BLG 540E
TEXT RETRIEVAL SYSTEMS

Evaluation of IR Systems
Relevance Feedback
Query Expansion

Arzucan Özgür

Faculty of Computer and Informatics, İstanbul Techical University

March 18, 2011

Evaluation of IR Systems

Measures for a search engine

- How fast does it index
  - Number of documents/hour
- How fast does it search
  - Latency as a function of index size
- How large is the document collection
- Expressiveness of query language
  - Ability to express complex information needs
  - Speed on complex queries
- Uncluttered User Interface
- Is it free?

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
  - Fast, but 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

- Web engine:
  - Users find what they want and return to the engine
    - Can measure rate of return users
  - Advertisers also users of modern search engines.
    - Happy when customers click through to their sites and make purchase

- eCommerce site: user finds what they want and buy
  - 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/government/academic): Care about "user productivity"
  - How much time do my users spend when looking for information?
  - Many other criteria having to do with secure access, etc.

Happiness: elusive to measure

- Most common proxy: *relevance* of search results

- Relevance measurement requires 3 elements:

1. A benchmark document collection
2. A benchmark suite of queries
3. A usually binary assessment of either Relevant or Nonrelevant for each query and each document

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 these words

Standard relevance benchmarks

- TREC - National Institute of Standards and Technology (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, Relevant or Nonrelevant
  - or at least for subset of docs that some system returned for that query

Sample TREC query

<top>
<num> Number: 305
<title> Most Dangerous Vehicles

<desc> Description:
Which are the most crashworthy, and least crashworthy, passenger vehicles?

<narr> Narrative:
A relevant document will contain information on the crashworthiness of a given vehicle or vehicles that can be used to draw a comparison with other vehicles. The document will have to describe/compare vehicles, not drivers. For instance, it should be expected that vehicles preferred by 16-25 year-olds would be involved in more crashes, because that age group is involved in more crashes. I would view number of fatalities per 100 crashes to be more revealing of a vehicle's crashworthiness than the number of crashes per 100,000 miles, for example.
</top>

LA031689-0177	LA042790-0172
FT922-1008	LA021790-0136
LA090190-0126	LA092289-0167
LA101190-0218	LA111189-0013
LA082690-0158	LA120189-0179
LA112590-0109	LA020490-0021
FT944-136	LA122989-0063
LA020590-0119	LA091389-0119
FT944-5300	LA072189-0048
LA052190-0048	FT944-15615
LA051689-0139	LA091589-0101
FT944-9371	LA021289-0208
LA032390-0172	

<DOCNO> LA031689-0177 </DOCNO>
<DOCID> 31701 </DOCID>
<DATE><P>March 16, 1989, Thursday, Home Edition </P></DATE>
<SECTION><P>Business; Part 4; Page 1; Column 5; Financial Desk </P></SECTION>
<LENGTH><P>586 words </P></LENGTH>
<HEADLINE><P>AGENCY TO LAUNCH STUDY OF FORD BRONCO II AFTER HIGH RATE OF ROLL-OVER ACCIDENTS </P></HEADLINE>
<BYLINE><P>By LINDA WILLIAMS, Times Staff Writer </P></BYLINE>
<TEXT>
<P>The federal government's highway safety watchdog said Wednesday that the Ford Bronco II appears to be involved in more fatal roll-over accidents than other vehicles in its class and that it will seek to determine if the vehicle itself contributes to the accidents. </P>
<P>The decision to do an engineering analysis of the Ford Motor Co. utility-sport vehicle grew out of a federal accident study of the Suzuki Samurai, said Tim Hurd, a spokesman for the National Highway Traffic Safety Administration. NHTSA looked at Samurai accidents after Consumer Reports magazine charged that the vehicle had basic design flaws. </P>
<P>Several Fatalities </P>
<P>However, the accident study showed that the "Ford Bronco II appears to have a higher number of single-vehicle, first event roll-overs, particularly those involving fatalities," Hurd said. The engineering analysis of the Bronco, the second of three levels of investigation conducted by NHTSA, will cover the 1984-1989 Bronco II models, the agency said. </P>
<P>According to a Fatal Accident Reporting System study included in the September report on the Samurai, 43 Bronco II single-vehicle roll-overs caused fatalities, or 19 of every 100,000 vehicles. There were eight Samurai fatal roll-overs, or 6 per 100,000; 13 involving the Chevrolet S10 Blazers or GMC Jimmy, or 6 per 100,000, and six fatal Jeep Cherokee roll-overs, for 2.5 per 100,000. After the accident report, NHTSA declined to investigate the Samurai. </P>
...
</TEXT>
<GRAPHIC><P> Photo, The Ford Bronco II "appears to have a higher number of single-vehicle, first event roll-overs," a federal official said. </P></GRAPHIC>
<SUBJECT>
<P>TRAFFIC ACCIDENTS; FORD MOTOR CORP; NATIONAL HIGHWAY TRAFFIC SAFETY ADMINISTRATION; VEHICLE INSPECTIONS; RECREATIONAL VEHICLES; SUZUKI MOTOR CO; AUTOMOBILE SAFETY </P>
</SUBJECT>
</DOC>

TREC (cont'd)

- http://trec.nist.gov/tracks.html
- http://trec.nist.gov/presentations/presentations.html

Unranked retrieval evaluation: Precision and Recall

- **Precision**: fraction of retrieved docs that are relevant
- **Recall**: fraction of relevant docs that are retrieved

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

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

Should we instead use the accuracy measure for evaluation?

- Given a query, an engine classifies each doc as "Relevant" or "Nonrelevant"
- The **accuracy** of an engine: the fraction of these classifications that are correct
  - (tp + tn) / (tp + fp + fn + tn)
- **Accuracy** is a commonly used evaluation measure in machine learning classification work
- 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….

- 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 the number of docs retrieved or recall increases
  - This is not a theorem, but a result with strong empirical confirmation

A combined measure: F

- Combined measure that assesses precision/recall 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 $\alpha = ½$

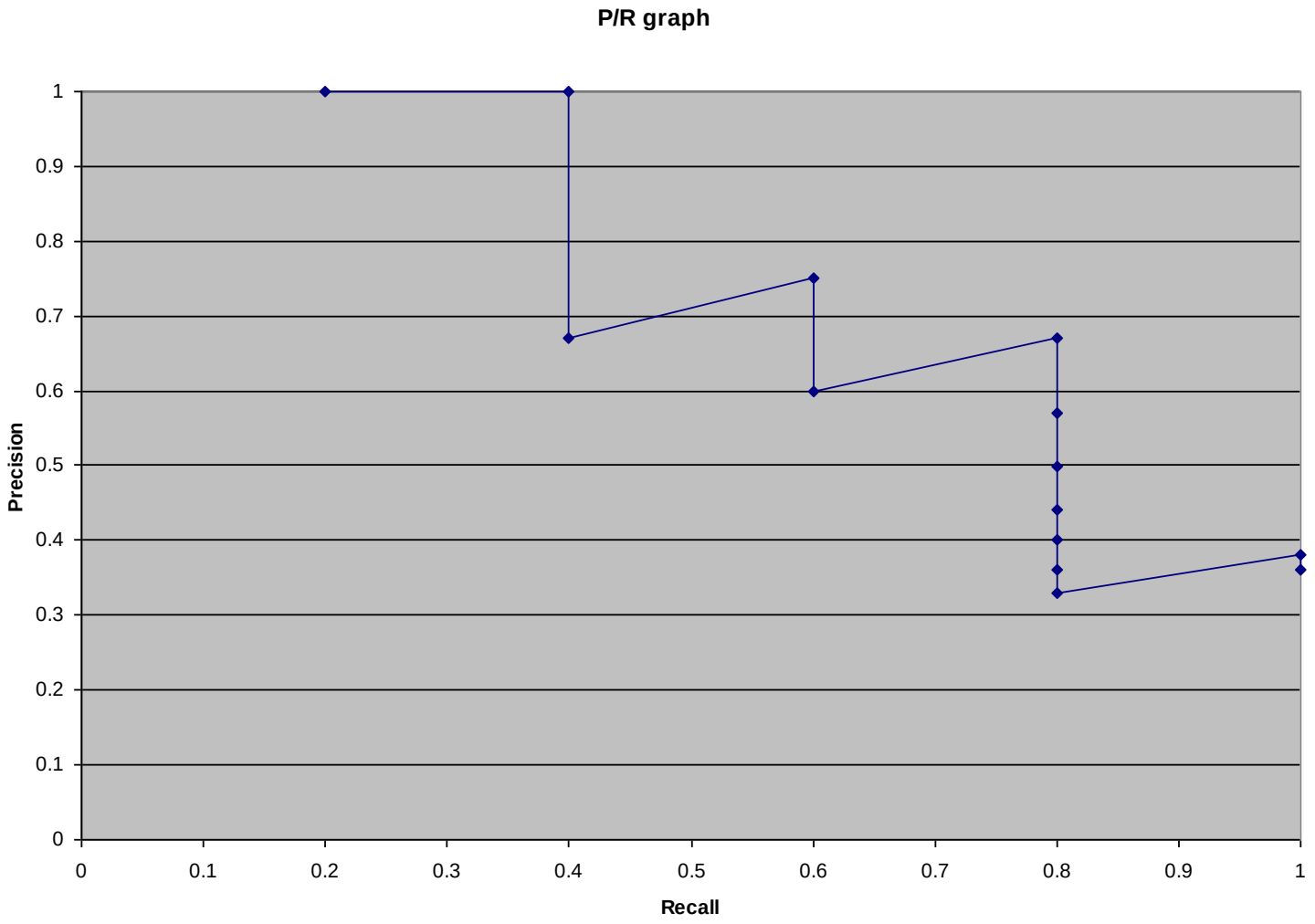
$$\mathbf{F_1} = \frac{2PR}{P+R}$$

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*

n	Doc. no	Relevant?	Recall	Precision
1	588	x	0.2	1.00
2	589	x	0.4	1.00
3	576		0.4	0.67
4	590	x	0.6	0.75
5	986		0.6	0.60
6	592	x	0.8	0.67
7	984		0.8	0.57
8	988		0.8	0.50
9	578		0.8	0.44
10	985		0.8	0.40
11	103		0.8	0.36
12	591		0.8	0.33
13	772	x	1.0	0.38
14	990		1.0	0.36

[From Salton's book]

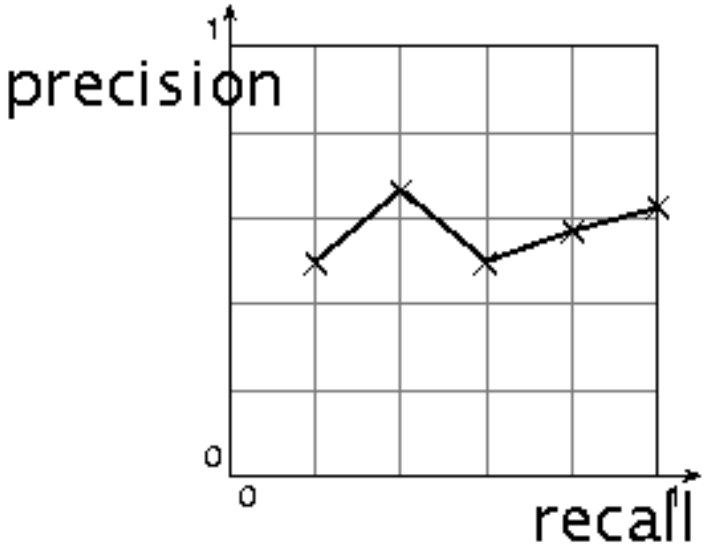

P/R graph
Precision
Recall
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

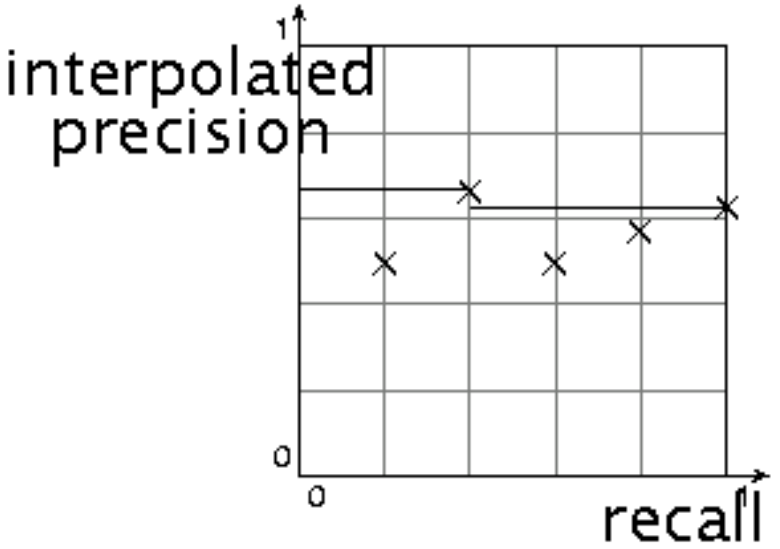
Averaging over queries

- 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?

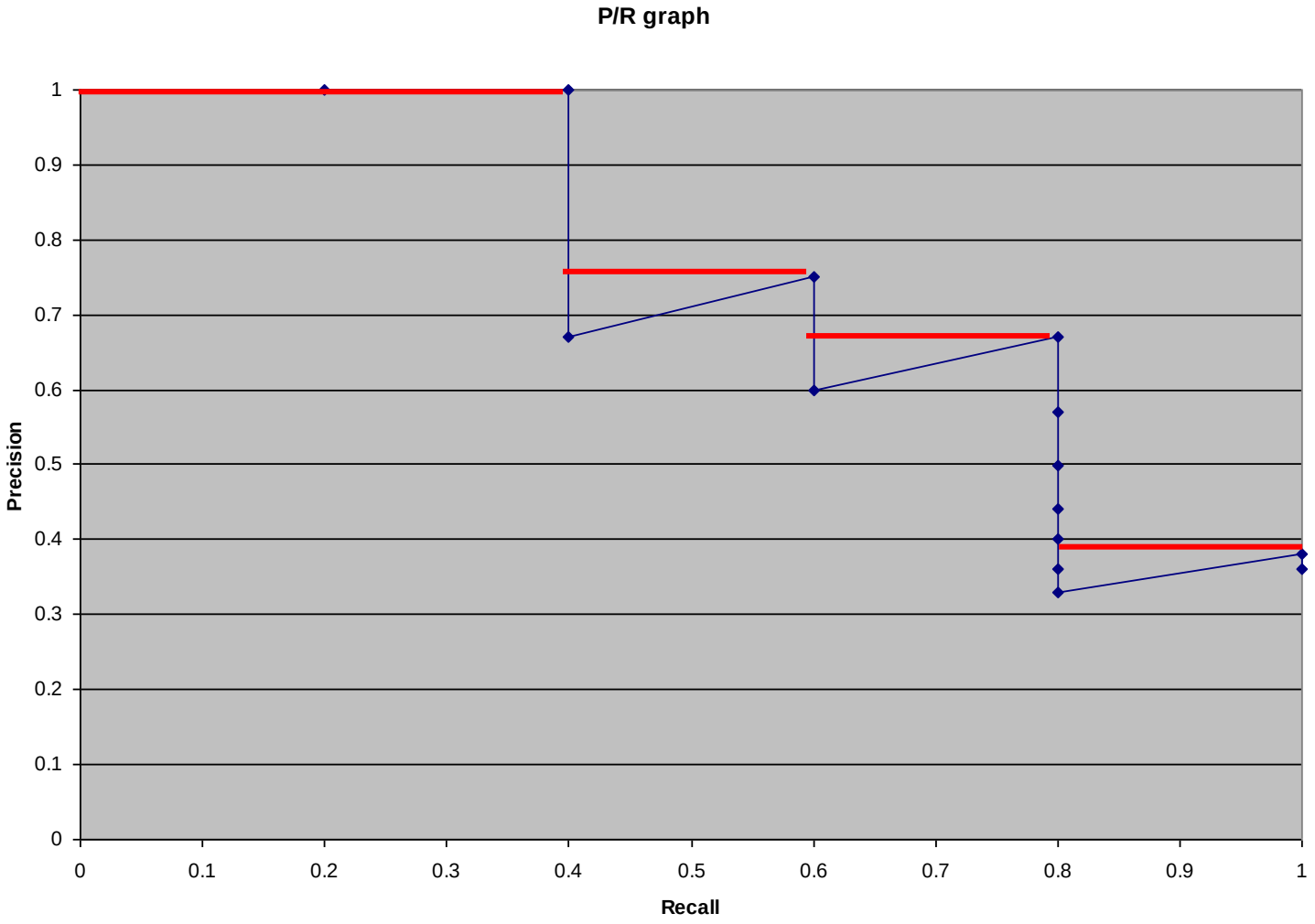
Interpolated precision

- Idea: If locally precision increases with increasing recall, then you should get to count that…
- So you max of precisions to right of value





Interpolated precision

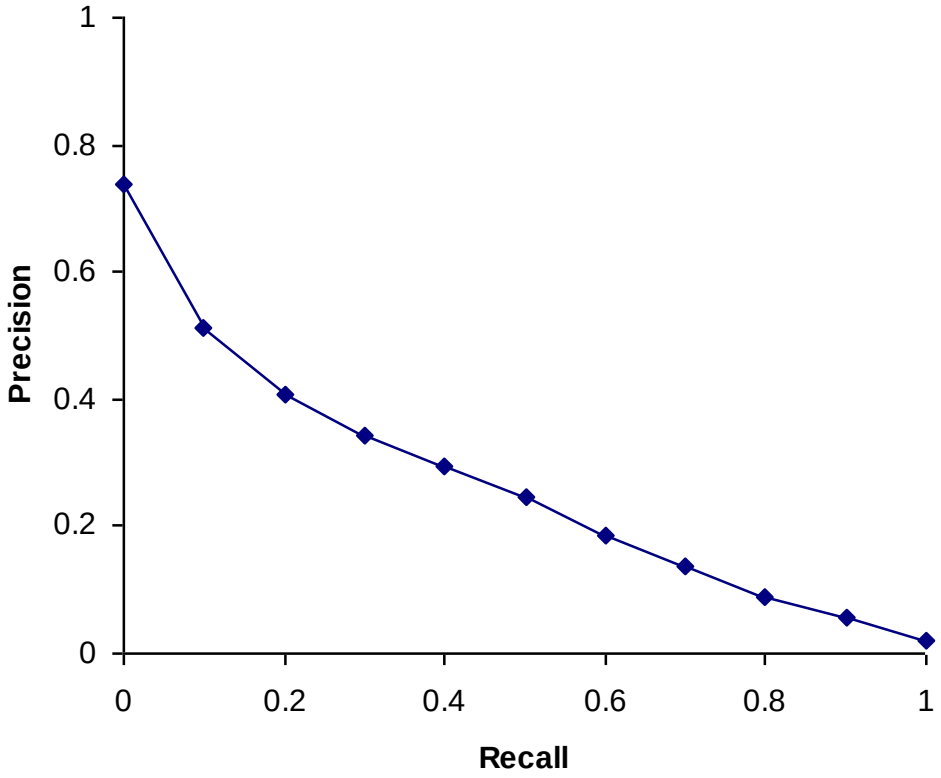


Evaluation

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

Typical (good) 11 point precisions

- SabIR/Cornell 8A1 11pt precision from TREC 8 (1999) across 50 queries.



Yet more evaluation measures…

- Mean average precision (MAP)
  - Average of the precision value obtained for the top *k* documents, each time a relevant doc is retrieved
  - MAP for query collection is arithmetic ave.

- R-precision
  - If have known (though perhaps incomplete) set of relevant documents of size *Rel*, then calculate precision of top *Rel* docs returned
  - Perfect system could score 1.0.

Variance

- For a test collection, it is usual that a system does crummily on some information needs (e.g., MAP = 0.1) and excellently on others (e.g., MAP = 0.7)
- Indeed, it is usually the case that the variance in performance of the same system across queries is much greater than the variance of different systems on the same query.

- That is, there are easy information needs and hard ones!

Creating Test Collections for IR Evaluation

From document collections 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?

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

		Judge 2 Relevance		
		Yes	No	Total
Judge 1 Relevance	Yes	300	20	320
	No	10	70	80
	Total	310	90	400

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 Example

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

TREC

- TREC Ad Hoc task from first 8 TRECs is standard IR task
  - 50 detailed information needs a year
  - Human evaluation of pooled results returned
  - More recently other related things: Web track
- A TREC query (TREC 5)

<top>

<num> Number: 225

<desc> Description:

What is the main function of the Federal Emergency Management Agency (FEMA) and the funding level provided to meet emergencies? Also, what resources are available to FEMA such as people, equipment, facilities?

</top>

Standard relevance benchmarks: Others

- GOV2
  - Another TREC/NIST collection
  - 25 million web pages
  - Largest collection that is easily available
  - But still much smaller than what Google/Yahoo/MSN index
- NTCIR
  - East Asian language and cross-language information retrieval
- Cross Language Evaluation Forum (CLEF)
  - This evaluation series has concentrated on European languages and cross-language information retrieval.
- Many others

Impact of Inter-judge Agreement

- Impact on absolute performance measure can be significant (0.32 vs 0.39)
- Little impact on ranking of different systems or relative performance
- Suppose we want to know if algorithm A is better than algorithm B
- A standard information retrieval experiment will give us a reliable answer to this question.

Evaluation at large search engines

- Search engines have test collections of queries and hand-ranked results
- Recall is difficult to measure on the web
- Search engines often use precision at top k, e.g., k = 10
- . . . or measures that reward you more for getting rank 1 right than for getting rank 10 right.
  - NDCG (Normalized Cumulative Discounted Gain)
- Search engines also use non-relevance-based measures.
  - Clickthrough on first result
    - Not very reliable if you look at a single clickthrough … but pretty reliable in the aggregate.
  - Studies of user behavior in the lab
  - A/B testing

A/B testing

- Purpose: Test a single innovation
- Prerequisite: You have a large search engine up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an "automatic" measure like clickthrough on first result
- Now we can directly see if the innovation does improve user happiness.
- Probably the evaluation methodology that large search engines trust most
- In principle less powerful than doing a multivariate regression analysis, but easier to understand

Results presentation

Result Summaries

- Having ranked the documents matching a query, we wish to present a results list
- Most commonly, a list of the document titles plus a short summary, aka "10 blue links"

Reşat Nuri Güntekin - Vikipedi
Reşat Nuri Güntekin (25 Kasım 1889;, İstanbul - 7 Aralık 1956; Londra), Cumhuriyet dönemi edebiyatında önemli bir yeri olan Çalıkuşu, Yeşil Gece ve Anadolu ...
Yaşamı - Eserleri Hakkında Bilgiler - Çalışma Yöntemi Hakkında - Romanları
tr.wikipedia.org/wiki/**Reşat_Nuri_Güntekin** - Önbellek - Benzer

reşat nuri güntekin in eserleri kimdir hayatı eserleri şiirleri ...
REŞAT NURİ GÜNTEKİN. 25 Kasım 1889'da İstanbul'da doğdu. 7 Aralık 1956'da Londra'da öldü. İlk öğrenimini Çanakkale'de Mekteb-i İptidai'de yaptı. ...
www.edebiyatogretmeni.net/**resat_nuri_guntekin**.htm - Önbellek - Benzer

Reşat Nuri Güntekin'in "Yaprak Dökümü" Adlı Romanının Özeti
Bu sayfada bir okul ödevim için hazırlamış olduğum, **Reşat Nuri Güntekin**'in -Yaprak Dökümü- adlı romanının özetini bulacaksınız. Alttaki küçük resim o ödev ...
www.islamisanat.net/ibrahim/yaprak_dokumu.html - Önbellek

Reşat Nuri Güntekin Biyografi.info
Reşat Nuri Güntekin (1889-1856) Kimdir? : **Reşat Nuri Güntekin** Biyografisi, **Reşat Nuri Güntekin** Fotoğrafları, **Reşat Nuri Güntekin** Hakkında herşey ...
www.biyografi.info/kisi/**resat-nuri-guntekin** - Önbellek - Benzer

REŞAT NURİ GÜNTEKİN İLKÖĞRETİM OKULU
Okulun tanıtımının yapıldığı sitede; tarihçesi, fotoğrafları, kadrosu, faaliyetleri, başarıları ve duyuruları yer alıyor.
www.rn**guntekin**.k12.tr/ - Önbellek - Benzer

Summaries

- The title is often automatically extracted from document metadata. What about the summaries?
  - This description is crucial.
  - User can identify good/relevant hits based on description.
- Two basic kinds:
  - Static
  - Dynamic
- A **static summary** of a document is always the same, regardless of the query that hit the doc
- A **dynamic summary** is a *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; text summarization work

Dynamic summaries

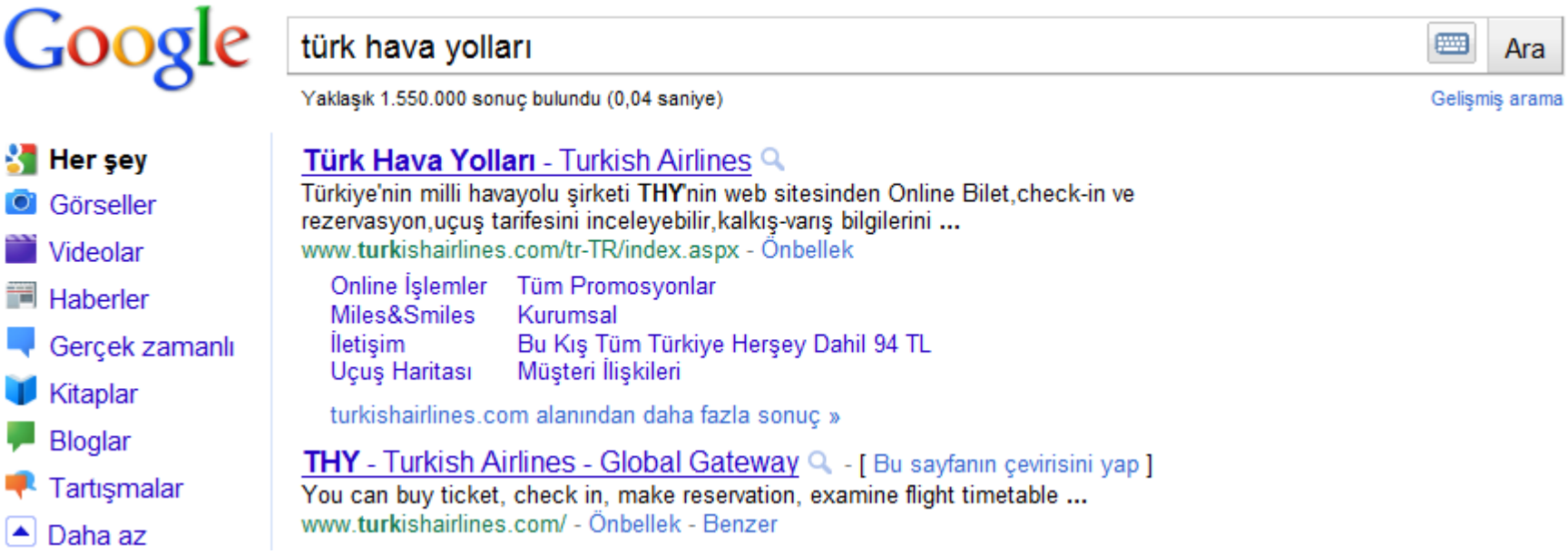
- Present one or more "windows" within the document that contain several of the query terms
  - "KWIC" snippets: Keyword in Context presentation

Techniques for dynamic summaries

- Find small windows in doc that contain query terms
  - Requires fast window lookup in a document cache
- Score each window wrt query
  - Use various features such as window width, position in document, etc.
  - Combine features through a scoring function

Quicklinks

- For a *navigational query* such as ***turk hava yollari*** user's need likely satisfied on *www.turkishairlines.com*
- Quicklinks provide navigational cues on that home page



Query expansion

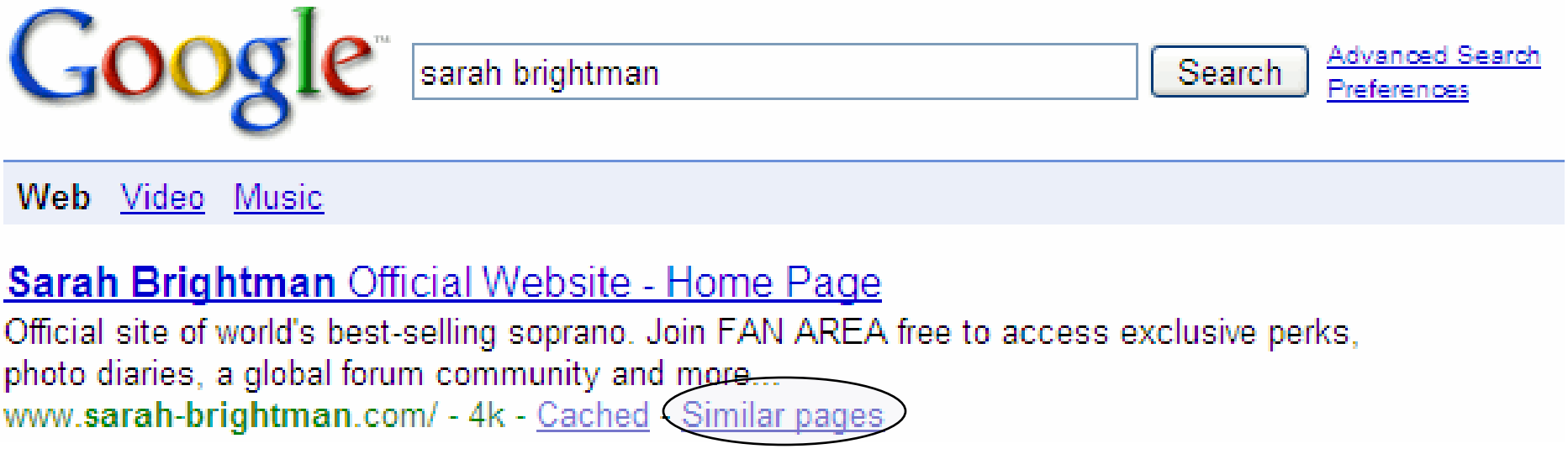
Improving results

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

Relevance Feedback

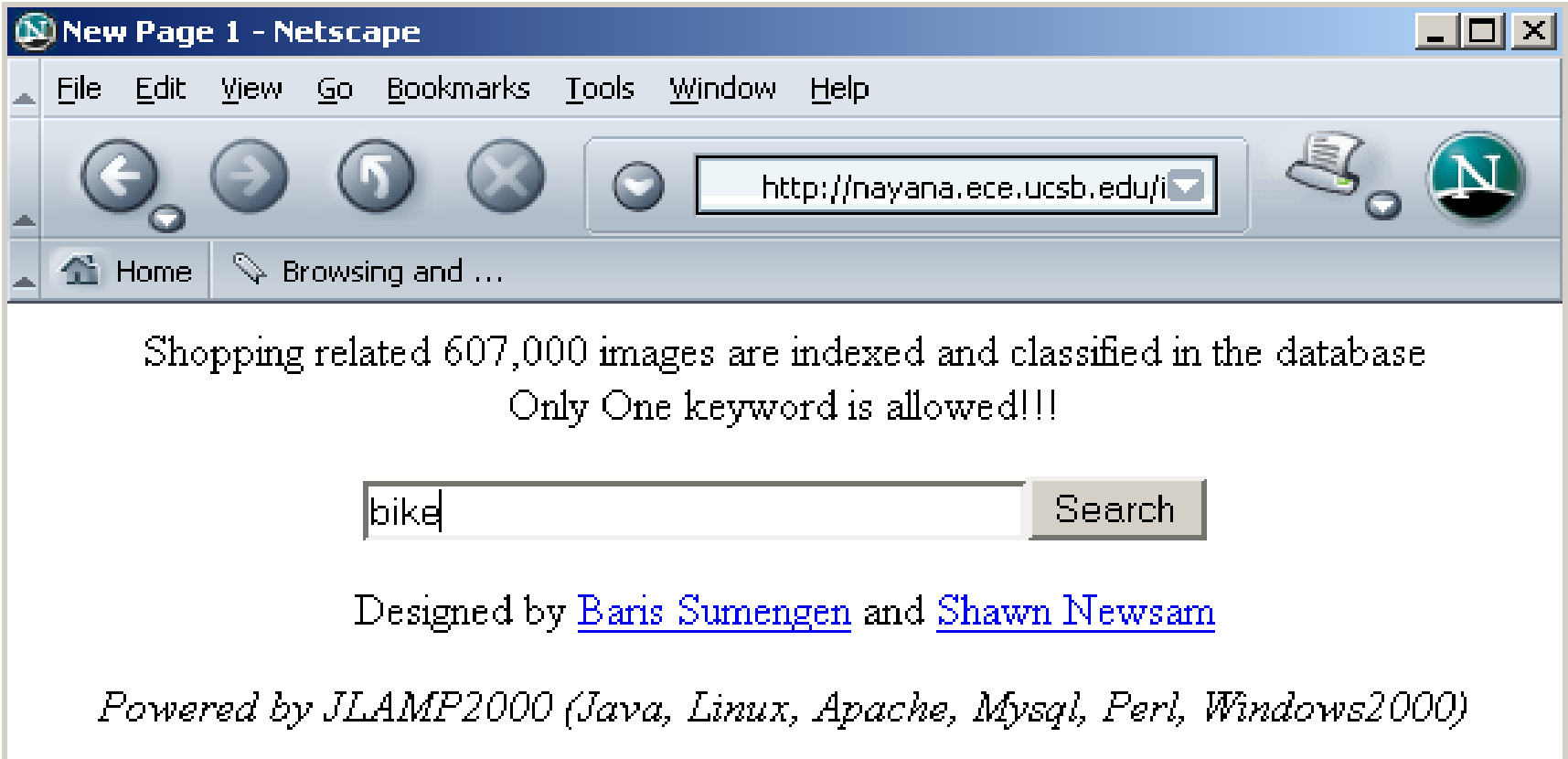
- Relevance feedback: user feedback on relevance of docs in initial set of results
  - User issues a (short, simple) query
  - The user marks some results as relevant or non-relevant.
  - The system computes a better representation of the information need based on feedback.
  - Relevance feedback can 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

Similar pages

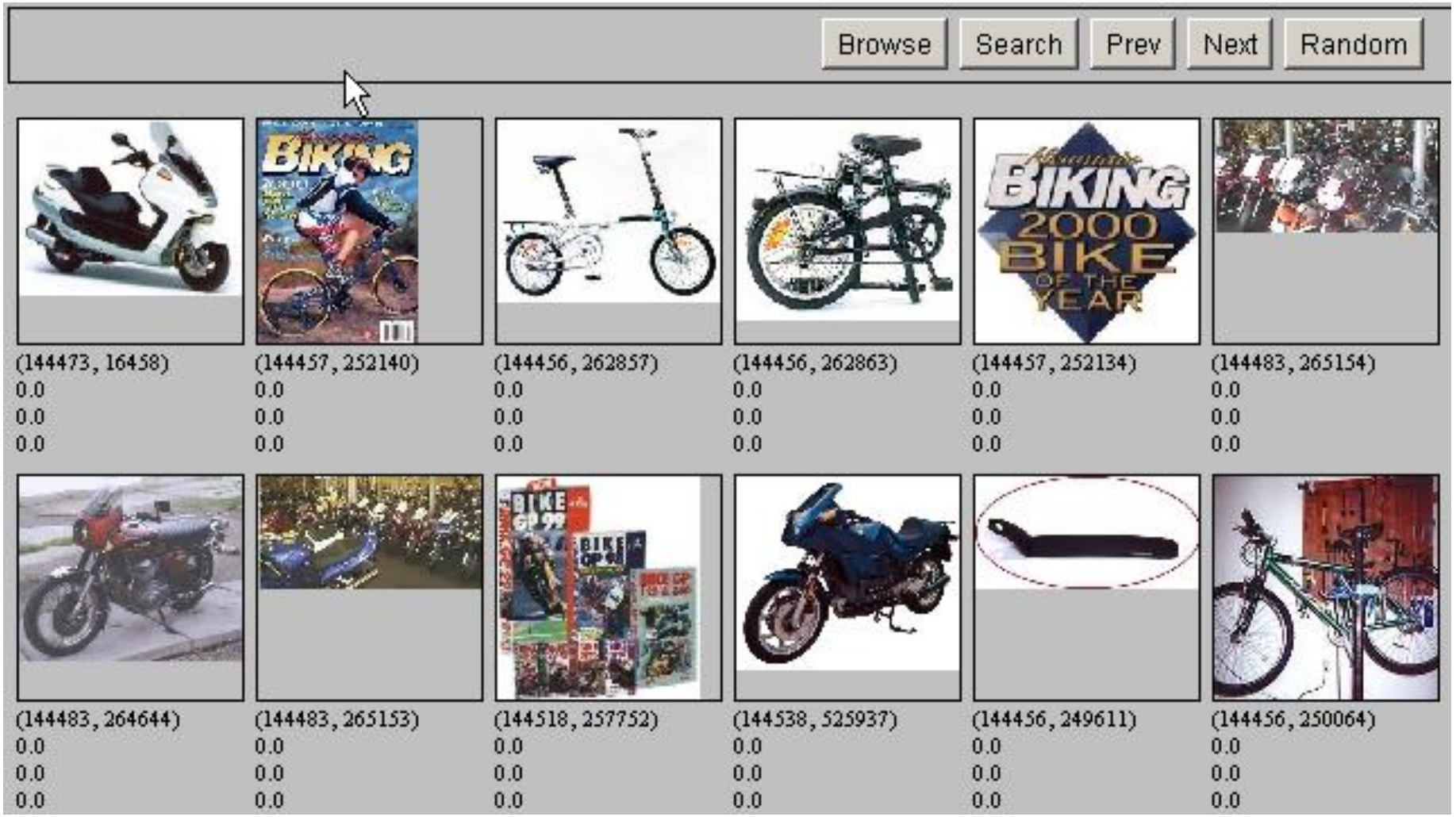


Relevance Feedback: Example

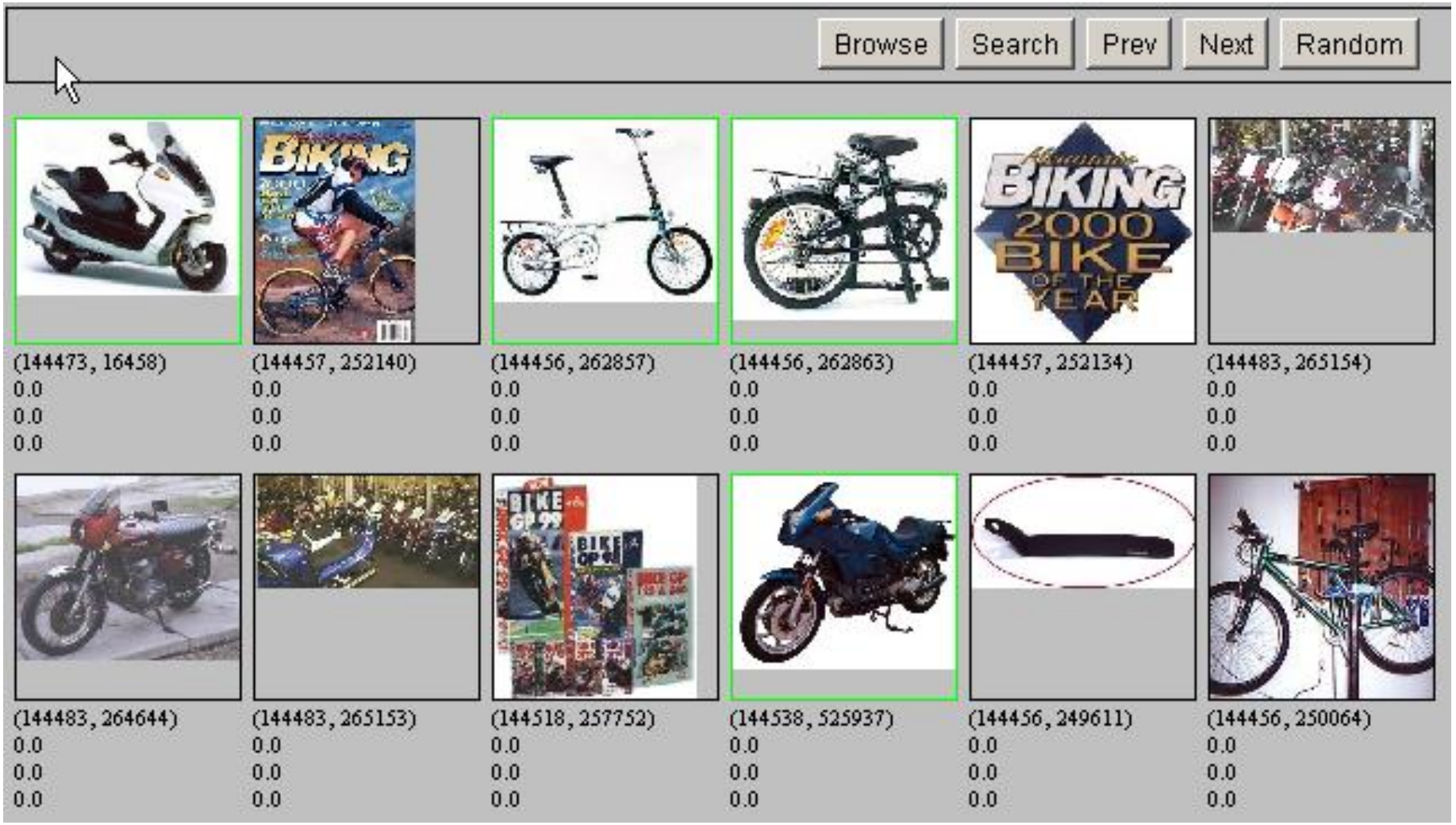
- Image search engine
  http://nayana.ece.ucsb.edu/imsearch/imsearch.html



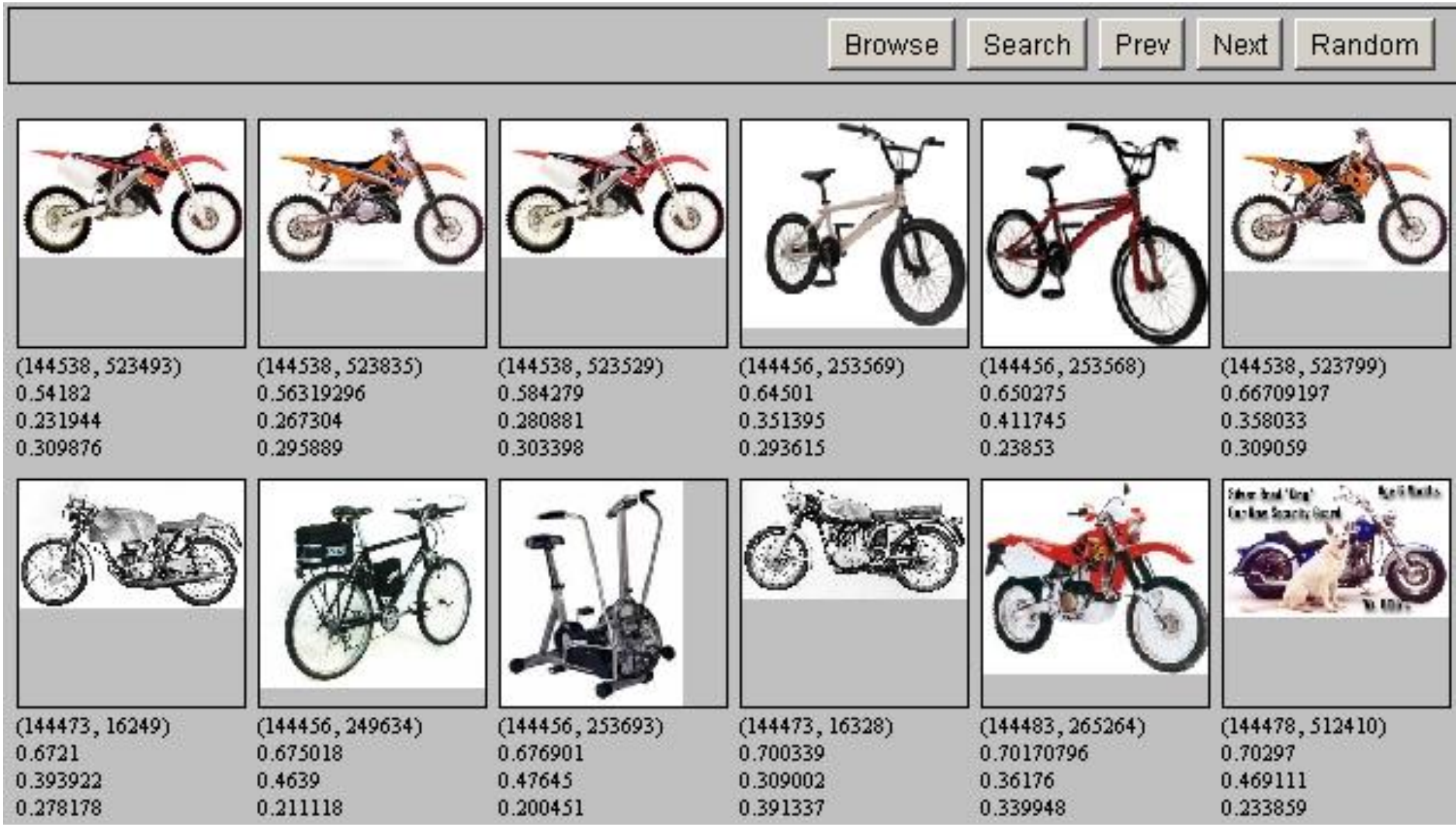
Results for Initial Query



Relevance Feedback



Results after Relevance Feedback



Initial query/results

- Initial 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

- User then marks relevant documents with "+".

Expanded query after relevance feedback

- 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

Results for expanded query

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

Key concept: Centroid

- The centroid is the center of mass of a set of points
- Recall that we represent documents as points in a high-dimensional space
- Definition: Centroid

$$\vec{\mu}(C) = \frac{1}{|C|}\sum_{d \in C}\vec{d}$$

where C is a set of documents.

Rocchio Algorithm

- The Rocchio algorithm uses the vector space model to pick a relevance feed-back query
- Rocchio seeks the query q_{opt} that maximizes

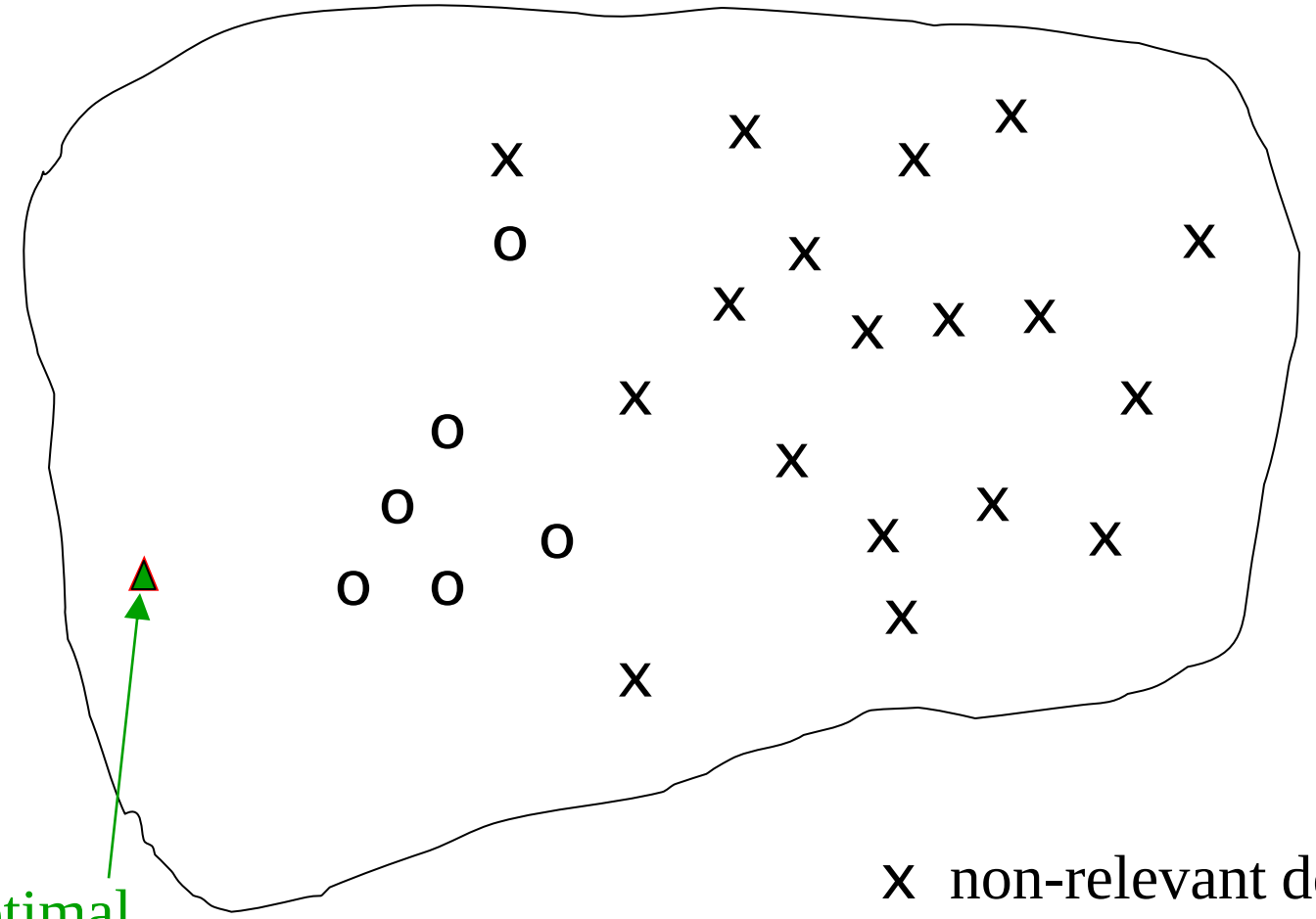
$$\vec{q}_{opt} = \arg\max_{\vec{q}} [\cos(\vec{q}, \vec{\mu}(C_r)) - \cos(\vec{q}, \vec{\mu}(C_{nr}))]$$

- Tries to separate docs marked relevant and non-relevant

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

- Problem: we don't know the truly relevant docs

The Theoretically Best Query



Rocchio 1971 Algorithm (SMART)

- Used in practice:

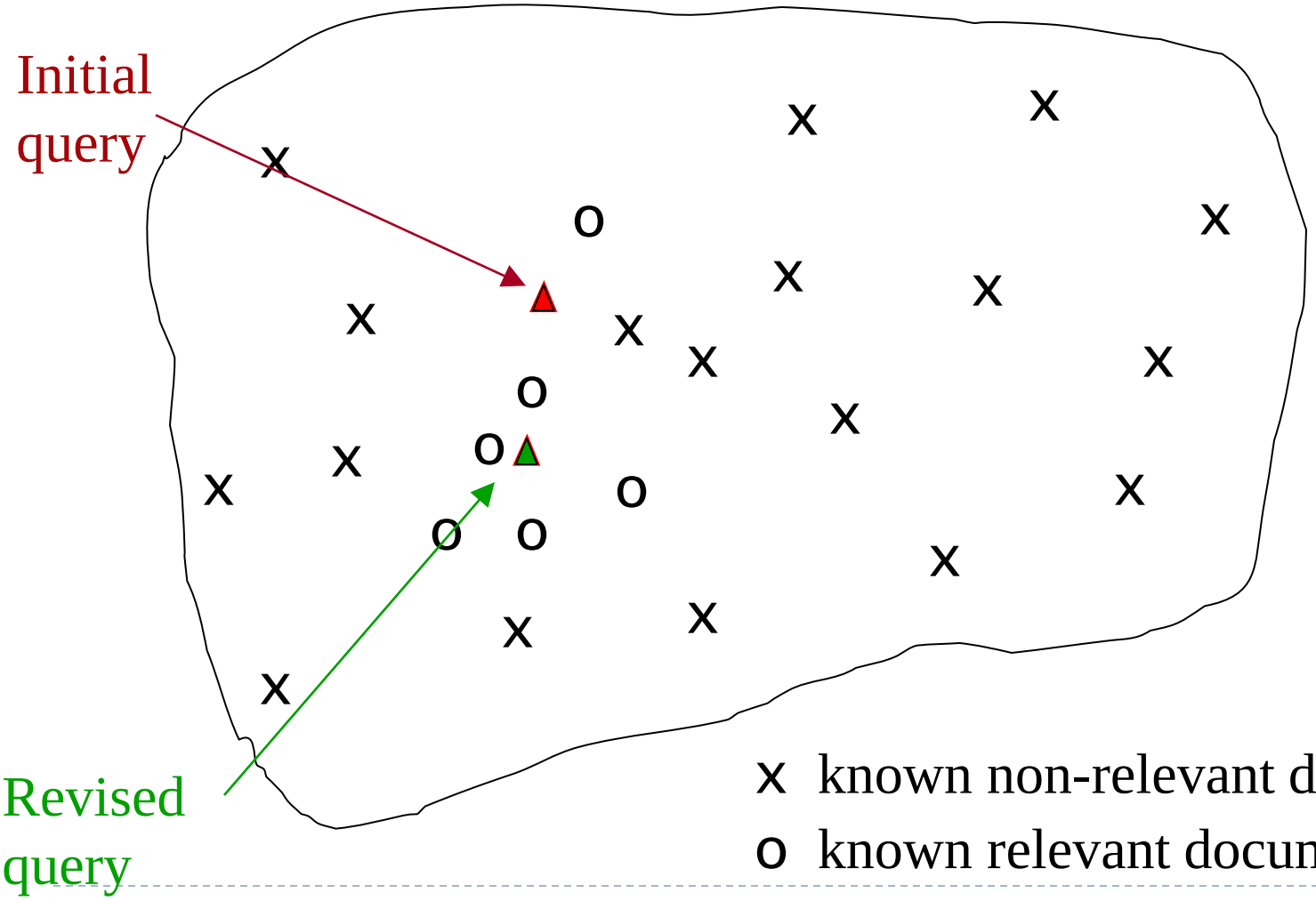
$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

- D_r = set of known relevant doc vectors
- D_{nr} = set of known irrelevant doc vectors
  - Different from C_r and C_{nr}
- q_m = modified query vector; q_0 = original query vector; α, β, γ: weights (hand-chosen or set empirically)
- New query moves toward relevant documents and away from irrelevant documents

Subtleties to note

- Tradeoff α vs. β/γ : If we have a lot of judged documents, we want a higher β/γ.
- Some weights in query vector 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 ($\gamma=0$).

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

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 applying relevance feedback

Evaluation of relevance feedback strategies

- Use q_0 and compute precision and recall graph
- Use q_m and compute precision recall graph
  - Assess on all documents in the collection
    - Spectacular improvements, but … it's cheating!
    - Partly due to known relevant documents ranked higher
    - Must evaluate with respect to documents not seen by user
  - Use documents in residual collection (set of documents minus those assessed relevant)
    - Measures usually then lower than for original query
    - But a more realistic evaluation
    - Relative performance can be validly compared
- Empirically, one round of relevance feedback is often very useful. Two rounds is sometimes marginally useful.

Evaluation of relevance feedback

- Second method – assess only the docs *not* rated by the user in the first round
  - Could make relevance feedback look worse than it really is
  - Can still assess relative performance of algorithms
- Most satisfactory – use two collections each with their own relevance assessments
  - q_0 and user feedback from first collection
  - q_m run on second collection and measured

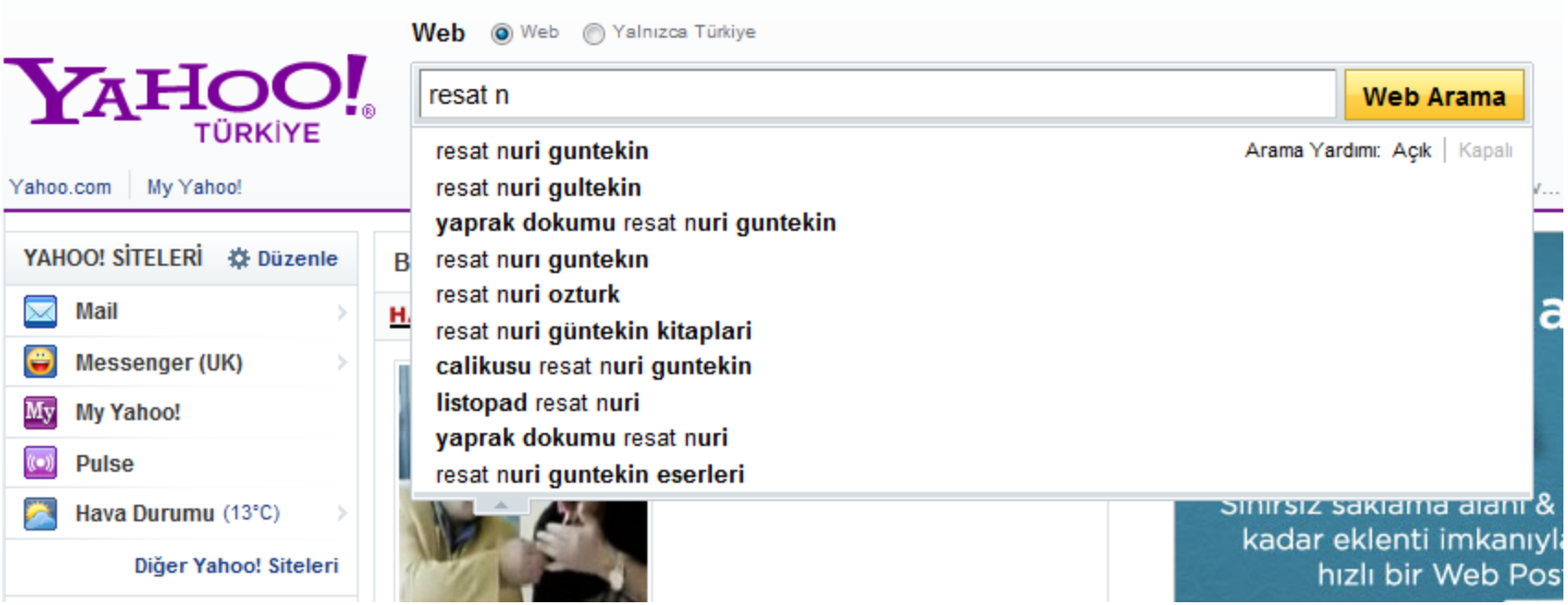
Pseudo relevance feedback

- Pseudo-relevance feedback automates the "manual" part of true relevance feedback.
- Pseudo-relevance algorithm:
  - Retrieve a ranked list of hits for the user's query
  - Assume that the top k documents are relevant.
  - Do relevance feedback (e.g., Rocchio)
- Works very well on average
- But can go horribly wrong for some queries.

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

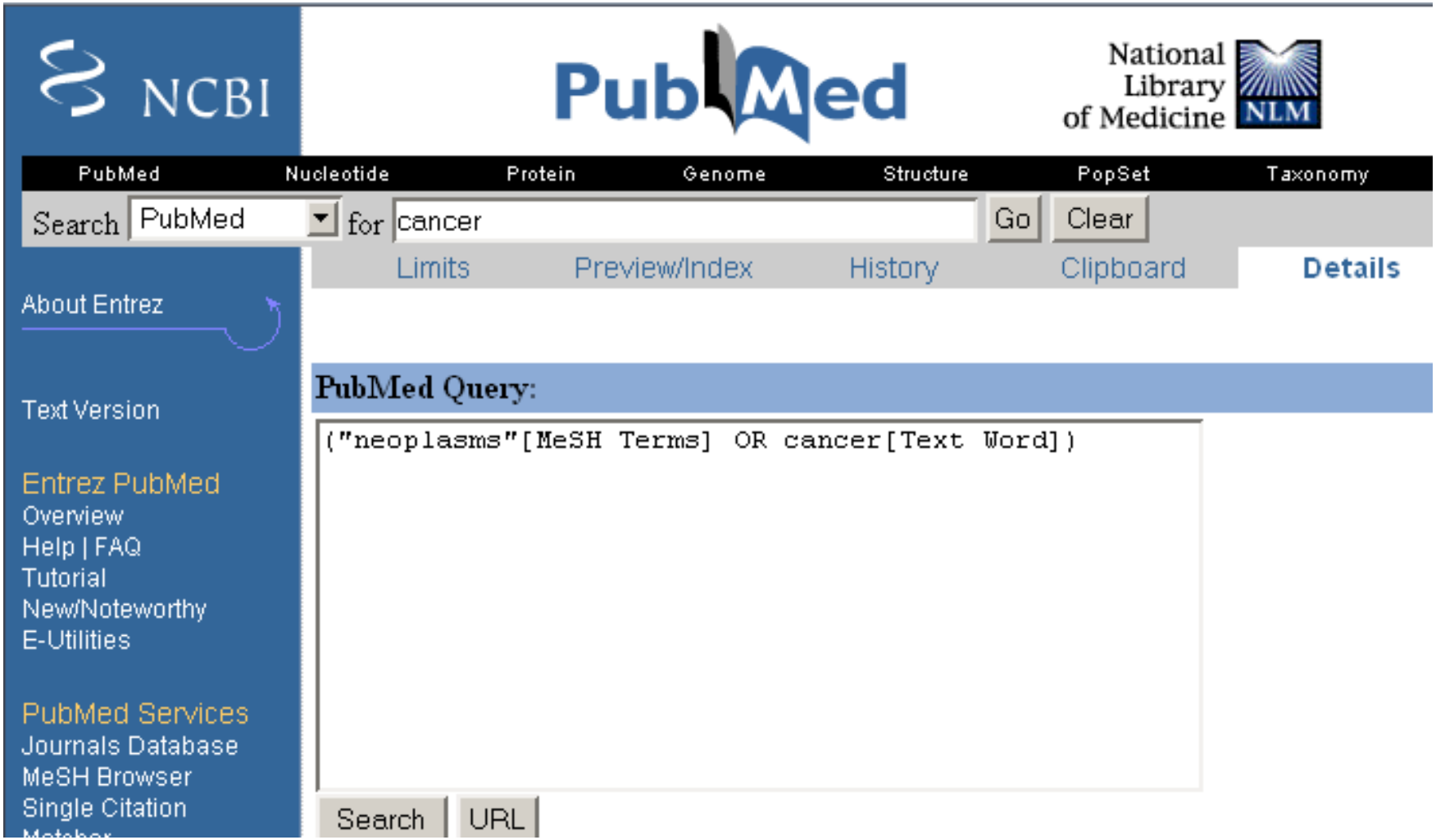
Query assist



How do we augment the user query?

- Manual thesaurus
  - E.g. MedLine: physician, syn: doc, doctor, MD, medico
- Global Analysis: (static; of all documents in collection)
  - 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

Example of manual thesaurus



Thesaurus-based query expansion

- 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

- Attempt to generate a thesaurus automatically by analyzing the collection of documents
- Fundamental notion: similarity between two words
- Definition 1: Two words are similar if they co-occur with similar words.
- Definition 2: Two words are similar if they occur in a given grammatical relation with the same words.
- You can harvest, peel, eat, prepare, etc. apples and pears, so apples and pears must be similar.

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 gazec
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 awl

Automatic Thesaurus Generation Discussion

- Quality of associations is usually a problem.
- Term ambiguity may introduce irrelevant statistically correlated terms.
  - "Apple computer" → "Apple red fruit computer"
- Problems:
  - False positives: Words deemed similar that are not
  - False negatives: Words deemed dissimilar that are similar
- Since terms are highly correlated anyway, expansion may not retrieve many additional documents.

Indirect relevance feedback

- On the web, DirectHit introduced a form of indirect relevance feedback.
- DirectHit ranked documents higher that users look at more often.
  - Clicked on links are assumed likely to be relevant
    - Assuming the displayed summaries are good, etc.
- Globally: Not necessarily user or query specific.
  - This is the general area of clickstream mining
- Today – handled as part of machine-learned ranking

References

- *Introduction to Information Retrieval*, chapters 8 & 9.
- The slides were adapted from
  - Prof. Dragomir Radev's lectures at the University of Michigan:
    - http://clair.si.umich.edu/~radev/teaching.html
  - the book's companion website:
    - http://nlp.stanford.edu/IR-book/information-retrieval-book.html
- See Russell http://dmrussell.googlepages.com/JCDL-talk-June-2007-short.pdf