Chapter 5
Query Operations

Motivation - Feast or famine

- Queries return ***either too few or too many results***
- Users are generally looking for ***the best document*** with a particular piece of information
- Users don't want to look through hundreds of documents to locate the information

$\Rightarrow$ Rank documents according to expected relevance!

Relevance Feedback

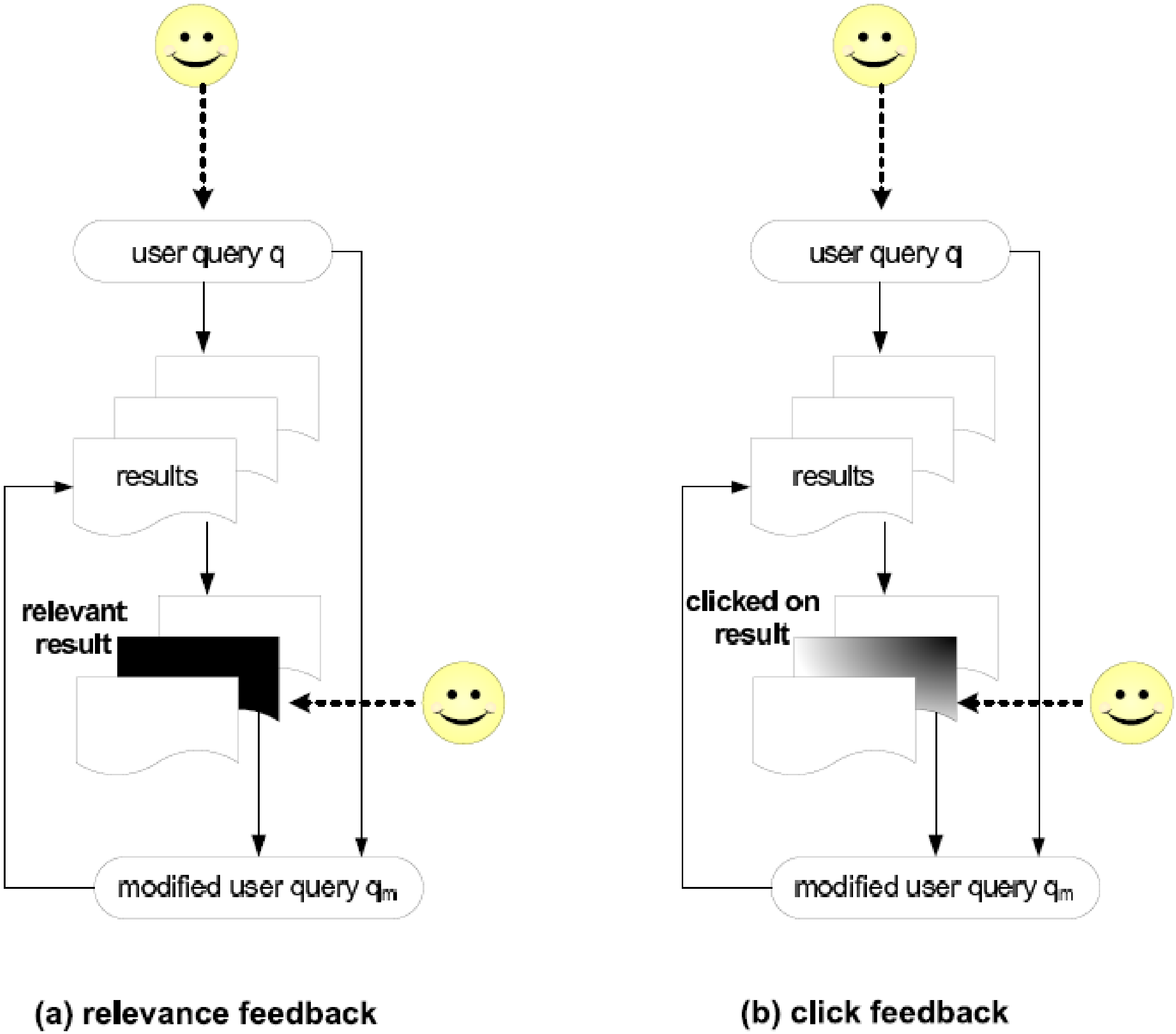
Queries

- Most queries are short
  - One to three words
- Many queries are ambiguous
  - "Saturn"
    - Saturn the planet?
    - Saturn the car?

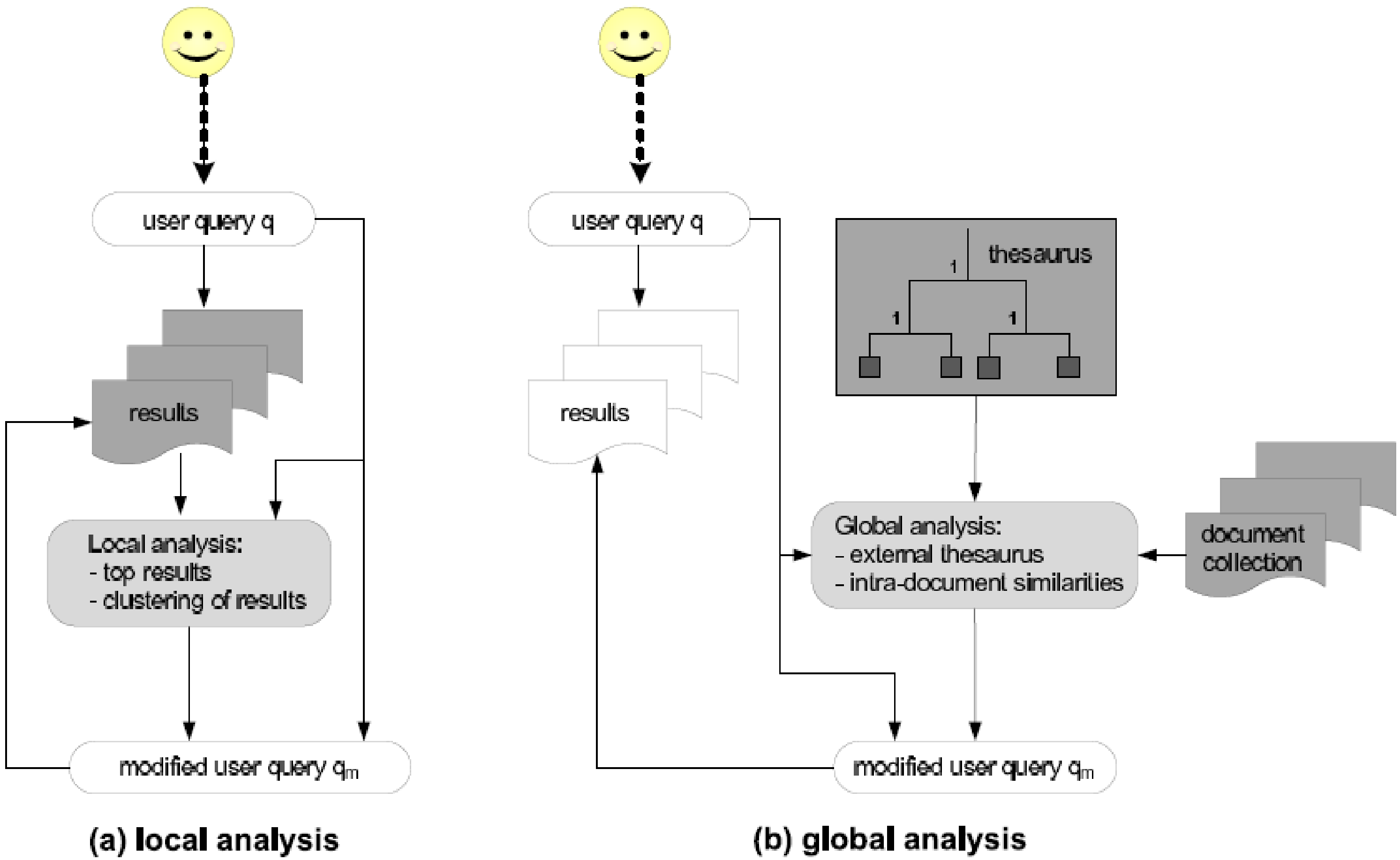
Relevance Feedback

- Two general approaches:
  - Create new queries with ***user feedback*** ***(explicit feedback)***
  - Create new queries ***automatically*** ***(implicit feedback)***
- Re-compute document weights with new information
- Expand or modify the query to more accurately reflect the user's desires

Explicit Feedback



Implicit Feedback



User Feedback

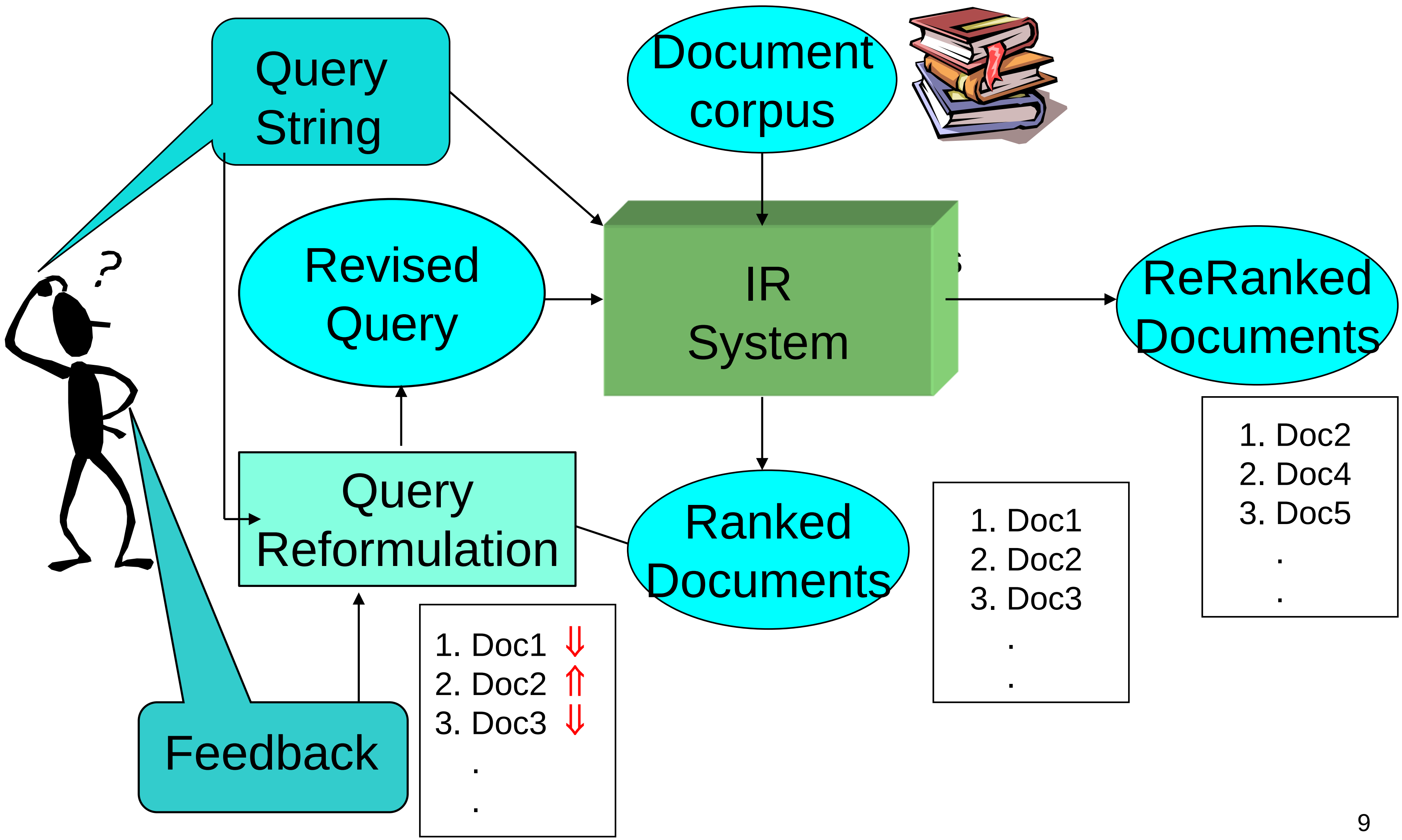
- After initial retrieval results are presented, allow the user to provide feedback on the relevance of one or more of the retrieved documents.
- Use this feedback information to reformulate the query.
- Produce new results based on reformulated query.
- Allows more interactive, ***multi-pass process***.

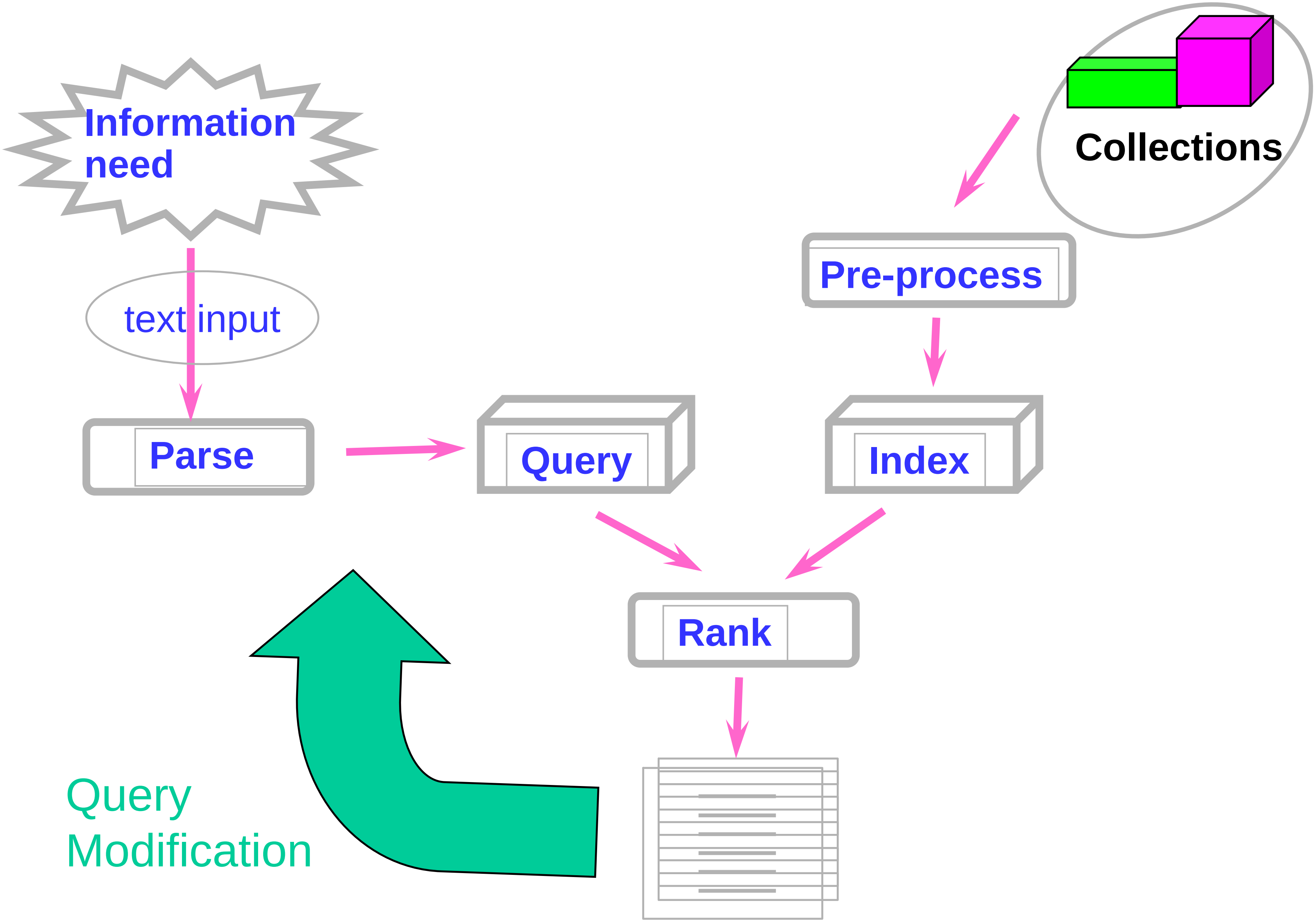
User Feedback

The main idea consists of

- selecting important terms from the documents that have been identified as relevant, and
- enhancing the importance of these terms in a new query formulation

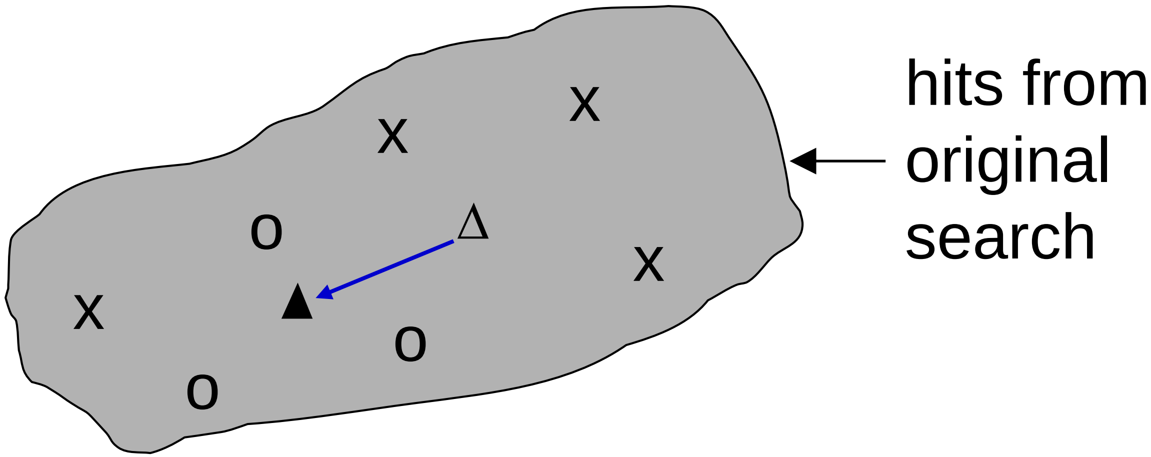
User Feedback Architecture





Information need
Collections
Pre-process
text input
Parse
Query
Index
Rank
Query Modification

User Feedback (concept)



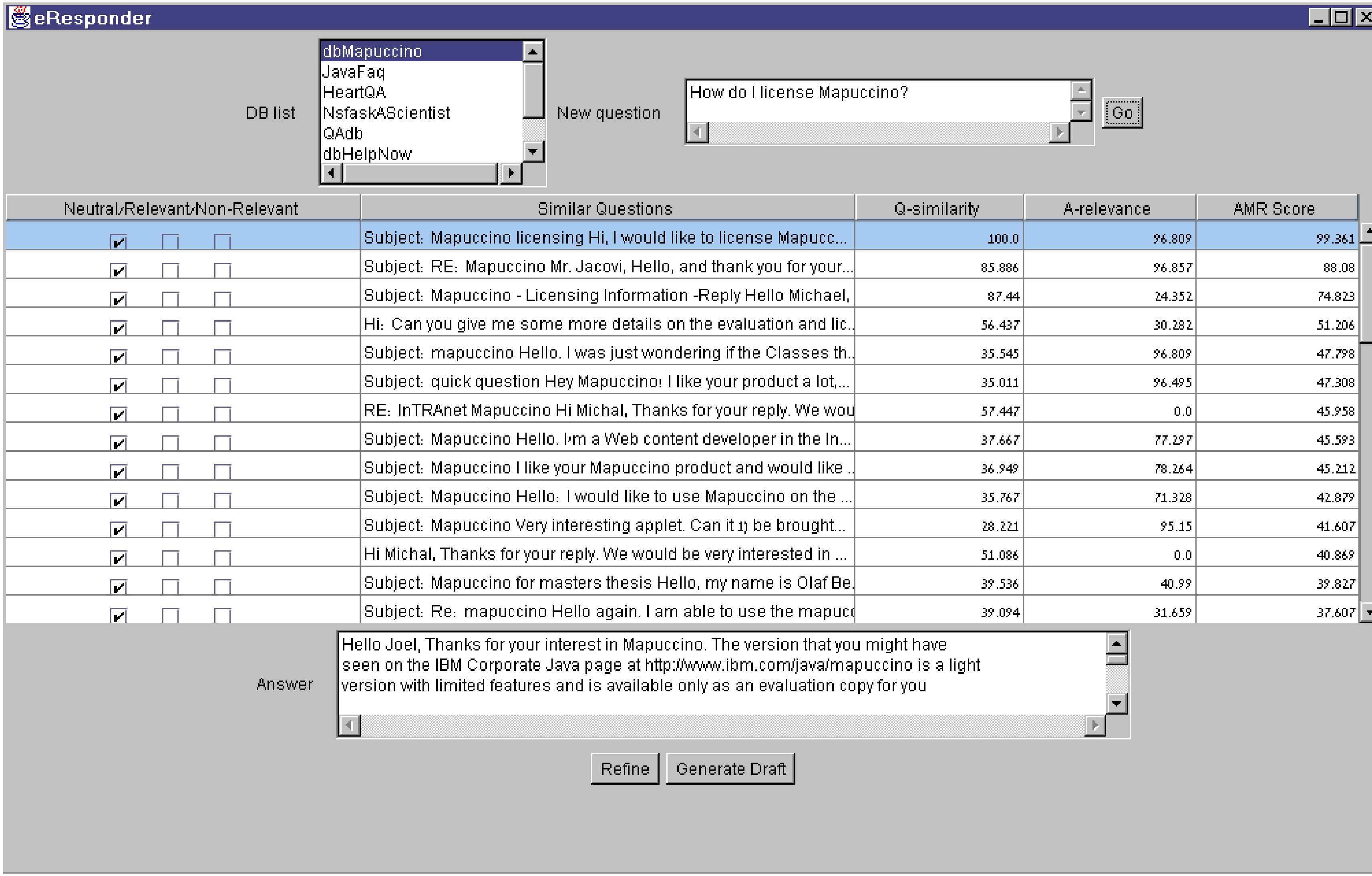
x documents identified as non-relevant

o documents identified as relevant

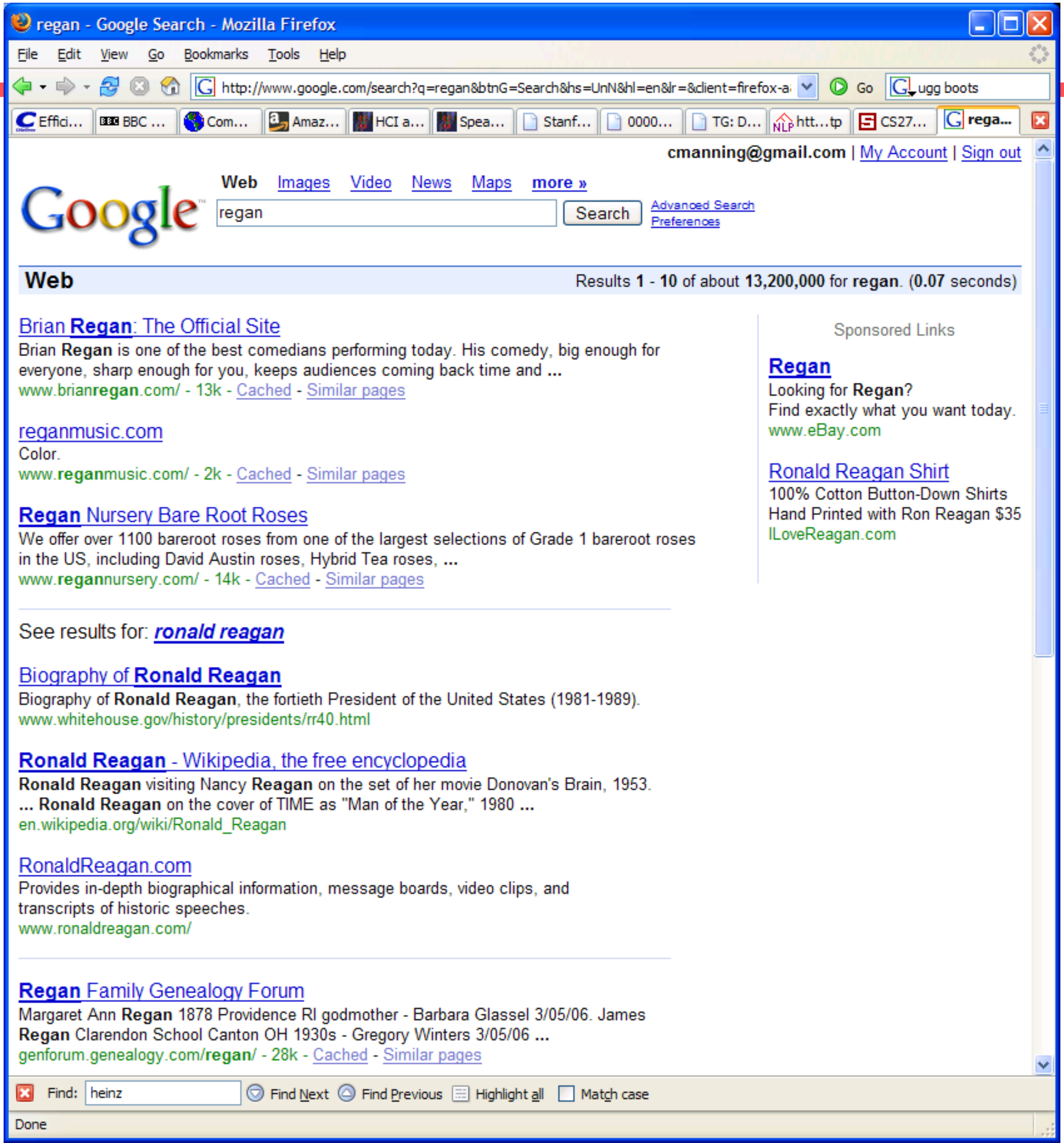
▲ original query

Δ reformulated query

User Feedback (concept)

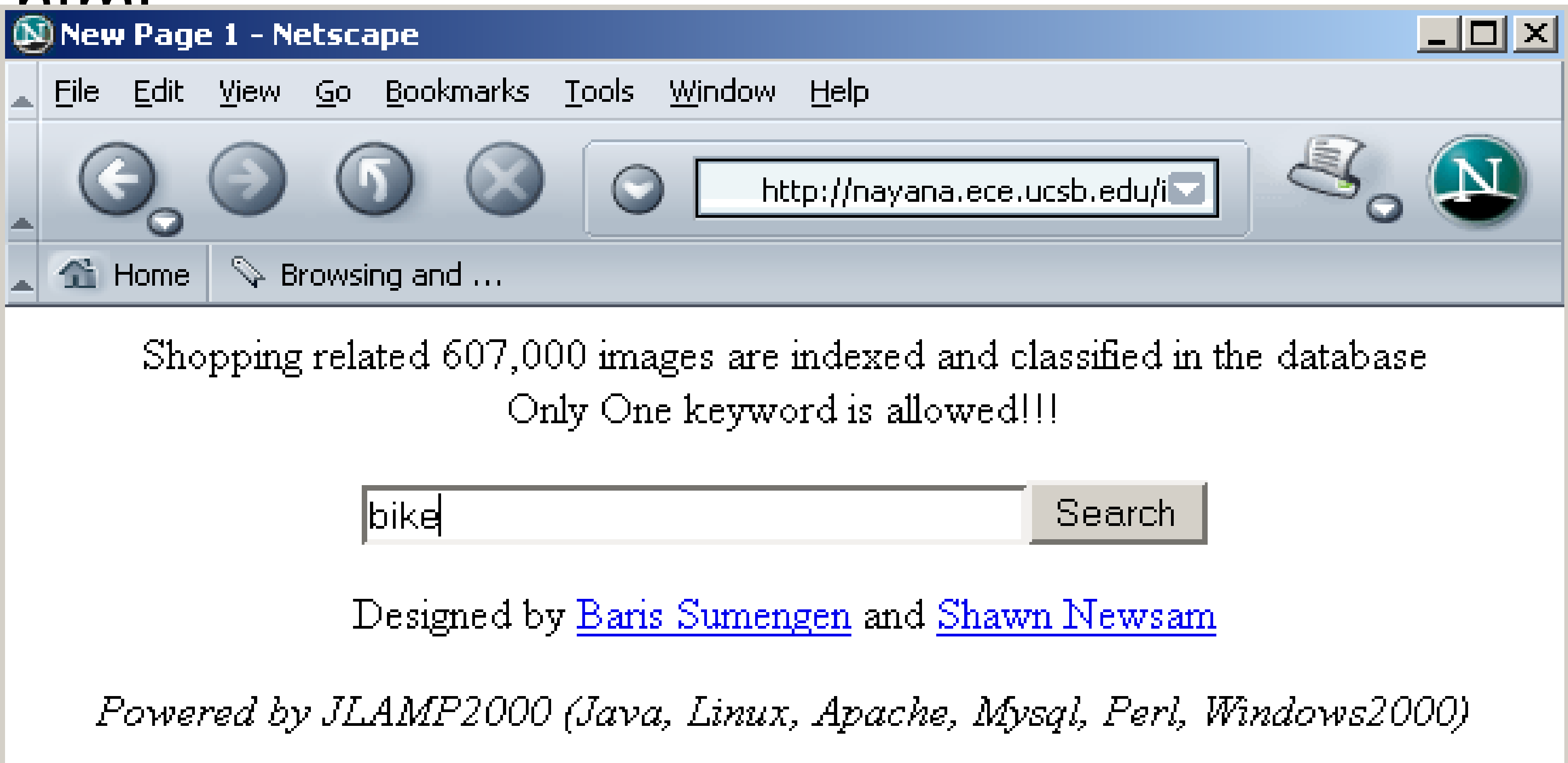


User Feedback (concept)

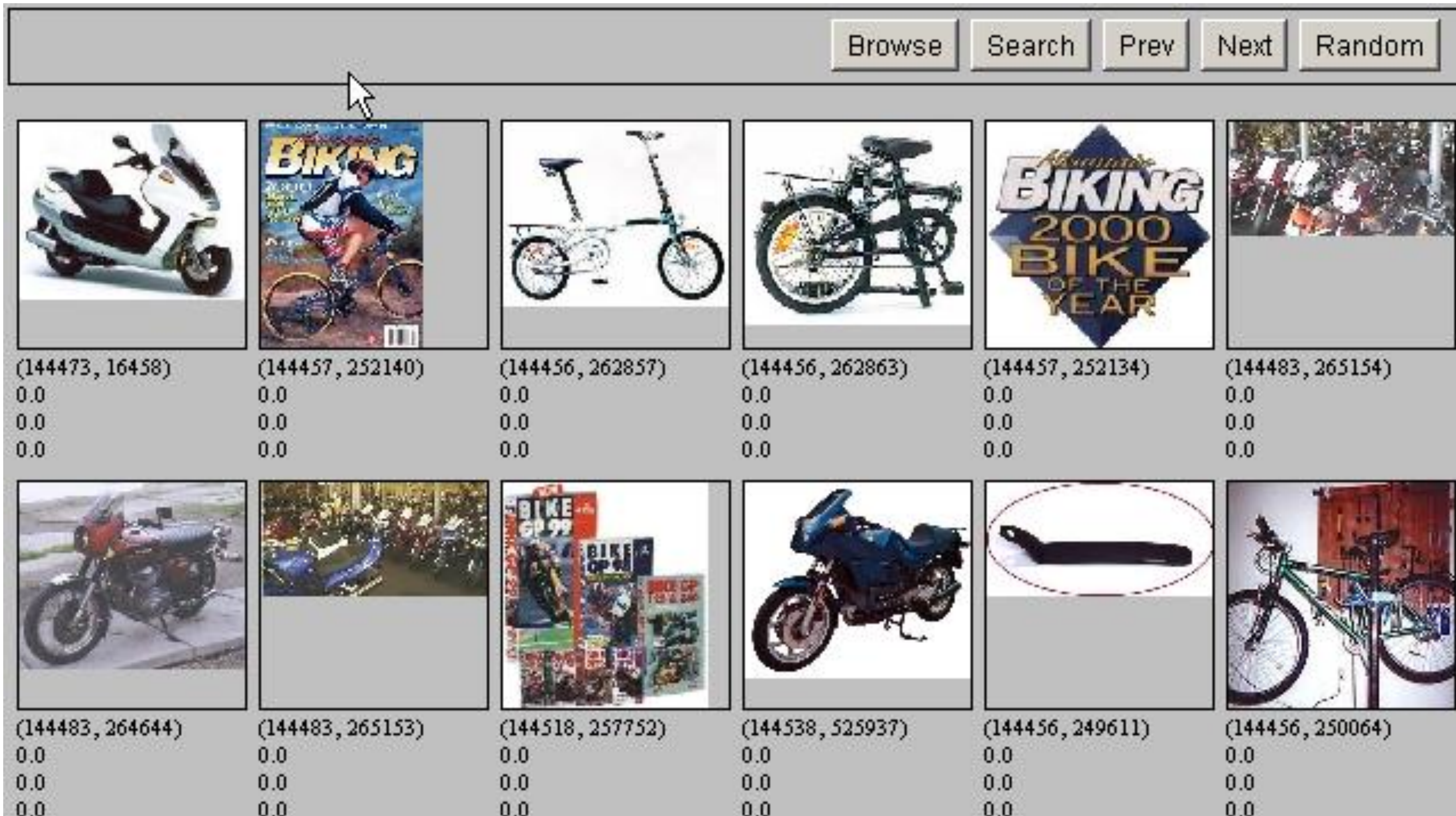


User Feedback: Example

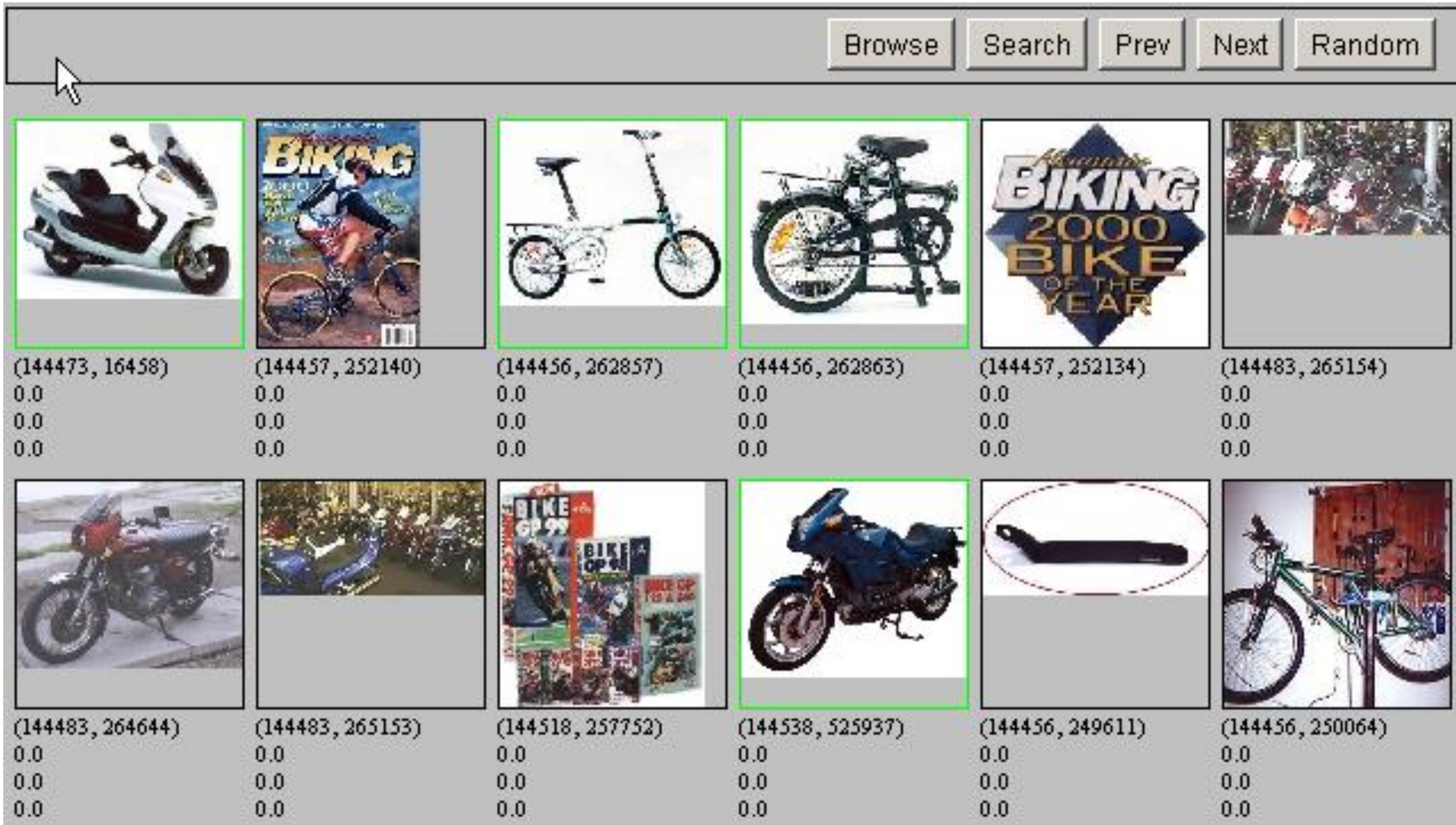
- Image search engine
  http://nayana.ece.ucsb.edu/imsearch/imsearch.html



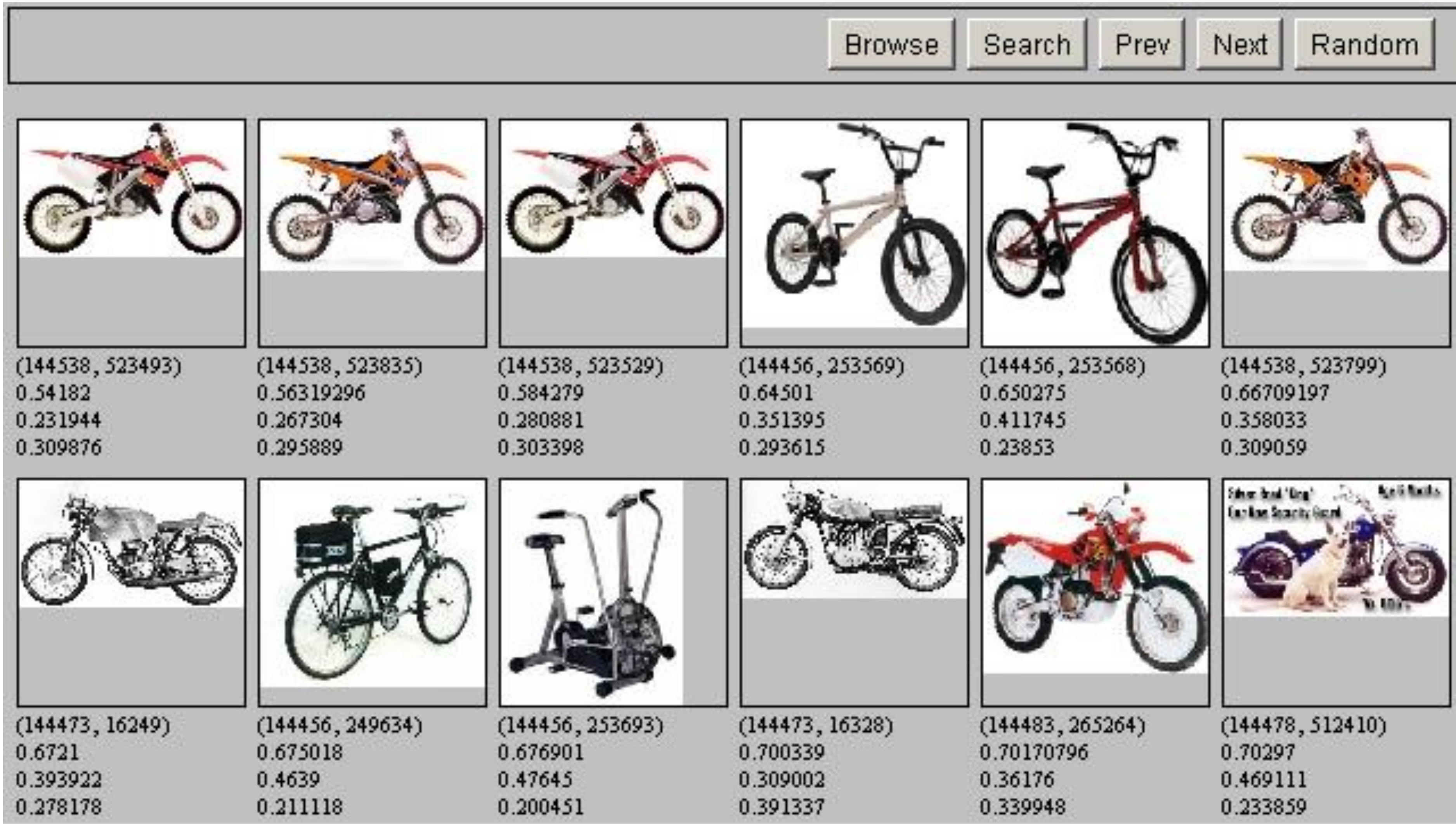
Results for Initial Query



User Feedback



Results after User Feedback



Query Reformulation

- Revise query to account for feedback:
  - ***Query Expansion***: Add new terms to query from relevant documents.
  - ***Term Reweighting***: Increase weight of terms in relevant documents and decrease weight of terms in irrelevant documents.
- Several algorithms for query reformulation.

Query Reformulation

- Change query vector using vector algebra.
- **Add** the vectors for the **relevant** documents to the query vector.
- **Subtract** the vectors for the **irrelevant** docs from the query vector.

Vector Space Re-Weighting

Rochio:

- $\boldsymbol{q'} = \alpha \boldsymbol{q} + (\beta/|\mathbf{D}_r|)\sum_{d_i \in \mathbf{D}_r} \boldsymbol{d}_i - (\gamma/|\mathbf{D}_n|)\sum_{d_i \in \mathbf{D}_n} \boldsymbol{d}_i$

Ide regular

- $\boldsymbol{q'} = \alpha \boldsymbol{q} + \beta\sum_{d_i \in \mathbf{D}_r} \boldsymbol{d}_i - \gamma\sum_{d_i \in \mathbf{D}_n} \boldsymbol{d}_i$

Ide Dec_hi

- $\boldsymbol{q'} = \alpha \boldsymbol{q} + \beta\sum_{d_i \in \mathbf{D}_r} \boldsymbol{d}_i - \gamma \max_{d_i \in \mathbf{D}_n} (\boldsymbol{d}_i)$

Rocchio Method

$$Q_1 = \alpha \; Q_0 + \frac{\beta}{n_1} \sum_{\forall d_j \in D_r} \vec{d}_j - \frac{\gamma}{n_2} \sum_{\forall d_j \in D_n} \vec{d}_j$$

where

Q_0 = the vector for the initial query

D_r = the set of relevant documents

D_n = the set of non-relevant documents

n_1 = the number of relevant documents chosen

n_2 = the number of non-relevant documents chosen

α, β and γ tune importance of relevant and nonrelevant terms

(in some studies best to set α to 1 β to 0.75 and γ to 0.25)

Example Rocchio Calculation

$$R_1 = (0.030,0,0,0.025,0.025,0.050,0,0,0.120)$$
$$R_2 = (0.020,0.009,0.020,0.002,0.050,0.025,0.100,0.100,0.120)$$

Relevant docs

$$S_1 = (0.030,0.010,0.020,0,0.005,0.025,0,0.020,0)$$

Non-rel doc

$$Q = (0,0,0,0,0.500,0,0.450,0,0.950)$$

Original Query

$$\alpha = 1$$
$$\beta = 0.75$$
$$\gamma = 0.25$$

Constants

$$Q_{new} = \alpha \times Q + \left(\frac{\beta}{2} \times (R_1 + R_2)\right) - \left(\frac{\gamma}{1} \times S_1\right)$$

Rocchio Calculation

$$Q_{new} = (0.011,0.000875,0.002,0.01,0.527,0.022,0.488,0.033,1.04)$$

Resulting feedback query

Rocchio Method - summary

- Rocchio automatically
  - re-weights terms
  - adds in new terms (from relevant docs)
    - have to be careful when dealing with negative terms
  - known to significantly improve results
- Quality
  - heavily dependent on test collection
  - heavily dependent on relevance quality

Ide Regular Method

- Since more feedback should perhaps increase the degree of reformulation, do not normalize for amount of feedback:

$$\vec{q}_1 = \alpha \vec{q}_0 + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

α: Tunable weight for initial query.
β: Tunable weight for relevant documents.
γ: Tunable weight for irrelevant documents.

Relevance Feedback

$$\vec{q}_m = \vec{q} + \alpha \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \beta \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

Original Query : (5,0,3,0,1)

Document D_1 Relevant : (2,1,2,0,0)

Document D_2 Nonrelevant : (1,0,0,0,2)

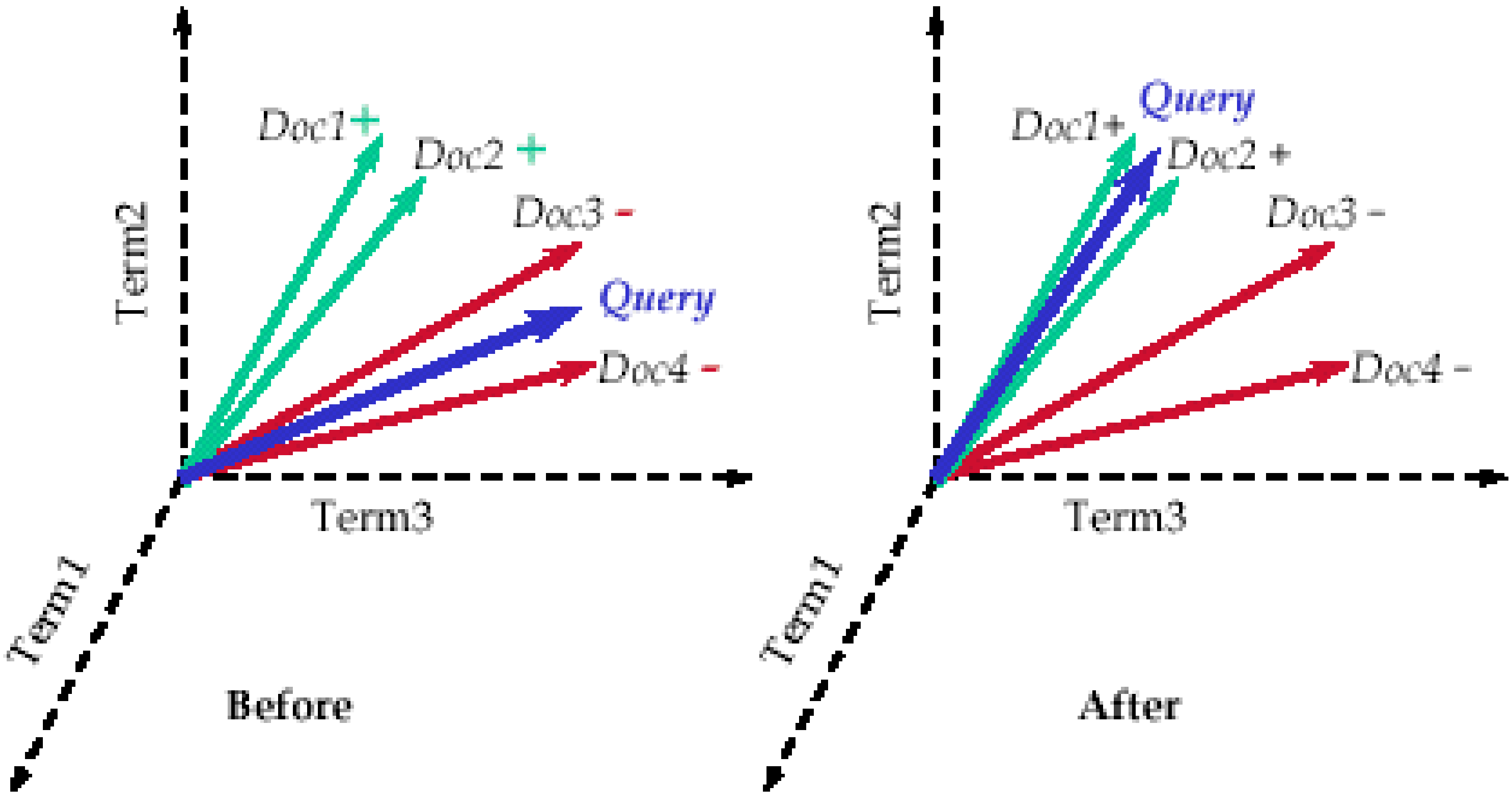
α **= 0.50** β **= 0.25**

$$q' = q + 0.5D_1 - 0.25D_2$$

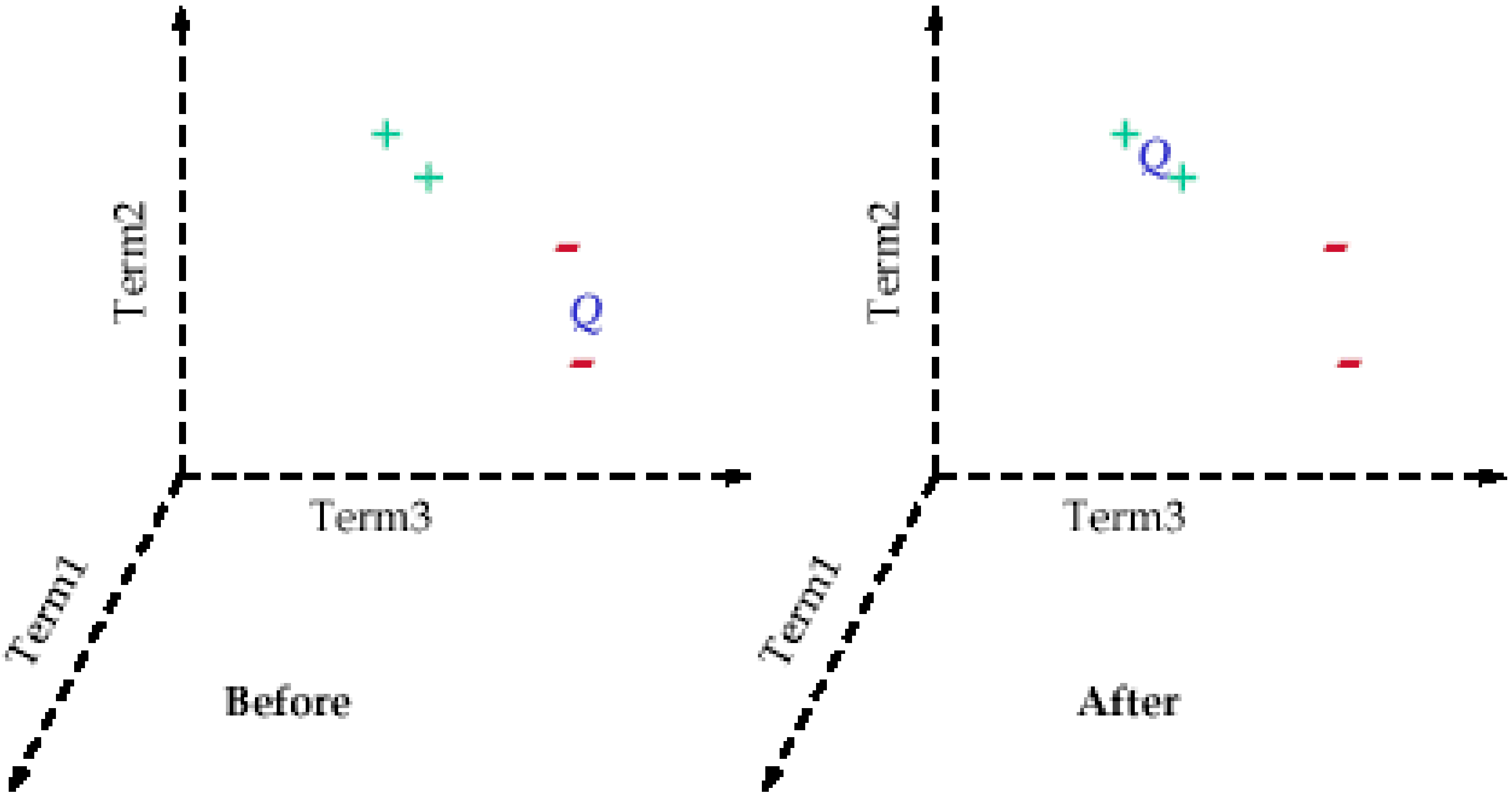
= (5,0,3,0,1) + 0.5(2,1,2,0,0) - 0.25(1,0,0,0,2)

= (5.75, 0.50, 4.0, 0.0, 0.5)

Relevance Feedback

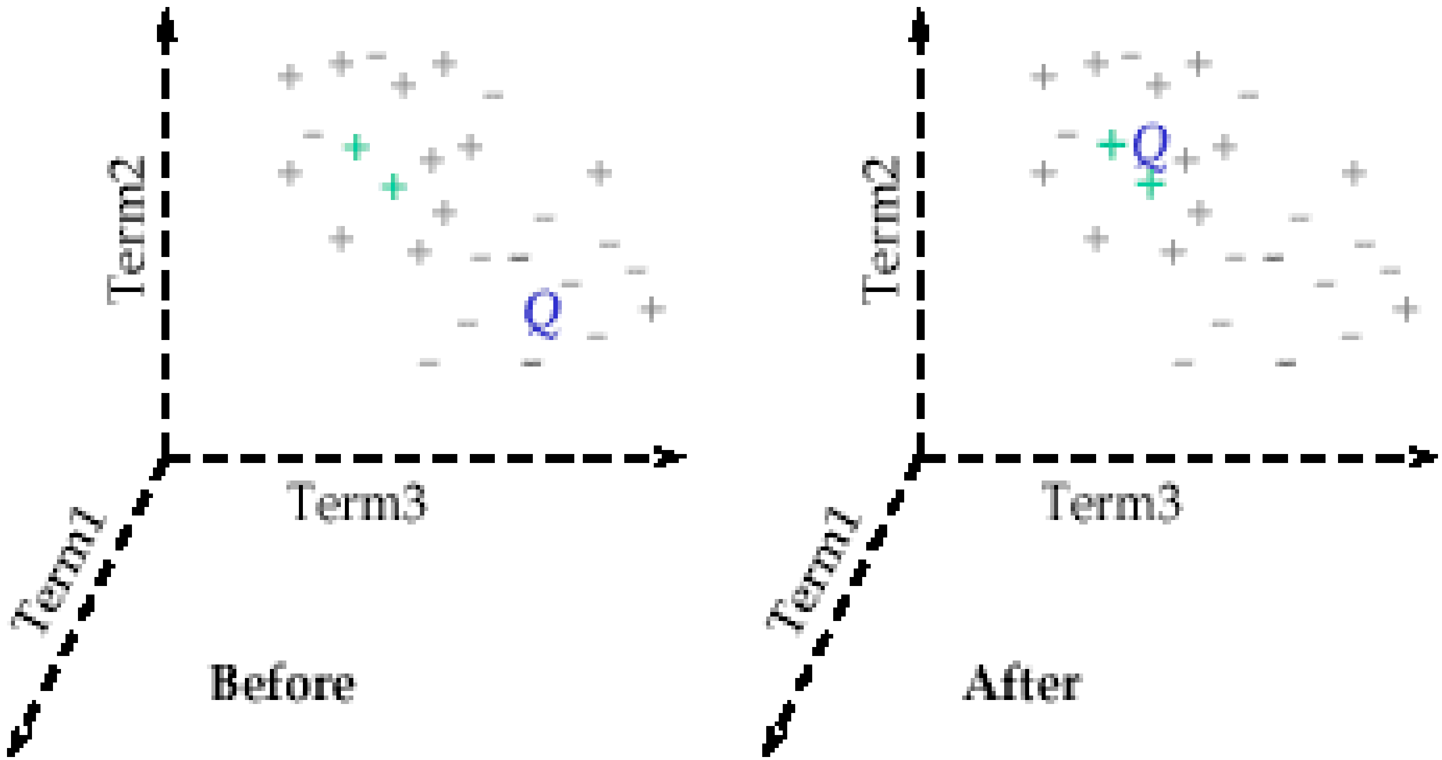


Relevance Feedback

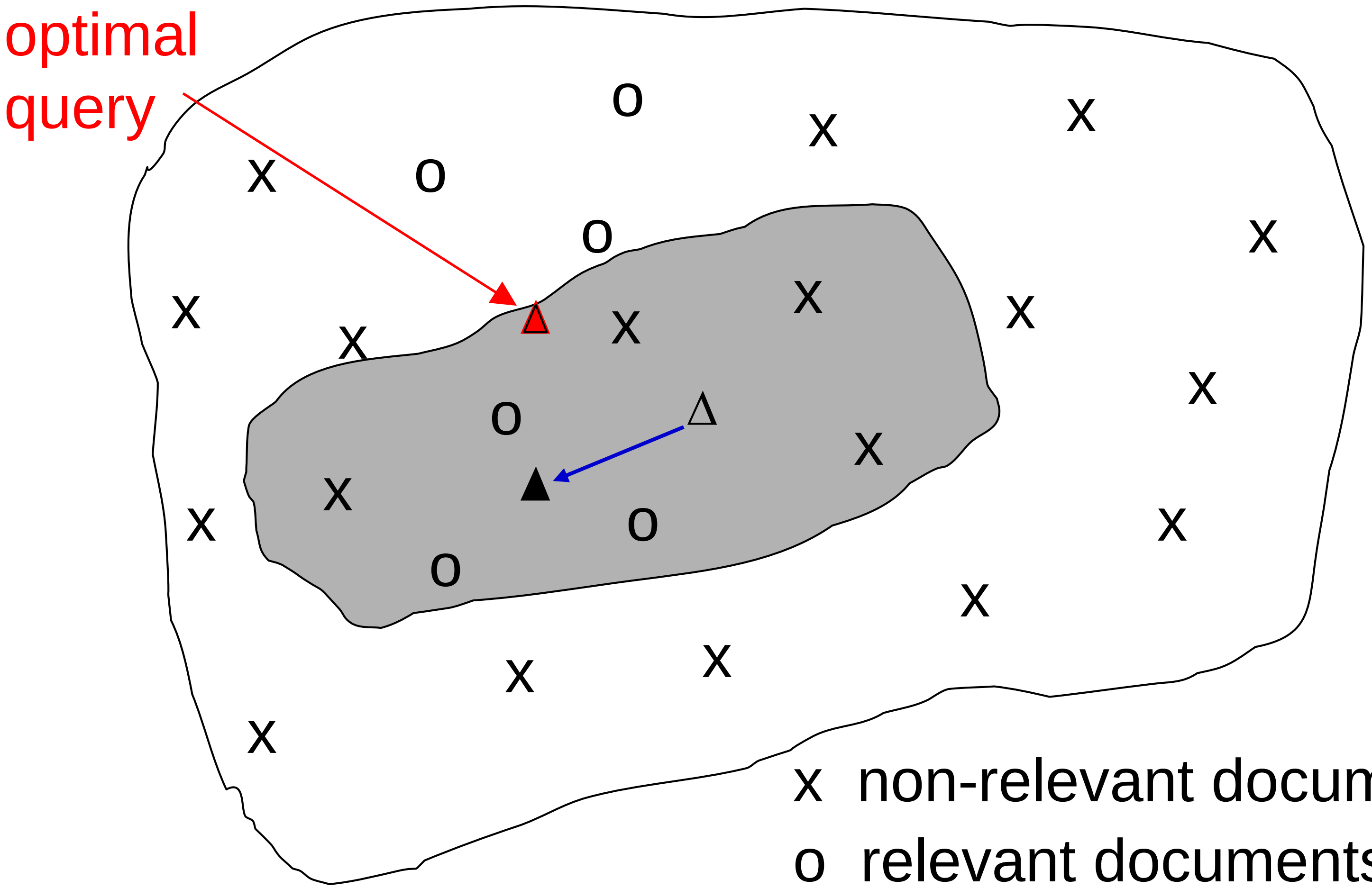


Relevance Feedback

How can relevance feedback save time if a person has to read documents?



Difficulties with Relevance Feedback



Hits from the initial query are contained in the gray shaded area

x non-relevant documents
o relevant documents
Δ original query
▲ reformulated query

Vector Space Re-Weighting

- The initial query vector $\boldsymbol{q}_0$ will have non-zero weights only for terms appearing in the query

- The query vector update process can add weight to terms that don't appear in the original query

- Some terms can **end up** having **negative** weight!
  - E.g., if you want to find information on the planet Saturn, "car" could have a negative weight…

Automatically (Implicit)

- **Automatic Global Analysis**
- **Automatic Local Analysis**

Automatic Global Analysis

- A thesaurus-like structure
- Short history
  - Until the beginning of the 1990s, global analysis was considered to be a technique which failed to yield consistent improvements in retrieval performance with ***general collections***
  - This perception has changed with the appearance of modern procedures for ***global analysis***

Query Expansion based on a Similarity Thesaurus

- **Idea by Qiu and Frei [1993]**
  - Similarity thesaurus is based on ***term to term relationships*** rather than on a matrix of co-occurrence
  - Terms for expansion are selected based on ***their similarity to the whole query*** rather than on their similarities to individual query terms
- **Definition**
  - N: total number of documents in the collection
  - t: total number of terms in the collection
  - $tf_{i,j}$: occurrence **frequency of term k_i** in the document d_j
  - t_j: the number of distinct index terms in the document d_j
  - itf_j : the inverse **term frequency** for document d_j

$$itf_j = \log \frac{t}{t_j}$$

Term weighting vs. Term concept space

$$
\begin{array}{ccccc}
 & K_1 & K_2 & \dots & K_t \\
D_1 & w_{11} & w_{21} & \dots & w_{t1} \\
D_2 & w_{12} & w_{22} & \dots & w_{t2} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
D_n & w_{1n} & w_{2n} & \dots & w_{tn}
\end{array}
\qquad
\begin{array}{ccccc}
 & D_1 & D_2 & \dots & D_n \\
K_1 & w_{11} & w_{12} & \dots & w_{1n} \\
K_2 & w_{21} & w_{22} & \dots & w_{2n} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
K_t & w_{t1} & w_{t2} & \dots & w_{tn}
\end{array}
$$

$$
w_{i,j} = \frac{(0.5 + 0.5\frac{tf_{i,j}}{\max_k\{tf_{k,j}\}})idf_i}{\sqrt{\sum_{k=1}^{t}(0.5 + 0.5\frac{tf_{k,j}}{\max_k\{tf_{k,j}\}})^2 idf_k^2}}
\qquad
w_{i,j} = \frac{(0.5 + 0.5\frac{tf_{i,j}}{\max_k\{tf_{i,k}\}})itf_j}{\sqrt{\sum_{k=1}^{N}(0.5 + 0.5\frac{tf_{i,k}}{\max_k\{tf_{i,k}\}})^2 itf_k^2}}
$$

$$
idf_i = \log\frac{N}{n_i}
\qquad
itf_j = \log\frac{t}{t_j}
$$

Similarity Thesaurus

- Each term is associated with a vector

$$\vec{k_i} = (w_{i,1}, w_{i,2}, \cdots, w_{i,N})$$

  - where **$w_{i,j}$ is a weight** associated to the index-document pair

$$w_{i,j} = \frac{(0.5 + 0.5 \frac{tf_{i,j}}{\max_k \{tf_{i,k}\}}) itf_j}{\sqrt{\sum_{k=1}^{N} (0.5 + 0.5 \frac{tf_{i,k}}{\max_k \{tf_{i,k}\}})^2 itf_k^2}}$$

- The ***relationship between two terms k_u and k_v*** is

$$c_{u,v} = \vec{k_u} \bullet \vec{k_v} = \sum_{j=1}^{N} w_{u,j} \times w_{v,j}$$

Query Expansion Procedure with Similarity Thesaurus

1. Represent the query in the concept space by using the representation of the index terms

$$\vec{q} = \sum_{k_u \in q} w_{u,q} \overrightarrow{k_u}$$

2. Compute the similarity sim(q,k_v) between each term k_v and the whole query

$$sim(q,k_v) = \vec{q} \bullet \overrightarrow{k_v} = \left(\sum_{k_u \in q} w_{u,q} \overrightarrow{k_u} \right) \bullet \overrightarrow{k_v} = \sum_{k_u \in Q} w_{u,q} \times c_{u,v}$$

3. Expand the query with the top r ranked terms according to $sim(q,k_v)$

$$w_{v,q'} = \frac{sim(q,k_v)}{\sum_{k_u \in q} w_{u,q}}$$

Query Expansion based on a Similarity Thesaurus

- A document d_j is represented term-concept space by $$\overrightarrow{d_j} = \sum_{k_v \in d_j} w_{v,j} \times \overrightarrow{k_v}$$
- If the original query q is expanded to include all the t index terms, then the similarity sim(q, d_j) between the document d_j and the query q can be computed as

$$sim(\vec{q}, \overrightarrow{d_j}) = \left(\sum_{k_u \in q} w_{u,q} \times \overrightarrow{k_u} \right) \bullet \left(\sum_{k_v \in d_j} w_{v,j} \times \overrightarrow{k_v} \right)$$

$$sim(\vec{q}, \overrightarrow{d_j}) = \sum_{k_v \in d_j} \sum_{k_u \in q} w_{v,j} \times w_{u,q} \times c_{u,v}$$

  - which is similar to the generalized vector space model

Automatic Global Analysis Example

$$
\begin{array}{cccccc}
 & D_1 & D_2 & \ldots. & D_n \\
K_1 & w_{11} & w_{12} & \ldots & w_{1n} \\
K_2 & w_{21} & w_{22} & \ldots & w_{2n} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
K_t & w_{t1} & w_{t2} & \ldots & w_{tn}
\end{array}
$$

$$
w_{i,j} = \frac{\left(0.5 + 0.5\frac{f_{i,j}}{\max_j(f_{i,j})}\right) itf_j}{\sqrt{\sum_{l=1}^{N}\left(0.5 + 0.5\frac{f_{i,j}}{\max_l(f_{i,l})}\right)^2 itf_l^2}}
$$

$$
itf_j = \log \frac{t}{t_j}
$$

Automatic Global Analysis Example

The relationship between two terms

C	1	2	3	…	m
1	C1,1	C1,2	C1,3		C1,m
2	C2,1	C2,2	C2,3	…	C2,m
3	C3,1	C3,2	C3,3	…	C3,m
…					
n	Cn,1	Cn,2	Cn,3	…	Cn,m

$$c_{u,v} = \overrightarrow{k_u} \bullet \overrightarrow{k_v} = \sum_{j=1}^{N} w_{u,j} \times w_{v,j}$$

Ex.

$$C_{1,3} = w_{1,1} * w_{3,1} + w_{1,2} * w_{3,2} + w_{1,3} * w_{3,3} + \ldots + w_{1,n} * w_{3,n}$$

Automatic Global Analysis Example

Original Query

$q = w_{1,q}K_1 + w_{2,q}K_2 + w_{3,q}K_3 + \ldots + w_{n,q}K_n$

•compute a similarity sim(q,kv) between each term kv correlated to the query terms and the whole query q

$$sim(q, k_v) = \vec{q} \cdot \vec{k}_v = \sum_{k_u \in q} w_{u,q} \times c_{u,v}$$

EX.

$sim(q,k_3) = w_{1,q} * c_{1,3} + w_{2,q} * c_{2,3} + w_{3,q} * c_{3,3} + \ldots + w_{n,q} * c_{n,3}$

Automatic Global Analysis Example

Arrange $sim(q,k_t)$

Ex.

$sim(q,k_1) = 0.53$

$sim(q,k_2) = 0.36$

$sim(q,k_3) = 3.98$

$sim(q,k_4) = 1.87$

$sim(q,k_3)$

$sim(q,k_4)$

$sim(q,k_1)$

$sim(q,k_2)$

$sim(q,k_2)$

$sim(q,k_4)$

$sim(q,k_3)$

$sim(q,k_1)$

Original Query

$q = K_1+K_4$

New Query

$q = K_1+K_3+K_4$

New Query

$q = K_1+K_2+K_3+K_4$

Automatic Global Analysis Example

Compute new weight terms for query

Original Query

$q = w_{1,q}K_1 + w_{2,q}K_2 + w_{3,q}K_3 + \ldots + w_{n,q}K_n$

$$w_{v,q'} = \frac{sim(q,k_v)}{\sum_{k_u \in q} w_{u,q}}$$

Ex.

$$w_{3,q'} = \frac{sim(q,k_3)}{(w_{1,q} + w_{2,q} + w_{3,q} + .. + w_{n,q})}$$

$w_{1,q'} = 2.6$

$w_{3,q'} = 5.4$

$w_{4,q'} = 4.8$

New Query

$q = 2.6K_1 + 5.4K_3 + 4.8K_4$

Automatic Global Analysis Example

Compute sim(q,d_j) for new relevance document

$$sim(q,d_j) \propto \sum_{k_v \in d_j} \sum_{k_u \in q} w_{i,j} \times w_{u,q} \times c_{u,v}$$

$sim(q,d_2) = w_{1,2}*w_{1,q}*c_{1,1} + w_{1,2}*w_{1,q}*c_{1,2} + w_{1,2}*w_{1,q}*c_{1,3} + .. + w_{1,2}*w_{1,q}*c_{1,m} +$

$w_{2,2}*w_{2,q}*c_{2,1} + w_{2,2}*w_{2,q}*c_{2,2} + w_{2,2}*w_{2,q}*c_{2,3} + .. + w_{2,2}*w_{2,q}*c_{2,m} +$

$w_{3,2}*w_{3,q}*c_{3,1} + w_{3,2}*w_{3,q}*c_{3,2} + w_{3,2}*w_{3,q}*c_{3,3} + .. + w_{3,2}*w_{3,q}*c_{3,m} +$

……

$w_{n,2}*w_{n,q}*c_{n,1} + w_{n,2}*w_{n,q}*c_{n,2} + w_{n,2}*w_{n,q}*c_{n,3} + .. + w_{n,2}*w_{n,q}*c_{n,m}$

$sim(q,d_2) = w_{1,2}*w_{1,q}*(c_{1,1} + c_{1,2} + c_{1,3} + .. + c_{1,m}) +$

$w_{2,2}*w_{2,q}*(c_{2,1} + c_{2,2} + c_{2,3} + .. + c_{2,m}) +$

$w_{3,2}*w_{3,q}*(c_{3,1} + c_{3,2} + c_{3,3} + .. + c_{3,m}) +$

……

$w_{n,2}*w_{n,q}*(c_{n,1} + c_{n,2} + c_{n,3} + .. + c_{n,m})$

Automatic Global Analysis Example

Example

D_1 = A,B,B,A,A,C

D_2 = D,D,C

D_3 = B,E,E

D_4 = D,E,A

Query = 2.3A+ C

$$w_{i,j} = \frac{\left(0.5+0.5\frac{f_{i,j}}{\max_j(f_{i,j})}\right) itf_j}{\sqrt{\sum_{l=1}^{N}\left(0.5+0.5\frac{f_{i,j}}{\max_l(f_{i,l})}\right)^2 itf_l^2}}$$

$$itf_j = \log\frac{t}{t_j}$$

Automatic Global Analysis Example

Example

D_1 = A,B,B,A,A,C

D_2 = D,D,C

D_3 = B,E,E

D_4 = A,D,E

Query = 2.3A+ C

Term = 5

$$itf_j = \log \frac{t}{t_j}$$

$$itf_4 = \log \frac{5}{3} = 0.222$$

Key/Doc	D1	D2	D3	D4
A	3	0	0	1
B	2	0	1	0
C	1	1	0	0
D	0	2	0	1
E	0	0	2	1
Max	3	2	2	1
t_j	3	2	2	3
itf(Doc)	0.222	0.398	0.398	0.222

Automatic Global Analysis Example

	D1	D2	D3	D4
A	3	0	0	1
B	2	0	1	0
C	1	1	0	0
D	0	2	0	1
E	0	0	2	1
Max	3	2	2	1
tj	3	2	2	3
itf	0.222	0.398	0.398	0.222

$$w_{i,j} = \frac{\left(0.5+0.5\frac{f_{i,j}}{\max_j(f_{i,j})}\right) itf_j}{\sqrt{\sum_{l=1}^{N}\left(0.5+0.5\frac{f_{i,j}}{\max_l(f_{i,l})}\right)^2 itf_l^{\,2}}}$$

$$w_{1,3} = \frac{\left(0.5+0.5\frac{f_{1,3}}{\max(fd_3)}\right) itf_3}{\sqrt{\left(0.5+0.5\frac{f_{1,1}}{\max(fd_1)}\right)^2 itf_1^{\,2} + \left(0.5+0.5\frac{f_{1,2}}{\max(fd_2)}\right)^2 itf_2^{\,2} + \left(0.5+0.5\frac{f_{1,3}}{\max(fd_3)}\right)^2 itf_3^{\,2} + \left(0.5+0.5\frac{f_{1,4}}{\max(fd_4)}\right)^2 itf_4^{\,2}}}$$

$$w_{1,3} = \frac{(0.5+0.5*\frac{0}{2})0.398}{\sqrt{(0.5+0.5*\frac{3}{3})^2 0.222^2 + (0.5+0.5*\frac{0}{2})^2 0.398^2 + (0.5+0.5*\frac{0}{2})^2 0.398^2 + (0.5+0.5*\frac{1}{1})^2 0.222^2}}$$

$$w_{1,3} = 1.509$$

Automatic Global Analysis Example

Term Weight

W	D_1	D_2	D_3	D_4
A	1.683	1.509	1.509	1.683
B	1.228	1.322	1.983	0.737
C	0.996	2.010	1.340	0.747
D	0.598	2.146	1.073	1.197
E	0.598	1.073	2.146	1.197

$$c_{u,v} = \overrightarrow{k_u} \bullet \overrightarrow{k_v} = \sum_{j=1}^{N} w_{u,j} \times w_{v,j}$$

$$C_{1,3} = w_{1,1} * w_{3,1} + w_{1,2} * w_{3,2} + w_{1,3} * w_{3,3} + w_{1,4} * w_{3,4}$$

$$= 1.683*0.996 + 1.509*2.010 + 1.509*1.340 + 1.683*0.747$$

$$= 7.987$$

Automatic Global Analysis Example

The relationship between two terms

C	A	B	C	D	E
A	10.218	8.293	7.987	7.879	7.879
B	8.293	7.728	7.085	6.581	7.290
C	7.987	7.085	7.383	7.241	6.522
D	7.879	6.581	7.241	7.548	6.397
E	7.879	7.290	6.522	6.397	7.548

Automatic Global Analysis Example

term similarity

C	A	B	C	D	E	Sim(q,K_j)
A	10.218	8.293	7.987	7.879	7.879	31.487
B	8.293	7.728	7.085	6.581	7.290	26.159
C	7.987	7.085	7.383	7.241	6.522	25.753
D	7.879	6.581	7.241	7.548	6.397	25.362
E	7.879	7.290	6.522	6.397	7.548	24.643
q	2.3	0	1	0	0	

ADD K_2 to Query

$$sim(q,k_v) = \vec{q} \cdot \vec{k}_v = \sum_{k_u \in q} w_{u,q} \times c_{u,v}$$

$$sim(q,k_3) = w_{1,q}*c_{1,3} + w_{2,q}*c_{2,3} + w_{3,q}*c_{3,3} + w_{4,q}*c_{4,3} + w_{5,q}*c_{5,3}$$

$$= 2.3*7.987 + 1*7.383 = 25.753$$

Automatic Global Analysis Example

Recompute term similarity

C	A	B	C	D	E
A	10.218	8.293	7.987	7.879	7.879
B	8.293	7.728	7.085	6.581	7.290
C	7.987	7.085	7.383	7.241	6.522
D	7.879	6.581	7.241	7.548	6.397
E	7.879	7.290	6.522	6.397	7.548
q	2.3	*1*	1	0	0

$Sim(q,K_i)$
39.780
33.887
32.838
31.942
31.933

$$sim(q,k_3) = w_{1,q}*c_{1,3} + w_{2,q}*c_{2,3} + w_{3,q}*c_{3,3} + w_{4,q}*c_{4,3} + w_{5,q}*c_{5,3}$$

$$= 2.3*7.987 + 1*7.085+1*7.383 =32.838$$

Automatic Global Analysis Example

Compute new weight terms for query

Original Query

$q = 2.3K_1 + K_2 + K_3$ Sum query weight = 2.3+1+1 = 4.3

$$w_{v,q'} = \frac{sim(q,k_v)}{\sum_{k_u \in q} w_{u,q}}$$

$w_{1,q'}$ = 39.780/4.3 = 9.251
$w_{2,q'}$ = 33.887/4.3 = 7.881
$w_{3,q'}$ = 32.838/4.3 = 7.637

	A	B	C	D	E
q'	9.251	7.881	7.637	-	-

Automatic Global Analysis Example

Arrange Relevance

q' =9.251A+7.881B+7.637C

W	D1	D2	D3	D4
A	1.683	1.509	1.509	1.683
B	1.228	1.322	1.983	0.737
C	0.996	2.010	1.340	0.747
D	0.598	2.146	1.073	1.197
E	0.598	1.073	2.146	1.197

C	A	B	C	D	E
A	10.22	8.293	7.987	7.879	7.879
B	8.293	7.728	7.085	6.581	7.290
C	7.987	7.085	7.383	7.241	6.522
D	7.879	6.581	7.241	7.548	6.397
E	7.879	7.290	6.522	6.397	7.548

$$sim(q,d_j) \propto \sum_{k_v \in d_j} \sum_{k_u \in q} w_{i,j} \times w_{u,q} \times c_{u,v}$$

$w_{1,2} = 1.509$ $\quad w_{1,q} = 9.251$

$w_{2,2} = 1.322$ $\quad w_{2,q} = 7.881$

$w_{3,2} = 2.010$ $\quad w_{3,q} = 7.637$

$w_{4,2} = 2.146$ $\quad w_{4,q} = 0$

$w_{5,2} = 1.073$ $\quad w_{5,q} = 0$

$sim(q,d_2) =$

$w_{1,2}*w_{1,q}*(c_{1,1}+ c_{1,2}+ c_{1,3}+ c_{1,4}+ c_{1,5}) +$

$w_{2,2}*w_{2,q}*(c_{2,1}+ c_{2,2}+ c_{2,3}+ c_{2,4}+ c_{2,5}) +$

$w_{3,2}*w_{3,q}*(c_{3,1}+ c_{3,2}+ c_{3,3}+ c_{3,4}+ c_{3,5}) +$

$w_{4,2}*w_{4,q}*(c_{4,1}+ c_{4,2}+ c_{4,3}+ c_{4,4}+ c_{4,5}) +$

$w_{5,2}*w_{5,q}*(c_{5,1}+ c_{5,2}+ c_{5,3}+ c_{5,4}+ c_{5,5})$

$sim(q,d_2) = 1531.123$

Automatic Global Analysis Example

Arrange Relevance

$q' = 9.251A + 7.881B + 7.637C$

	D_1	D_2	D_3	D_4
$Sim(q,d_j)$	1,291.282	1,531.123	1,538.429	1,079.324

Answer = D_3, D_2, D_1, D_4

Automatic Local analysis

- Basic concept
  - Expanding the query with terms correlated to the query terms
  - The correlated terms are presented in the local clusters built from ***the local document set***

Automatic Local Analysis

- Definition
  - local document set D_l : the set of ***documents retrieved*** by a query
  - local vocabulary V_l : the set of ***all distinct words*** in D_l
  - stemed vocabulary S_l : the set of ***all distinct stems*** derived from V_l
- Building local clusters
  - association clusters
  - metric clusters
  - scalar clusters

Association Clusters

- idea
  - Based on the co-occurrence of stems (or terms) **inside documents**
- association matrix
  - $f_{s_i,j}$: the frequency of a **stem** s_i in a document $\vec{d}_j$ ($\in D_l$)
  - $m=(f_{s_i,j})$: an association matrix with $|S_l|$ rows and $|D_l|$ columns
  - $\vec{s} = \vec{m}\vec{m}^t$: a local **stem-stem** association matrix

Association Clusters

- Idea
  - **co-occurrence** of **stems** (or terms) inside documents **(frequency of stems in doc)**

$$c(k_u, k_v) = \sum_{j=1}^{|D|} f_{u,j} \times f_{v,j}$$

    - $f_{u,j}$: the frequency of a **stem** k_u in a document d_j
  - local association cluster for a stem k_u
    - the set of k largest values $c(k_u, k_v)$
  - given a query q, find clusters for the $|q|$ query terms
  - normalized form $s(k_u, k_v) = \dfrac{c(k_u, k_v)}{c(k_u, k_u) + c(k_v, k_v) - c(k_u, k_v)}$

Metric Clusters

- Idea
  - consider the **distance between two terms** in the same cluster
- Definition
  - $V(k_u)$: the set of keywords which have the same stem form as k_u
  - distance $r(k_i, k_j)$=the number of words between term k_i and k_j

$$c(k_u, k_v) = \sum_{i \in V(k_u)} \sum_{j \in V(k_v)} \frac{1}{r(k_i, k_j)}$$

  - normalized form

$$s(k_u, k_v) = \frac{c(k_u, k_v)}{|V(k_u)| \times |V(k_v)|}$$

Scalar Clusters

- Idea
  - two stems with similar neighborhoods have some **synonymity** relationships
- *Definition*
  - $c_{u,v}=c(k_u, k_v)$
  - vectors of correlation values for stem k_u and k_v

$$\vec{s_u} = (c_{u,1}, c_{u,2}, \cdots, c_{u,t}) \qquad \vec{s_v} = (c_{v,1}, c_{v,2}, \cdots, c_{v,t})$$

  - scalar association matrix

$$S_{u,v} = \frac{\vec{s_u} \bullet \vec{s_v}}{|\vec{s_u}| \times |\vec{s_v}|}$$

  - scalar clusters
    - the set of k largest values of scalar association

Association Clusters

- Idea
  - **co-occurrence** of stems (or terms) inside documents **(frequency of stems in doc)**

$$c(k_u, k_v) = \sum_{j=1}^{|D|} f_{u,j} \times f_{v,j}$$

    - $f_{u,j}$: the frequency of a stem k_u in a document d_j
  - local association cluster for a stem k_u
    - the set of k largest values $c(k_u, k_v)$
  - given a query q, find clusters for the $|q|$ query terms
  - normalized form $s(k_u, k_v) = \dfrac{c(k_u, k_v)}{c(k_u, k_u) + c(k_v, k_v) - c(k_u, k_v)}$

Association Clusters

$$c_{u,v} = \sum_{dj \in Dl} f_{su,j} \times f_{sv,j}$$: a correlation between the stems s_u and s_v

an element in $\vec{m}\vec{m}^t$

$s_{u,v} = c_{u,v}$: ***unnormalized matrix***

$$s_{u,v} = \frac{c_{u,v}}{c_{u,u} + c_{v,v} - c_{u,v}}$$: ***normalized matrix***

$s_u(n)$: local association cluster around the stem s_u

- Take u-th row
- Return the set of n **largest values** $s_{u,v}$ ($u \neq v$)

Association Clusters Example

q = A+B

{B,D,C}→ A

d_1 = A,A,B,D
d_2 = B,A,C,C,D
d_3 = A,B
d_4 = B,C,D
d_5 = D
d_6 = A,B,D
d_7 = B,B,A

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
A	2	1	1	0	0	1	1
B	1	1	1	1	0	1	2
C	0	2	0	1	0	0	0
D	1	1	0	1	1	1	0

$$c_{u,v} = \sum_{dj \in Dl} f_{s_u, j} \times f_{s_v, j}$$

$C_{1,4} = (f_{1,1}*f_{4,1})+(f_{1,2}*f_{4,2})+(f_{1,3}*f_{4,3})+(f_{1,4}*f_{4,4})+(f_{1,5}*f_{4,5})+(f_{1,6}*f_{4,6})+(f_{1,7}*f_{4,7})$

= 2*1 + 1*1 + 1*0 + 0*1 + 0*1 + 1*1 + 1*0
= **4**

Association Clusters Example

Correlation Matrix (C)

	A	*B*	*C*	*D*
A	8	7	2	4
B	7	9	3	4
C	2	3	5	3
D	4	4	3	5

Association Clusters Example

Other way to compute the Correlation Matrix

$$c = \vec{m}\vec{m}^{t}$$

	d_1	d_2	d_3	d_4	d_5	d_6	d_7
A	2	1	1	0	0	1	1
B	1	1	1	1	0	1	2
C	0	2	0	1	0	0	0
D	1	1	0	1	1	1	0

m

	A	B	C	D
d_1	2	1	0	1
d_2	1	1	2	1
d_3	1	1	0	0
d_4	0	1	1	1
d_5	0	0	0	1
d_6	1	1	0	1
d_7	1	2	0	0

m^t

$C_{1,4} = (m_{1,1}*mt_{1,4})+(m_{1,2}*mt_{2,4})+(m_{1,3}*mt_{3,4})+(m_{1,4}*mt_{4,4})+(m_{1,5}*mt_{5,4})+ (m_{1,6}*mt_{6,4})+(m_{1,7}*mt_{7,4})$

$= 2*1+ 1*1+1*0+0*1+0*1+1*1+1*0$

$=$ **4**

Association Clusters Example

Correlation Matrix (C)

	A	*B*	*C*	*D*
A	8	7	2	4
B	7	9	3	4
C	2	3	5	3
D	4	4	3	5

Association Clusters Example

Normalized Correlation Matrix (S)

$$s_{u,v} = \frac{c_{u,v}}{c_{u,u} + c_{v,v} - c_{u,v}}$$

$$s_{1,2} = \frac{c_{1,2}}{c_{1,1} + c_{2,2} - c_{1,2}} = \frac{7}{8+9-7} = 0.70$$

	A	***B***	***C***	***D***
A	**8**	**7**	**2**	**4**
B	**7**	**9**	**3**	**4**
C	**2**	**3**	**5**	**3**
D	**4**	**4**	**3**	**5**

Association Clusters Example

Normalized Correlation Matrix

	A	*B*	*C*	*D*
A	1	0.70	0.18	0.44
B	0.70	1	0.27	0.40
C	0.18	0.27	1	0.43
D	0.44	0.40	0.43	1

Take u-th row
Return the set of n largest values $s_{u,v}$ ($u \neq v$)

Term Relation

1. *{A,B}*
2. *{B,A}*
3. *{C,D}*
4. *{D,A}*

Original Query

q = A+B

New Query

q' = (A + 0.7B) + (0.7A + B)
= 1.7A + 1.7B
= A + B

Association Clusters Example (other case)

Normalized Correlation Matrix

	A	*B*	*C*	*D*
A	**1**	**0.70**	**0.18**	**0.44**
B	**0.70**	**1**	***0.85***	***0.63***
C	**0.18**	***0.85***	**1**	***0.63***
D	**0.44**	***0.63***	***0.63***	**1**

Term Relation

1. *{A,B}*
2. *{B,C}*
3. *{C,B}*
4. *{D,B,C}*

Original Query

q = A+B

New Query

q' = (A+0.7B)+(B+0.85C)
= A+1.7B+0.85C

Original Query

q = C+2D

New Query

q' = (0.85B+C)+2*(0.63B+0.63C+D)
=2.11B+2.26C+2D

Metric Clusters

- Idea
  - consider the **distance between two terms** in the same cluster
- Definition
  - $V(k_u)$: the set of keywords which have the same stem form as k_u
  - distance $r(k_i, k_j)$=the number of words between term k_i and k_j

$$c(k_u, k_v) = \sum_{i \in V(k_u)} \sum_{j \in V(k_v)} \frac{1}{r(k_i, k_j)}$$

  - normalized form

$$s(k_u, k_v) = \frac{c(k_u, k_v)}{|V(k_u)| \times |V(k_v)|}$$

Metric Clusters

$s_{u,v}=c_{u,v}$: unnormalized matrix

$$s_{u,v} = \frac{c_{u,v}}{|V(s_u)| \times |V(s_v)|}$$: normalized matrix

$s_u(n)$: local metric cluster around the stem s_u

- Take u-th row
- Return the set of n **largest values** $s_{u,v}$ $(u \neq v)$

Metric Clusters Example

q = A+2D

k_n = A,B,C,D,E,F

A,B,C base on S_1 stem
D,E base on S_2 stem
F base on S_3 stem

Then

$V(S_1) = \{A,B,C\}$
$V(S_2) = \{D,E\}$
$V(S_3) = \{F\}$

ระยะห่าง	*A*	*B*	*C*	*D*	*E*	*F*
A	0	5	∞	∞	1	2
B	5	0	3	2	1	1
C	∞	3	0	3	4	∞
D	∞	2	3	0	∞	5
E	1	1	4	∞	0	1
F	2	1	∞	5	1	0

Metric Clusters Example

ระยะห่าง	*A*	*B*	*C*	*D*	*E*	*F*
A	0	5	∞	∞	1	2
B	5	0	3	2	1	1
C	∞	3	0	3	4	∞
D	∞	2	3	0	∞	5
E	1	1	4	∞	0	1
F	2	1	∞	5	1	0

	A	*B*	*C*	*D*	*E*	*F*
A	-	0.20	0	0	1	0.50
B	0.20	-	0.33	0.50	1	1
C	0	0.33	-	0.33	0.25	0
D	0	0.50	0.33	-	0	0.20
E	1	1	0.25	0	-	1
F	0.50	1	0	0.20	1	-

Metric Clusters Example

$V(S_1) = \{A,B,C\}$
$V(S_2) = \{D,E\}$
$V(S_3) = \{F\}$

	A	*B*	*C*	*D*	*E*	*F*
A	-	0.20	0	0	1	0.50
B	0.20	-	0.33	0.50	1	1
C	0	0.33	-	0.33	0.25	0
D	0	0.50	0.33	-	0	0.20
E	1	1	0.25	0	-	1
F	0.50	1	0	0.20	1	-

$$c_{u,v} = \sum_{ki \in V(su)} \sum_{kj \in V(sv)} \frac{1}{r(k_i, k_j)}$$

$$c_{1,2} = c(A,D) + c(A,E) + c(B,D) + c(B,E) + c(C,D) + c(C,E)$$

= 0 + 1 + 0.50 + 1 + 0.33 + 0.25

= 3.08

Metric Clusters Example

Correlation Matrix (C)

	S_1	S_2	S_3
S_1	0	3.08	1.50
S_2	3.08	0	1.20
S_3	1.50	1.20	0

Metric Clusters Example

Normalized Correlation Matrix (S)

	S_1	S_2	S_3
S_1	0	3.08	1.50
S_2	3.08	0	1.20
S_3	1.50	1.20	0

$$s_{u,v} = \frac{c_{u,v}}{|V(s_u)| \times |V(s_v)|}$$

$$s_{2,3} = \frac{c_{2,3}}{|V(s_2)| \times |V(s_3)|} = \frac{1.2}{2x1} = 0.6$$

$V(S_1)$ = {A,B,C} = 3
$V(S_2)$ = {D,E} = 2
$V(S_3)$ = {F} = 1

Metric Clusters Example

Normalized Correlation Matrix (S)

	S_1	S_2	S_3
S_1	0	0.51	0.50
S_2	0.51	0	0.60
S_3	0.50	0.60	0

Stem Relation

1. $\{S_1, S_2\}$
2. $\{S_2, S_3\}$
3. $\{S_3, S_2\}$

Original Query

$q = A + 2D$

New Query

$$q' = (S_1 + 0.51S_2) + 2*(S_2 + 0.60S_3)$$
$$= S_1 + 2.51S_2 + 1.2S_3$$

Scalar Clusters

- Idea
  - two stems with similar neighborhoods have some ***synonymity relationships***
- *Definition*
  - $c_{u,v}=c(k_u, k_v)$
  - vectors of correlation values for stem k_u and k_v

$$\vec{s_u} = (c_{u,1}, c_{u,2}, \cdots, c_{u,t}) \qquad \vec{s_v} = (c_{v,1}, c_{v,2}, \cdots, c_{v,t})$$

  - scalar association matrix

$$S_{u,v} = \frac{\vec{s_u} \bullet \vec{s_v}}{|\vec{s_u}| \times |\vec{s_v}|}$$

  - scalar clusters
    - the set of k **largest values** of scalar association

Database , Math , Set
Tree, Water , Fertilizer
Flower, Letter , Lover

Scalar Clusters

$$\overrightarrow{s_u} = (c_{u,1}, c_{u,2}, \cdots, c_{u,t})$$

$$\overrightarrow{s_v} = (c_{v,1}, c_{v,2}, \cdots, c_{v,t})$$

$$\overrightarrow{s_1} = (c_{1,1}, c_{1,2}, \cdots, c_{1,t})$$

$$\overrightarrow{s_3} = (c_{3,1}, c_{3,2}, \cdots, c_{3,t})$$

$$\overrightarrow{s_1} = (c_{1,1}, c_{1,2}, c_{1,3}) = (5,6,1) \Rightarrow C_{\text{Database, Algebra}}, C_{\text{Database,Math}}, C_{\text{Database,Set}}$$

$$\overrightarrow{s_2} = (c_{2,1}, c_{2,2}, c_{2,3}) = (6,9,0) \Rightarrow C_{\text{AI, Algebra}}, C_{\text{AI,Math}}, C_{\text{AI,Set}}$$

$$\overrightarrow{s_3} = (c_{3,1}, c_{3,2}, c_{3,3}) = (1,0,2) \Rightarrow C_{\text{Network , Algebra}}, C_{\text{Network,Math}}, C_{\text{Network,Set}}$$

Network={Set(2), Algebra (1),Math(0)}***idea

Scalar Clusters

$$\overrightarrow{s_u} = (c_{u,1}, c_{u,2}, \cdots, c_{u,t})$$

$$\overrightarrow{s_v} = (c_{v,1}, c_{v,2}, \cdots, c_{v,t})$$

$$\overrightarrow{s_1} = (c_{1,1}, c_{1,2}, \cdots, c_{1,t})$$

$$\overrightarrow{s_3} = (c_{3,1}, c_{3,2}, \cdots, c_{3,t})$$

$$\overrightarrow{s_1} = (c_{1,1}, c_{1,2}, c_{1,3}) = (5,6,1)$$

$$\overrightarrow{s_2} = (c_{2,1}, c_{2,2}, c_{2,3}) = (6,9,0)$$

$$\overrightarrow{s_3} = (c_{3,1}, c_{3,2}, c_{3,3}) = (1,0,2)$$

$$S_{u,v} = \frac{\overrightarrow{s_u} \bullet \overrightarrow{s_v}}{|\overrightarrow{s_u}| \times |\overrightarrow{s_v}|}$$

$$S_{1,3} = \frac{\overrightarrow{s_1} \bullet \overrightarrow{s_3}}{|\overrightarrow{s_1}| \times |\overrightarrow{s_3}|}$$

$$|S_1| = \sqrt{25+36+1} = 7.874$$

$$|S_2| = \sqrt{36+81+0} = 10.817$$

$$|S_3| = \sqrt{1+0+4} = 2.236$$

$$S_{1,3} = \frac{7}{7.874 \times 2.236} = 0.398$$

Scalar Clusters

Normalized Correlation Matrix (S)

S	S_1	S_2	S_3
S_1	**1**	**0.986**	**0.398**
S_2	**0.986**	**1**	**0.248**
S_3	**0.398**	**0.248**	**1**

Stem Relation

1. $\{S_1, S_2\}$
2. $\{S_2, S_1\}$
3. $\{S_3, S_1\}$

Original Query

$$q = 3S_1 + S_3$$

Database

Network

New Query

$$q' = 3*(S_1 + 0.986S_2) + (0.398S_1 + S_3)$$
$$= 3.398S_1 + 2.958S_2 + S_3$$

Discussion

- Query expansion
  - useful
  - little explored technique
- Trends and research issues
  - The combination of local analysis, global analysis, visual displays, and interactive interfaces is also a current and important research problem