Multimedia Content Management

Evaluation and Query Expansion

Ralf Moeller
Hamburg Univ. of Technology

Acknowledgements

- Slides taken from:
  - Introduction to Information Retrieval
    Christopher Manning and Prabhakar Raghavan

This lecture

- Results summaries:
  - Making our good results usable to a user
- How do we know if our results are any good?
  - Evaluating a search engine
    - Benchmarks, Precision and recall
- Query Reformulation/Expansion

Results summaries

Summaries

- Having ranked the documents matching a query, we wish to present a results list
- Most commonly, the document title plus a short summary
- The title is typically automatically extracted from document metadata
- What about the summaries?

Summaries

- A **static summary** of a document is always the same, regardless of the query that hit the doc
- **Dynamic summaries** are *query-dependent* attempt to explain why the document was retrieved for the query at hand

Static summaries

- In typical systems, the static summary is a subset of the document
- Simplest heuristic: the first 50 (or so – this can be varied) words of the document
  - Summary cached at indexing time
- More sophisticated: extract from each document a set of “key” sentences
  - Simple NLP heuristics to score each sentence
  - Summary is made up of top-scoring sentences.
- Most sophisticated: NLP used to synthesize a summary
  - Seldom used in IR (hard to automatize)

Dynamic summaries

- Present one or more “windows” within the document that contain several of the query terms
  - “KWIC” snippets: Keyword in Context presentation
- Generated in conjunction with scoring
  - If query found as a phrase, the/some occurrences of the phrase in the doc
  - If not, windows within the doc that contain multiple query terms
- The summary itself gives the entire content of the window – all terms, not only the query terms

Generating dynamic summaries

- If we have only a positional index, we cannot (easily) reconstruct context surrounding hits
- If we *cache the documents* at index time, can run the window through it, cueing to hits found in the positional index
  - E.g., positional index says "the query is a phrase in position 4378" so we go to this position in the cached document and stream out the content
- Most often, cache a fixed-size prefix of the doc
  - Note: Cached copy can be outdated

Dynamic summaries

- Producing good dynamic summaries is a tricky optimization problem
  - The real estate for the summary is normally small and fixed
  - Want short item, so show as many KWIC matches as possible, and perhaps other things like title
  - Want snippets to be long enough to be useful
  - Want linguistically well-formed snippets: users prefer snippets that contain complete phrases
  - Want snippets maximally informative about doc
- But users really like snippets, even if they complicate IR system design

Evaluating search engines

Task:

Which measures can you think of?

Measures for a search engine

- How fast does it index
  - Number of documents/hour
  - (Average document size)
- How fast does it search
  - Latency as a function of index size
- Expressiveness of query language
  - Ability to express complex information needs
  - Speed on complex queries

Measures for a search engine

- All of the preceding criteria are *measurable*: we can quantify speed/size; we can make expressiveness precise
- The key measure: user happiness
  - What is this?
  - Speed of response/size of index are factors
  - But blindingly fast, useless answers won't make a user happy
- Need a way of quantifying user happiness

Measuring user happiness

- Issue: who is the user we are trying to make happy?
  - Depends on the setting
- <u>Web engine</u>: user finds what they want and return to the engine
  - Can measure rate of return users
- <u>eCommerce site</u>: user finds what they want and make a purchase
  - Is it the end-user, or the eCommerce site, whose happiness we measure?
  - Measure time to purchase, or fraction of searchers who become buyers?

Measuring user happiness

- Enterprise (company/govt/academic): Care about "user productivity"
  - How much time do my users save when looking for information?
  - Many other criteria having to do with breadth of access, secure access, etc.
- To sum up: this is really hard!

Evaluating an IR system

- Note: the **information need** is translated into a **query**
- Relevance is assessed relative to the **information need** *not* the **query**
- E.g., Information need: *I'm looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.*
- Query: ***wine red white heart attack effective***
- You evaluate whether the doc addresses the information need, not whether it has those words

Standard relevance benchmarks

- TREC - National Institute of Standards and Testing (NIST) has run a large IR test bed for many years
- Reuters and other benchmark doc collections used
- “Retrieval tasks” specified
  - sometimes as queries
- Human experts mark, for each query and for each doc, <u>Relevant</u> or <u>Irrelevant</u>
  - or at least for subset of docs that some system returned for that query

Unranked retrieval evaluation: Precision and Recall

- **Precision**: fraction of retrieved docs that are relevant = P(relevant|retrieved)
- **Recall**: fraction of relevant docs that are retrieved = P(retrieved|relevant)

	Relevant	Not Relevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision P = tp/(tp + fp)
- Recall R = tp/(tp + fn)

Accuracy

- Given a query an engine classifies each doc as “Relevant” or “Irrelevant”.
- Accuracy of an engine: the fraction of these classifications that is correct.
- Why is this not a very useful evaluation measure in IR?

Why not just use accuracy?

- How to build a 99.9999% accurate search engine on a low budget….

snoogle.com

Search for:

0 matching results found.

- People doing information retrieval *want to find something* and have a certain tolerance for junk.

Precision/Recall

- You can get high recall (but low precision) by retrieving all docs for all queries!
- Recall is a non-decreasing function of the number of docs retrieved
- In a good system, precision decreases as either number of docs retrieved or recall increases
  - A fact with strong empirical confirmation

Difficulties in using precision/recall

- Should average over large corpus/query ensembles
- Need human relevance assessments
  - People aren’t reliable assessors
- Assessments have to be binary
  - Nuanced assessments?

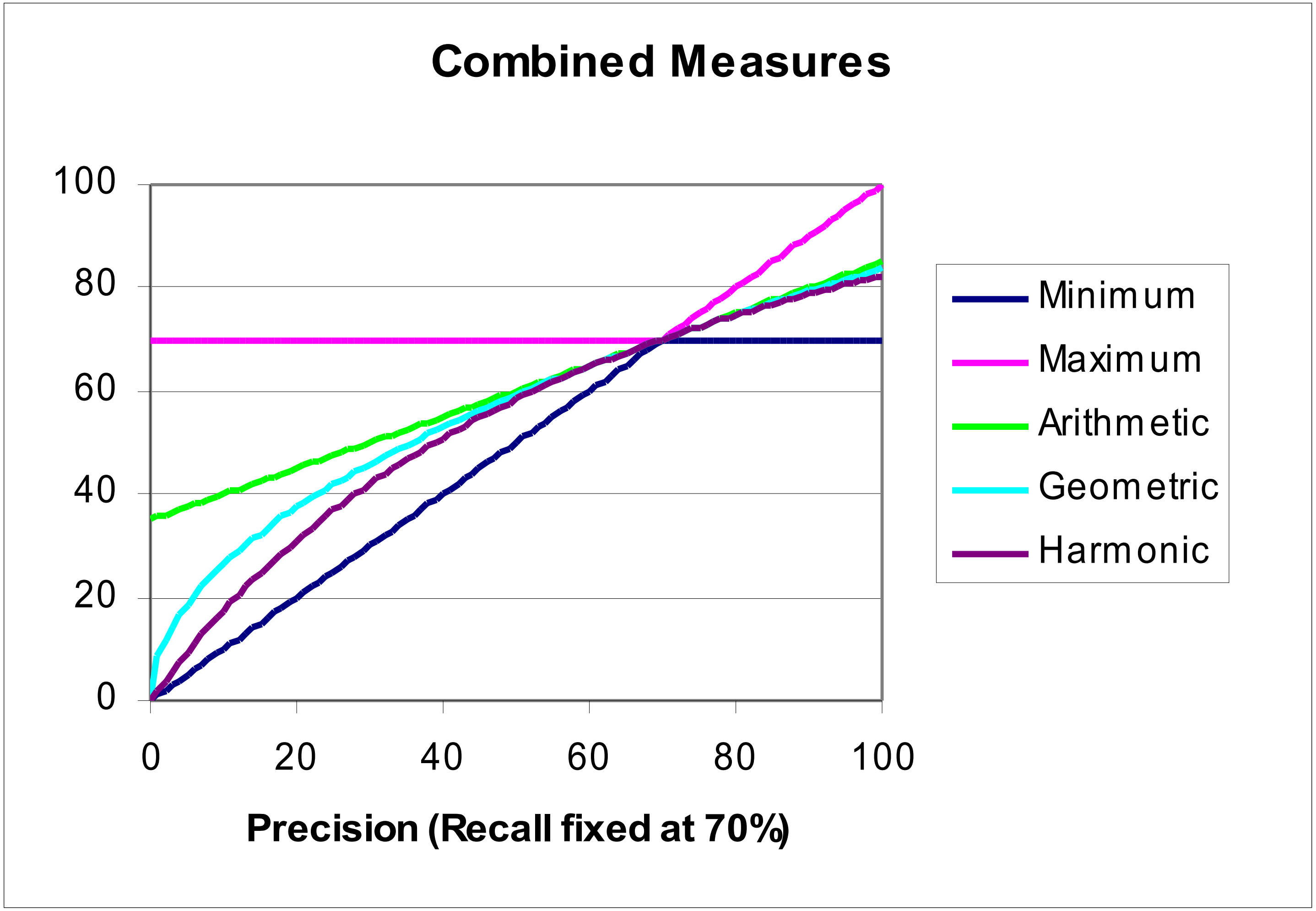
A combined measure: *F*

- Combined measure that assesses this tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha)\frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

- People usually use balanced F_1 measure
  - i.e., with $\beta = 1$ or $\alpha = \frac{1}{2}$
- Harmonic mean is a conservative average
  - See CJ van Rijsbergen, *Information Retrieval*

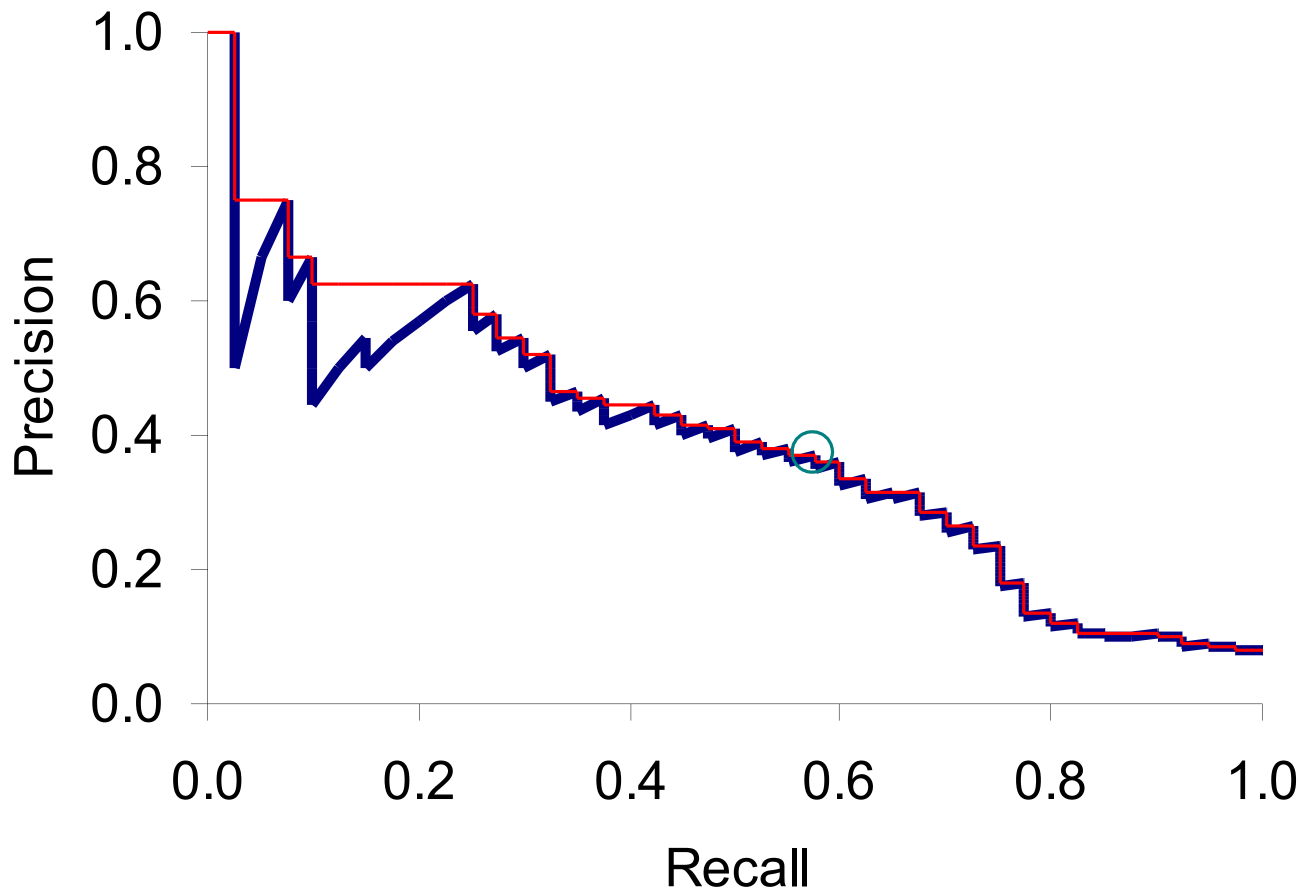
F_1 and other averages



Evaluating ranked results

- Evaluation of ranked results:
  - The system can return any number of results
  - By taking various numbers of the top returned documents (levels of recall), the evaluator can produce a *precision-recall curve*

A precision-recall curve



Averaging over queries

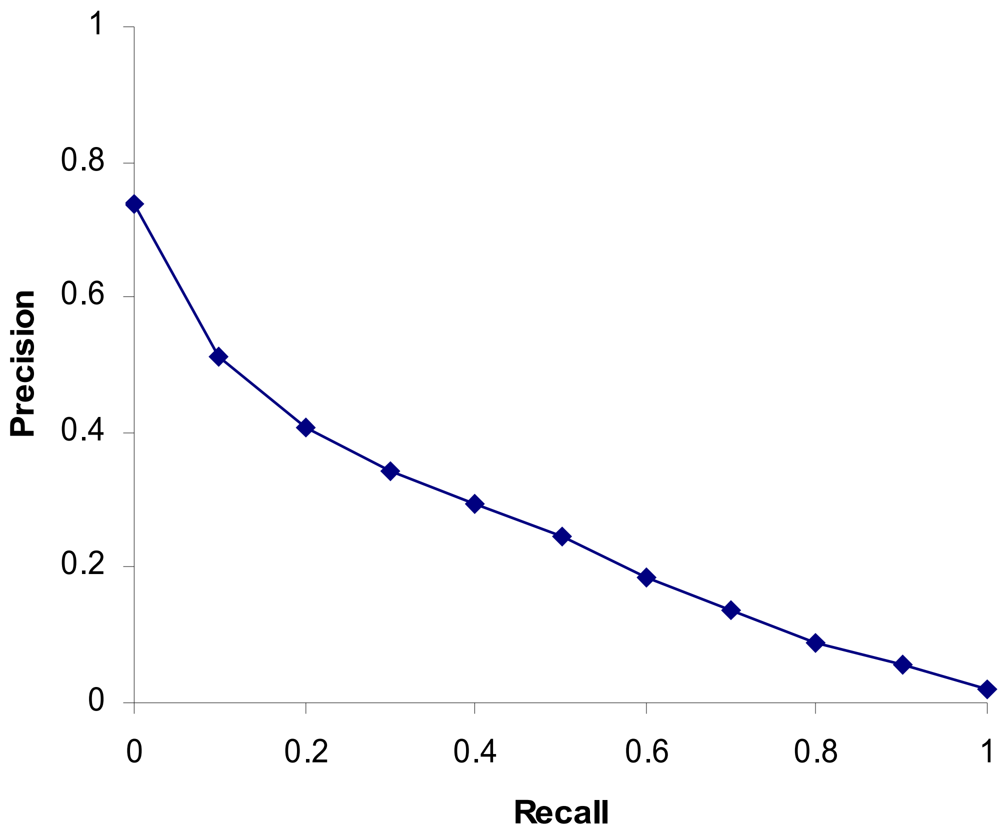
- A precision-recall graph for one query isn't a very sensible thing to look at
- You need to average performance over a whole bunch of queries.
- But there's a technical issue:
  - Precision-recall calculations place some points on the graph
  - How do you determine a value (interpolate) between the points?

Evaluation

- Graphs are good, but people want summary measures!
  - Precision at fixed retrieval level
    - Perhaps most appropriate for web search: all people want are good matches on the first one or two results pages
    - But has an arbitrary parameter of *k*
  - 11-point interpolated average precision
    - The standard measure in the TREC competitions: you take the precision at 11 levels of recall varying from 0 to 1 by tenths of the documents, using interpolation (the value for 0 is always interpolated!), and average them
    - Evaluates performance at all recall levels

Typical (good) 11 point precisions

- SabIR/Cornell 8A1 11pt precision from TREC 8 (1999)



Creating Test Collections for IR Evaluation

Test Corpora

TABLE 4.3 Common Test Corpora

Collection	*NDocs*	*NQrys*	*Size (MB)*	*Term/Doc*	*Q-D RelAss*
ADI	82	35			
AIT	2109	14	2	400	>10,000
CACM	3204	64	2	24.5	
CISI	1460	112	2	46.5	
Cranfield	1400	225	2	53.1	
LISA	5872	35	3		
Medline	1033	30	1		
NPL	11,429	93	3		
OSHMED	34,8566	106	400	250	16,140
Reuters	21,578	672	28	131	
TREC	740,000	200	2000	89-3543	» 100,000

From corpora to test collections

- Still need
  - Test queries
  - Relevance assessments
- Test queries
  - Must be germane to docs available
  - Best designed by domain experts
  - Random query terms generally not a good idea
- Relevance assessments
  - Human judges, time-consuming
  - Are human panels perfect?

Unit of Evaluation

- We can compute precision, recall, F, and ROC curve for different units.
- Possible units
  - Documents (most common)
  - Facts (used in some TREC evaluations)
  - Entities (e.g., car companies)
- May produce different results. Why?

Kappa measure for inter-judge (dis)agreement

- Kappa measure
  - Agreement measure among judges
  - Designed for categorical judgments
  - Corrects for chance agreement
- Kappa = [P(A) – P(E)] / [1 – P(E)]
- P(A) – proportion of time judges agree
- P(E) – what agreement would be by chance
- Kappa = 0 for chance agreement, 1 for total agreement.

Kappa Measure: Example

P(A)? P(E)?

Number of docs	Judge 1	Judge 2
300	Relevant	Relevant
70	Nonrelevant	Nonrelevant
20	Relevant	Nonrelevant
10	Nonrelevant	relevant

Kappa Example

- P(A) = 370/400 = 0.925
- P(nonrelevant) = (10+20+70+70)/800 = 0.2125
- P(relevant) = (10+20+300+300)/800 = 0.7878
- P(E) = 0.2125^2 + 0.7878^2 = 0.665
- Kappa = (0.925 – 0.665)/(1-0.665) = 0.776

- Kappa > 0.8 = good agreement
- 0.67 < Kappa < 0.8 -> "tentative conclusions" (Carletta '96)
- Depends on purpose of study
- For >2 judges: average pairwise kappas

Can we avoid human judgment?

- Not really
- Makes experimental work hard
  - Especially on a large scale
- In some very specific settings, can use proxies
- Example below, approximate vector space retrieval
- But once we have test collections, we can reuse them (so long as we don't overtrain too badly)

Approximate vector retrieval

- Given *n* document vectors and a query, find the *k* doc vectors closest to the query.
  - Exact retrieval – we know of no better way than to compute cosines from the query to every doc
  - Approximate retrieval schemes
- Given such an approximate retrieval scheme, how do we measure its goodness?

Approximate vector retrieval

- Let $G(q)$ be the "ground truth" of the actual k closest docs on query q
- Let $A(q)$ be the k docs returned by approximate algorithm A on query q
- For performance we would measure $A(q) \cap G(q)$
  - Is this the right measure?

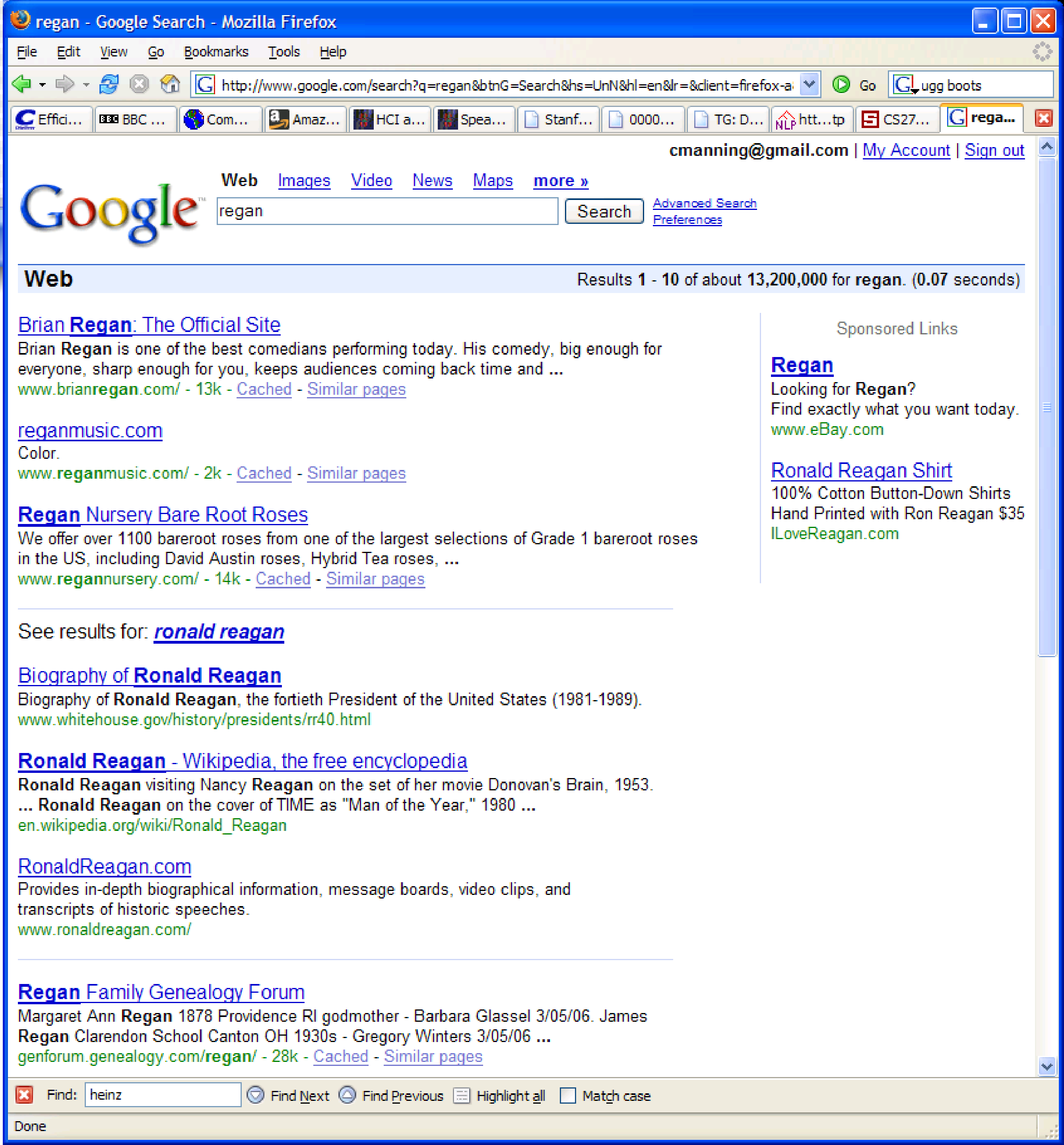
Alternative proposal

- Focus instead on how $A(q)$ compares to $G(q)$.
- Goodness can be measured here in cosine proximity to q: we sum up $q \bullet d$ over $d \in A(q)$.
- Compare this to the sum of $q \bullet d$ over $d \in G(q)$.
  - Yields a measure of the relative "goodness" of A vis-à-vis G.

What now?

- Improving results
  - For high recall. E.g., searching for *aircraft* doesn't match with *plane;* nor *thermodynamic* with *heat*
- Options for improving results…
  - Focus on relevance feedback
  - The complete landscape
    - Global methods
      - Query expansion
        - Thesauri
        - Automatic thesaurus generation
    - Local methods
      - Relevance feedback
      - Pseudo relevance feedback

Query expansion

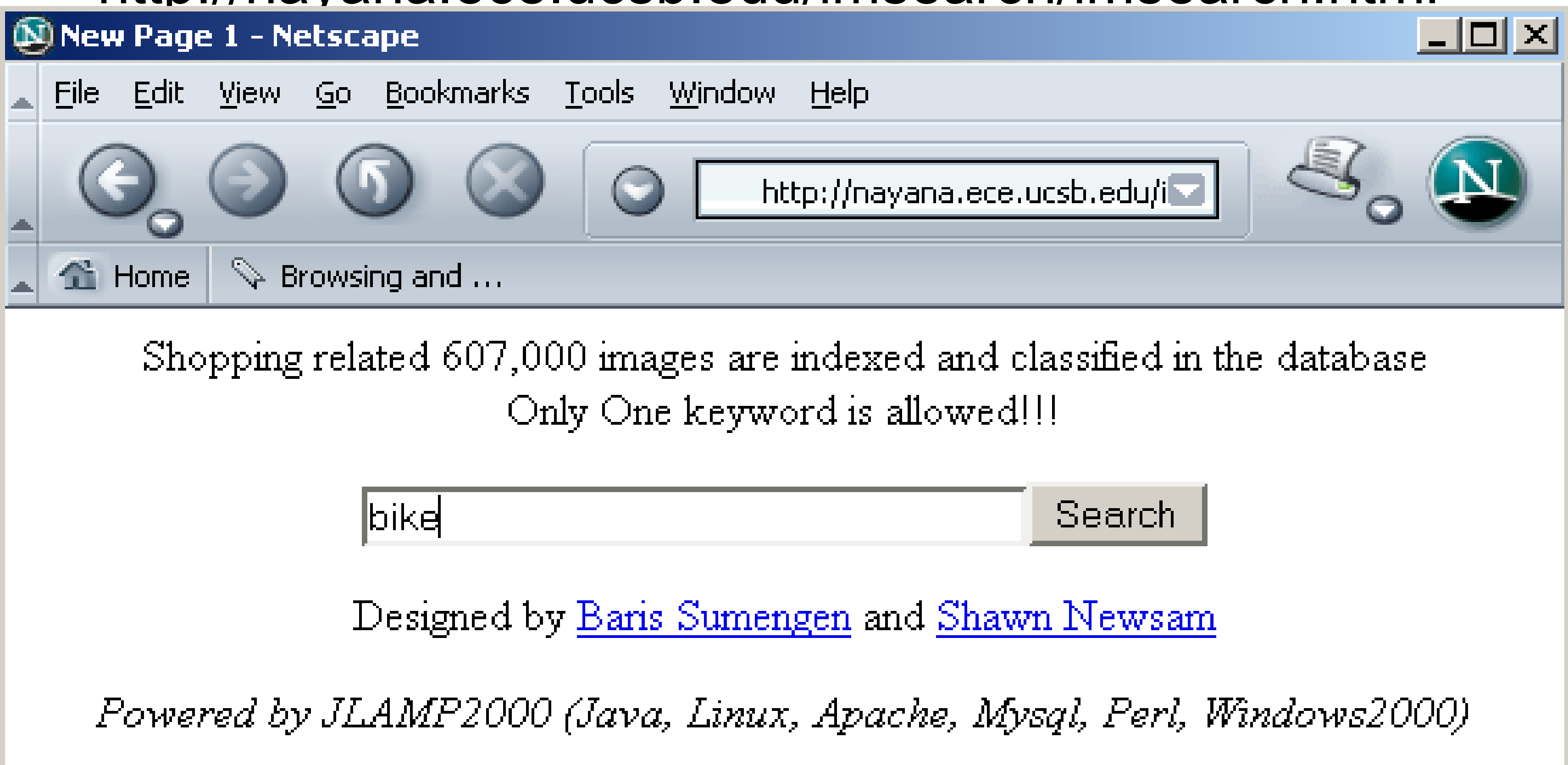


Relevance Feedback

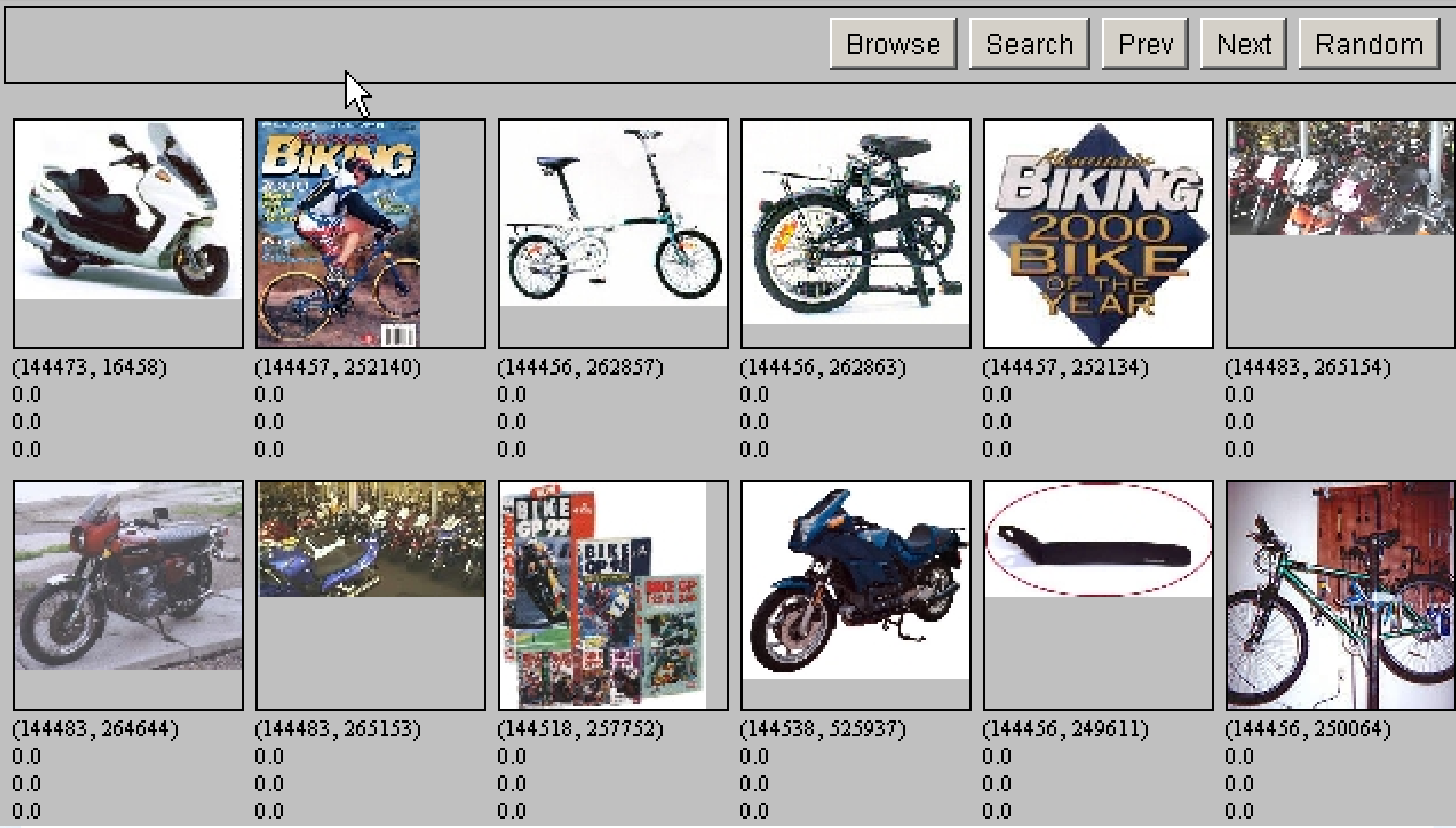
- Relevance feedback: user feedback on relevance of docs in initial set of results
  - User issues a (short, simple) query
  - The user marks returned documents as relevant or non-relevant.
  - The system computes a better representation of the information need based on feedback.
  - Relevance feedback can go through one or more iterations.
- Idea: it may be difficult to formulate a good query when you don't know the collection well, so iterate

Relevance Feedback: Example

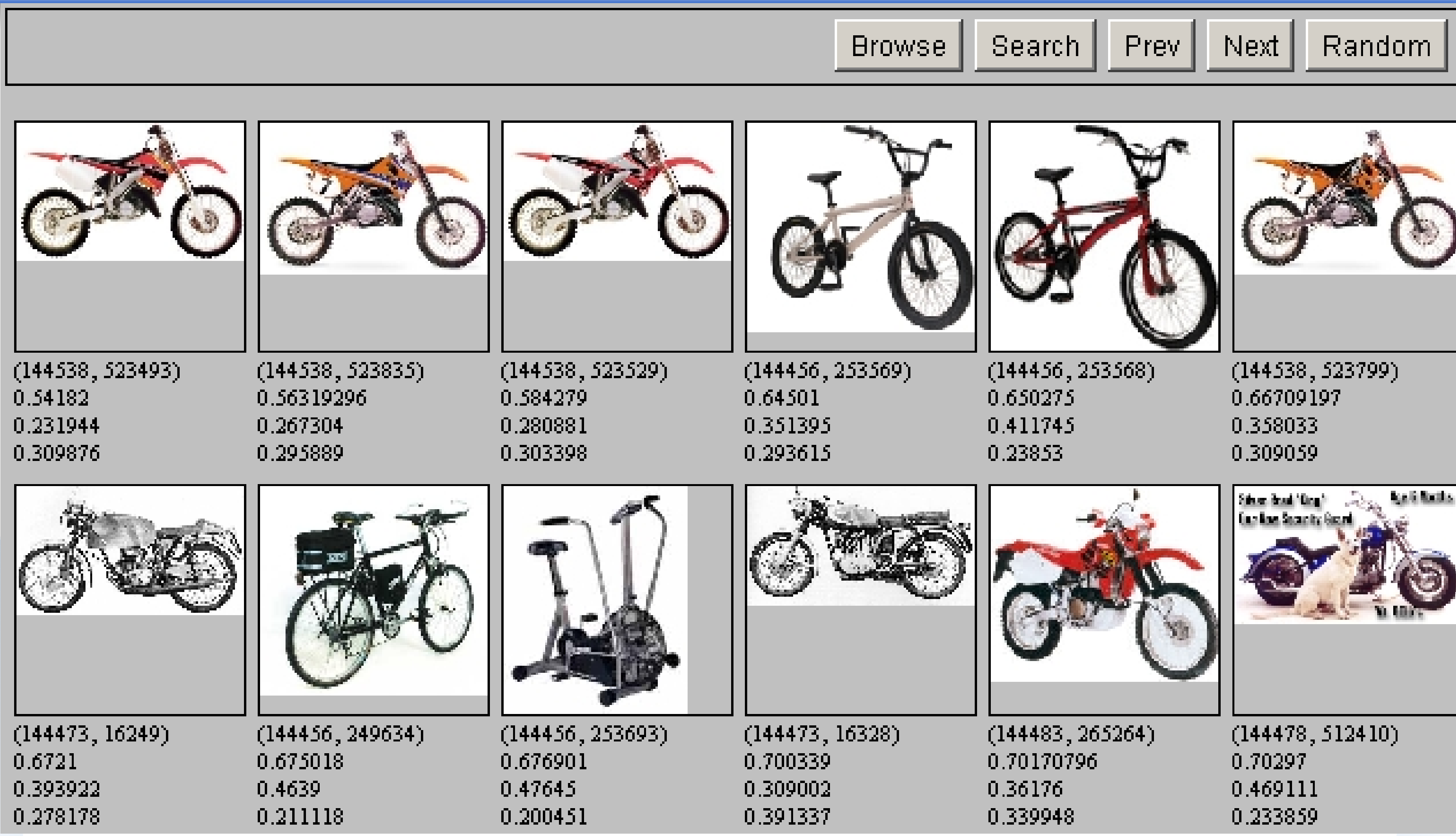
- Image search engine
  http://nayana.ece.ucsb.edu/imsearch/imsearch.html



Results for Initial Query



Results after Relevance Feedback



Rocchio Algorithm

- The Rocchio algorithm incorporates relevance feedback information into the vector space model.
- Want to maximize $sim(Q, C_r) - sim(Q, C_{nr})$
- The optimal query vector for separating relevant and non-relevant documents (with cosine sim.):

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

- Q_{opt} = optimal query; C_r = set of rel. doc vectors; N = collection size
- Unrealistic: we don't know relevant documents.

The Theoretically Best Query

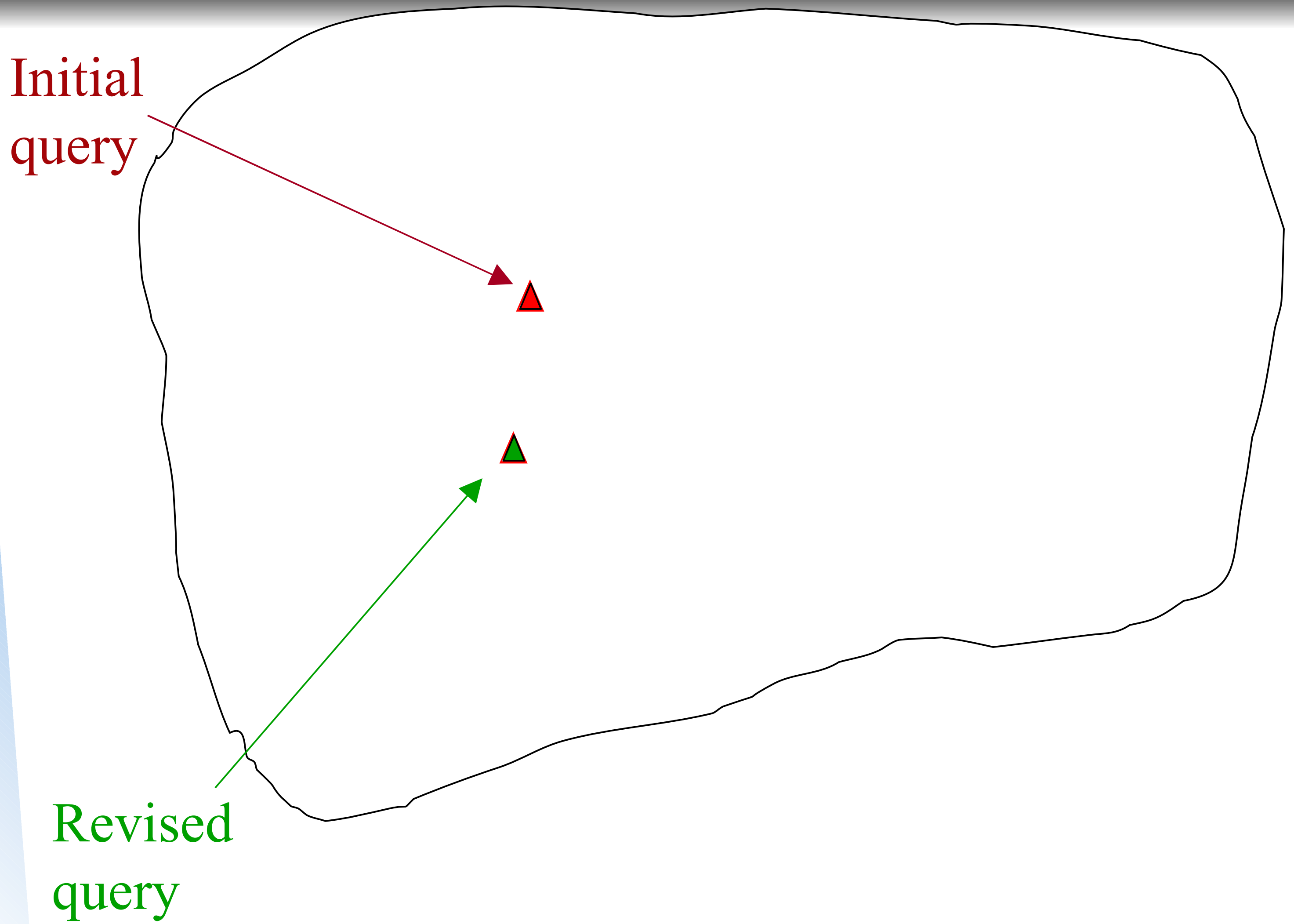
Rocchio 1971 Algorithm (SMART)

- Used in practice:

$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

- q_m = modified query vector; q_0 = original query vector; α,β,γ: weights (hand-chosen or set empirically); D_r = set of known relevant doc vectors; D_{nr} = set of known irrelevant doc vectors
- New query moves toward relevant documents and away from irrelevant documents
- Tradeoff α vs. β/γ : If we have a lot of judged documents, we want a higher β/γ.
- Term weight can go negative
  - Negative term weights are ignored (set to 0)

Relevance feedback on initial query

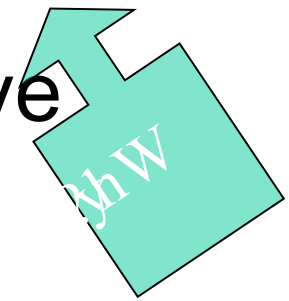


Relevance Feedback in vector spaces

- We can modify the query based on relevance feedback and apply standard vector space model.
- Use only the docs that were marked.
- Relevance feedback can improve recall and precision
- Relevance feedback is most useful for increasing *recall* in situations where recall is important
  - Users can be expected to review results and to take time to iterate

Positive vs Negative Feedback

- Positive feedback is more valuable than negative feedback (so, set $\gamma < \beta$; e.g. $\gamma = 0.25$, $\beta = 0.75$).
- Many systems only allow positive feedback (γ=0).



High-dimensional Vector Spaces

- The queries "cholera" and "john snow" are far from each other in vector space.
- How can the document "John Snow and Cholera" be close to both of them?
- Our intuitions for 2- and 3-dimensional space don't work in >10,000 dimensions.
- 3 dimensions: If a document is close to many queries, then some of these queries must be close to each other.
- Doesn't hold for a high-dimensional space.

Relevance Feedback: Assumptions

- A1: User has sufficient knowledge for initial query.
- A2: Relevance prototypes are “well-behaved”.
  - Term distribution in relevant documents will be similar
  - Term distribution in non-relevant documents will be different from those in relevant documents
    - Either: All relevant documents are tightly clustered around a single prototype.
    - Or: There are different prototypes, but they have significant vocabulary overlap.
    - Similarities between relevant and irrelevant documents are small

Violation of A1

- User does not have sufficient initial knowledge.
- Examples:
  - Misspellings (Brittany Speers).
  - Cross-language information retrieval (hígado).
  - Mismatch of searcher’s vocabulary vs. collection vocabulary
    - Cosmonaut/astronaut

Violation of A2

- There are several relevance prototypes.
- Examples:
  - Burma/Myanmar
  - Contradictory government policies
  - Pop stars that worked at Burger King
- Often: instances of a general concept
- Good editorial content can address problem
  - Report on contradictory government policies

Relevance Feedback: Problems

- Why do most search engines not use relevance feedback?

Relevance Feedback: Problems

- Long queries are inefficient for typical IR engine.
  - Long response times for user.
  - High cost for retrieval system.
  - Partial solution:
    - Only reweight certain prominent terms
      - Perhaps top 20 by term frequency
- Users are often reluctant to provide explicit feedback
- It’s often harder to understand why a particular document was retrieved after apply relevance feedback

Why?

Relevance Feedback Example: Initial Query and Top 8 Results

Note: want high recall

- Query: New space satellite applications

- + 1. 0.539, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
- + 2. 0.533, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
- 3. 0.528, 04/04/90, Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes
- 4. 0.526, 09/09/91, A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget
- 5. 0.525, 07/24/90, Scientist Who Exposed Global Warming Proposes Satellites for Climate Research
- 6. 0.524, 08/22/90, Report Provides Support for the Critics Of Using Big Satellites to Study Climate
- 7. 0.516, 04/13/87, Arianespace Receives Satellite Launch Pact From Telesat Canada
- + 8. 0.509, 12/02/87, Telecommunications Tale of Two Companies

Relevance Feedback Example: Expanded Query

- 2.074 new 15.106 space
- 30.816 satellite 5.660 application
- 5.991 nasa 5.196 eos
- 4.196 launch 3.972 aster
- 3.516 instrument 3.446 arianespace
- 3.004 bundespost 2.806 ss
- 2.790 rocket 2.053 scientist
- 2.003 broadcast 1.172 earth
- 0.836 oil 0.646 measure

Top 8 Results After Relevance Feedback

- \+ 1. 0.513, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
- \+ 2. 0.500, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
- 3. 0.493, 08/07/89, When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
- 4. 0.493, 07/31/89, NASA Uses 'Warm‘ Superconductors For Fast Circuit
- \+ 5. 0.492, 12/02/87, Telecommunications Tale of Two Companies
- 6. 0.491, 07/09/91, Soviets May Adapt Parts of SS-20 Missile For Commercial Use
- 7. 0.490, 07/12/88, Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers
- 8. 0.490, 06/14/90, Rescue of Satellite By Space Agency To Cost $90 Million

Relevance Feedback on the Web

[in 2003: now less major search engines, but same general story]

- Some search engines offer a similar/related pages feature (this is a trivial form of relevance feedback)
  - Google (link-based)
  - Altavista
  - Stanford WebBase

$\alpha/\beta/\gamma$??

- But some don't because it's hard to explain to average user:
  - Alltheweb
  - msn
  - Yahoo
- Excite initially had true relevance feedback, but abandoned it due to lack of use.

Relevance Feedback Summary

- Relevance feedback has been shown to be very effective at improving relevance of results.
  - Requires enough judged documents, otherwise it's unstable ($\geq$ 5 recommended)
  - Requires queries for which the set of relevant documents is medium to large
- Full relevance feedback is painful for the user.
- Full relevance feedback is not very efficient in most IR systems.
- Other types of interactive retrieval may improve relevance by as much with less work.

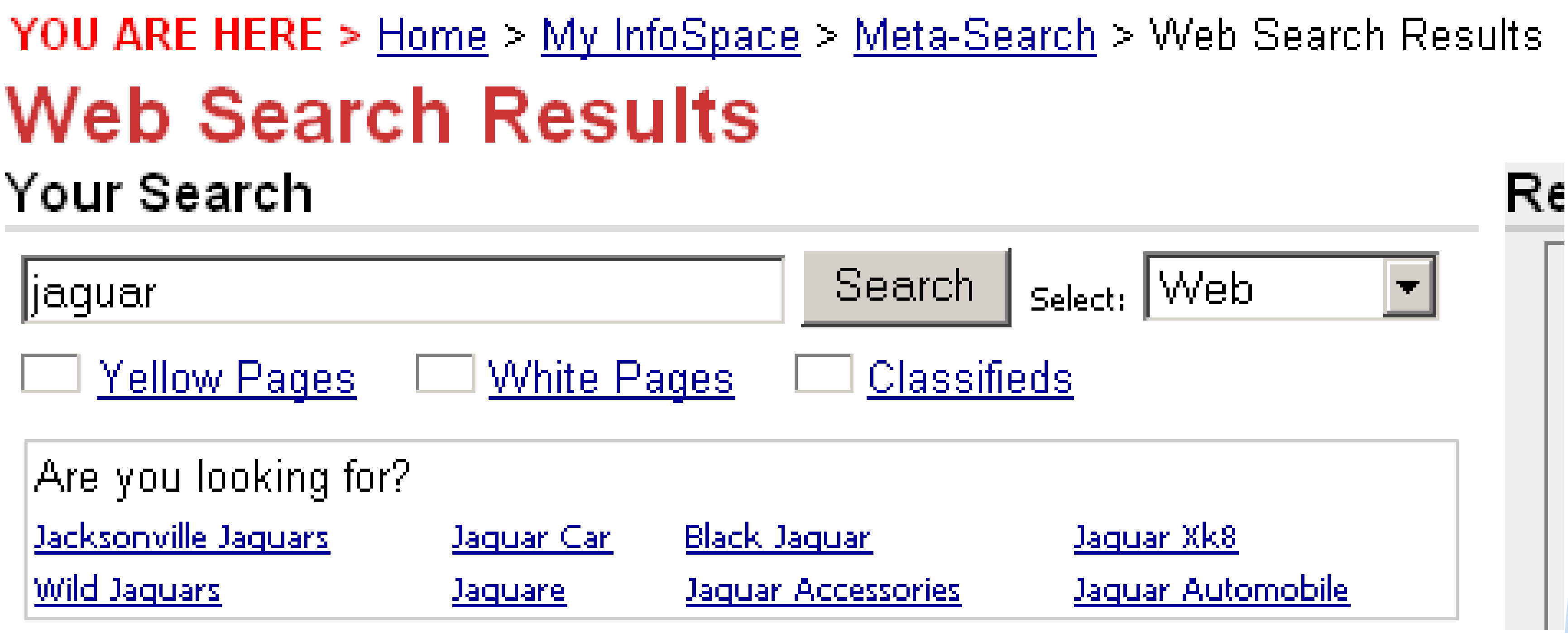
The complete landscape

- Global methods
  - Query expansion/reformulation
    - Thesauri (or WordNet)
    - Automatic thesaurus generation
  - Global indirect relevance feedback
- Local methods
  - Relevance feedback
  - Pseudo relevance feedback

Query Expansion

- In relevance feedback, users give additional input (relevant/non-relevant) on documents, which is used to reweight terms in the documents
- In query expansion, users give additional input (good/bad search term) on words or phrases.

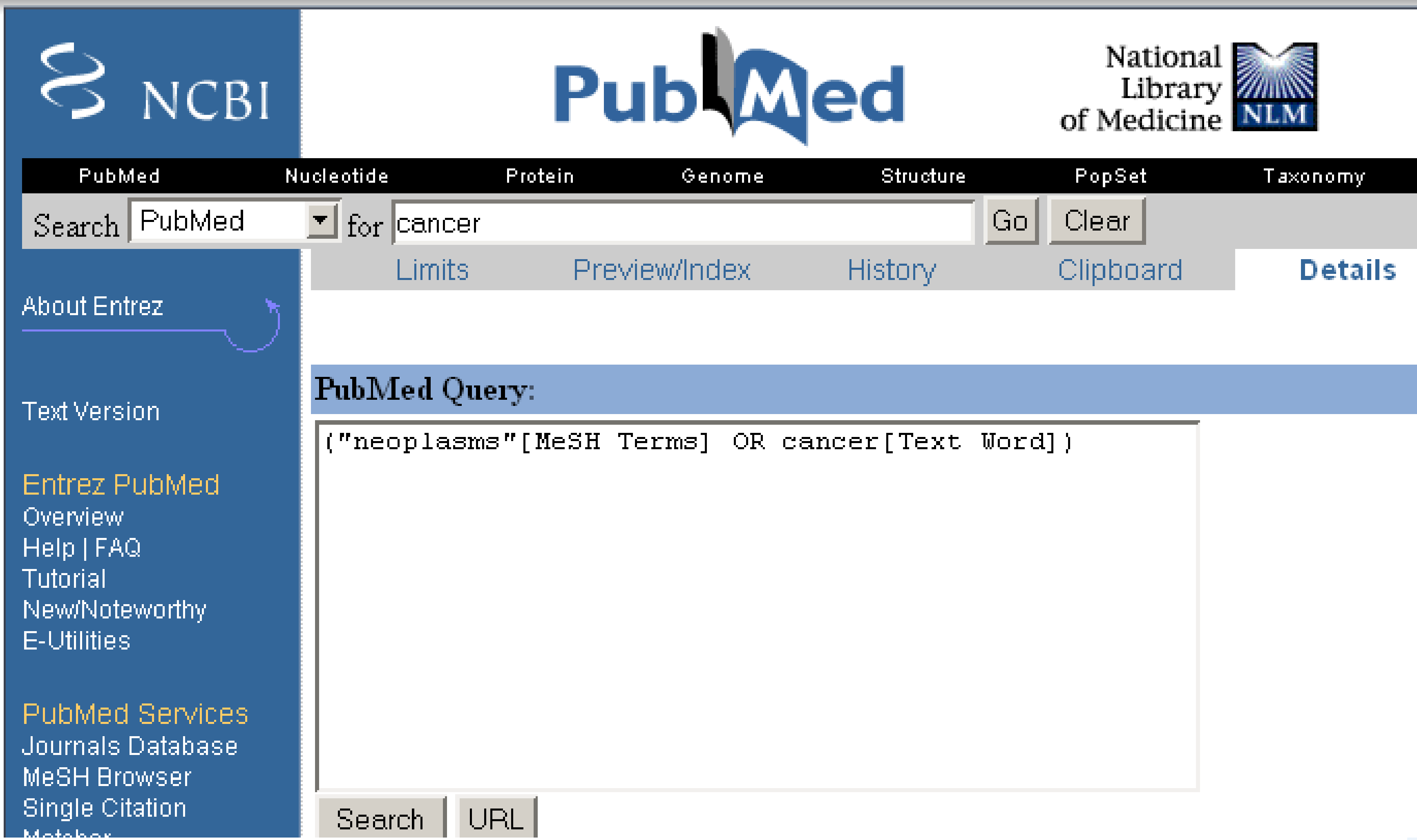
Query Expansion: Example



Types of Query Expansion

- Global Analysis: (static; of all documents in collection)
  - Controlled vocabulary
    - Maintained by editors (e.g., medline)
  - Manual thesaurus
    - E.g. MedLine: physician, syn: doc, doctor, MD, medico
  - Automatically derived thesaurus
    - (co-occurrence statistics)
  - Refinements based on query log mining
    - Common on the web
- Local Analysis: (dynamic)
  - Analysis of documents in result set

Controlled Vocabulary



Thesaurus-based Query Expansion

- This doesn't require user input
- For each term, *t*, in a query, expand the query with synonyms and related words of *t* from the thesaurus
  - feline → feline cat
- May weight added terms less than original query terms.
- Generally increases recall.
- Widely used in many science/engineering fields
- May significantly decrease precision, particularly with ambiguous terms.
  - "interest rate" → "interest rate fascinate evaluate"
- There is a high cost of manually producing a thesaurus
  - And for updating it for scientific changes

Automatic Thesaurus Generation Example

word	ten nearest neighbors
absolutely	absurd whatsoever totally exactly nothing
bottomed	dip copper drops topped slide trimmed slig
captivating	shimmer stunningly superbly plucky witty
doghouse	dog porch crawling beside downstairs gazed
Makeup	repellent lotion glossy sunscreen Skin gel p
mediating	reconciliation negotiate cease conciliation p
keeping	hoping bring wiping could some would othe
lithographs	drawings Picasso Dali sculptures Gauguin
pathogens	toxins bacteria organisms bacterial parasite
senses	grasp psyche truly clumsy naive innate awk

Query Expansion: Summary

- Query expansion is often effective in increasing recall.
  - Not always with general thesauri
  - Fairly successful for subject-specific collections
- In most cases, precision is decreased, often significantly.
- Overall, not as useful as relevance feedback; may be as good as