Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών
Άνοιξη 2008

HY463 - Συστήματα Ανάκτησης Πληροφοριών
Information Retrieval (IR) Systems

Προχωρημένες Λειτουργίες Επερώτησης
Advanced Query Operations

Γιάννης Τζίτζικας

Διάλεξη : 11
Ημερομηνία :

Διάρθρωση Διάλεξης

- Κίνητρο
- **Ανάδραση Συνάφειας (Relevance Feedback)**
- **Αναδιατύπωση Επερωτήσεων (Query Reformulation)**
  - Αναβάρυνση Όρων (Term Reweighting)
  - Επέκταση (Διαστολή) Επερώτησης (Query Expansion),
  - Αναδιατύπωση Επερωτήσεων για το Διανυσματικό Μοντέλο
    - Optimal Query, Rocchio Method, Ide Method, DeHi Method
  - Η έννοια του Optimal (or Best) Query
  - Αξιολόγηση
- **Ψευδο-ανάδραση συνάφειας (Pseudo relevance feedback)**
- **Επέκταση Επερωτήσεων**
  - Αυτόματη Τοπική (Επιτόπια) Ανάλυση (Automatic Local Analysis)
  - Καθολική Ανάλυση
  - Επέκταση Επερώτησης βάσει Θησαυρού (Thesaurus-based Query Expansion)
  - Αυτόματη Καθολική Ανάλυση (Automatic Global Analysis)
  - Στατιστικοί Θησαυροί (Statistical Thesaurus)
  - Κατασκευή Θησαυρών
- //Γενετικοί Αλγόριθμοι

Κίνητρο

- Έχει παρατηρηθεί ότι οι χρήστες των ΣΑΠ δαπανούν πολύ χρόνο αναδιατυπώνοντας την αρχική τους επερώτηση προκειμένου να βρουν ικανοποιητικά έγγραφα
- Πιθανές αιτίες
  - ο χρήστης δεν γνωρίζει το περιεχόμενο των υποκείμενων εγγράφων
  - το λεξιλόγιο του χρήστη μπορεί να διαφέρει από αυτό της συλλογής
  - η αρχική επερώτηση μπορεί να είναι πιο γενική ή πιο ειδική από αυτή που θα έπρεπε (καταλήγοντας είτε σε πάρα πολλά ή σε πολύ λίγα έγγραφα)
- Η αρχική επερώτηση μπορεί να θεωρηθεί ως η πρώτη προσπάθεια έκφρασης της πληροφοριακής ανάγκης του χρήστη
- Ανάγκη για τεχνικές αντιμετώπισης αυτού του προβλήματος

Τρόποι Αντιμετώπισης

(1) Βελτίωση της αρχικής επερώτησης
(2) Χρήση Προφίλ Χρήστη
(3) Βελτίωση παράστασης κειμένων
(4) Βελτίωση αλγορίθμου (μοντέλου) ανάκτησης

Παρατηρήσεις

- Τα (2) ,(3),(4) έχουν πιο μόνιμο αποτέλεσμα (επηρεάζουν την απάντηση και των επόμενων επερωτήσεων)
- Εδώ θα εστιάσουμε στο (1)

Τεχνικές Βελτίωσης της Αρχικής Επερώτησης

Κατηγορίες:

(α) τεχνικές που απαιτούν **είσοδο από τον χρήστη**

(β) τεχνικές που **δεν απαιτούν** είσοδο

(β1) που βασίζονται στα **κορυφαία έγγραφα** που ανακτήθηκαν

(β2) που βασίζονται σε **όλα τα έγγραφα** της συλλογής

Ανάδραση Συνάφειας (Relevance Feedback): Η βασική ιδέα

Βήματα:

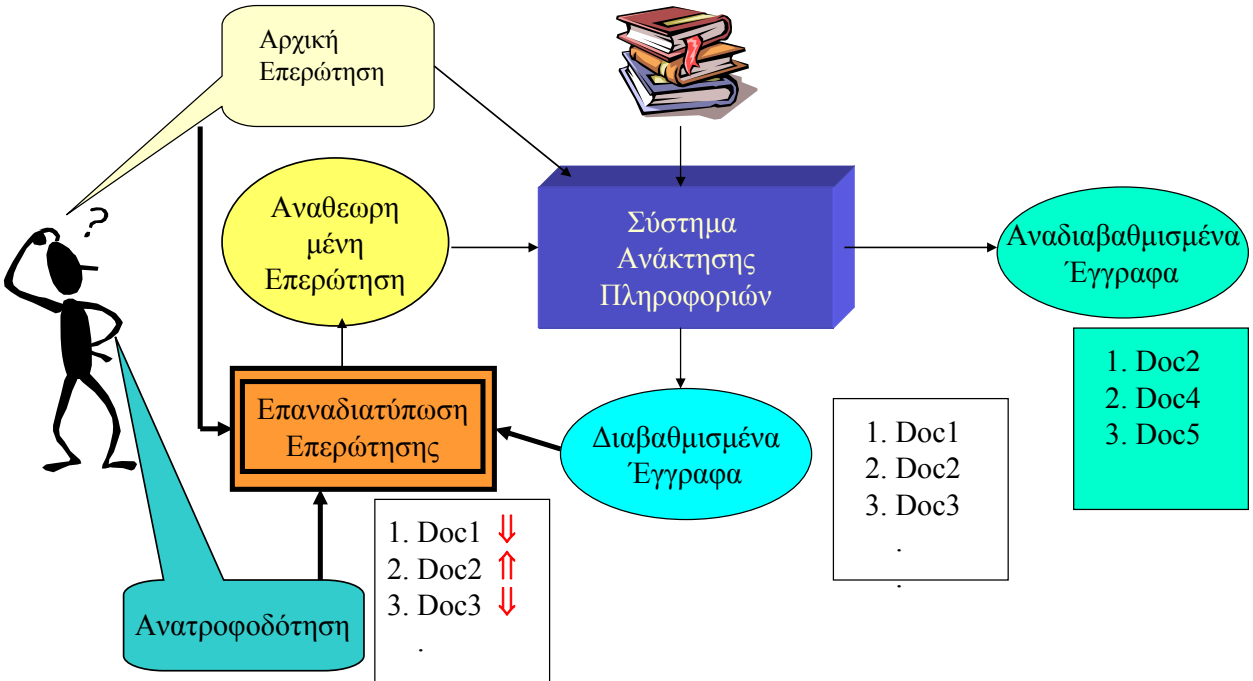
1/ Μετά την παρουσίαση των αποτελεσμάτων, επιτρέπουμε στο χρήστη **να κρίνει (θετικά ή αρνητικά) την συνάφεια** ενός ή περισσότερων εγγράφων της απάντησης

2/ Αξιοποιούμε αυτήν την πληροφορία για να **αναδιατυπώσουμε** την επερώτηση

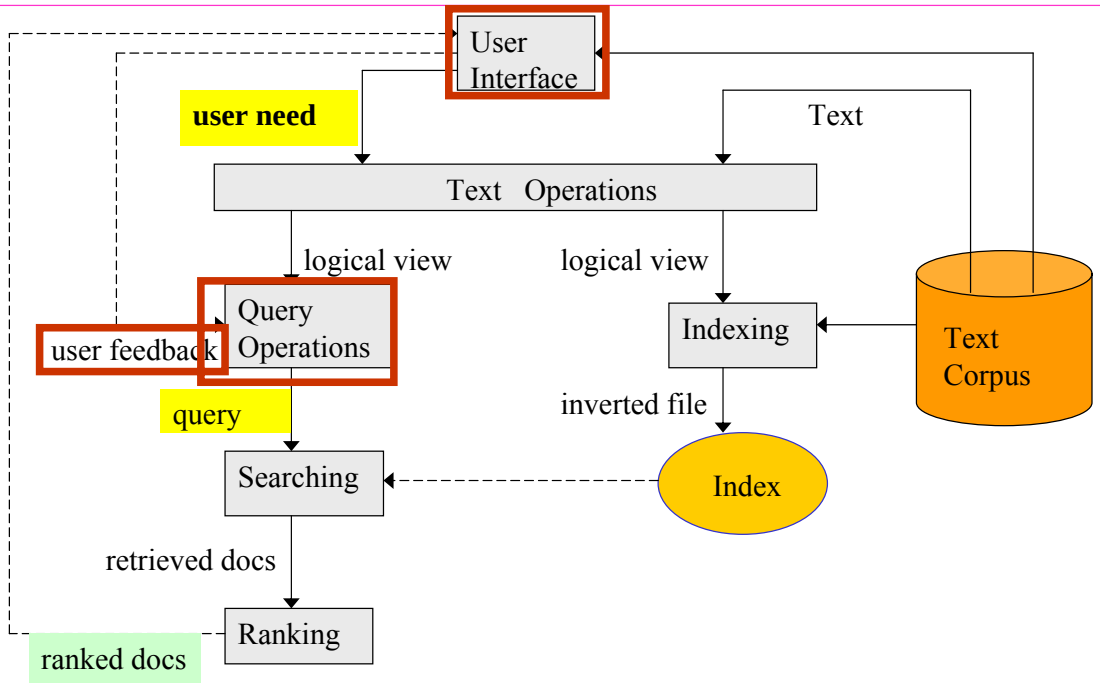
3/ Κατόπιν δίδουμε στο χρήστη την απάντηση της αναδιατυπωμένης επερώτησης

4/ Πήγαινε στο βήμα 1/

Ανάδραση Συνάφειας (Relevance Feedback): Η βασική ιδέα

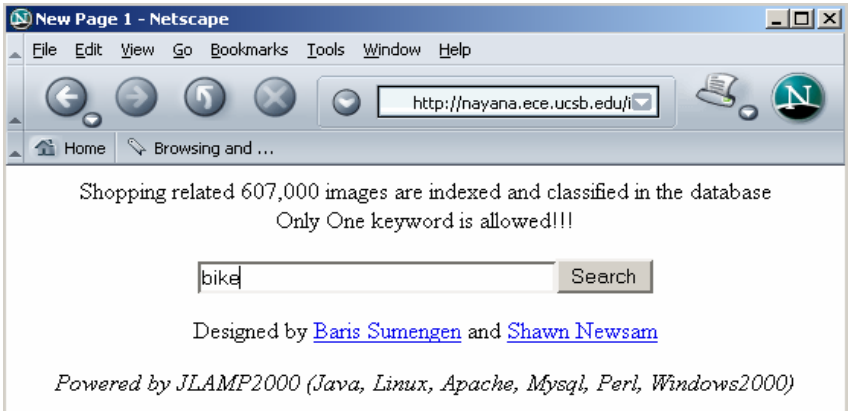


Τμήματα της Αρχιτεκτονικής που Εμπλέκονται



Παράδειγμα ανατροφοδότησης συνάφειας σε σύστημα ανάκτησης εικόνων

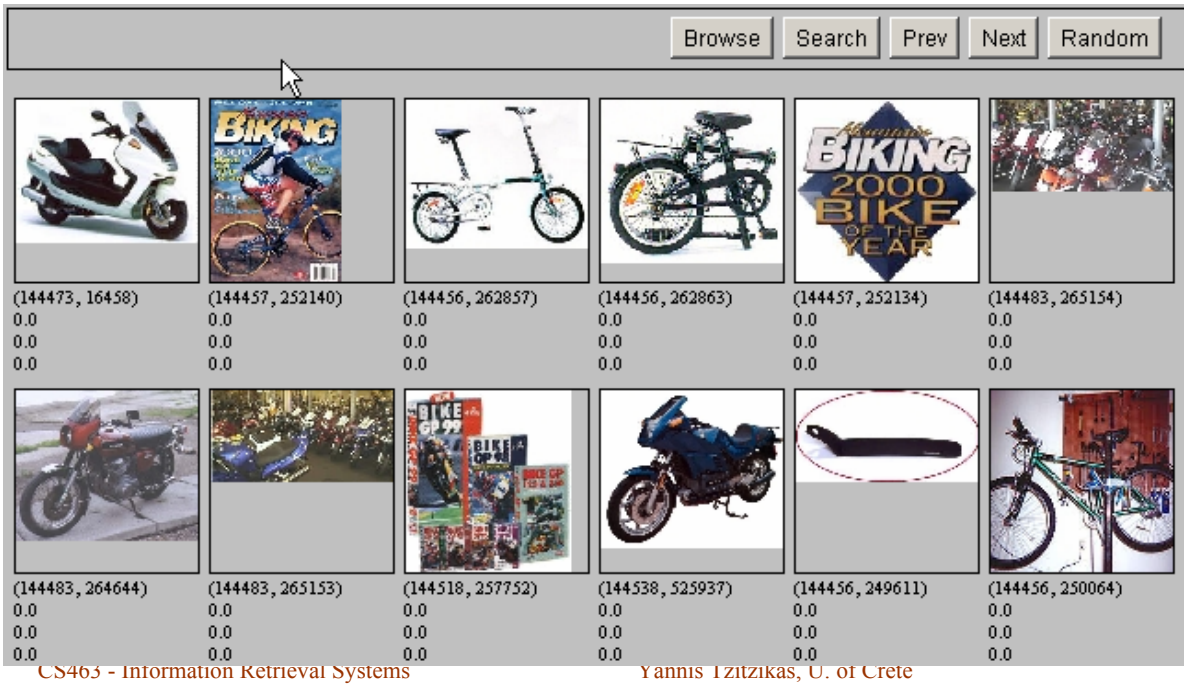
q=bike



(http://nayana.ece.ucsb.edu/imsearch/imsearch.html)

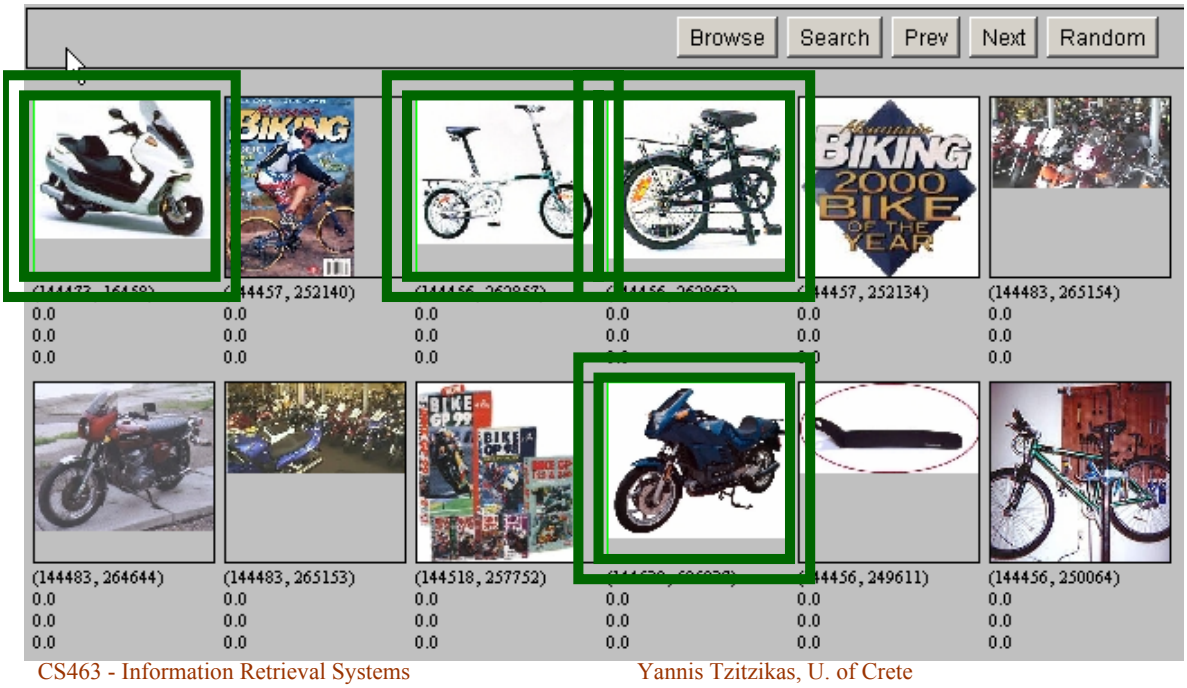
Παράδειγμα ανατροφοδότησης συνάφειας σε σύστημα ανάκτησης εικόνων

Answer("bike")=



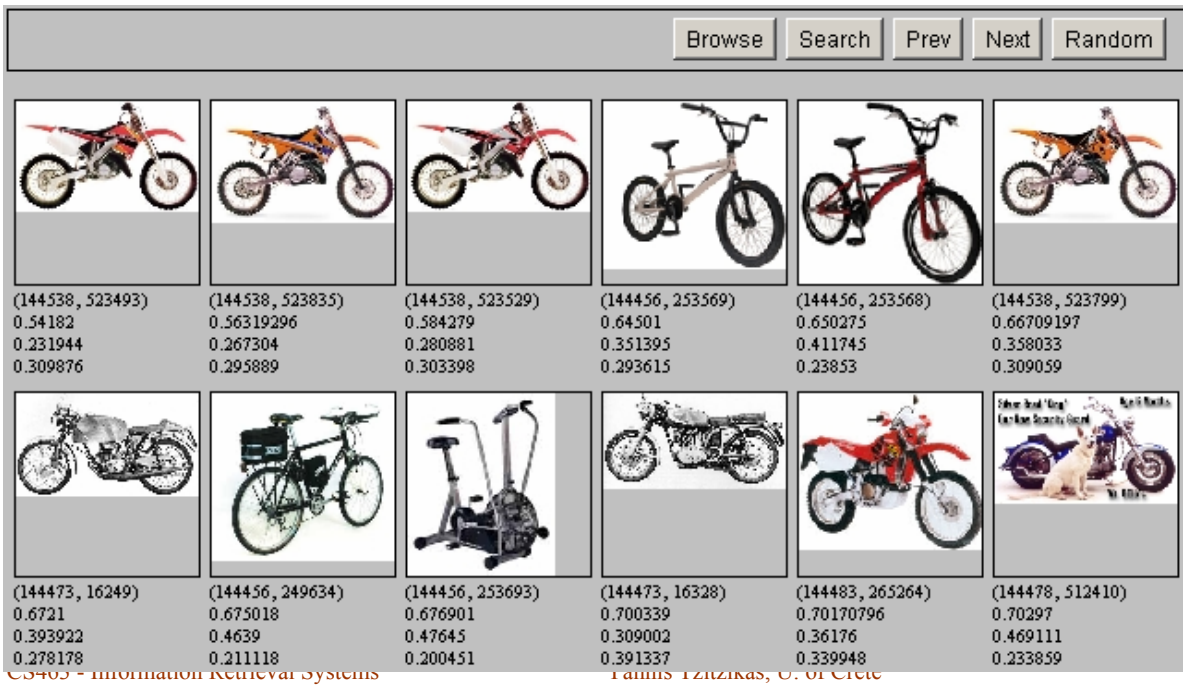
Παράδειγμα ανατροφοδότησης συνάφειας σε σύστημα ανάκτησης εικόνων

Μαρκάρισμα των Συναφών (η Επιθυμητών) από τον Χρήστη



Παράδειγμα ανατροφοδότησης συνάφειας σε σύστημα ανάκτησης εικόνων

Απάντηση της αναδιατυπωμένης απάντησης =



Αναδιατύπωση επερώτησης βάσει Ανάδρασης Συνάφειας (Relevance Feedback: Query Reformulation)

Τρόποι αναδιατύπωσης της επερώτησης μετά την ανάδραση:

- **Αναβάρυνση των Όρων (Term Reweighting)**:
  - Αύξηση των βαρών των όρων που εμφανίζονται στα συναφή/επιθυμητά έγγραφα και μείωση των βαρών των όρων που εμφανίζονται στα μη-συναφή/επιθυμητά έγγραφα.
- **Επέκταση επερώτησης (Query Expansion):**
  - **Προσθήκη νέων όρων** στην επερώτηση (π.χ. από γνωστά συναφή έγγραφα)
- Υπάρχουν πολλοί αλγόριθμοι για επαναδιατύπωση επερώτησης

Αναδιατύπωση επερώτησης στο **Διανυσματικό Χώρο** Η έννοια της **βέλτιστης επερώτησης**

Η βέλτιστη επερώτηση (Optimal Query)

- Ας υποθέσουμε ότι γνωρίζουμε το σύνολο C_r **όλων** των συναφών (με την πληροφοριακή ανάγκη του χρήστη) εγγράφων.
- Η «καλύτερη επερώτηση» (αυτή που κατατάσσει στην κορυφή όλα τα συναφή έγγραφα), βάσει του διανυσματικού μοντέλου, θα ήταν:

$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\forall \vec{d}_j \in C_r} \vec{d}_j - \frac{1}{N - |C_r|} \sum_{\forall \vec{d}_j \notin C_r} \vec{d}_j$$

Where N is the total number of documents.

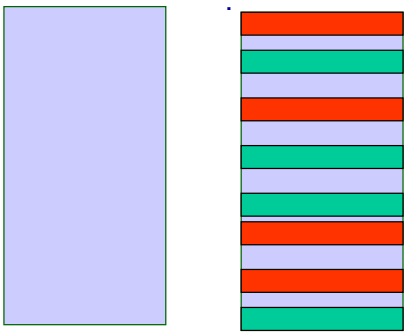
(θα αναλύσουμε περισσότερο αυτό το ζήτημα αργότερα)

→ Αφού όμως δεν γνωρίζουμε το σύνολο C_r,θα λάβουμε υπόψη την αρχική επερώτηση και την είσοδο/ανατροφοδότηση του χρήστη.

Αναδιατύπωση επερώτησης στο **Διανυσματικό Χώρο**

Αφού όμως δεν γνωρίζουμε το σύνολο C_r, θα λάβουμε υπόψη την αρχική επερώτηση και την είσοδο του χρήστη.

Answer(q)= Answer (q) + user feedback = :



Κόκκινα: ο χρήστης έδωσε αρνητική ανάδραση
Πράσινα: ο χρήστης έδωσε θετική ανάδραση
Μπλε: ο χρήστης δεν έδωσε ανάδραση

Τρόποι αξιοποίησης της ανατροφοδότησης του χρήστη
(I) **Rocchio** Method
(II) **Ide** Method
(III) **DeHi** Method

(I) Standard **Rocchio** Method

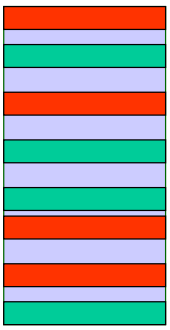
Αφού το σύνολο όλων των συναφών είναι άγνωστο,

χρησιμοποίησε τα **γνωστά** συναφή (D_r) και **γνωστά** μη-συναφή (D_n) έγγραφα (από την απάντηση της αρχικής επερώτησης και βάσει της εισόδου από τον χρήστη) και επίσης συμπεριέλαβε την αρχική επερώτηση *q*.

Αναδιατυπωμένη επερώτηση:

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

answer(q):



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

Usually γ < β (the relevant docs are more important)
If γ=0 then we have positive feedback only

(II) **IDE** Regular Method

Περισσότερη ανάδραση => μεγαλύτερος βαθμός αναδιατύπωσης.

Για αυτό, κατά την IDE Regular μέθοδο **δεν** κάνουμε κανονικοποίηση (βάσει του ποσού ανάδρασης)

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

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

(III) IDE "**Dec Hi**" Method

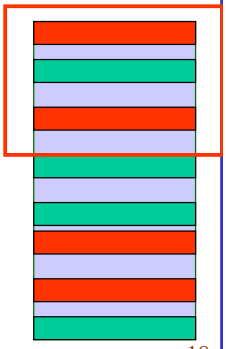
Τάση για απόρριψη **μόνο** των μη-συναφών εγγράφων που έχουν υψηλό σκορ

(Bias towards rejecting **just** the highest ranked of the irrelevant documents:)

$$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \max_{non-relevant} (\vec{d}_j)$$

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

answer(q):



Σύγκριση μεθόδων (I) (II) (III)

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

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

$$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \max_{non-relevant} (\vec{d}_j)$$

- Γενικά, τα πειραματικά δεδομένα δεν δίνουν καθαρό προβάδισμα σε κάποια τεχνική.
- Όλες οι τεχνικές βελτιώνουν την απόδοση (recall & precision)
- Συνήθως α=β=γ=1

Αξιολόγηση Αποτελεσματικότητας Τεχνικών Ανάδρασης Συνάφειας

Remarks

- By construction, reformulated query will rank **explicitly-marked relevant** documents higher and **explicitly-marked irrelevant** documents lower.
- When evaluating such methods, a method should not get credit for improvement on ***these*** documents, since it was told their relevance.
- In machine learning, this error is called “testing on the training data.”
- Evaluation should focus on generalizing to **other** un-rated documents.

Fair Process for Evaluating the Effectiveness of Relevance Feedback

- **Remove** from the corpus any document for which feedback was provided.
- Measure recall/precision performance on the remaining ***residual collection***.
- *Compared to complete corpus, specific recall/precision numbers may decrease since relevant documents were removed*.
- Measure recall/precision after relevance feedback (on the residual collection)
- Relative performance on the residual collection provides fair data on the effectiveness of relevance feedback

Relevance Feedback Evaluation

TABLE 4. Evaluation of typical relevance feedback methods for five collections (weighted documents, weighted queries).

Relevance Feedback Method	Rank of Method and Avg Precision	CACM 3204 docs 64 queries	CISI 1460 docs 112 docs	CRAN 1397 docs 225 queries	INSPEC 12684 docs 84 queries	MED 1033 docs 30 queries	Average
Initial Run (reduced collection)		.1459	.1184	.1156	.1368	.3346	
Ide (dec hi) expand by	Rank		+44%	+92%	+32%	+79%	+59%
Rocchio (standard $\beta = .75$, $\alpha = .25$ expand by all terms	Rank	2	39	8	14	17	16
	Precision	.2552	.1404	.2955	.1821	.5630	
	Improvement	+75%	+19%	+156%	+33%	+68%	+70%
expand by most common terms	Rank	3	12	12	10	24	
	Precision	.2491	.1623	.2534	.1861	.5279	
	Improvement	+71%	+37%	+119%	+36%	+55%	+64%
Probabilistic (adjusted revised							

Simulated interactive retrieval consistently outperforms non-interactive retrieval (70% here).

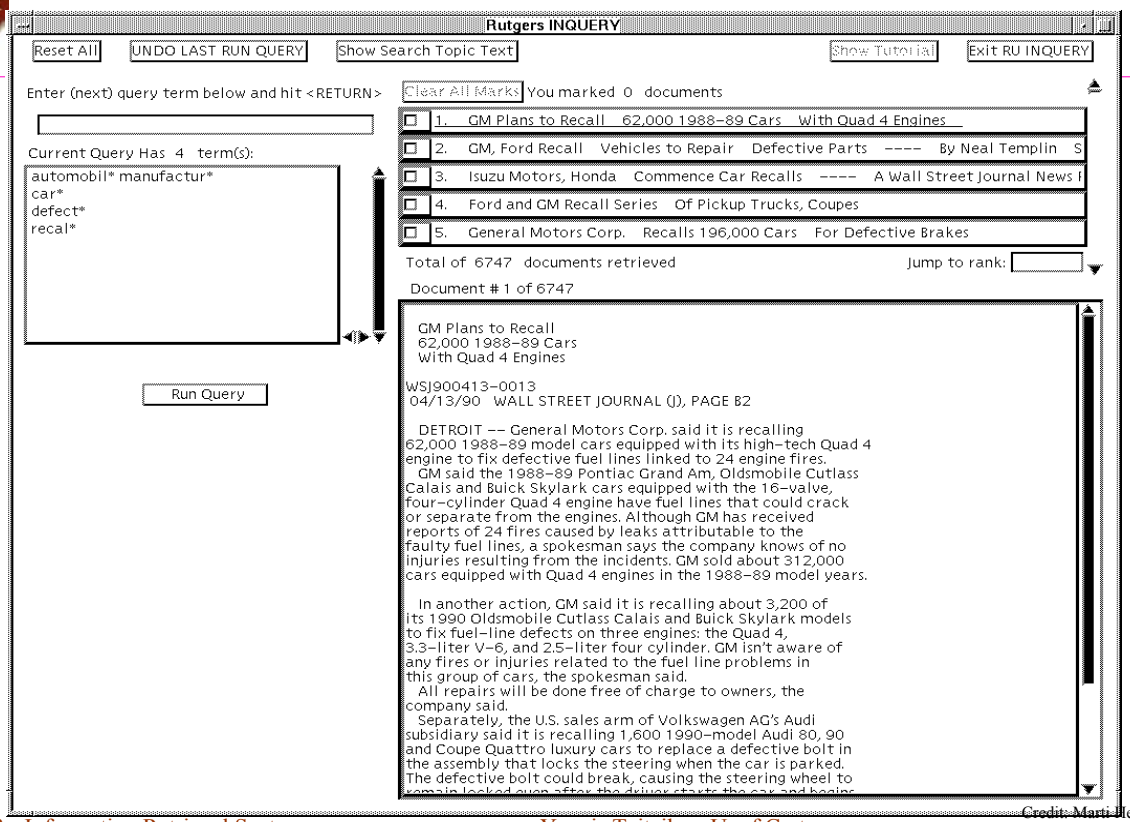
Relevance Feedback Evaluation: Case Study

Example of evaluation of interactive information retrieval [Koenemann & Belkin 1996]

Goal of study: show that relevance feedback improves retrieval effectiveness

Details

- 64 novice searchers (43 female, 21 male, native English)
- TREC test bed (Wall Street Journal subset)
- Two search topics
  - Automobile Recalls
  - Tobacco Advertising and the Young
- Relevance judgements from TREC and experimenter
- System was INQUERY (vector space with some bells and whistles)
- Subjects had a tutorial session to learn the system
- Their goal was to keep modifying the query until they have developed one that gets high precision
- Reweighting of terms similar to but different from Rocchio



Credit: Marti Hearst

Evaluation:Precision vs. RF condition (from Koenemann & Belkin 96)

Criterion: p@30 (precision at 30 documents)

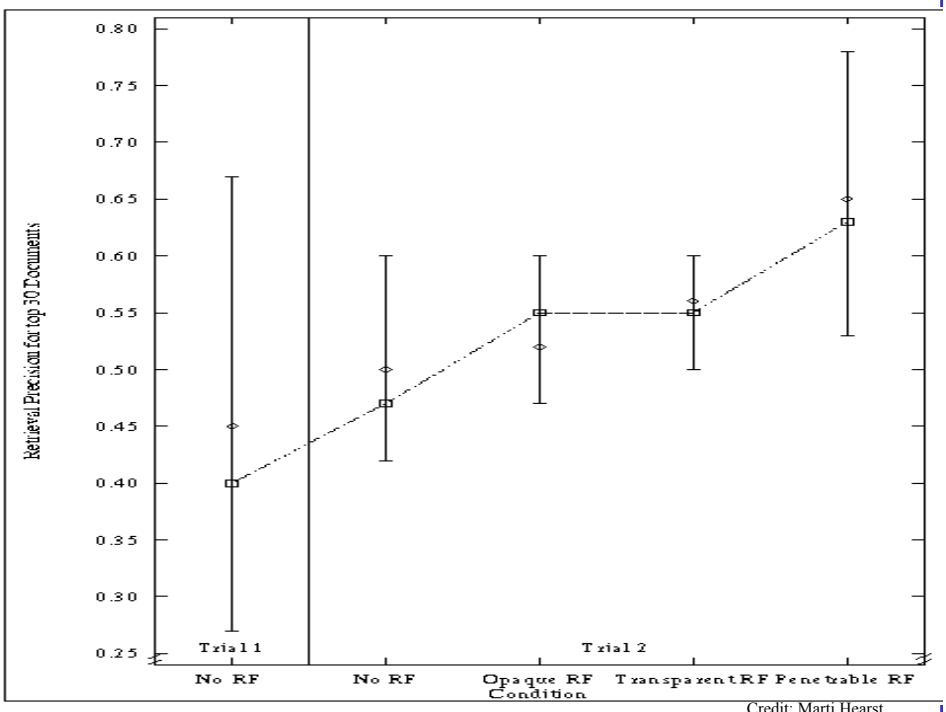
Compare:

- p@30 for users with relevance feedback
- p@30 for users without relevance feedback

Goal: show that users with relevance feedback do better

Results:

- Subjects with relevance feedback had 17-34% better performance
- But: Difference in precision numbers not statistically significant. Search times approximately equal



Credit: Marti Hearst

Σιωπηρές Υποθέσεις της Ανάδρασης Συνάφειας

A1: ***User has sufficient knowledge for formulating the initial query***.

- However:
  - User does not always have sufficient initial knowledge.
  - Examples: Misspellings, Mismatch of searcher's vocabulary vs collection vocabulary.

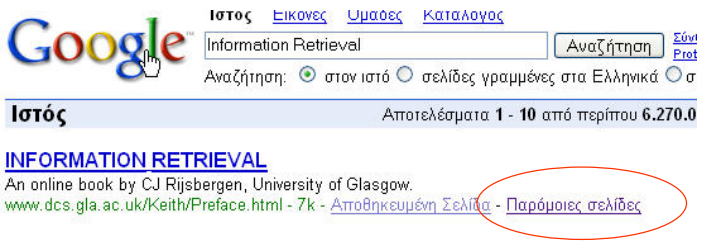
A2: ***Relevance prototypes are "well-behaved".***

- Either: All relevant documents are similar to a single prototype.
- Or: There are different prototypes, but they have significant vocabulary overlap.
- However:
  - There are several relevance prototypes.

Γιατί η ανάδραση συνάφειας δεν χρησιμοποιείται ευρέως;

- Οι χρήστες συχνά διστάζουν να δώσουν είσοδο
- Η ανάδραση έχει ως αποτέλεσμα μεγάλες επερωτήσεις των οποίων ο υπολογισμός απαιτεί περισσότερο χρόνο
  - σε σύγκριση με τις συνηθισμένες επερωτήσεις που διατυπώνουν οι χρήστες οι οποίες αποτελούνται από 2-3 λέξεις
  - (search engines process lots of queries and allow little time for each one)
- Μερικές φορές η νέα απάντηση περιέχει έγγραφα τα οποία δεν μπορούμε να καταλάβουμε πως προέκυψαν

Ανάδραση Συνάφειας στον Παγκόσμιο Ιστό



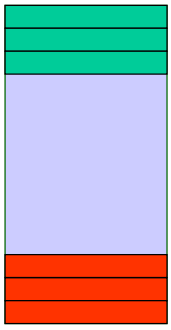
- Some search engines offer a *similar/related pages* feature (simplest form of relevance feedback)
  - **Πολλές φορές ο υπολογισμός αυτών των όμοιων/σχετικών σελίδων δεν γίνεται βάσει του περιεχομένου αλλά βάσει της δομής του γράφου (θυμηθείτε την ανάλυση συνδέσμων). Ο υπολογισμός είναι αρκετά πιο γρήγορος.**
- But some don't because it's hard to explain to average user.
  - "Excite" initially had true relevance feedback, but abandoned it due to lack of use.

Ψευδοανάδραση Συνάφειας

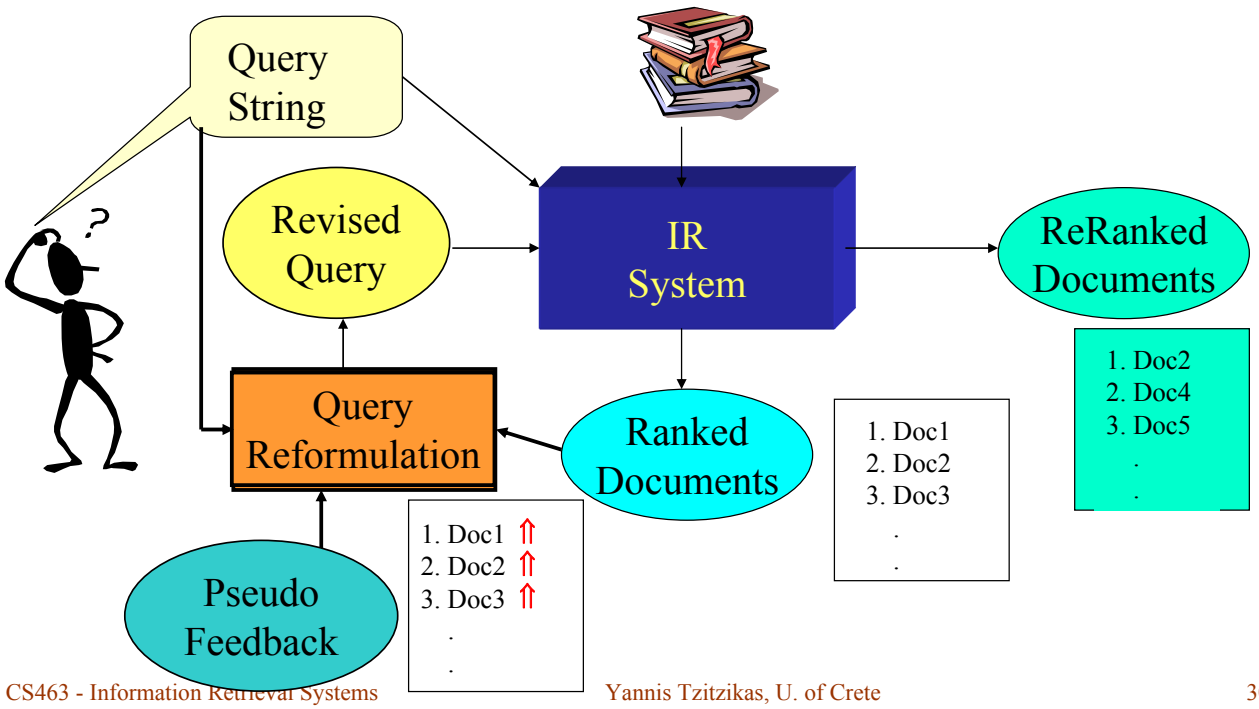
Ψευδοανάδραση Συνάφειας
Pseudo Relevance Feedback

- Χρήση μεθόδων ανάδρασης αλλά **χωρίς είσοδο από το χρήστη**
- **Υπόθεση** ότι τα κορυφαία m από τα ανακτημένα έγγραφα είναι συναφή (και χρήση αυτών για ανάδραση)
  - Μπορούμε επίσης να χρησιμοποιήσουμε τα τελευταία έγγραφα για αρνητική ανάδραση
- Επιτρέπει την επέκταση της επερώτησης με όρους που σχετίζονται με τους όρους της επερώτησης

answer(q):



Ψευδοανάδραση Συνάφειας



Αξιολόγηση Ψευδοανάδρασης

- Βρέθηκε να βελτιώνει την απόδοση στο διαγωνισμό του TREC (ad-hoc retrieval task)
- Δουλεύει ακόμα καλύτερα αν τα κορυφαία έγγραφα πρέπει να ικανοποιούν και μια boolean έκφραση προκειμένου να χρησιμοποιηθούν για ανάδραση
  - (π.χ. να περιέχουν όλους του όρους της επερώτησης)

Αναλύοντας περισσότερο την έννοια της **βέλτιστης επερώτησης** (optimal query)

Πηγή:

Yannis Tzitzikas and Yannis Theoharis, **Naming Functions for the Vector Space Model**, *29th European Conference on Information Retrieval, Rome 2-5 April 2007*

The **Naming Problem**

We can view an IR system as a function from set of Queries to set of Answers

$$S: Queries \longrightarrow Answers$$

If q ∈Queries, S(q) denotes the answer of q.
Classically IR systems are good at "solving" the equation *S(q)=A* wrt *A*, *i.e.*:

$$S(q) = ?$$

The **naming problem** is the problem of solving the equation wrt *q*, *i.e.* :

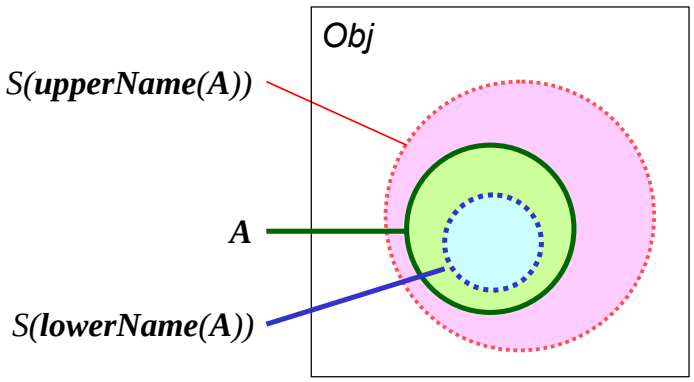
$$S(?) = A$$

We can distinguish two formulations of the naming problem:
• For **unordered** answers (here A is a subset of *Obj*)
• For **ordered** answers (here A is an ordered subset of Obj).
where Obj is the set of stored objects (e.g. documents, web pages, etc)

The Naming Problem for **Unordered** Sets

Basic Notions

- **Exact** name
- **Upper** name
  - **Best Upper** Name
- **Lower** name
  - **Best Lower** Name
- **Relaxed** name



The Naming Problem for **Ordered** Sets

Basic Notions

- **Exact** name
- **Upper** name
  - **Best Upper** Name
- **Lower** name
  - **Best Lower** Name
- **Relaxed** name

Example:

A = ⟨**d1**, **d2**, **d3**⟩

$S(exactName(A))$ = ⟨**d1**, **d2**, **d3**, d8, d9, …⟩

$S(lowerName(A))$ = ⟨**d1**, **d2**, d5, d7, …⟩

$S(upperName(A))$ = ⟨**d1**, d9, **d2**, d5, **d3**, d8, …⟩

Notations

Let A be an answer. Some notations that will be used

- **A(k)** : the **ordered** set comprising the **first k** elements of **A**
- **A{k}** : the **set** of elements that appear in **A(k)**
- $A_{|F}$: the **restriction** of A on the set F, i.e. the **ordered set** obtained if we **exclude** from A those elements that do not belong to F,

Example

if A = ⟨d1, d2, d3⟩ then

- A(2)= ⟨d1, d2⟩
- A{2}= {d1, d2}
- {A} = A{|A|} = {d1, d2, d3}
- if F={d1,d3}, then $A_{|F}$= ⟨d1, d3⟩

Defining Formally Relaxed/Upper/Lower/Exact Names

A query **q** is a **relaxed name** of an answer A iff :

Case: A is a set : $|S(\mathbf{q})\{m\} \cap A| = j$ where $m \geq j > 0$.

Case: A is an ordered set: $S(\mathbf{q})(m)_{|\{A(j)\}} = A(j)$ where $m \geq j > 0$.

- If m=j=|A| then q is an **exact** name
- If m>j=|A| then q is an **upper** name (the **best upper** name if m is the least possible)
- If m=j<|A| then q is a **lower** name (the **best lower** name if m is the greatest possible)

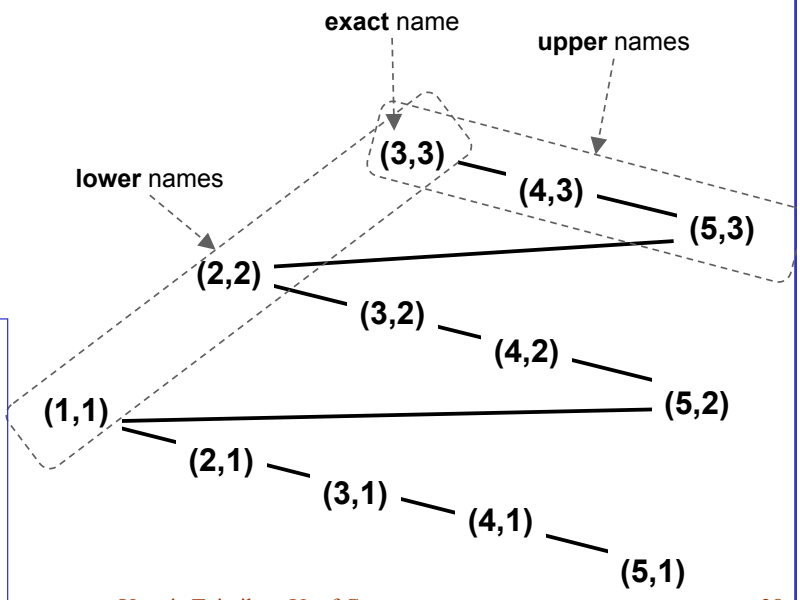
So each query can be characterized by a pair (m,j). We can now define an ordering over these pairs. The ordering should reflect the quality of the queries (as solutions for the naming problem).

Defining a total ordering of queries

(m,j) > (m',j') iff: (j>j') or (j=j' and m<m')

Let |Obj|=5 and |A|=3

Let |Obj|=5 and |A|=3.
The ordering of the corresponding pairs are:



So far, we have defined what exact, upper, lower, relaxed names are, and we defined an ordering over them.

The next question is whether and how we can compute these names for a given answer A.

Investigating possible approaches for solving the Naming Problem for **Unordered** Sets

Let A = {**d1**, **d2**, **d3,…, dn**}

Possible name queries

- qa= ½(di,dj) where (di,dj) = arg max {dist(d,d') | d,d' ∈A}
- qb=1/|A| Σ { d | d ∈A}
- qc=1/|A| Σ { a | d ∈A} - 1/|Obj-A| Σ { d | d ∉ A}

Notes

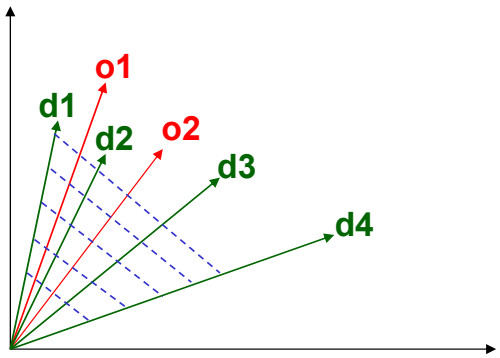
- qa: minimizes the maximum dist from the elements of A
- qb: minimizes the average dist from the elements of A
- qc: is the Rocchio method (avg A, - avg Obj-A)

Important remarks:

- **None** is guaranteed to be the exact name (they are all **relaxed** names)
- However, if an exact name does not exist, then qa is the **best upper** name
- Moreover, the evaluation of qa (i.e. the computation if its answer) is **faster** than the evaluation of qb, qc.

The Naming Problem for **Unordered** Sets (III)

Method (a): Let A = {**d1**, **d2**, **d3, d4**}



1st step:

Find the pair of most distant documents

Here: (**d1, d4**)

2nd step:

Find whether other documents lie in the area specified by (**d1, d4**)

Here: {**o1, o2**} != ∅

Thus, qa = ½(**d1, d4**) is an **upper name** of A

If there were no other documents then qa would be an exact name

- Cost
  - 1st step: O(|A|²) computations of similarity
  - 2nd step: O(|Obj|) or compute the answer of qa and check if S(qa){|A|}=A

The Naming Problem for **Ordered** Sets

Let A = ⟨**d1**, **d2**, **d3**,…, **dn**⟩

Let ***i*** the max integer for which it holds:

sim(**d1**,d2) > sim(**d1**,d3) > … > sim(**d1**,di)

d1 is a
- **relaxed** name, always
- **upper** name, if **i**=n
- **exact** name, if **i**=n and S(**d1**)(n)=A
- **lower** name, if **i**<n and S(**d1**)(**i**)=A(**i**)

Computational cost:

- O(n) computations of similarity to find i.
- One query evaluation in order to decide whether d1 is lower/exact name.

Experimental Evaluation

- Experiments conducted over the experimental web search engine GRoogle
  - http://groogle.csd.uoc.gr:8080/groogle (2006-2007)
- The set A was selected randomly, the average times are listed in the Table below
  - |O|: number of documents, |T|: number of terms

Collection			Naming Functions (in sec)					
			Unordered			Ordered		
$\|O\|$	$\|T\|$	$\|\{A\}\|$	t_a	$t_b(query\ terms)$	t_c	t_a	$t_b(query\ terms)$	t_c
1K	40K	10	0.015	1.566 (5) - 3.174 (10)	0.001	0	1.878 (5) - 3.575 (10)	0.008
1K	40K	100	0.328	1.56 (5) - 3.176 (10)	0.005	0	1.624 (5) - 3.192 (10)	0.004
5K	255K	10	0.015	112.1 (5) - 262.5 (10)	0.001	0	131.2 (5) - 264.7 (10)	0.048
5K	255K	100	0.328	116.0 (5) - 251.1 (10)	0.006	0	153.4 (5) - 271.1 (10)	0.152

ta: time to find the query (fast, depends on |A|, not the size of the db)

tb: time to evaluate the query (this is the only expensive task)

tc: time to decide what kind of name it is (fast)

Concluding Remarks and Further Research

- The naming problem has several applications (e.g. for relevance feedback or for providing a flexible interaction scheme between the system and the users)
- We expressed formally several variations of this problem and related optimality criteria (to best of our knowledge this analysis is novel)
- We provided optimal solutions
  - E.g. we provided a method that certainly returns the best upper name for unordered sets (this is not true for the Rocchio method)
- We described (polynomial) algorithms for solving these problems
- Future research
  - Extend the problem statement with an additional parameter: the maximum number of words that a name could have, e.g. “find the best upper name with no more than 3 words”.
  - Indexing structures for efficient computation of names
- For more see
  http://www.ics.forth.gr/~tzitzik/publications/2007_TzitzikasTheoharisNaming.pdf