Relevance Feedback and Query Expansion

SEEM5680

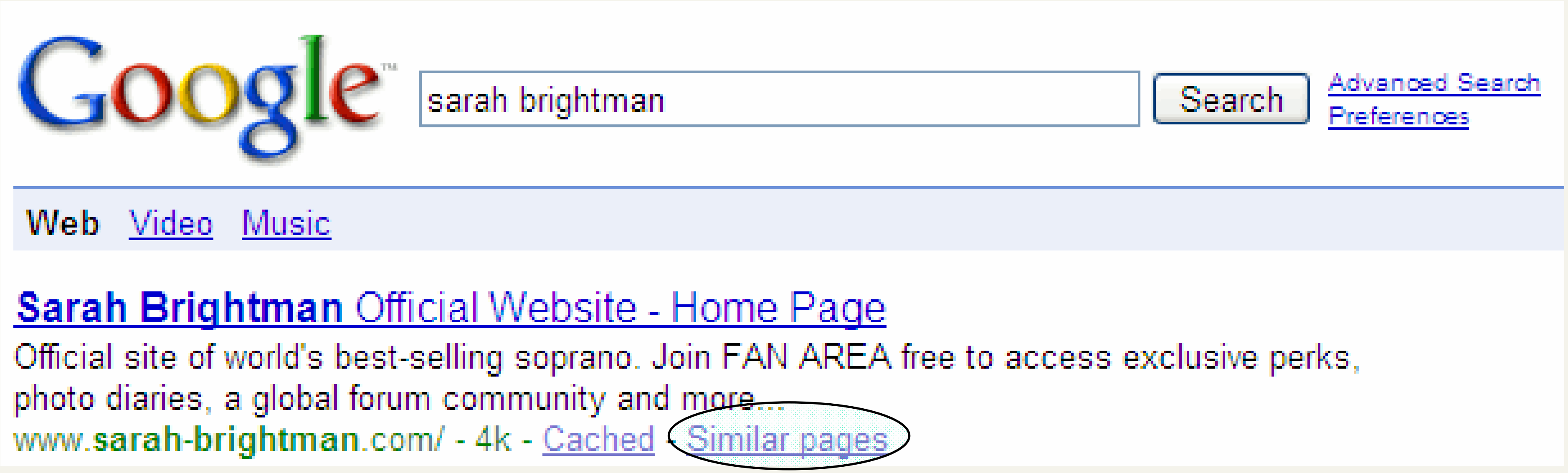
Relevance Feedback

- 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

Relevance feedback

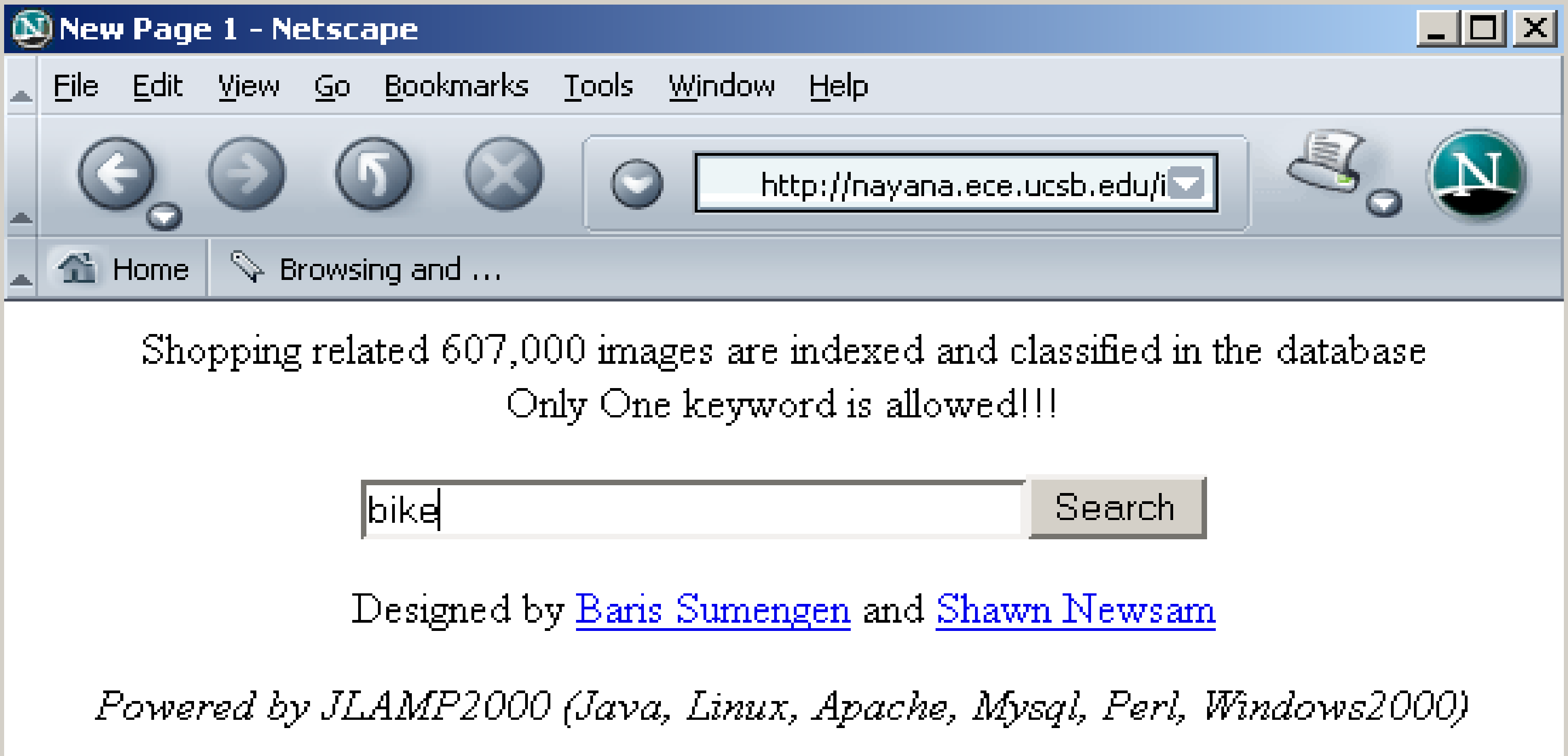
- We will use *ad hoc retrieval* to refer to regular retrieval without relevance feedback.
- We now look at some examples of relevance feedback.

Similar pages

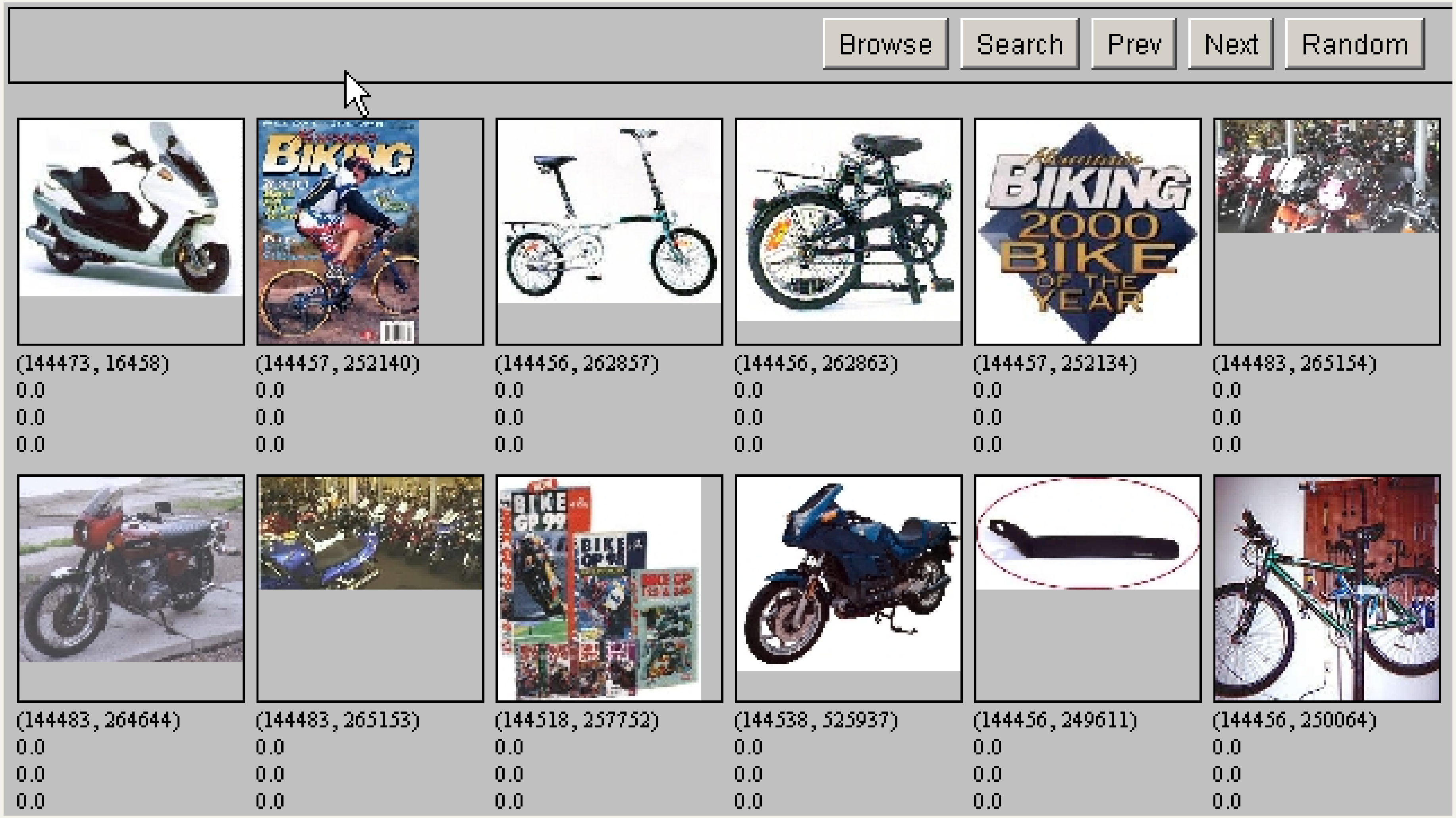


Relevance Feedback: Example

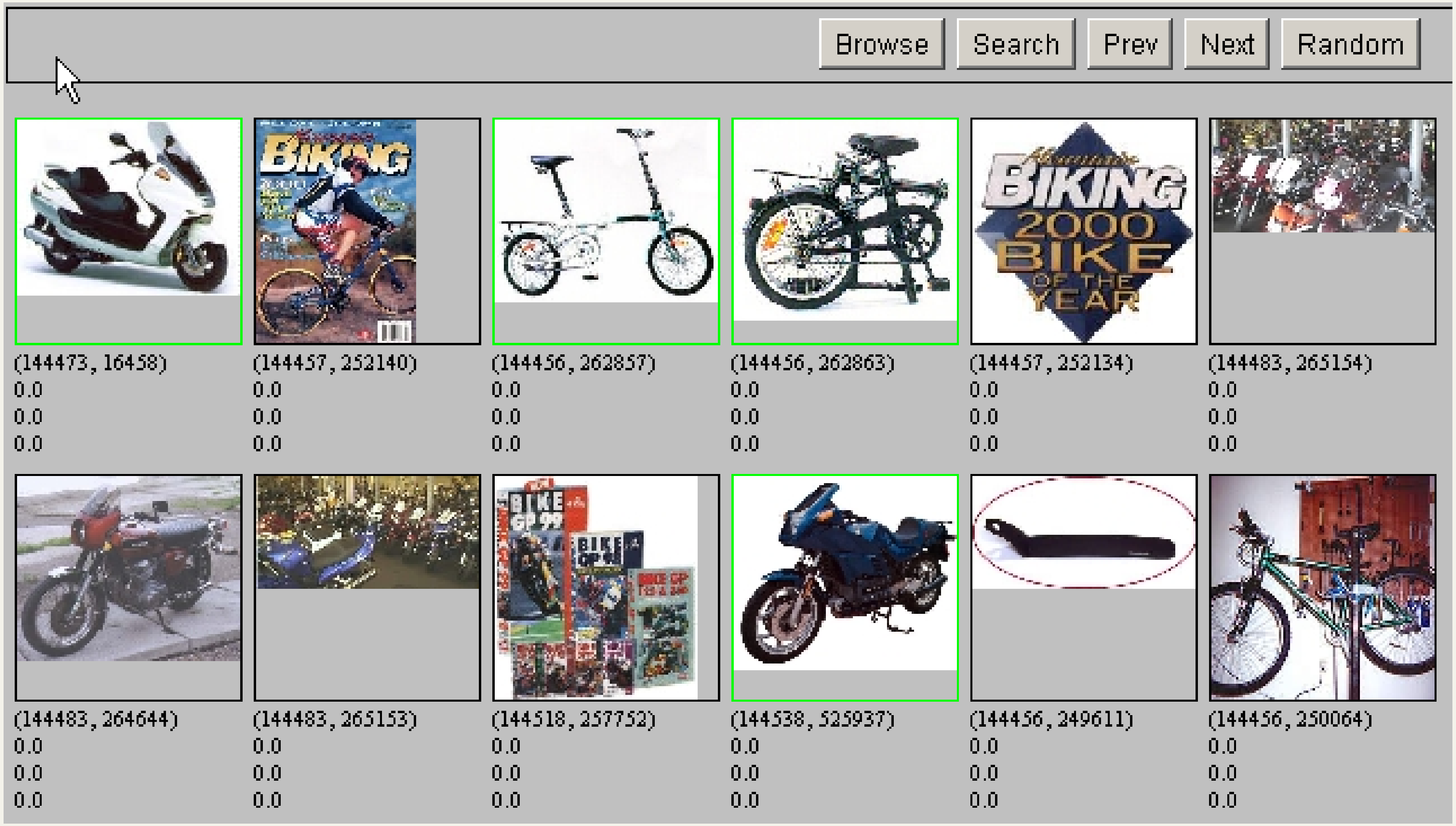
- Image search engine http://nayana.ece.ucsb.edu/imsearch/imsearch.html



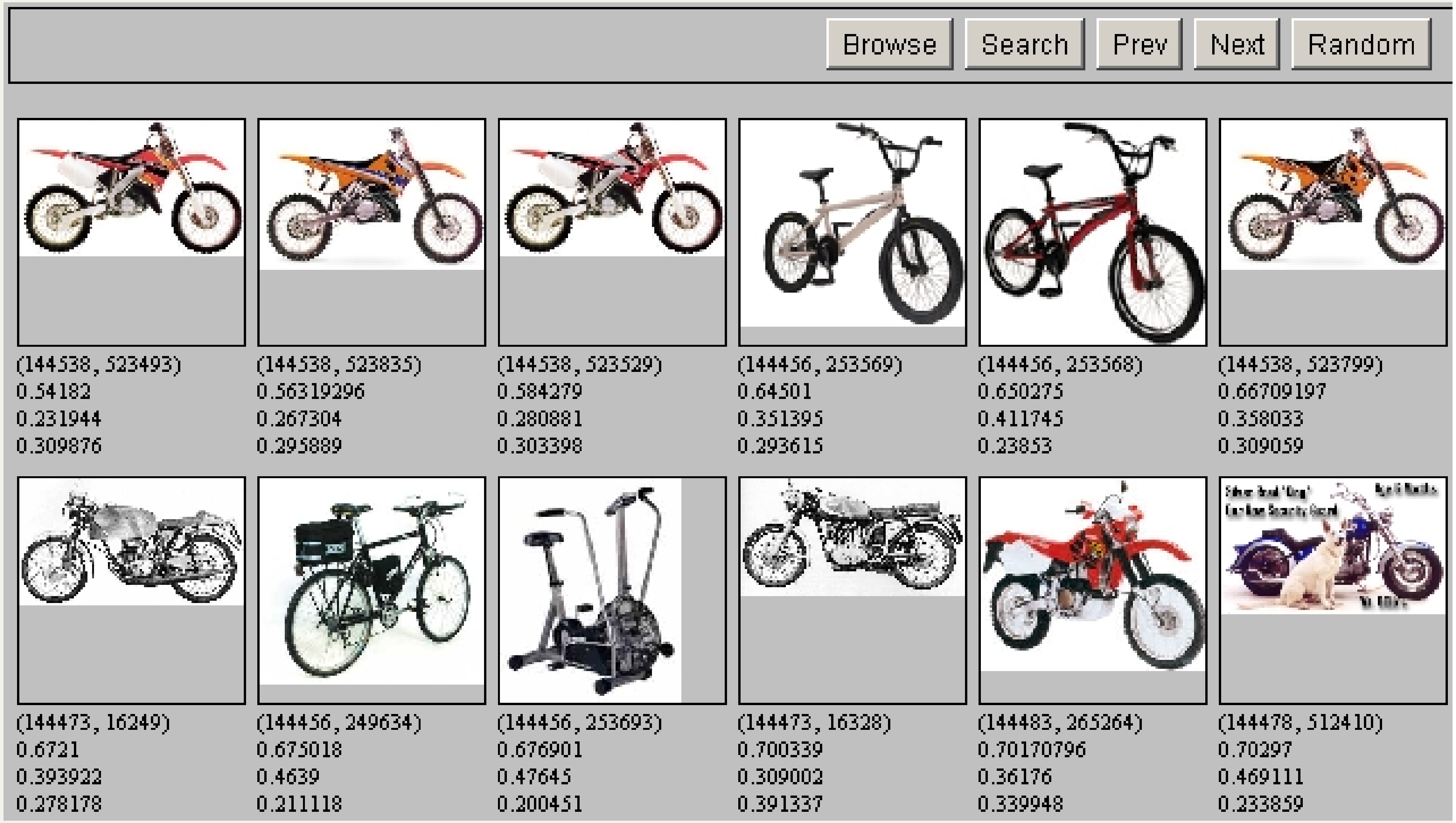
Results for Initial Query



Relevance Feedback

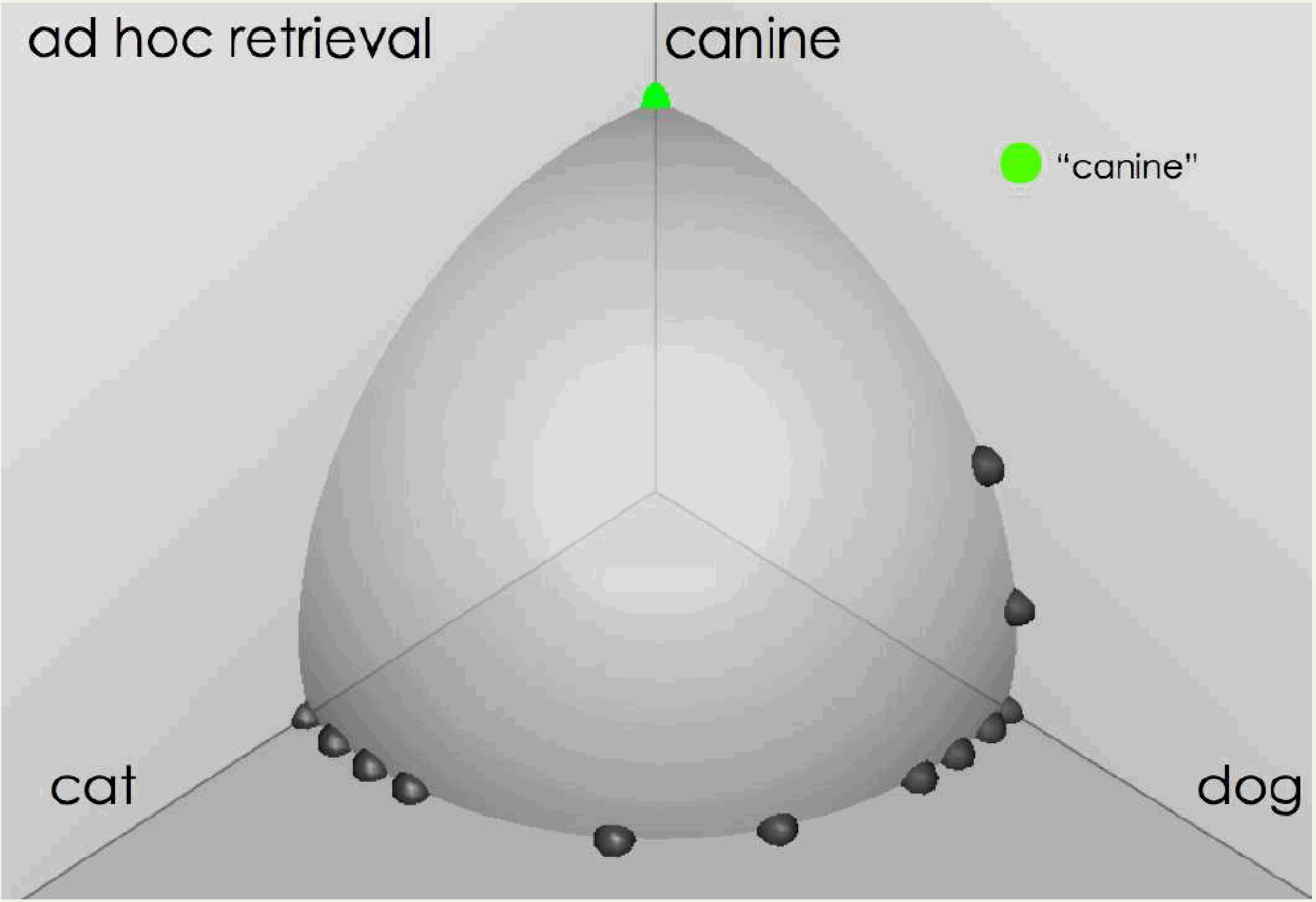


Results after Relevance Feedback



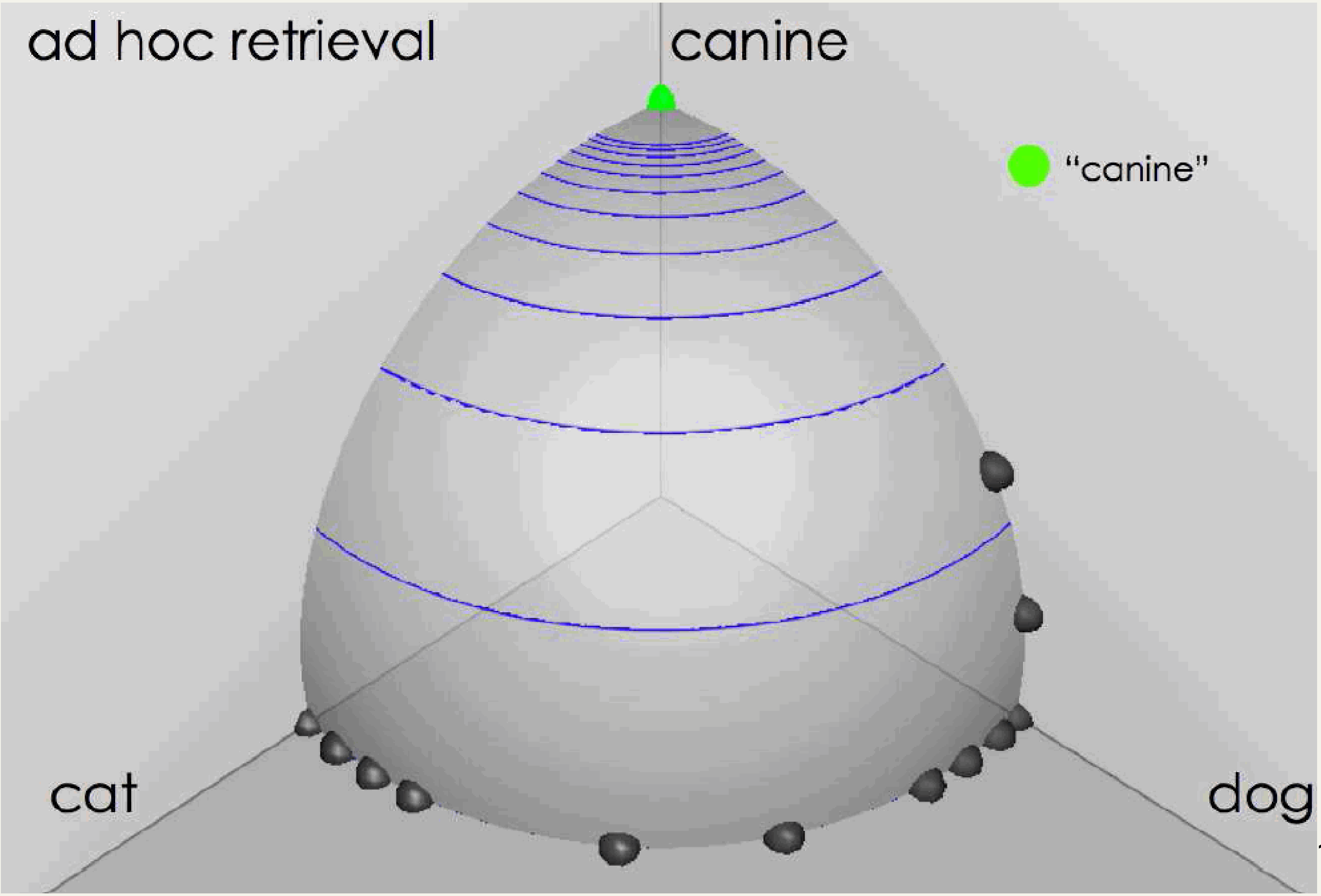
Ad hoc results for query *canine*

source: Fernando Diaz



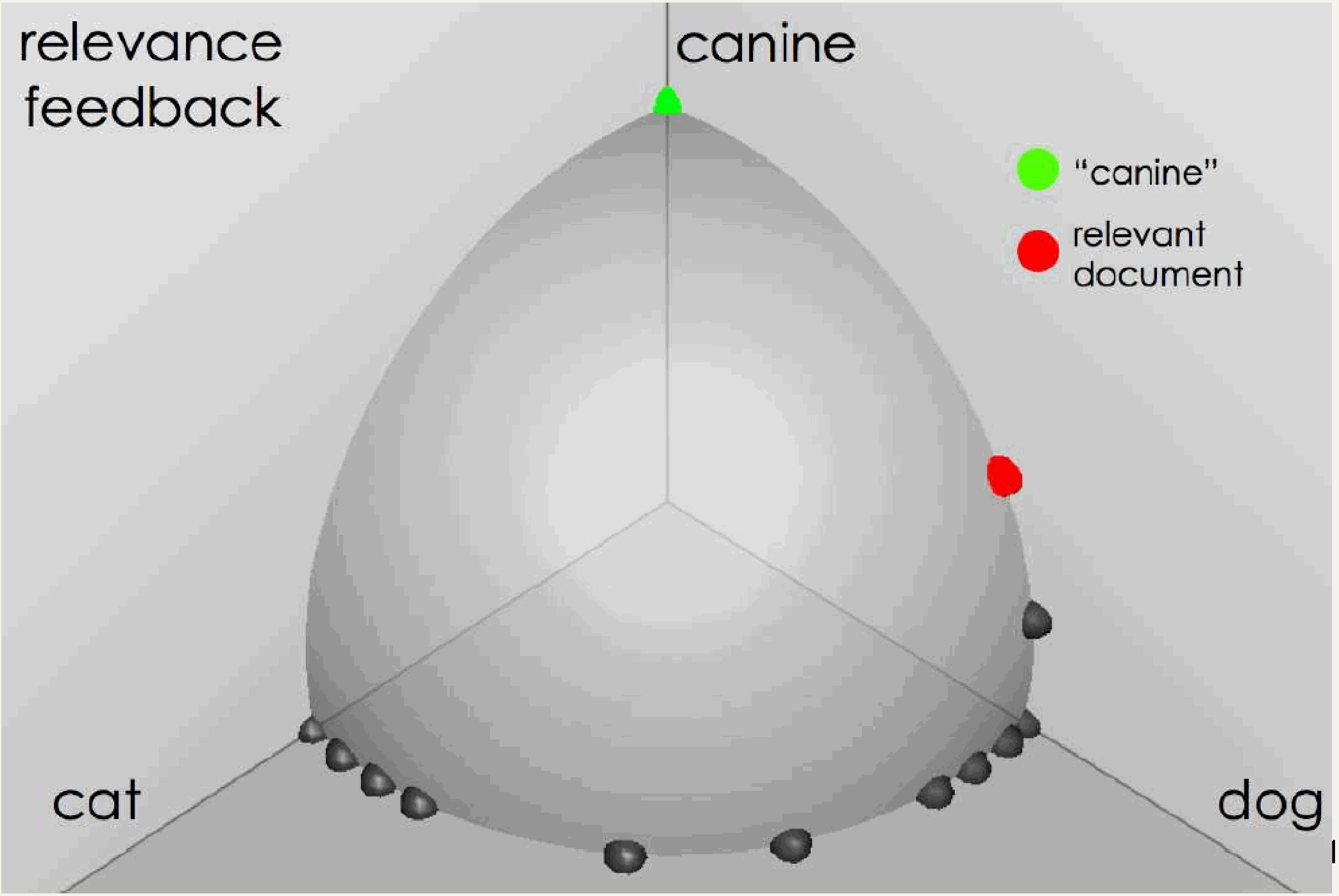
Ad hoc results for query *canine*

source: Fernando Diaz



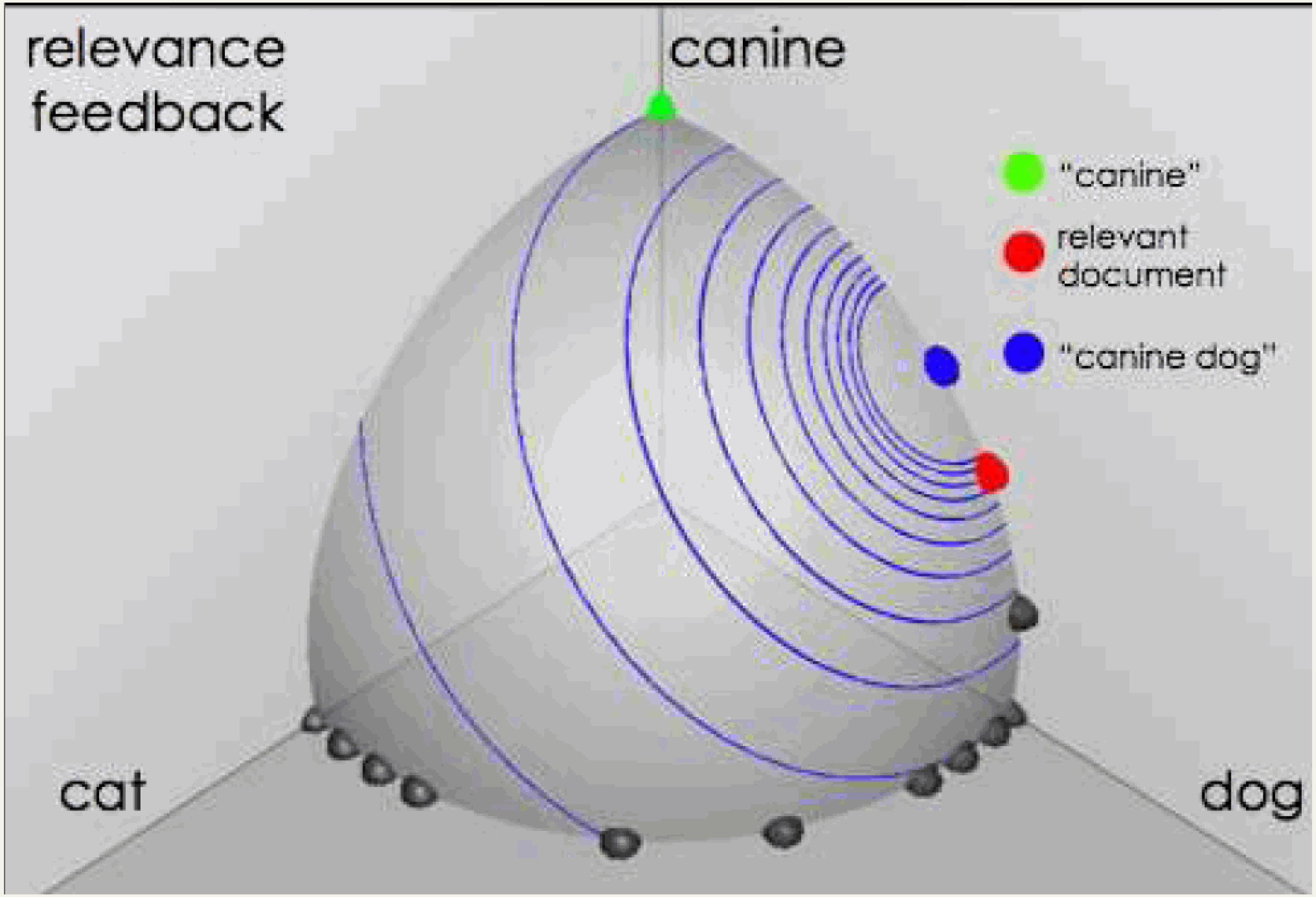
User feedback: Select what is relevant

source: Fernando Diaz



Results after relevance feedback

source: Fernando Diaz



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
- 30.816 satellite
- 5.991 nasa
- 4.196 launch
- 3.516 instrument
- 3.004 bundespost
- 2.790 rocket
- 2.003 broadcast
- 0.836 oil

15.106 space
5.660 application
5.196 eos
3.972 aster
3.446 arianespace
2.806 ss
2.053 scientist
1.172 earth
0.646 measure

Results for expanded query

2 1. 0.513, 07/09/91, NASA Scratches Environment Gear From Satellite Plan

1 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

8 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 $\vec{q}_{opt}$ that maximizes

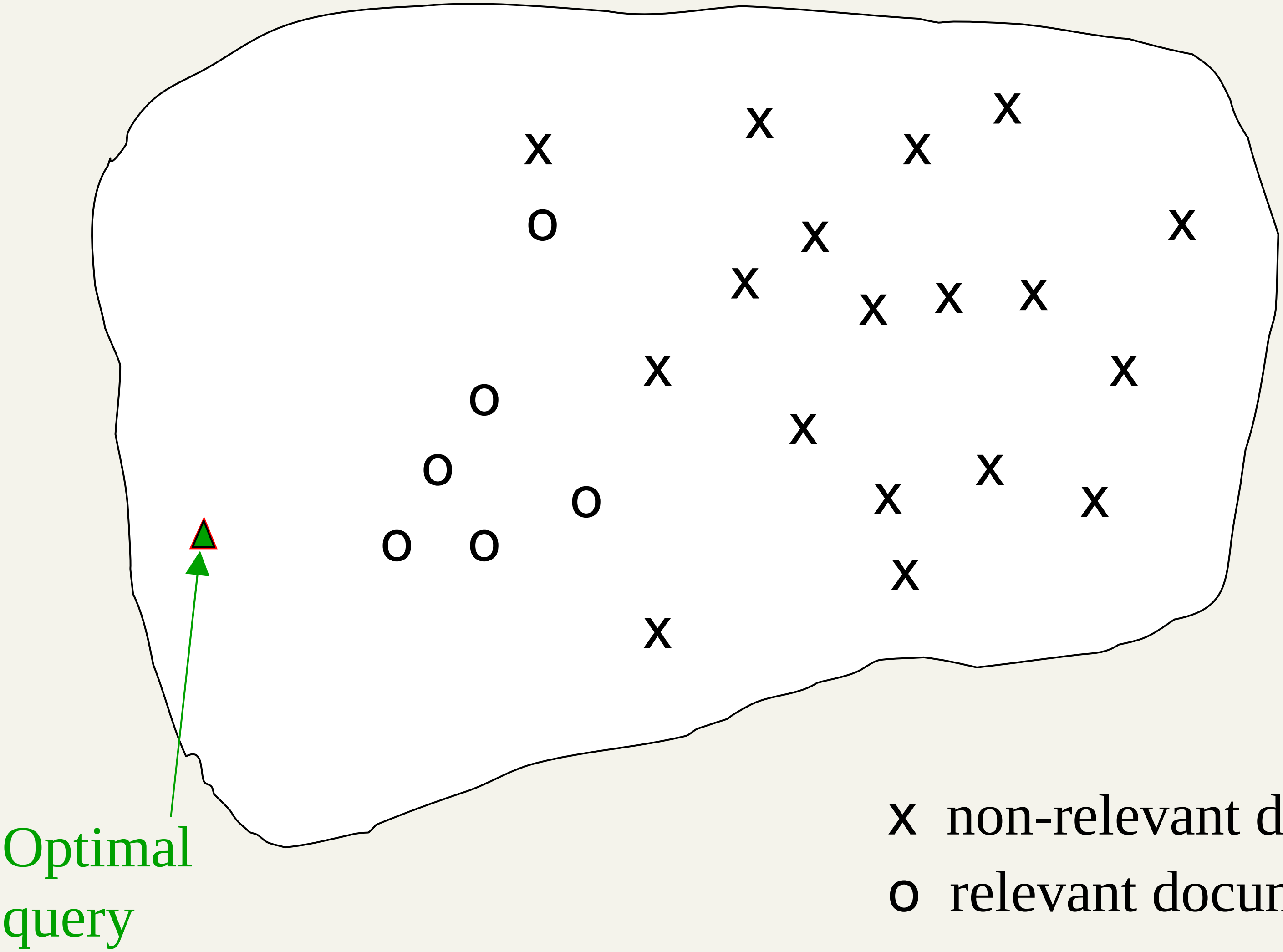
$$\vec{q}_{opt} = \arg\max_{\vec{q}} [\cos(\vec{q}, \vec{\mu}(C_r)) - \cos(\vec{q}, \vec{\mu}(C_{nr}))]$$

- 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



x non-relevant documents

o relevant documents

Rocchio Algorithm (SMART)

- Used in practice:

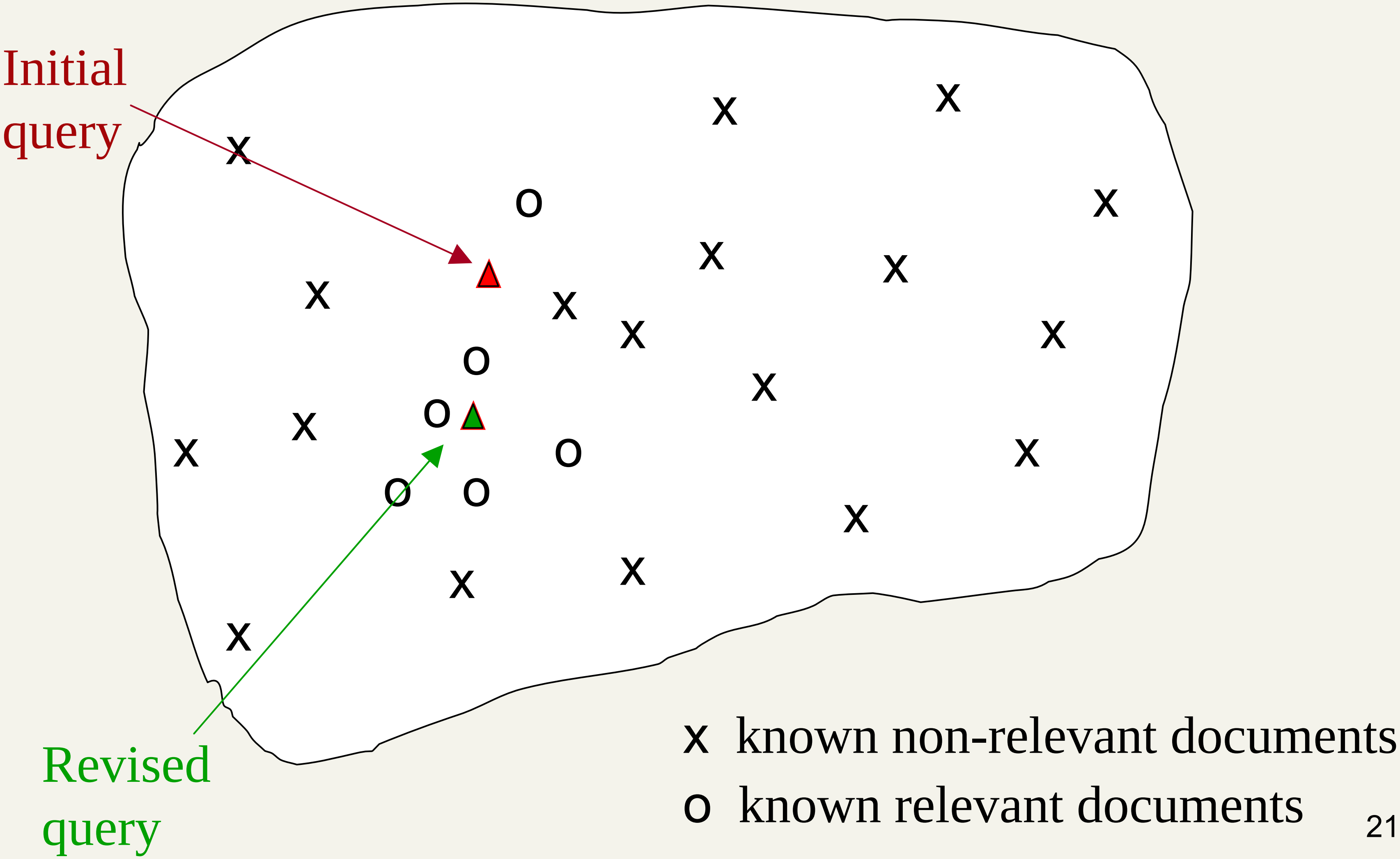
$$\vec{q}_m = \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j$$

- 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 (γ=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

Evaluation of relevance feedback strategies

- Use q_0 and compute precision-recall graph
- Use q_m and compute precision-recall graph
- Assess on all documents in the collection – not a good method
  - 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 for different relevance feedback algorithms

Evaluation of relevance feedback

- 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
- Empirically, one round of relevance feedback is often very useful. Two rounds is sometimes marginally useful.

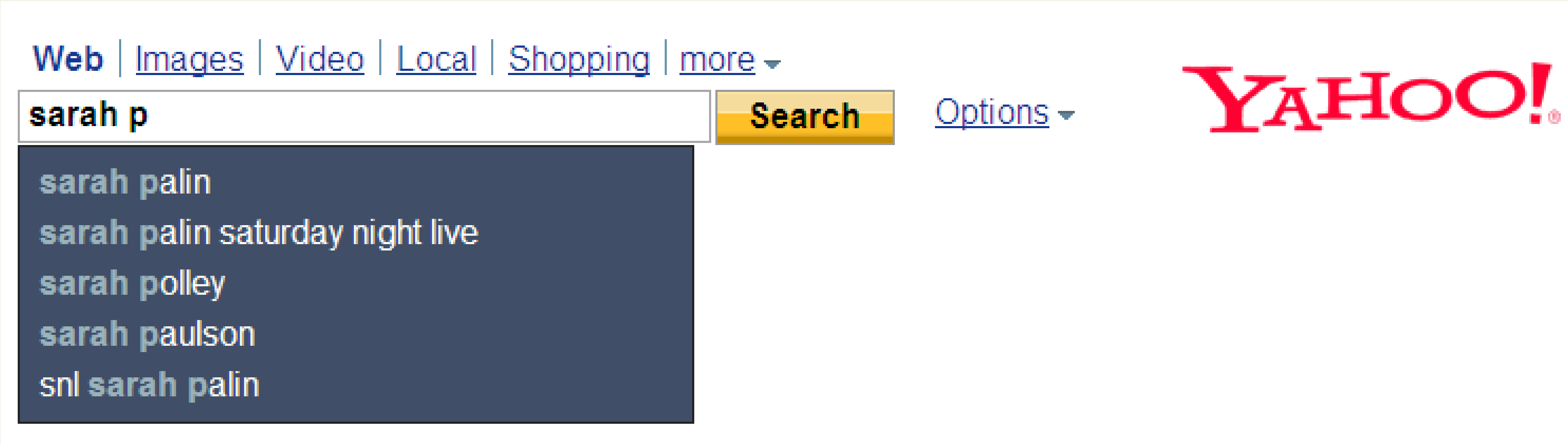
Pseudo relevance feedback

- 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.
- Several iterations can cause query drift.
- Why?

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



Would you expect such a feature to increase the query volume at a search engine?

How do we augment the user query?

- Manual thesaurus
  - E.g. MedLine: physician, syn: doc, doctor, MD, medico
  - Can be query rather than just synonyms
- 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

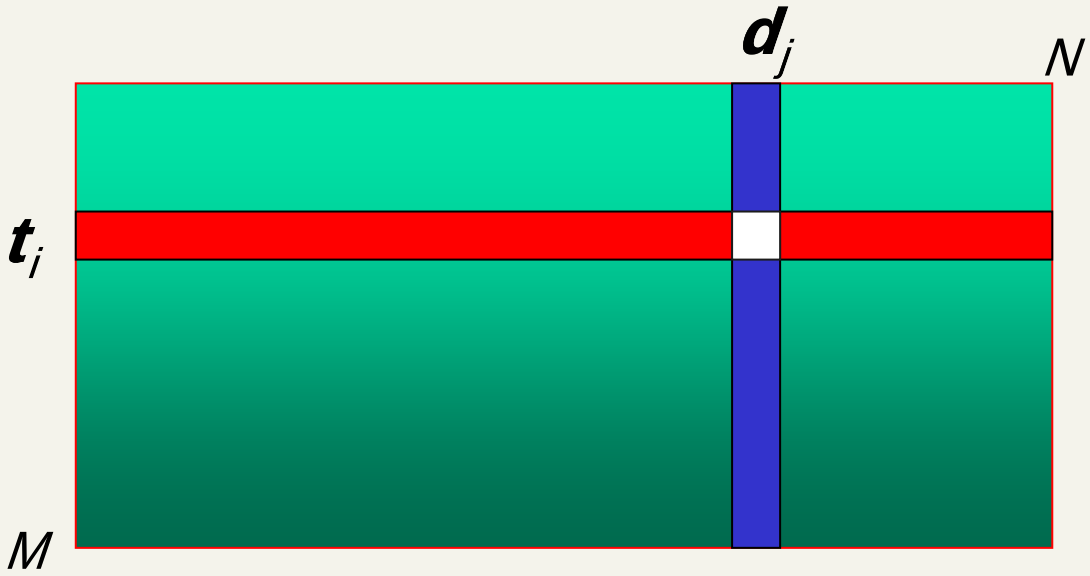
- For each term, *t*, in a query, expand the query with synonyms and related words of *t* from the thesaurus
  - feline → feline cat
- May weight added terms less than original query terms.
- Generally increases recall
- Widely used in many science/engineering fields
- May significantly decrease precision, 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.

Co-occurrence Thesaurus

- Simplest way to compute one is based on term-term similarities in $C = AA^T$ where A is term-document matrix.



- What does C contain if A is a term-doc incidence (0/1) matrix?
- For each t_i, pick terms with high values in C

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

Automatic Thesaurus Generation Discussion

- Quality of associations is usually a problem.
- Term ambiguity may introduce irrelevant statistically correlated terms.
  - "Apple computer" $\rightarrow$ "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.

Query assist

- Generally done by query log mining
- Recommend frequent recent queries that contain partial string typed by user
- A ranking problem! View each prior query as a doc – Rank-order those matching partial string …