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:

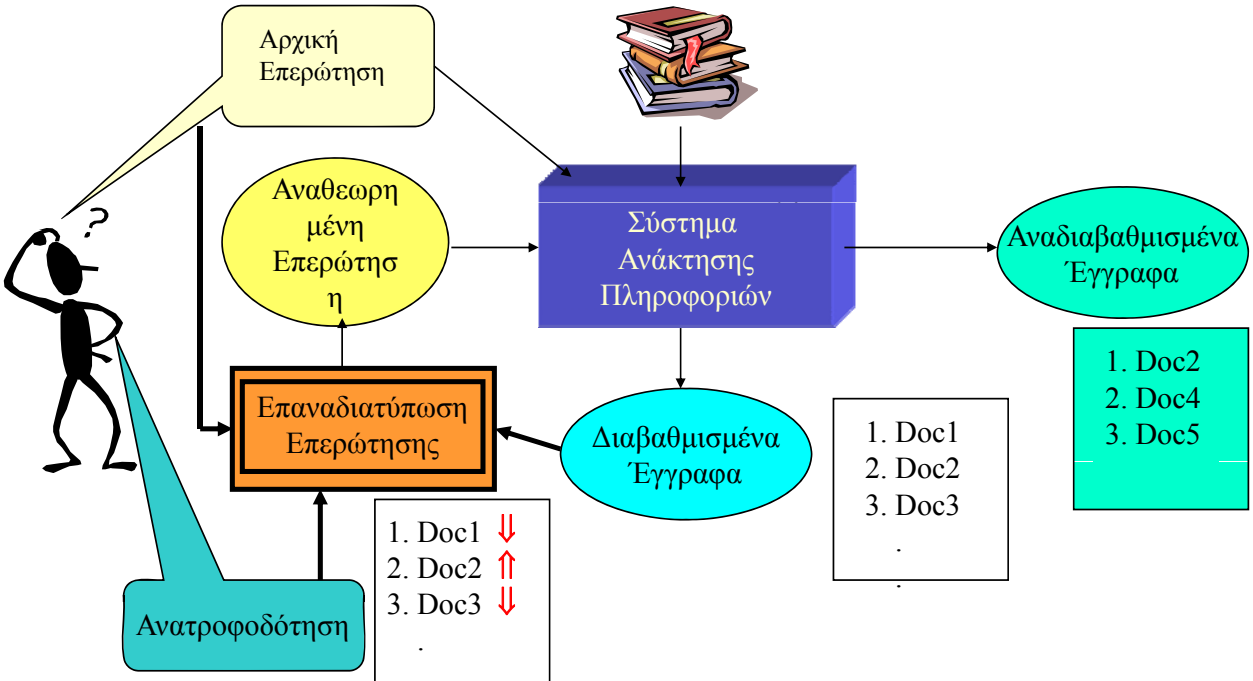
1/ Μετά την παρουσίαση των αποτελεσμάτων, επιτρέπουμε στο χρήστη **να κρίνει (θετικά ή αρνητικά) την συνάφεια** ενός ή περισσότερων εγγράφων της απάντησης

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

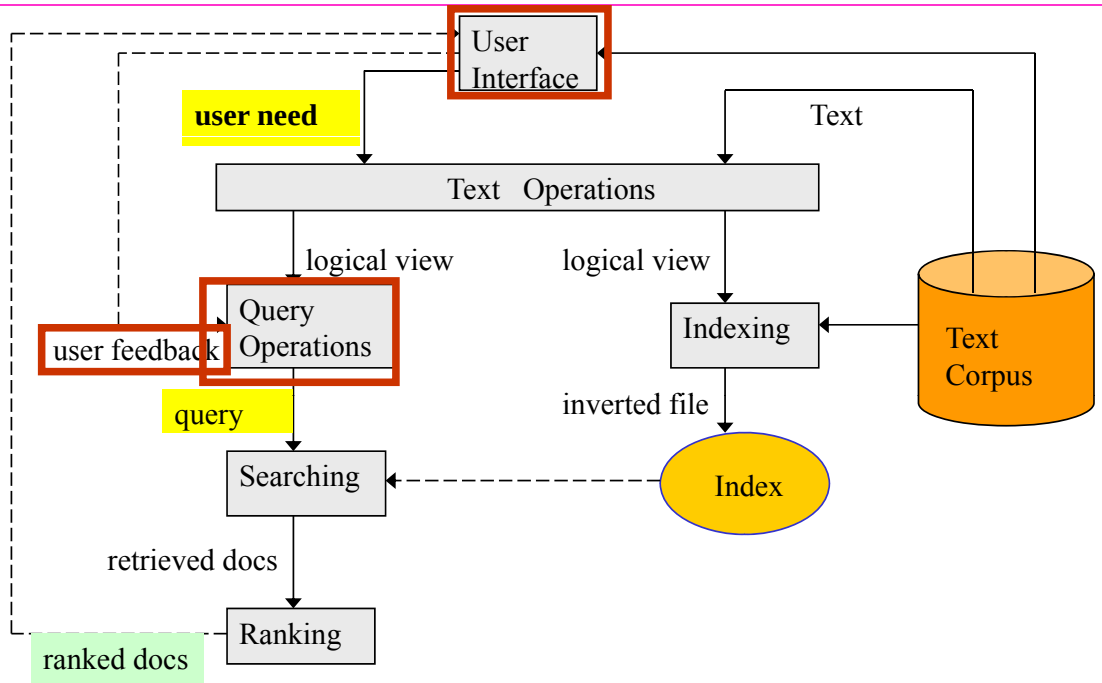
3/ Κατόπιν δίδουμε στο χρήστη την απάντηση της αναδιατυπωμένης επερώτησης

4/ Πήγαινε στο βήμα 1/

Ανάδραση Συνάφειας (Relevance Feedback): Η βασική ιδέα

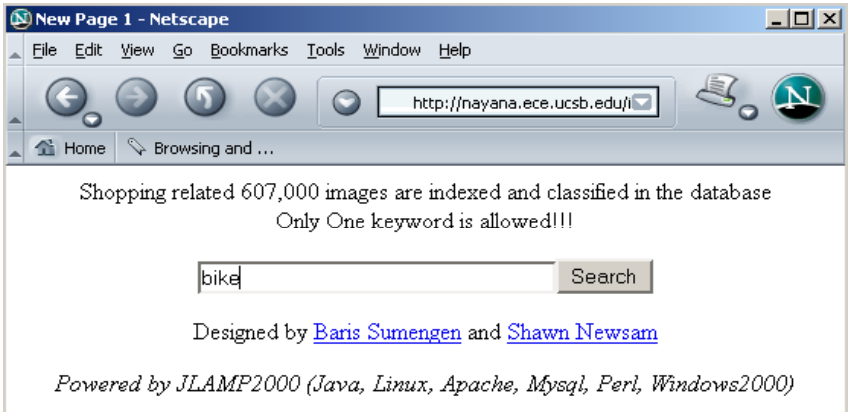


Τμήματα της Αρχιτεκτονικής που Εμπλέκονται



Παράδειγμα ανατροφοδότησης συνάφειας σε σύστημα ανάκτησης εικόνων

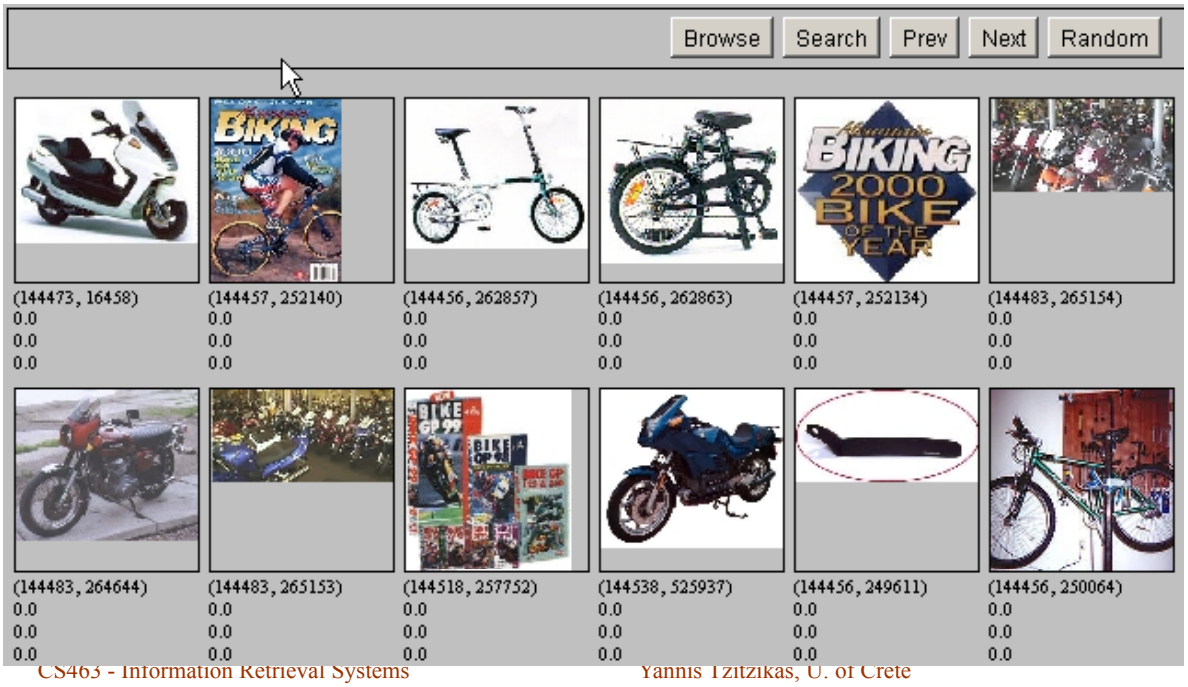
q=bike



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

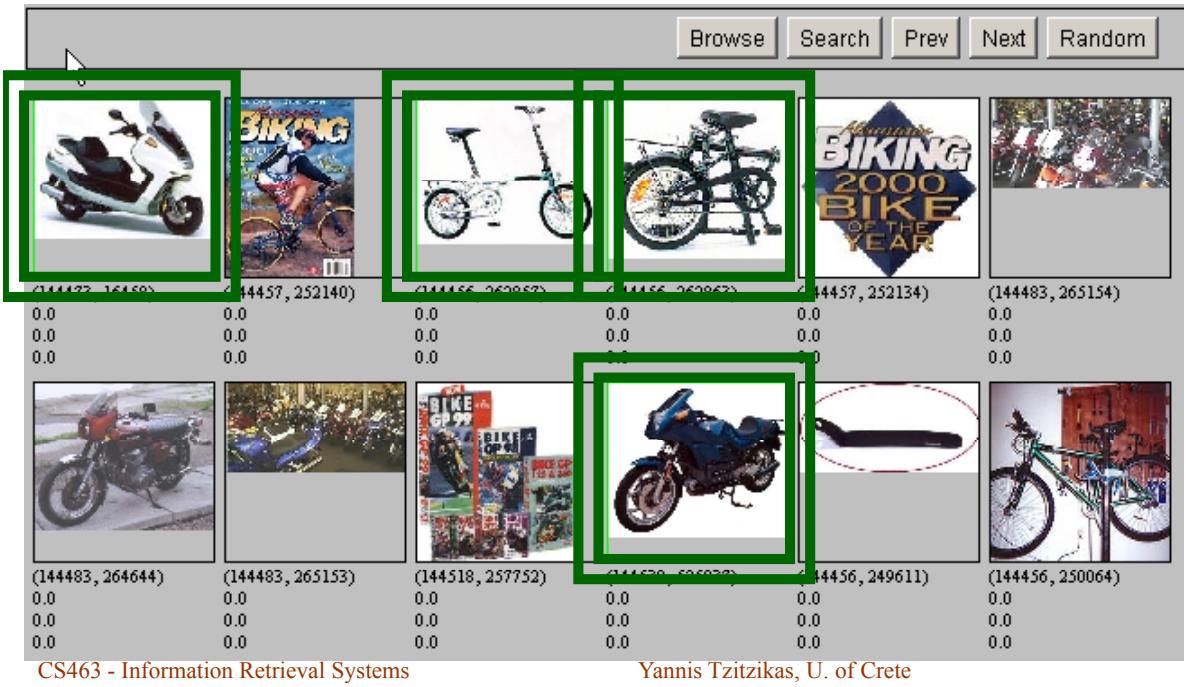
Παράδειγμα ανατροφοδότησης συνάφειας σε σύστημα ανάκτησης εικόνων

Answer("bike")=



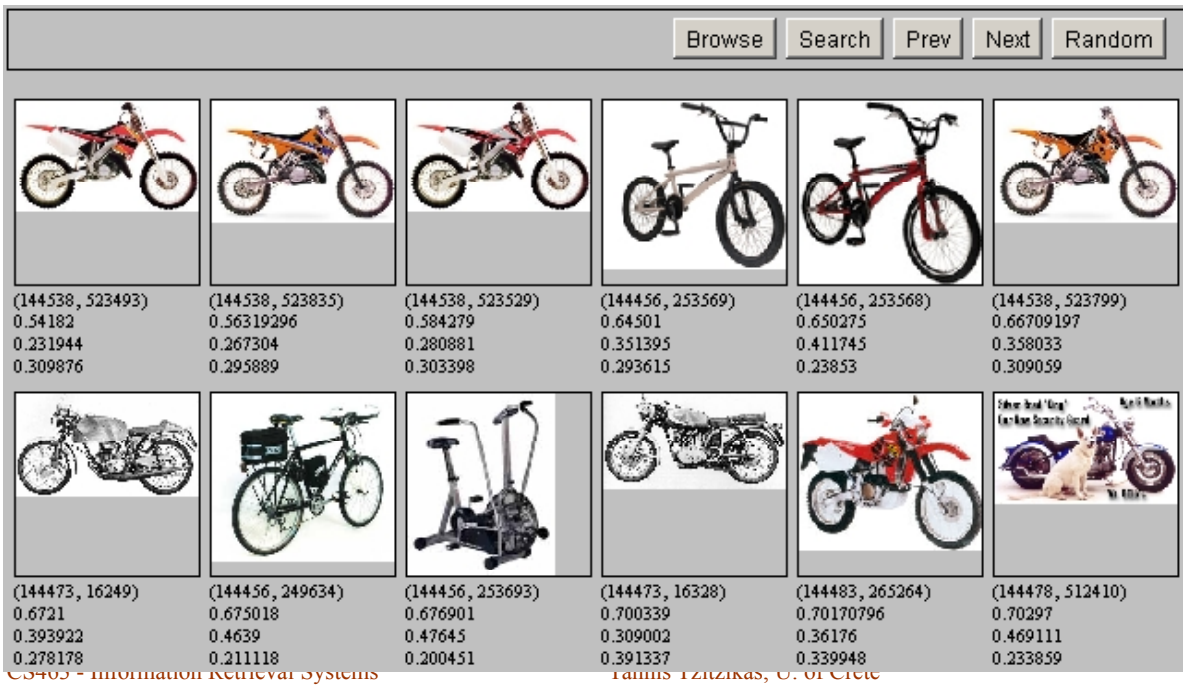
Παράδειγμα ανατροφοδότησης συνάφειας σε σύστημα ανάκτησης εικόνων

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.

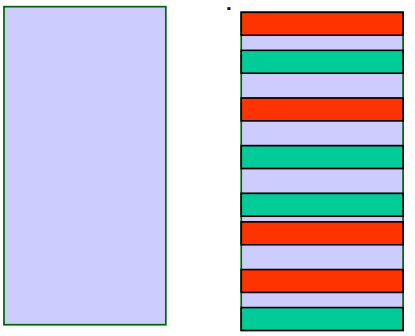
(θα αναλύσουμε περισσότερο αυτό το ζήτημα αργότερα)

→ 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

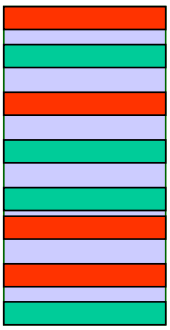
Αφού το σύνολο όλων των συναφών είναι άγνωστο,

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

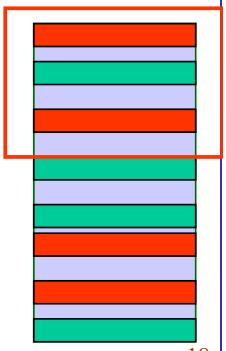
Τάση για απόρριψη **μόνο** των μη-συναφών εγγράφων που έχουν υψηλό σκορ

(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).

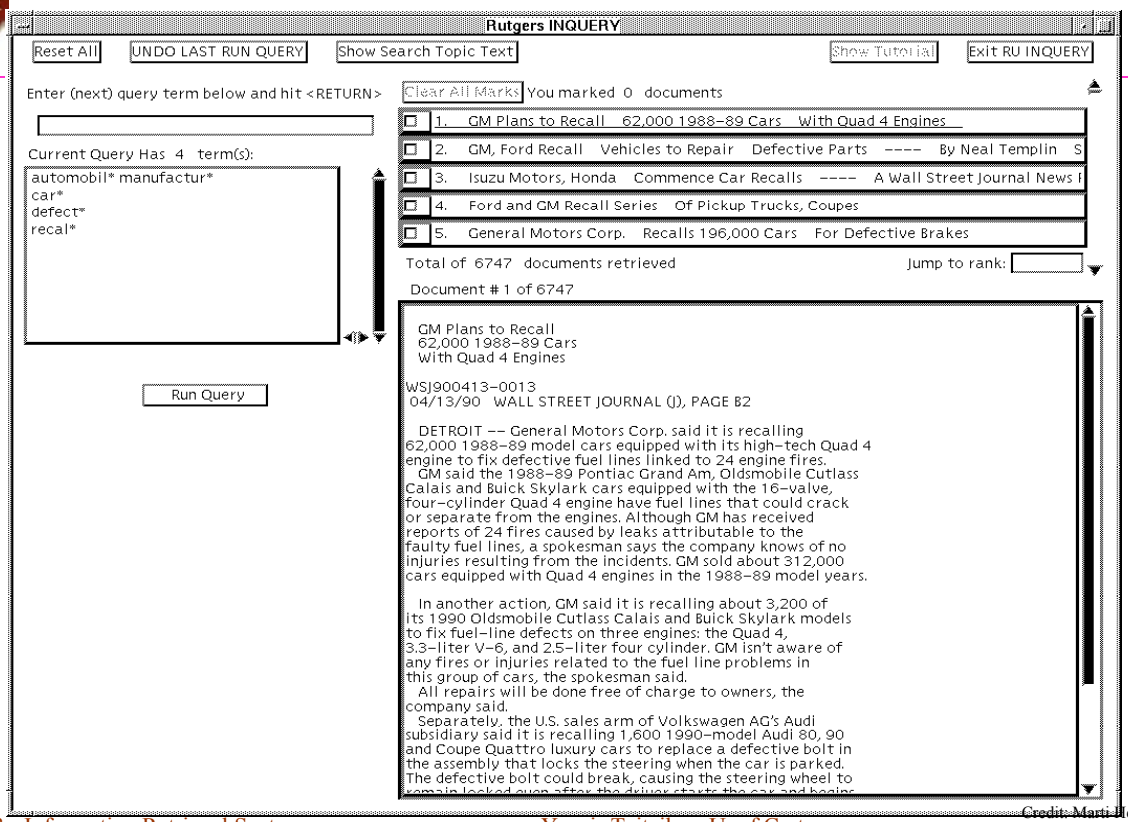
Relevance Feedback Evaluation: Case Study

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)

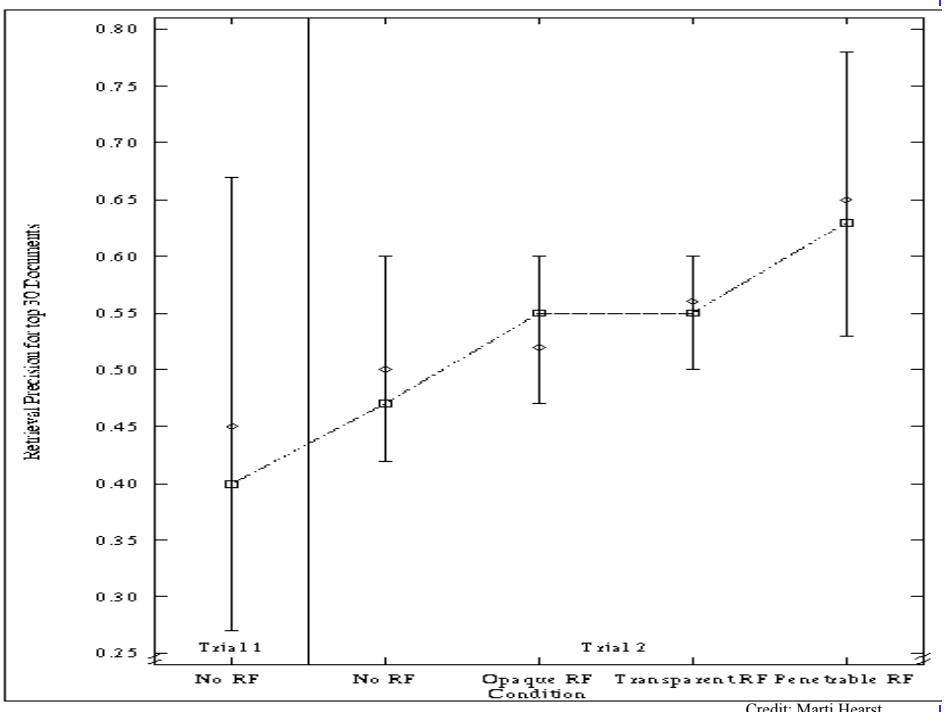
Compare:

- 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

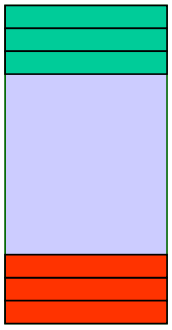
- Some search engines offer a *similar/related pages* feature (simplest form of relevance feedback)
  - **Πολλές φορές ο υπολογισμός αυτών των όμοιων/σχετικών σελίδων δεν γίνεται βάσει του περιεχομένου αλλά βάσει της δομής του γράφου (θυμηθείτε την ανάλυση συνδέσμων). Ο υπολογισμός είναι αρκετά πιο γρήγορος.**
- 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

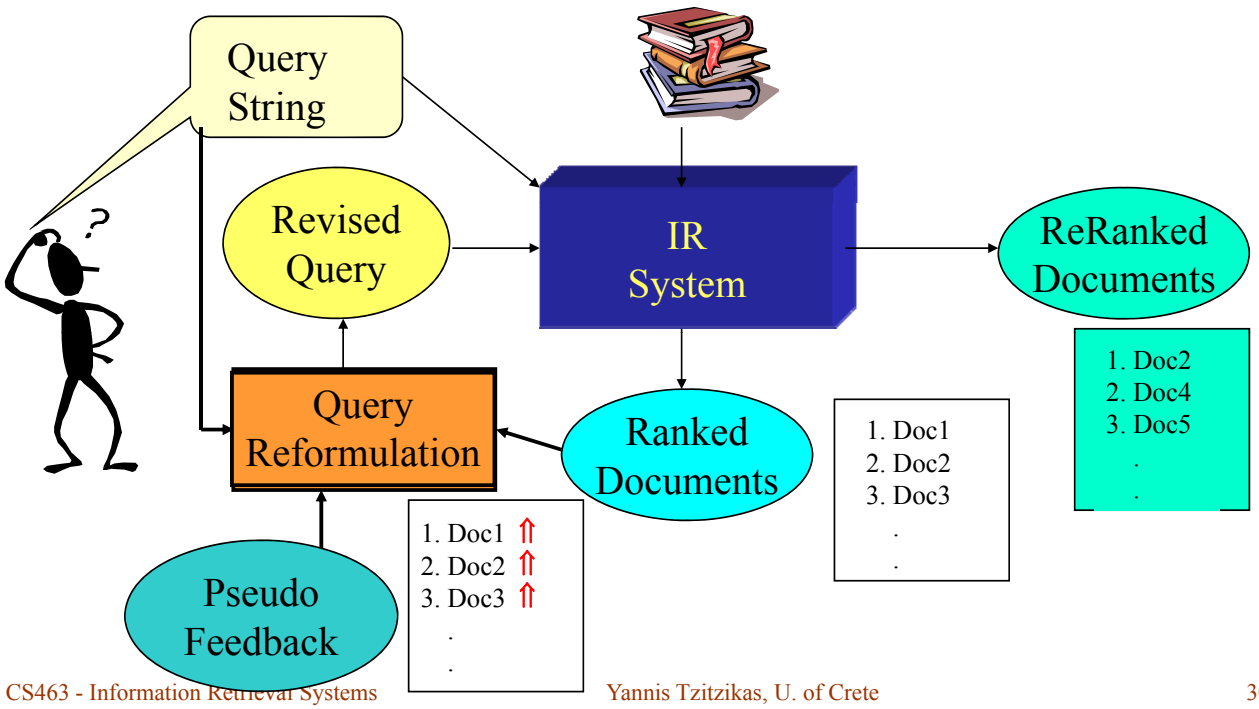
Ψευδοανάδραση Συνάφειας
Pseudo Relevance Feedback

- 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.* :

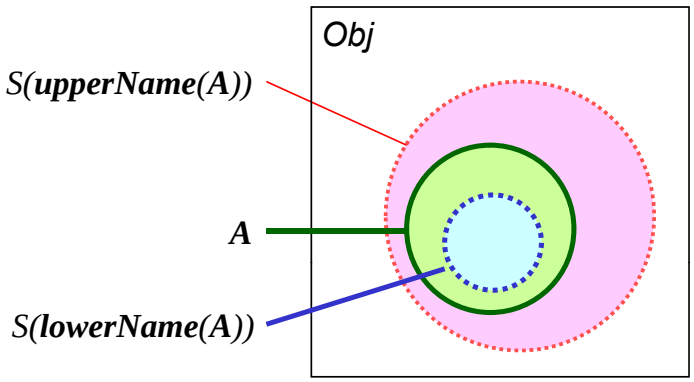
$$S(?) = A$$

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)**
- $\mathbf{A_{|F}}$: 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)

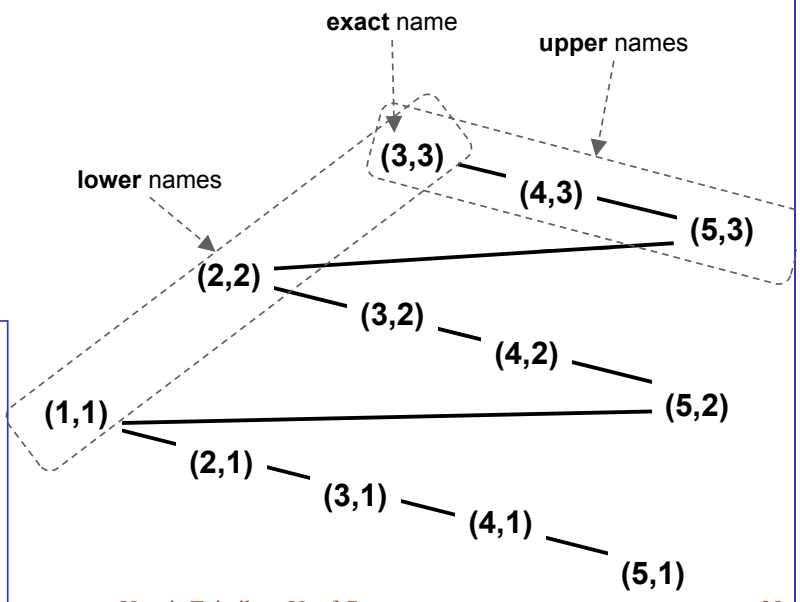
So each query can be characterized by a pair (m,j). We can now define an ordering over these pairs. The ordering should reflect the quality of the queries (as solutions for the naming problem).

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 { d | d ∈A} - 1/|Obj-A| Σ { d | d ∉ A}

Notes

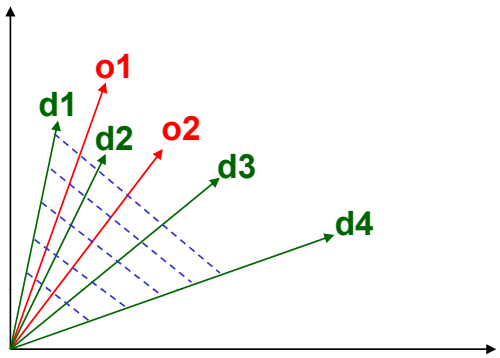
- qa: minimizes the maximum dist from the elements of A
- qb: minimizes the average dist from the elements of A
- qc: is the Rocchio method (avg A, - avg Obj-A)

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
- 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

Πανεπιστήμιο Κρήτης, Τμήμα Επιστήμης Υπολογιστών
Άνοιξη 2009

ΗΥ463 - Συστήματα Ανάκτησης Πληροφοριών
Information Retrieval (IR) Systems

Προχωρημένες Λειτουργίες Επερώτησης (ΙΙ)
Advanced Query Operations

Γιάννης Τζίτζικας

Διάλεξη : 12
Ημερομηνία :

Επέκταση Επερώτησης
(Query Expansion)

In *relevance feedback*, users give additional input (relevant/non-relevant) on documents.

In *query expansion*, users give additional input (good/bad search term) on words or phrases

Επέκταση Επερώτησης
(Query Expansion)

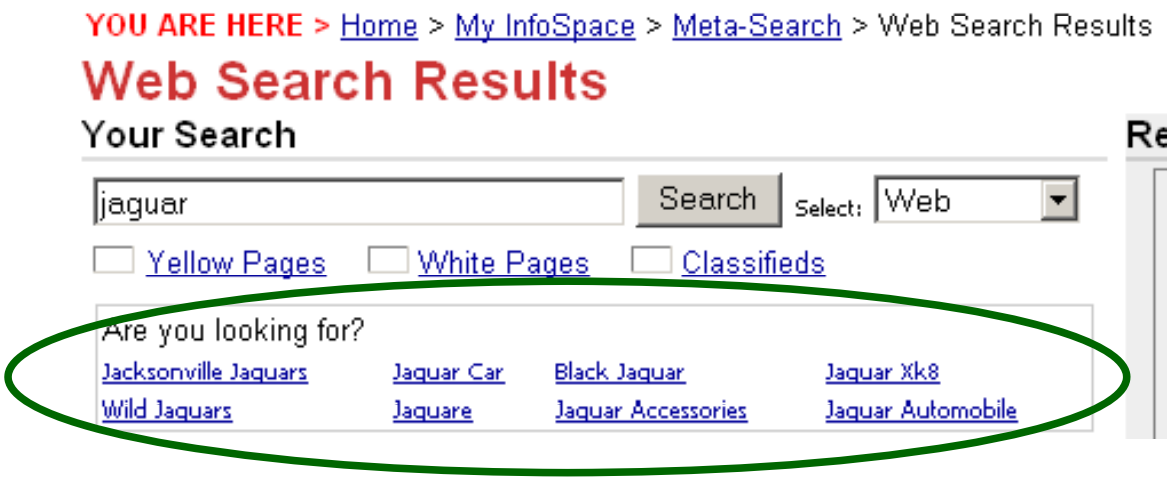
- Τοπική Ανάλυση
  - Αναλύουμε τα (κορυφαία) έγγραφα της απάντησης
- Καθολική Ανάλυση
  - Αναλύουμε όλα τα έγγραφα της συλλογής

Επέκταση Επερώτησης (Query Expansion)
Τοπική Ανάλυση (Local Analysis)

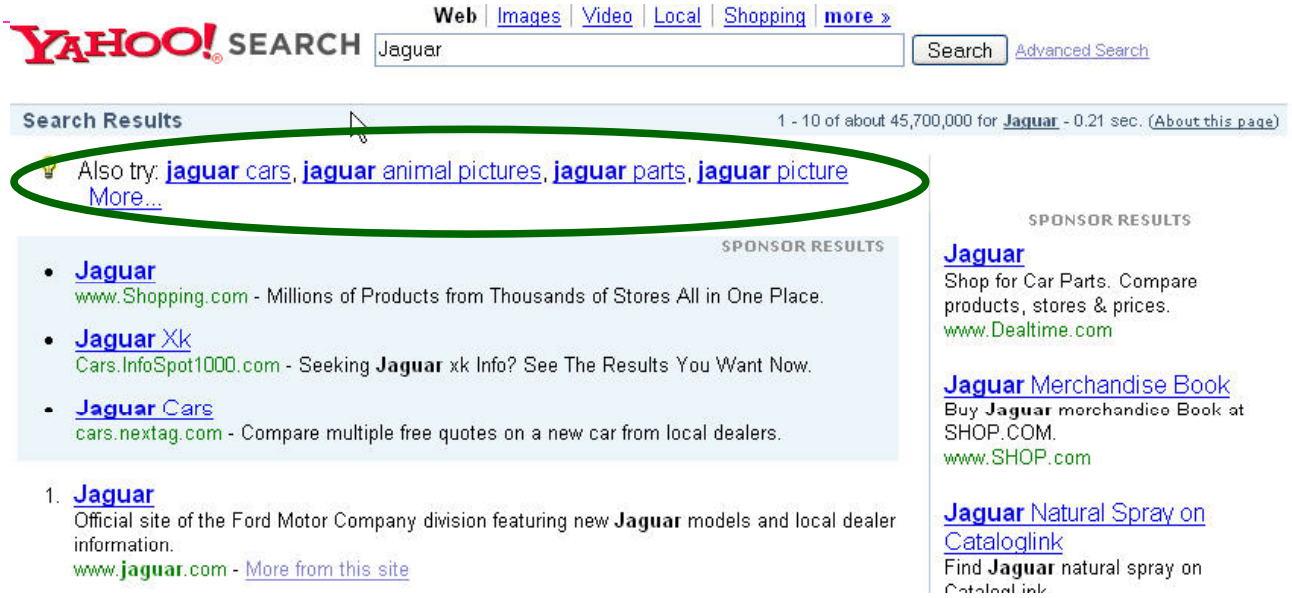
Αυτόματη Τοπική (Επιτόπια) Ανάλυση
Automatic Local Analysis

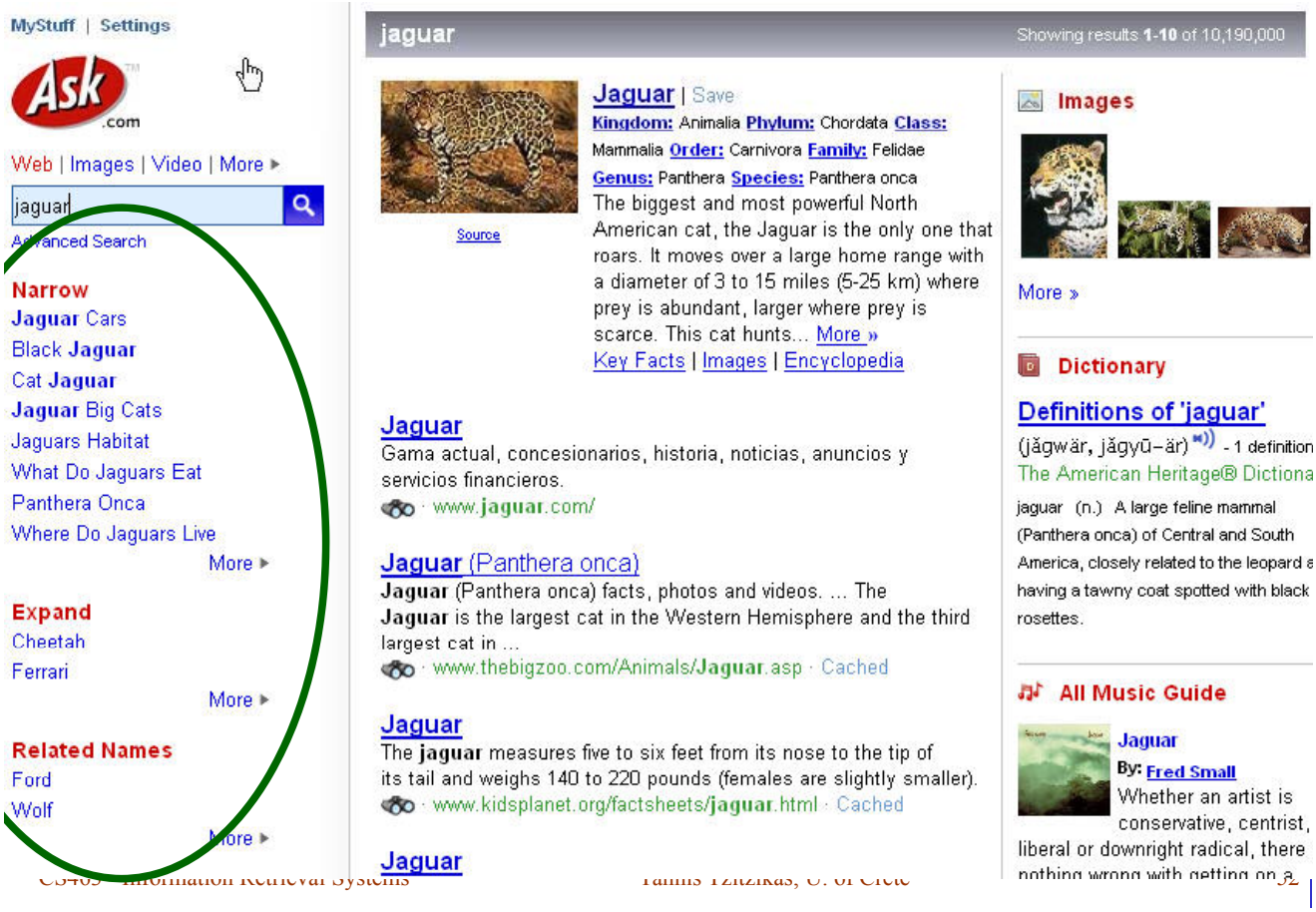
- Μετά την διατύπωση της επερώτησης, ανάλυσε (στατιστικά) τις λέξεις που εμφανίζονται **μόνο** στα κορυφαία ανακτημένα έγγραφα
  - π.χ. επιλέγουμε τις 10 πιο συχνά εμφανιζόμενες λέξεις των κορυφαίων 5 εγγράφων
- Το σύστημα παρουσιάζει στο χρήστη τις πιο συχνά εμφανιζόμενες λέξεις και ο αυτός επιλέγει εκείνες που θέλει να προστεθούν στην επερώτηση
  - εναλλακτικά η επιλογή μπορεί να γίνει αυτόματα (χωρίς την παρέμβαση ή συγκατάθεση του χρήστη)
- Επίδραση στην αποτελεσματικότητα της ανάκτησης
  - Οι ασαφείς (ή αμφίσημες) λέξεις δημιουργούν λιγότερα προβλήματα (απ' ότι στην καθολική ανάλυση – την οποία θα αναλύσουμε παρακάτω)
  - Παράδειγμα: με τοπική ανάλυση η επερώτηση "Apple computer" μπορεί να επεκταθεί στην "Apple computer Powerbook laptop"

Παράδειγμα εφαρμογής



Also: see altavista, teoma





Στο groogle/mitos

A very simple technique is currently supported:

- For each term t_i that appears in the top L (by default L=5) documents returned by the Query Evaluator, we sum its term frequencies (i.e. all tf_{ij} where j in top-L documents) and we recommend to the user the S terms (by default S=5) with the highest accumulative frequency.

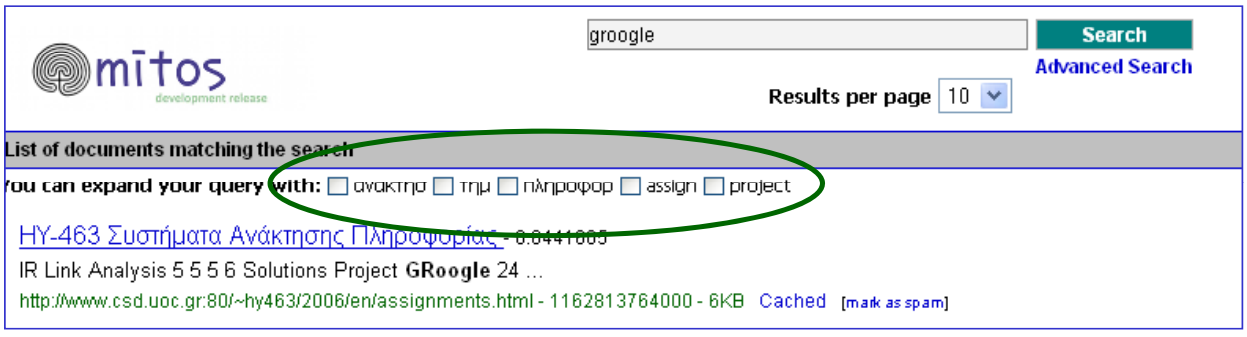


Table 9. Query Expansion Examples

	Initial Query	Expanded Terms				
1	retrieval	imag	medic	index	storag	system
2	web	system	servic	page	process	cours
3	user	interfac	layer	system	develop	softwar

Table 10. Query Expansion Average Times

L	Time (sec)
5	0.002
10	0.003
15	0.004
20	0.004

Αυτόματη Τοπική (Επιτόπια) Ανάλυση

Τεχνικές αυτόματης τοπικής ανάλυσης

- Association Matrix
  - based on the co-occurrence of terms in documents
- Metric Correlation Matrix
  - based on the co-occurrence and proximity of terms in documents
- //Scalar Clusters
- //Local context analysis

(a) Association Matrix and Normalized Association Matrix

D: τα έγγραφα της απάντησης και $w_1 ... w_n$ οι όροι που εμφανίζονται σε αυτά.

	w_1	w_2	w_3	.....	w_n
w_1	c_{11}	c_{12}	c_{13}	.....	c_{1n}
w_2	c_{21}				
w_3	c_{31}				
.	.				
.	.				
w_n	c_{n1}				

c_{ij}: Correlation factor between term i and term j:

$$c_{ij} = \sum_{d_k \in D} f_{ik} \times f_{jk}$$

f_{ik} : frequency of term i in document k

- Normalized Association Matrix
  - Frequency based correlation factor favors more frequent terms.
  - Normalize association scores: $$s_{ij} = \frac{c_{ij}}{c_{ii} + c_{jj} - c_{ij}}$$
  - Normalized score is 1 if two terms have the same frequency in all documents in D.

*Από αυτόν τον πίνακα μπορούμε να βρούμε τους όρους που είναι **<u>πιο κοντά σε αυτούς της επερώτησης</u>** (θυμηθείτε και τον πίνακα συσχέτισης στο fuzzy model του μαθήματος 4)*

(b) Metric Correlation Matrix

- Association correlation does not account for the **proximity** of terms in documents, just co-occurrence frequencies within documents.
- **Metric correlations** account for term proximity.

V_i: Set of all occurrences of term i in any document in D.
$r(k_u,k_v)$: Distance in words between word occurrences k_u and k_v ($=\infty$ if k_u and k_v are occurrences in different documents).

$$c_{ij} = \sum_{k_u \in V_i} \sum_{k_v \in V_j} \frac{1}{r(k_u, k_v)}$$

- Normalized Metric Correlation Matrix
  - to account for term frequencies:

$$s_{ij} = \frac{c_{ij}}{|V_i| \times |V_j|}$$

Query Expansion with (Association or Metric) Correlation Matrix

	w_1	w_2	w_3	.....	w_n
w_1	c_{11}	c_{12}	c_{13}	.....	c_{1n}
w_2	c_{21}				
w_3	c_{31}				
.	.				
.	.				
w_n	c_{n1}				

- For each term i in the query q, expand query with n terms, those with the highest value of c_{ij}.
- This adds semantically related terms in the "neighborhood" of the query terms.

Αυτόματη Καθολική Ανάλυση
(Automatic Global Analysis)

Αυτόματη Καθολική Ανάλυση
Automatic Global Analysis

- Προσδιορισμός βαθμού ομοιότητας μεταξύ των όρων βάσει στατιστικής ανάλυσης ολόκληρης της συλλογής
  - Υπολογισμός πινάκων συσχέτισης (association matrices) που ποσοτικοποιούν την ομοιότητα μεταξύ των όρων ανάλογα με το πόσο συχνά συνεμφανίζονται
- Επέκταση επερώτησης με τους πιο όμοιους όρους.
- Επίδραση στην αποτελεσματικότητα της ανάκτησης
  - Οι ασαφείς (ή αμφίσημες) λέξεις δημιουργούν περισσότερα προβλήματα (απ' ότι στην τοπική ανάλυση)
  - Παράδειγμα: με καθολική ανάλυση η επερώτηση "`Apple computer`" μπορεί να επεκταθεί στην "`Apple red fruit orange computer`"
- Μια λύση:
  - Query Expansion Based on a Similarity Thesaurus

Query Expansion Based on a **Similarity Thesaurus**

- Βασική ιδέα
  - Οι όροι που προστίθενται στην επερώτηση καθορίζονται με βάση την απόσταση τους **από ολόκληρη την επερώτηση** (και όχι βάσει της απόστασής τους από κάθε όρο της επερώτησης ξεχωριστά)
- Στην αντίθετη περίπτωση θα είχαμε:
  - "Apple computer" → "Apple red fruit computer"
- Ενώ τώρα
  - "fruit" not added to "Apple computer" since it is far from "computer."
  - "fruit" added to "apple pie" since "fruit" close to both "apple" and "pie."

Query Expansion Based on a Similarity Thesaurus

- Τρόπος
  - Έστω N έγγραφα, t όροι K={k1,…,kt}
  - Παριστάνουμε **κάθε όρο** με ένα διάνυσμα στον χώρο των N διαστάσεων
    - Είναι σαν να έχουμε αντιστρέψει το ρόλο των όρων και των εγγράφων: έχουμε λοιπόν μια διανυσματική παράσταση των όρων (κάθε έγγραφο αποτελεί μια διάσταση στο χώρο των διανυσμάτων). Προσαρμόζουμε το σχήμα βάρυνσης TF-IDF βάσει αυτής της θεώρησης.

$$\vec{k}_i = (w_{i1},...,w_{iN})$$

$$w_{ij} = \frac{(0.5+0.5\frac{f_{ij}}{\max_j(f_{ij})})itf_j}{\sqrt{\sum_{l=1}^{N}(0.5+0.5\frac{f_{il}}{\max_l(f_{il})})^2 itf_j^2}}$$

itf: Inverse term frequency (το ανάλογο του idf):

$$itf_j = \frac{\text{Num of terms in the collection}}{\text{Num of distinct terms in } d_j}$$

←ανάλογο της βάρυνσης TF*IDF μόνο που εδώ χρησιμοποιούμε το το inverse term frequency.

Query Expansion Based on a Similarity Thesaurus (II)

- Υπολογισμός ομοιότητας δυο όρων
  - (π.χ. με εσωτερικό γινόμενο)

$$c_{u,v} = \vec{k}_u \cdot \vec{k}_v$$

- Τα βήματα για την επέκταση της επερώτησης
  - (1) Represent query in the concept space that we used to represent terms

$$\vec{q} = \sum_{ki \in q} w_{iq} \vec{k}_i$$

  - (2) Compute sim(q,ku) for each ku

$$sim(q, k_u) = \vec{q} \cdot \vec{k}_u$$

  - (3) Expand q with the top r ranked terms. The weight of each added term ku is set

$$w_{uq'} = \frac{sim(q, k_u)}{\sum_{k_i \in q} w_{iq}}$$

- Results
  - 20% improved retrieval performance

Καθολική vs. Επιτόπια Ανάλυση

- Η καθολική ανάλυση έχει μεγάλο υπολογιστικό κόστος αλλά μόνο στην αρχή
  - υποθέτοντας ότι τα έγγραφα της συλλογής είναι σταθερά
- Η τοπική ανάλυση έχει αρκετό υπολογιστικό κόστος για κάθε επερώτηση
  - (παρόλο που το πλήθος των όρων και των εγγράφων είναι μικρότερο αυτού της καθολικής)
- Η τοπική ανάλυση δίδει καλύτερα αποτελέσματα

Επέκταση επερωτήσεων: Συμπεράσματα

- Η επέκταση των επερωτήσεων με σχετιζόμενους όρους μπορεί να βελτιώσει την αποτελεσματικότητα της ανάκτησης, ιδιαίτερα την ανάκληση (recall).
- Η αλόγιστη επιλογή σχετιζόμενων όρων μπορεί να μειώσει την ακρίβεια (precision).

Θησαυροί Όρων και Καθολική Ανάλυση

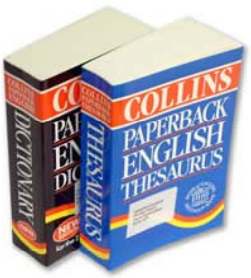
Θησαυροί Όρων

- Ένας θησαυρός παρέχει πληροφορίες για συνώνυμα και σημασιολογικά κοντινές λέξεις και φράσεις [see also Sec 7.2.5]

- Παράδειγμα:

```
physician
  syn: ||croaker, doc, doctor, MD, medical, mediciner, medico,||sawbones
  rel: medic, general practitioner, surgeon,
```

- Online-θησαυροί:
  - **Roget's** thesaurus
  - **INSPEC** thesaurus
  - **WordNet** (http://wordnet.princeton.edu/)
  - The free dictionary http://www.thefreedictionary.com/



Χρήσεις Θησαυρού

- **Ευρετηρίαση** κειμένων/βιβλίων με επιλογή όρου από θησαυρό

- **Αναζήτηση** χρησιμοποιώντας όρους του θησαυρού
  - (αυτόματη ή ύστερα από επιλογή του χρήστη)

- Για **βελτίωση** της **ανάκτησης**
  - Αν η απάντηση μιας επερώτησης είναι **μικρή**, μπορούμε να προσθέσουμε όρους βάσει των σχέσεων του θησαυρού (συνώνυμα, ..)
  - Αν απάντηση είναι **πολύ μεγάλη**, μπορούμε να συμβουλευτούμε το θησαυρό και να αντικαταστήσουμε κάποιους όρους της επερώτησης με **πιο ειδικούς**.

Διάκριση Θησαυρών

- **Γλωσσικοί** Θησαυροί
  - Designed to assist the writer in creatively selecting vocabulary
  - Παράδειγμα:
    - **Roget's** thesaurus.

- Θησαυροί κατάλληλοι για **Information Retrieval**
  - για το «συντονισμό» των διαδικασιών ευρετηρίασης και αναζήτησης
  - σχεδιάζονται για συγκεκριμένες θεματικές περιοχές (π.χ. βιβλιογραφία πληροφορικής, αρχιτεκτονικής, κλπ)
  - Παράδειγμα:
    - **INSPEC**
  - Στην κατηγορία αυτή μπορούμε να πούμε οτι εντάσσονται και οι Οντολογίες του Σημασιολογικού Ιστού (Semantic Web Ontologies)

Παράδειγμα: **INSPEC** thesaurus (for IR)

- Πεδίο: **physics, electrical engineering, electronics, computers**

- **Τύποι Συσχετίσεων** μεταξύ δύο όρων:
  - **UF:** ακρώνυμο του **Used For (converse: USE)** // π.χ. USE X σημαίνει οτι ο X είναι ο δόκιμος όρος
  - **BT:** ακρώνυμο του **Broader Term** (converse **NT)**
  - **TT:** ακρώνυμο του **Top Term** (δείχνει στον κορυφαίο όρο της ιεραρχίας)
  - **RT:** ακρώνυμο του **Related Term**
- Παράδειγμα:
  - **computer-aided instruction**
    - **see also education**
    - **UF teaching machines**
    - **BT educational computing**
    - **TT computer applications**
    - **RT education , teaching**

Παράδειγμα: **WordNet** (http://wordnet.princeton.edu/)

- A detailed database of semantic relationships between English words that was developed by the famous cognitive psychologist George Miller and a team at Princeton University.
- About 144,000 English words.Nouns, adjectives, verbs, and adverbs grouped into about 109,000 synonym sets called ***synsets***.

Synset Relationships (τύποι συσχετίσεων μεταξύ synsets)

• **Antonym**:	front → back	
• **Attribute**:	benevolence → good	(noun to adjective)
• **Pertainym**:	alphabetical → alphabet	(adjective to noun)
• **Similar**:	unquestioning → absolute	
• **Cause**:	kill → die	
• **Entailment**:	breathe → inhale	
• **Holonym**:	chapter → text	(part-of)
• **Meronym**:	computer → cpu	(whole-of)
• **Hyponym**:	tree → plant	(specialization)
• **Hypernym**:	fruit → apple	(generalization)

Παράδειγμα: **AAT** (Art and Architecture Thesaurus)

- Πεδίο: fine art, architecture, decorative art, and material culture.
- Almost 120,000 terms for objects, textual materials, images, architecture and culture from all periods and all cultures.
- Used by archives, museums, and libraries to describe items in their collections.
- Used to search for materials.
- Used by computer programs, for information retrieval, and natural language processing.

Χαρακτηριστικά Θησαυρών

- Coordination Level (βαθμός συντονισμού)
  - refers to the construction of phrases from individual terms
  - ***Pre-coordination***: the thesaurus **contains phrases**
    - + the vocabulary is very precise
    - - the user has to be aware of the phrase construction rules, large size
  - ***Post-coordination***: the thesaurus **does not contain phrases**. They are constructed while indexing/searching
    - + user does not worry about the order of the words
    - - precision may fall
- Term Relationships
  - **equivalence** relations (e.g. synonymy)
  - **hierarchical** relations (e.g. dogs BT animals,)
  - **nonhierarchical** relations (e.g. RT)

Χαρακτηριστικά Θησαυρών (2)

- Number of Entries per Term
  - preferably: a single entry for each thesaurus term
  - however homonyms does not make this possible
    - solution: usage of **parenthetical qualifiers**:
      - bonds(chemical), bonds(adhesive) // χημικός δεσμός / υλικό συγκόλλησης
- Specificity of Vocabulary
  - high specificity -> large vocabulary size
- Control of Term Frequency of Class Members (for statistical thesauri)
  - the terms of a thesaurus should have roughly equal frequencies
  - the total frequency in each class (of terms) should be equal
- Normalization of Vocabulary
  - terms should be in noun form
  - other rules related to singularity of terms, spelling, capitalization, abbreviations, initials, acronyms, punctuation

Τρόποι Κατασκευής Θησαυρών

[A] Χειροποίητη Δημιουργία

[B] Αυτόματη Κατασκευή

[B.1] από **συλλογή κειμένων**

Προϋπόθεση: Να υπάρχει μια μεγάλη και αντιπροσωπευτική συλλογή κειμένων

[B.2] από **συγχώνευση** άλλων θησαυρών

Προϋπόθεση: Να υπάρχουν >2 διαθέσιμοι θησαυροί για την περιοχή που μας ενδιαφέρει

[A] Χειροποίητοι Θησαυροί (διαδικασία κατασκευής)

Διαδικασία κατασκευής

1. Προσδιορισμός εμβέλειας (define subject boundaries)
2. Διαμέριση σε θεματικές ενότητες (partition into divisions and subject areas)
3. Συλλογή Όρων (collection of terms)
   - Πηγές: encyclopedias, handbooks, textbooks, journal titles, catalogues, other thesauri, subject experts, potential users
4. Ανάλυση Όρων
   - συνώνυμα, ιεραρχική δόμηση όρων, ορισμοί όρων, περιγραφή εφαρμοσιμότητας όρων (scope notes)
5. Αξιολόγηση και Αναθεώρηση (reviewing phase)
6. Παράδοση (εκτύπωση) του Θησαυρού σε ιεραρχική και αλφαβητική οργάνωση
7. Συντήρηση Θησαυρού (προσθήκη νέων όρων, αλλαγές στη δομή, κλπ)

→ ***Πολύ χρονοβόρα, κοπιαστική και ακριβή διαδικασία***

Επέκταση επερωτήσεων βάσει Θησαυρού
Thesaurus-based Query Expansion

- Τρόπος:
  - Για κάθε όρο t της επερώτησης, πρόσθεσε στην επερώτηση τα συνώνυμα και τις σχετικές λέξεις (related terms) του t
  - Τα βάρη των νέων λέξεων μπορεί να είναι χαμηλότερα των βαρών των λέξεων της αρχικής επερώτησης
  - E.g. of a WordNet-based Query Expansion
    - Add synonyms in the same synset.
    - Add hyponyms to add specialized terms.
    - Add hypernyms to generalize a query.
    - Add other related terms to expand query.
- Αποτέλεσμα
  - Αυξάνει την ανάκληση (recall.)
  - Μπορεί να μειώσει την ακρίβεια (precision), ιδιαίτερα όταν η επερώτηση περιέχει αμφίσημες λέξεις. Παράδειγμα μη επιτυχούς επέκτασης:
    - "interest rate" → "interest rate fascinate evaluate"

[B1] Αυτόματη Κατασκευή Θησαυρών από Κείμενα

- Η κατασκευή (από ανθρώπους) ενός θησαυρού είναι πολύ **χρονοβόρα** και δεν υπάρχουν θησαυροί για όλες τις γλώσσες
- Οι πληροφορίες που μπορούμε να χρησιμοποιήσουμε από έναν θησαυρό περιορίζονται στις σχέσεις που είναι ρητώς καταγεγραμμένες στον θησαυρό
- **Ιδέα: Μπορούμε να ανακαλύψουμε σημασιολογικές σχέσεις μεταξύ λέξεων αναλύοντας στατιστικά μια μεγάλη συλλογή κειμένων**
- Στάδια ενός αυτόματου τρόπου κατασκευής θησαυρών
  - ***1/ Κατασκευή λεξιλογίου***
  - ***2/ Υπολογισμός ομοιότητας μεταξύ όρων***
  - ***3/ Οργάνωση (συνήθως ιεραρχική) του λεξιλογίου***

Αυτόματη Κατασκευή Θησαυρών από Κείμενα (II)

1/ Κατασκευή Λεξιλογίου

- Απόφαση: Ποιος θέλουμε να είναι ο βαθμός εξιδίκευσης (desired specificity)
  - if high then emphasis will be given on identifying precise phrases
- Οι όροι (terms) μπορούν να επιλεγούν από τους *τίτλους,* τις *περιλήψεις (*abstracs), ή ακόμα και από το *πλήρες κείμενο (*full text)
- Normalization: stemming, stoplists
- Criteria for selecting a term:
  - frequency of occurrence (**divide words to 3 categories: low, medium, high, select terms with medium frequency**)
  - discrimination value ~ idf
- Κατασκευή φράσεων (phrase construction) αν κάτι τέτοιο είναι επιθυμητό (θυμηθείτε coordination level)

2/ Υπολογισμός Ομοιότητας μεταξύ όρων

- Παραδείγματα μετρικών: Cosine, Dice

Αυτόματη Κατασκευή Θησαυρών από Κείμενα (III)

3/ Οργάνωση (συνήθως ιεραρχική) του λεξιλογίου

- Οποιοσδήποτε αλγόριθμος clustering μπορεί να χρησιμοποιηθεί για αυτό το βήμα

Ένας αλγόριθμος για ιεραρχική οργάνωση ενός λεξιλογίου:

1/ Identify a set of frequency ranges

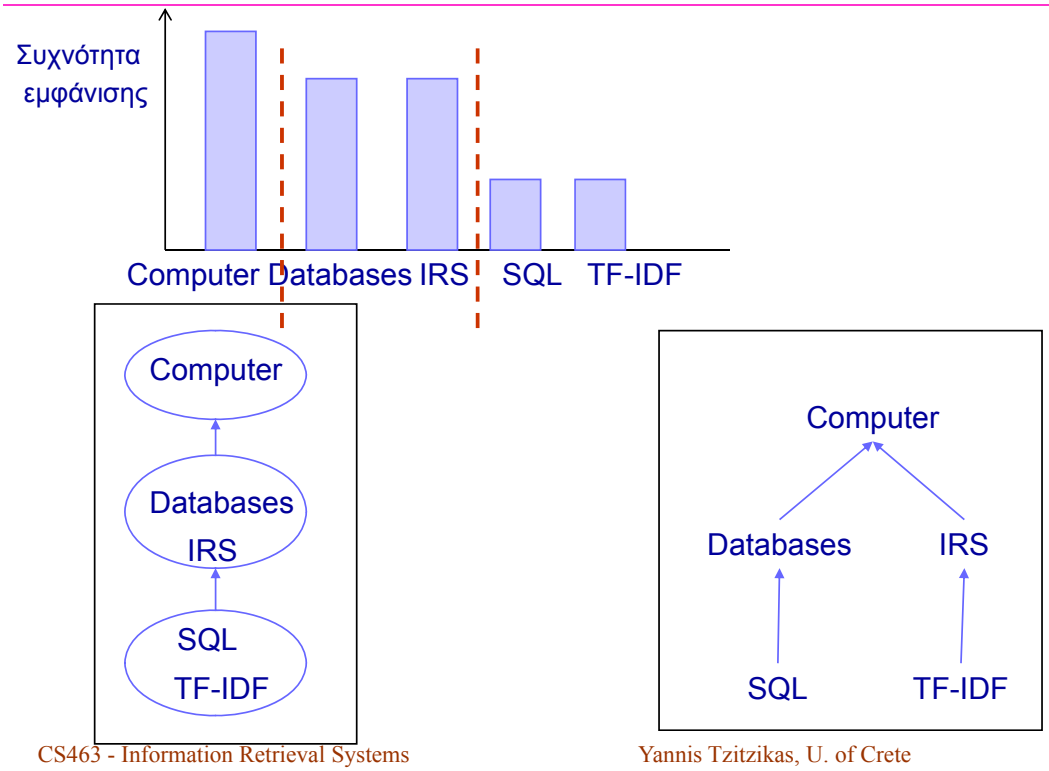
2/ Group the vocabulary terms into different classes based on their frequencies and the ranges selected in Step 1. There will be one term class for each frequency range

3/ The highest frequency class is assigned level 0, the next level 1, and so on

4/ Parent-child links: The parent(s) of a term at level i is the most similar term in level i-1 (a term is allowed to have multiple parents)

5/ Continue until reaching level 1

Παράδειγμα με 3 κλάσεις συχνοτήτων



Case: grOOGLE'2007

(1) Compute the minimum and maximum frequency of the words in the lexicon (denoted by df_{mn} and df_{mx} respectively).
(2) Partition the interval $[df_{mn}, df_{mx}]$ into L successive intervals (where L is administrator-provided), i.e. $[df_{mn}, df_1], \ldots, [df_{L-1}, df_{mx}]$. We will refer to them with $lev_1, \ldots, lev_L$ respectively.
(3) Ignore the intervals corresponding to low frequencies, specifically keep only the M intervals with the highest frequencies (M is administrator-provided and it should be $M < L$), i.e. keep only $lev_{L-M-1}, \ldots, lev_L$.
(4) Assign to each of these M intervals those words whose frequency falls to that interval.
(5) For each word w_i of level z (where $z \leq L-1$) connect it with the most "correlated" word of the level $z+1$ (that word will be the "parent" of w_i).

Regarding step (5), the correlation c_{ij} between two words w_i and w_j is computed using the formula:

$$c_{ij} = \sum_{d_k \in D} tf_{ik} \times tf_{jk} \quad (1)$$

where tf_{ik} is the frequency of term i in document k.

(cont)

As an example, Table 11 describes the partitioning obtained assuming $L = 20$ (for each level the table shows the number of words that belong to that level). To construct the taxonomy we have considered only the last 5 groups (empty groups, like level 19, are considered as non existant). So the taxonomy includes 35 words in total. After creating the connections between words we realized that each word has an average of 1.4 child nodes.

	Low frequency															High frequency				
Level	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Num. Of Words	217142	1103	523	292	199	128	83	83	53	52	25	18	18	14	14	7	8	2	0	4

The reason for partitioning words into groups (according to their frequency) is for avoiding computing the correlation matrix between all pairs of words (which would be formidably expensive[7]). In addition, ignoring those words that occur rarely further improves efficiency (as more than 95% of the vocabulary has a very small document frequency) and does not harm the quality of the result as these words do not describe the main concepts of the document corpus, and we have not anyway adequate statistical information to connect them right in a hierarchy.

(cont)

Resulting taxonomy:

<1> http
<2> system
<3> us
<3.1> new
<3.1.1> url
<3.1.1.1> map
<3.1.1.2> network
<3.1.1.2.1> commun
<3.1.1.2.2> gener
<3.1.1.2.3> site
<3.1.1.2.4> scienc
<3.1.1.2.5> web
<3.1.1.3> page
<3.1.1.3.1> link
<3.1.1.3.2> applic
<3.1.1.4> document
<3.1.1.5> access
<3.1.1.6> univers
<3.1.1.7> data
<3.1.1.8> process
<3.1.2> time
<3.1.3> base
<4> inform

As you can see this taxonomy is not very good/useful

Possible improvements:

- Better vocabulary construction
  - The terms with high frequency are not very informative as you can see (e.g.. http, system, url, ...). Therefore we should try the **middle** levels.
  - Furthermore if we had selected words that appear only in titles/abstracts then we would avoid words like: http, url, ..
  - The user at run-time could even specify how specific/general the taxonomy should be (his/her choice would determine the visible part of the taxonomy)
- Other improvements
  - It's better to show the original words (rather than stems)
  - Use phrases instead of single words as terms

	Low frequency															High frequency				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Of Words	217142	1103	523	292	199	128	83	83	53	52	25	18	18	14	14	7	8	2	0	4

Αυτόματη Κατασκευή Ιεραρχιών

[M. Sanderson and W. B. Croft. Deriving concept hierarchies from text. In SIGIR'1999]

- Given two terms x and y from a document collection, we say that x *subsumes* y, and we write $x \rightarrow y$ if: **$P(x|y) > 0.8$ and $P(y|x) < 1$**

  P(x|y) is the probability that term x occurs in a document, given that term y does

- This technique leads to creation of a hierarchy of terms, where
  - General terms appear as top-level categories
  - More specific terms appear as lower-level categories
- Pros
  - Simple and effective
- Cons
  - Requires n^2 computations of conditional probabilities, where n is the number of terms in the collection
  - Requires the terms to have a unique meaning
    - However If we use this technique only on query results and by using only terms that appear more frequently in the query results than in the whole collection. then this lessens the problem of ambiguity and reduces the number of terms that form the subsumption hierarchy.

Αυτόματη Κατασκευή Πολυεδρικών Ιεραρχιών

[Automatic Construction of Multifaceted Browsing Interfaces,W. Dakka, P. Ipeirotis, K. Wood, CICK'05]

- It describes an approach for constructing multifaceted hierarchies.
- Includes methods for selecting the best parts of the generated hierarchies when it is not possible to fit all the categories on screen
- Experiments with real-life data sets indicate that automatic construction of multifaceted interfaces is feasible, and generates high-quality hierarchies

A Data Mining approach (to organize the set of terms hierarchically)

- Let I={i_1,…i_m} be a set of items
- Let D be a set of transactions where each transaction is a subset of I
- An association rule is an implication of the form X→Y where X, Y are subsets of I and X ∩Y=∅
- A rule X→Y holds in the transaction set D with
  - confidence c if c% of the transactions in D that contain X also contain Y
  - support s if s% of the transactions in D contain X ∪Y

Consider the case of an IR system. In that case

- The set I could be the set of all terms (the vocabulary)
- The set D could be the set of binary vectors of the documents
- A rule X→Y would be an implication between set of terms
  - If |X|=|Y|=1 then the implications are between single terms
  - If |X|=|Y|=2 then the implications are between pairs of terms
- So we could exploit **data mining** algorithms to get a taxonomy from an IR system