recipe), (chocolate cookie)(recipe), (chocolate cookie recipe)
- Second, exploit a generalization hierarchy H = (E, R), where R ⊆ E × E
  - In the paper, H is the WordNet 3.0 hypernymy hierarchy

chocolate → food
chocolate → drink
cookie → dessert
chocolate cookie → dessert
dessert → food
food → substance
recipe → instruction

I. Szpektor, A. Gionis, Y. Maarek, **"Improving recommendation for long-tail queries via templates"** in WWW 2011.
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **The query-flow graph: model and applications.** CIKM 2008: 609-618

Query suggestion

- Third, apply the generalization H to produce the templates for "chocolate cookie recipe"

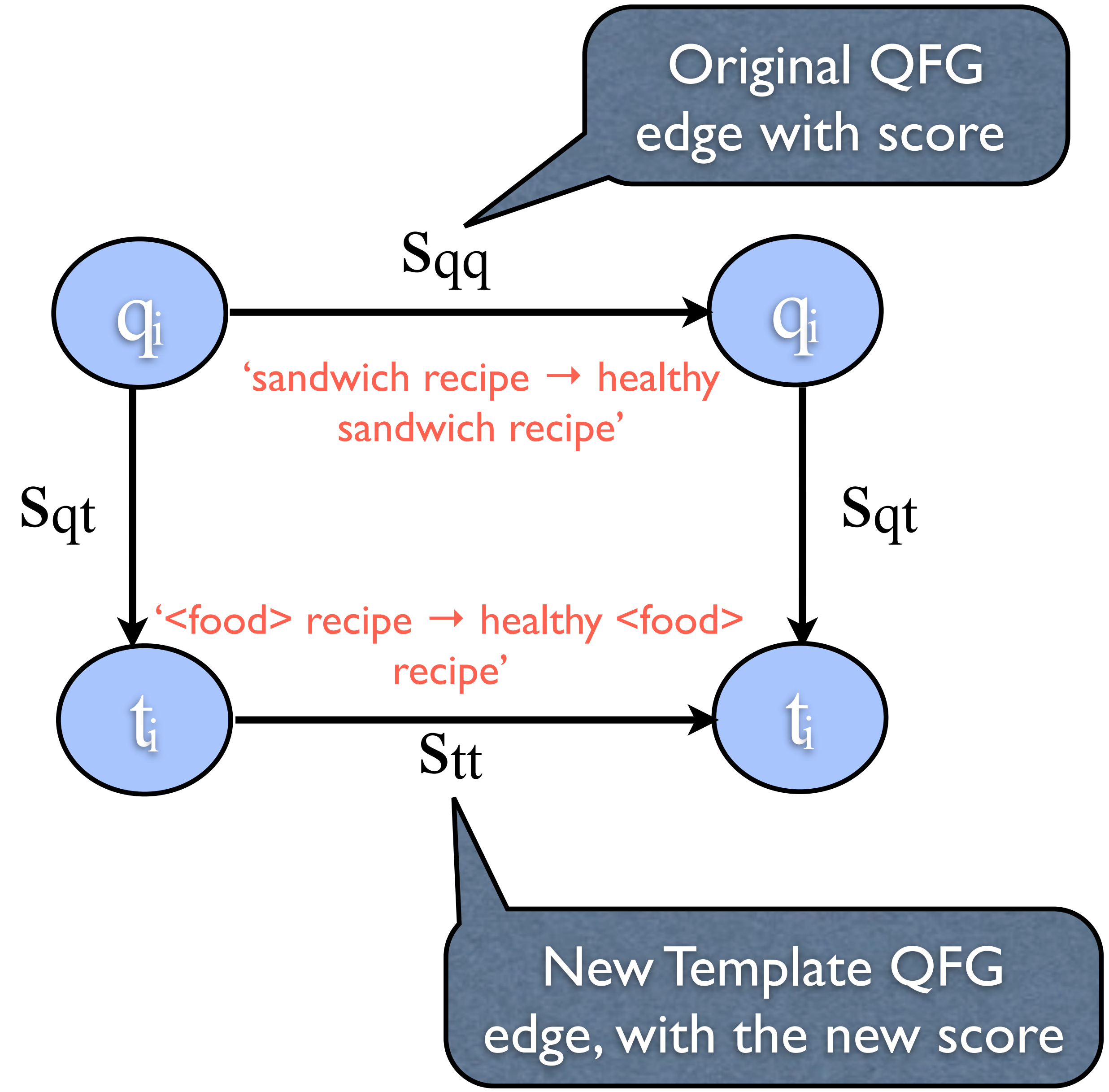
<food> cookie recipe
<drink> cookie recipe
<food> recipe
<substance> recipe
chocolate cookie <instruction>

- Fourth, mine the frequent transitions (rules) among templates, and build the Query-Template Flow Graph

I. Szpektor, A. Gionis, Y. Maarek, **"Improving recommendation for long-tail queries via templates"** in WWW 2011.
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **The query-flow graph: model and applications.** CIKM 2008: 609-618

Query suggestion

- How do we recommend
  - given query "Adele Astaire", generalize by using: 'Adele Astaire → <artist>'
  - suggest "Adele Astaire biography", based on the transition: '<artist> → <artist> biography'
- Note that the transition 'Adele Astaire → Adele Astaire biography' does not appear in the log, and then in the original QFG



I. Szpektor, A. Gionis, Y. Maarek, **"Improving recommendation for long-tail queries via templates"** in WWW 2011.
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **The query-flow graph: model and applications.** CIKM 2008: 609-618

Query suggestions

- User study
  - 100 queries, for which no recommendation could be provided by QFG (queries that were not seen before), were randomly sampled from the test-query-log
  - one of the top 10 recommendations by QTFG was randomly selected for each, and the accuracy of suggestions was 94.38%
- Prediction wrt a test query log
  - Extract pairs $\{q_i, q_{i+1}\}$ from a new test query log
  - How many of pairs could be proposed by the recommendation system based on QTFG?
    - upper-bound coverage (QFG vs. QTFG): 22.655 vs. 28.17%
  - Mean Average Precision (MAP) of the test-pairs, viewing no more than 100 recommendations per query (if q_{i+1} is not in the top 100, its precision is 0).
    - MAP (QFG vs. QTFG): 0.050 vs. 0.137

I. Szpektor, A. Gionis, Y. Maarek, **"Improving recommendation for long-tail queries via templates"** in WWW 2011.
P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, S. Vigna: **The query-flow graph: model and applications.** CIKM 2008: 609-618

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.* aims to optimize the categorization of 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 user history
  - This user-based categorization can be used to personalize results, since it aims to highlight 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*

- *User Profile*
  - User 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

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

- *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.
Not only queries/documents, but also categories, can be 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

- 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 partitions, 10 tests, each time choosing a partition for testing, and the other 9 for training

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 c_i, i.e. an integer ranging from 1 to K (3), computed using function $sim(q; c_i)$ *(cosine function)*

$ideal\_rank_{ci}$ 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

- *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

- 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 also 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

- 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

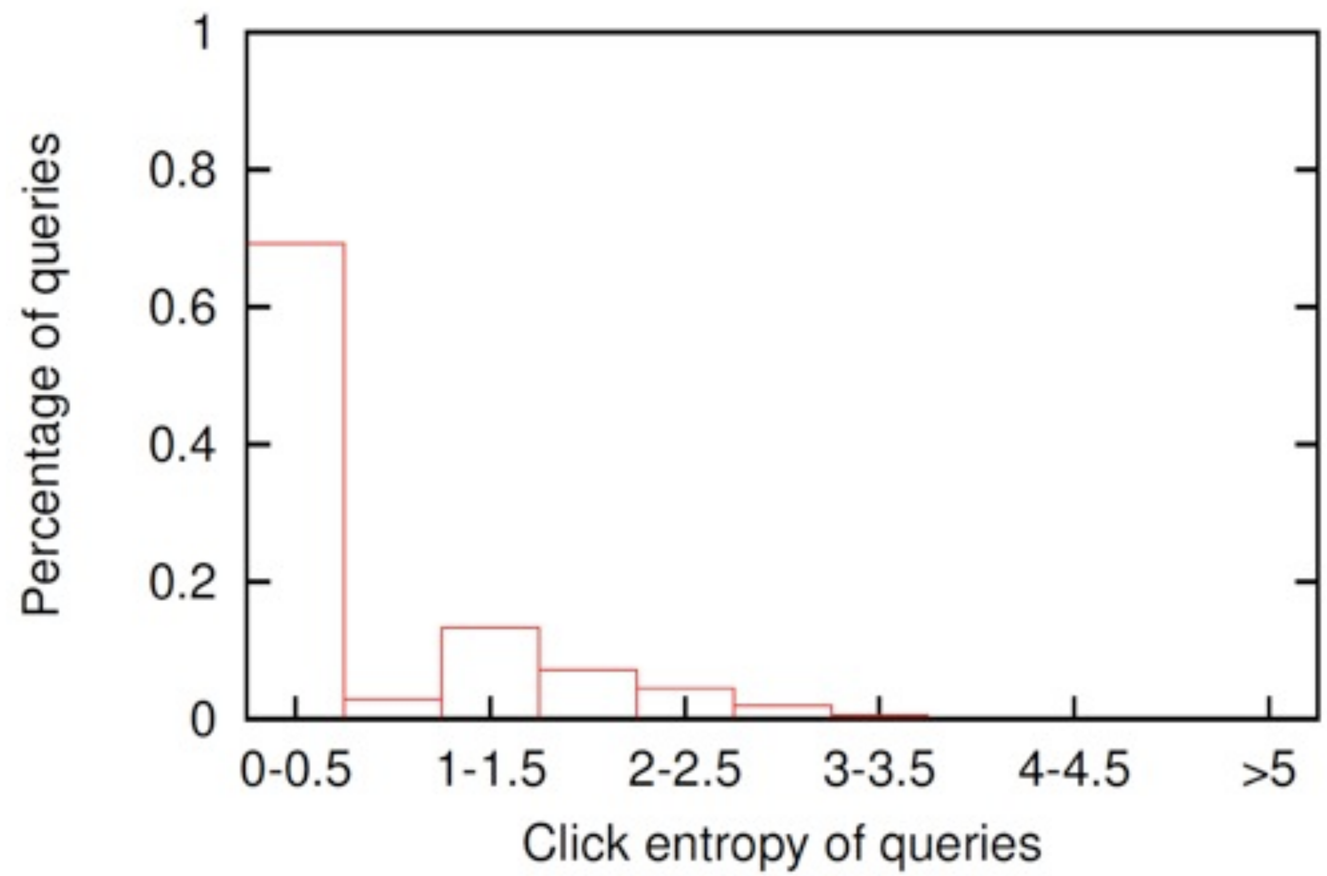
$$\text{ClickEntropy}(q) = \sum_{p \in \mathcal{P}(q)} -P(p|q) \log_2 P(p|q)$$

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

- *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

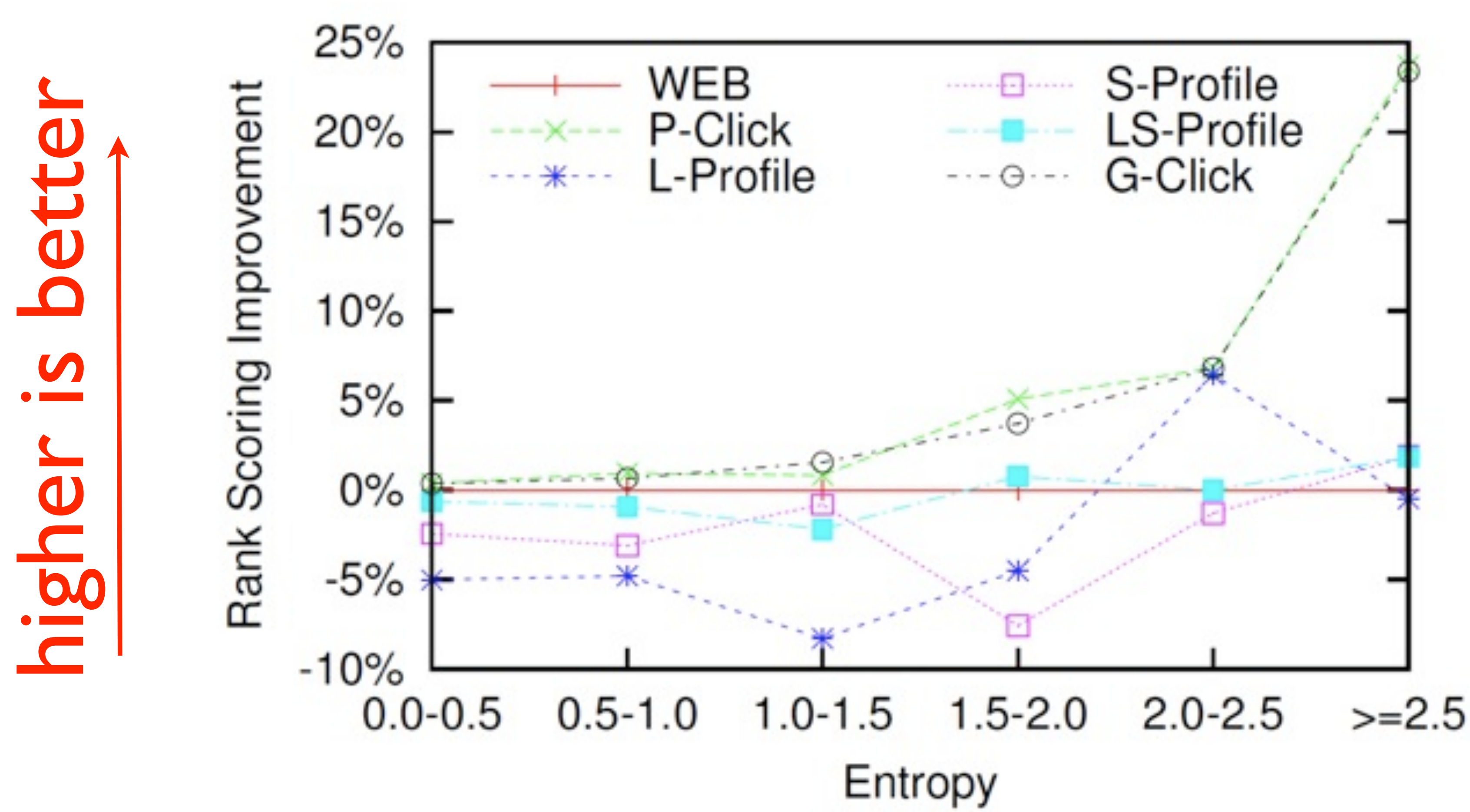
Query personalization



- 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



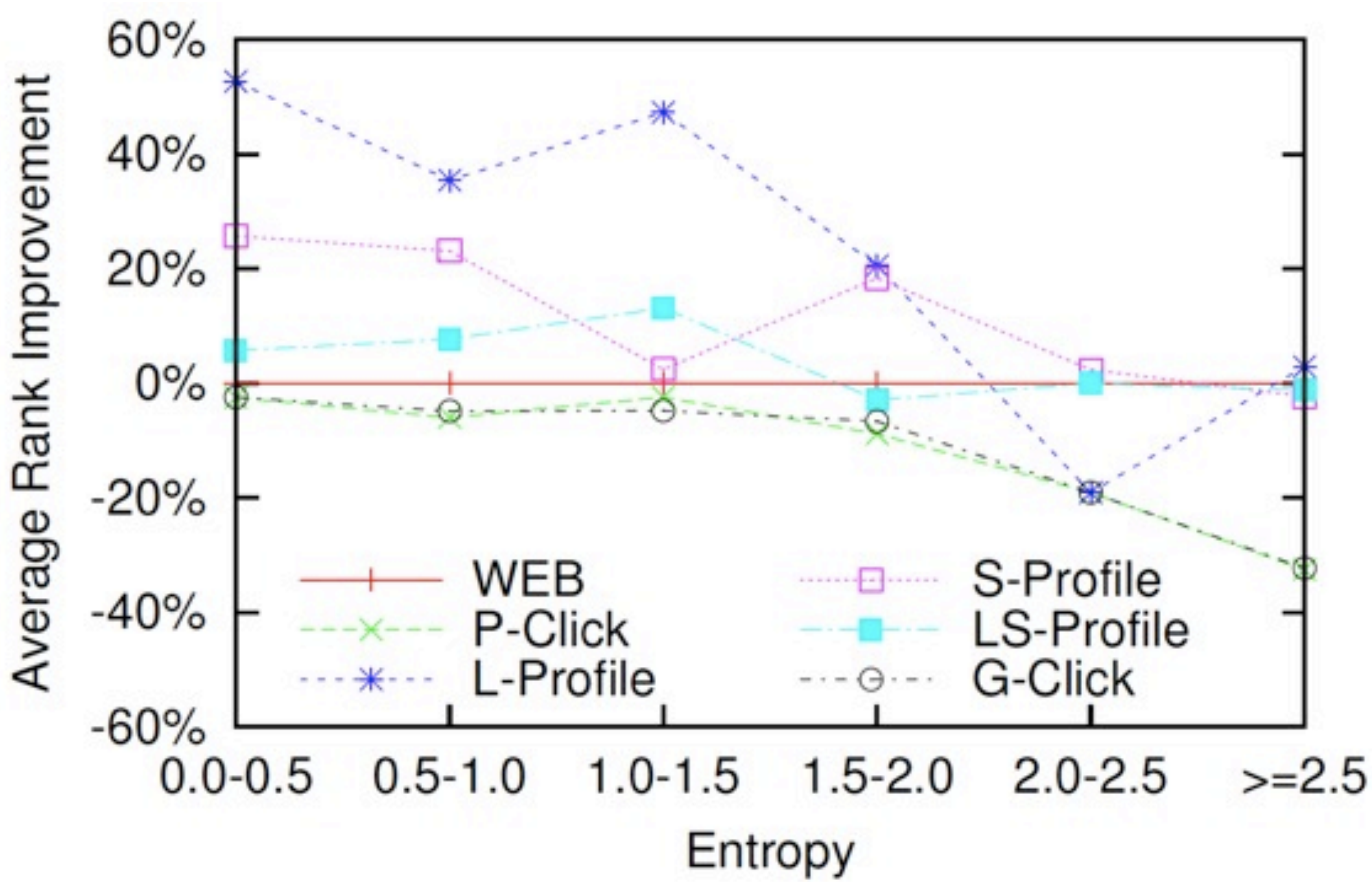
- Both Ranking scoring, and Average Rank perform better at higher entropy levels
    - P-Click and G-Click resulted the best ones

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
- We deal with "learning to rank" methods that make use of query logs as knowledge base to learn the optimal rank of the results

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 considered 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)

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

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

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

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

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

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

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:

~~$p_{11}, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}$~~

$p^*_{21}, p_{22}, p^*_{23}, p_{24}, p^*_{25},$ ~~p_{26}, p_{27}~~

$p_{31}, p^*_{32},$ ~~$p_{33}, p_{34}, p_{35}, p_{26}, p_{37}$~~

$p^*_{41},$ ~~$p_{42}, p_{43}, p_{44}, p_{45}, p_{36}, p_{47}$~~

- *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 $D=\{d1,..., dm\}$, 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 $D \times D$
- 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

- We need to define a similarity between the two orderings:
  - r^* => obtained by the clickthrough data
  - $r_{f(q)}$ => learnt ordering
- We need a metric that measures the similarity between two ranked lists => *Kendall's distance* 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, $\tau < 1$

  - 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

- *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*

$$(\mathrm{q}_1, \mathrm{r}_1^*), (\mathrm{q}_2, \mathrm{r}_2^*), \ldots, (\mathrm{q}_n, \mathrm{r}_n^*)$$

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.