Part 8: Relevance Feedback

Francesco Ricci

Most of these slides comes from the course:

Information Retrieval and Web Search,

Christopher Manning and Prabhakar Raghavan

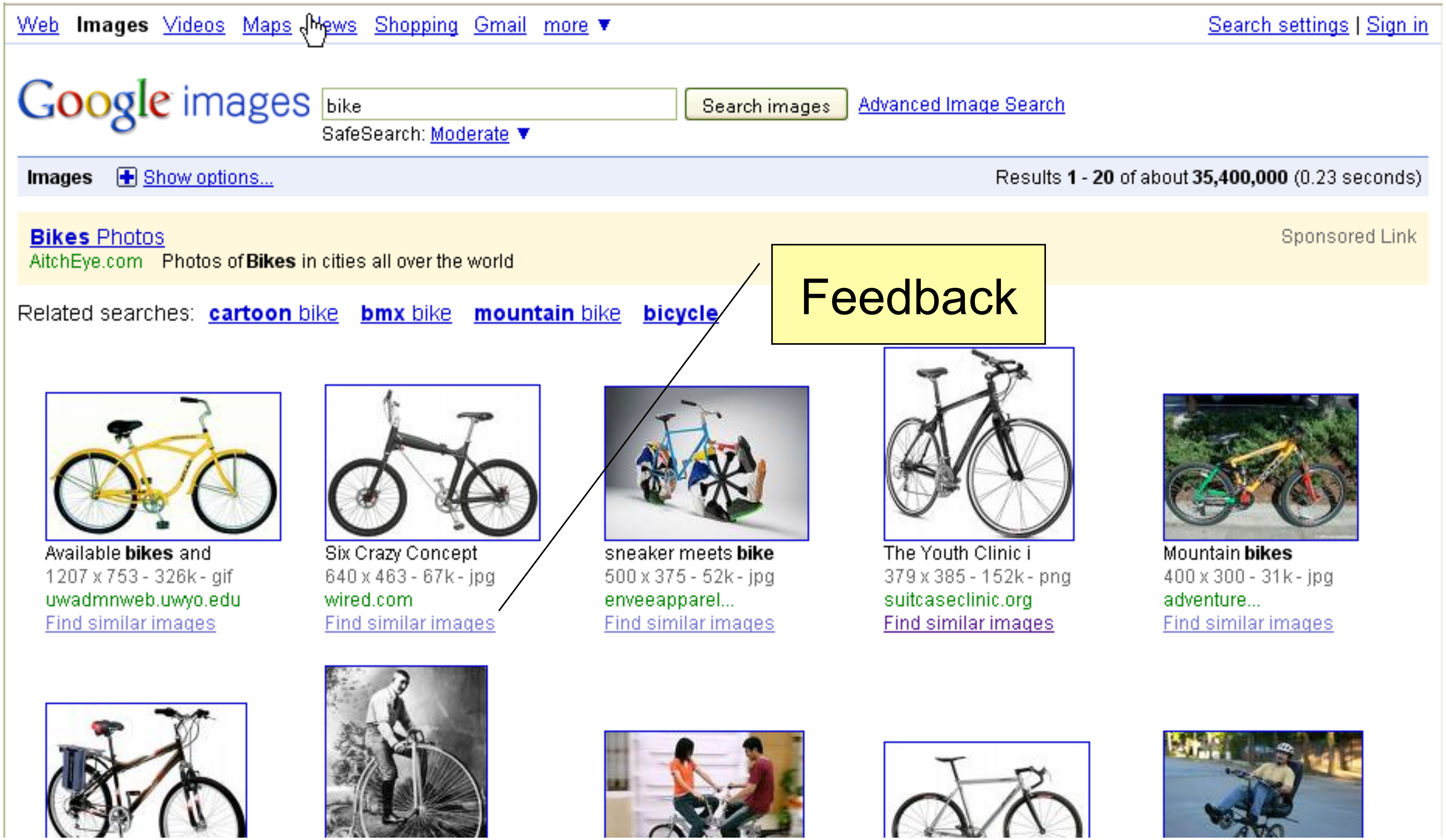
Content

- Local methods
  - Relevance feedback
  - Pseudo relevance feedback
- Global methods
  - Query expansion
    - Thesauri
    - Automatic thesaurus generation

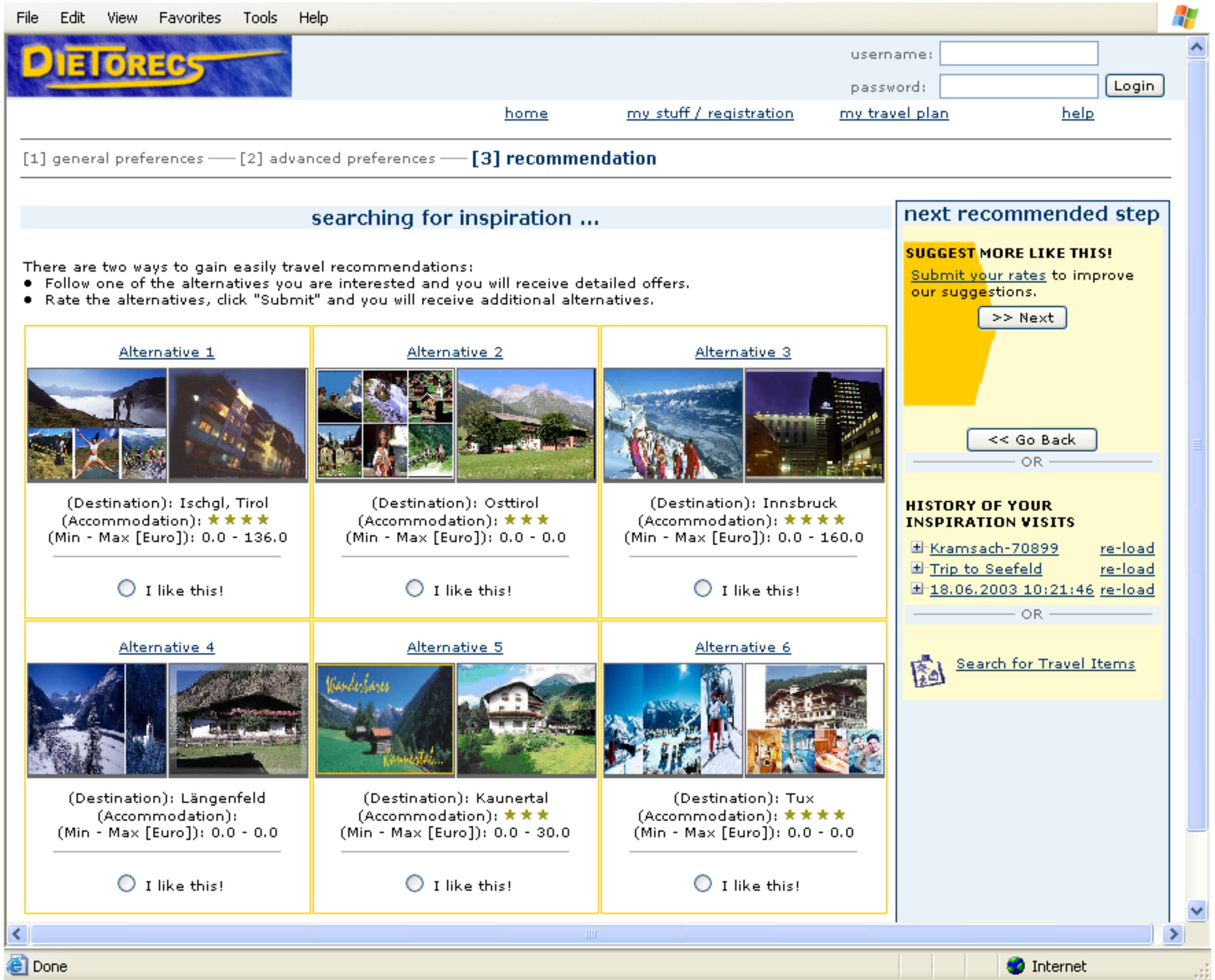
Relevance Feedback

- 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.

Example: search images



Preference-based: DieToRecs



[Ricci et al., 2006]

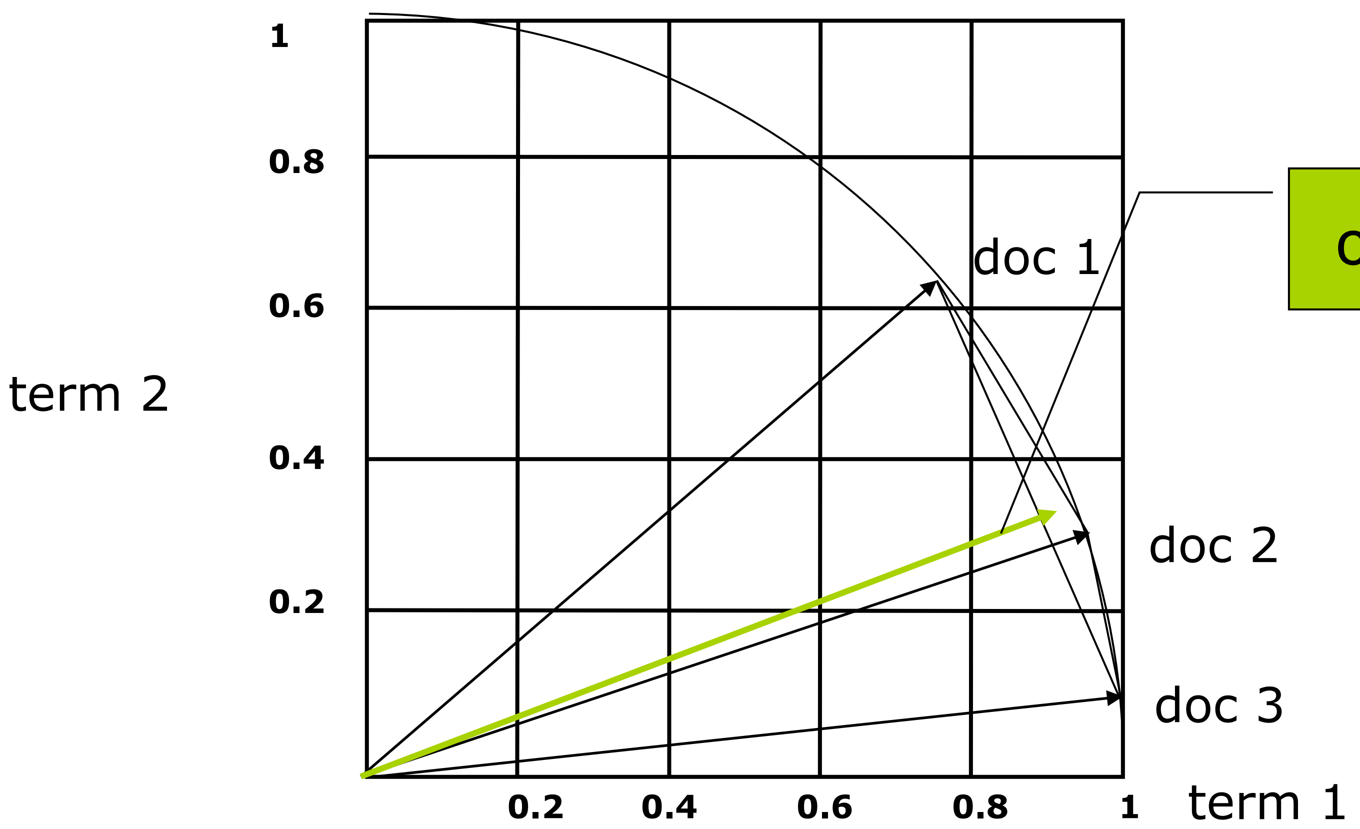
Key concept: Centroid

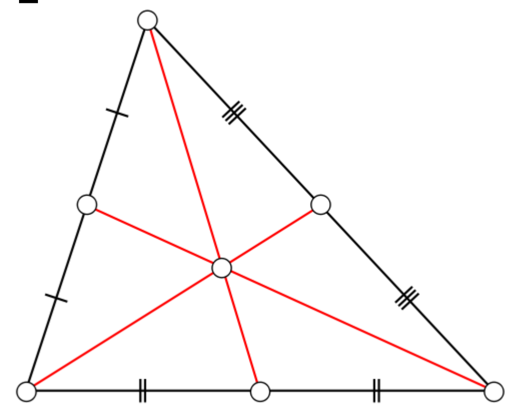
- The **centroid** is the center of mass of a set of points
- Recall that we represent documents as points in a high-dimensional space
- Definition: Centroid

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

where C is a set of documents.

Example





- The centroid is not normalized

Rocchio Algorithm

- The Rocchio algorithm uses the vector space model to pick a relevance feedback query
- Rocchio seeks the query $\vec{q}_{opt}$ that maximizes

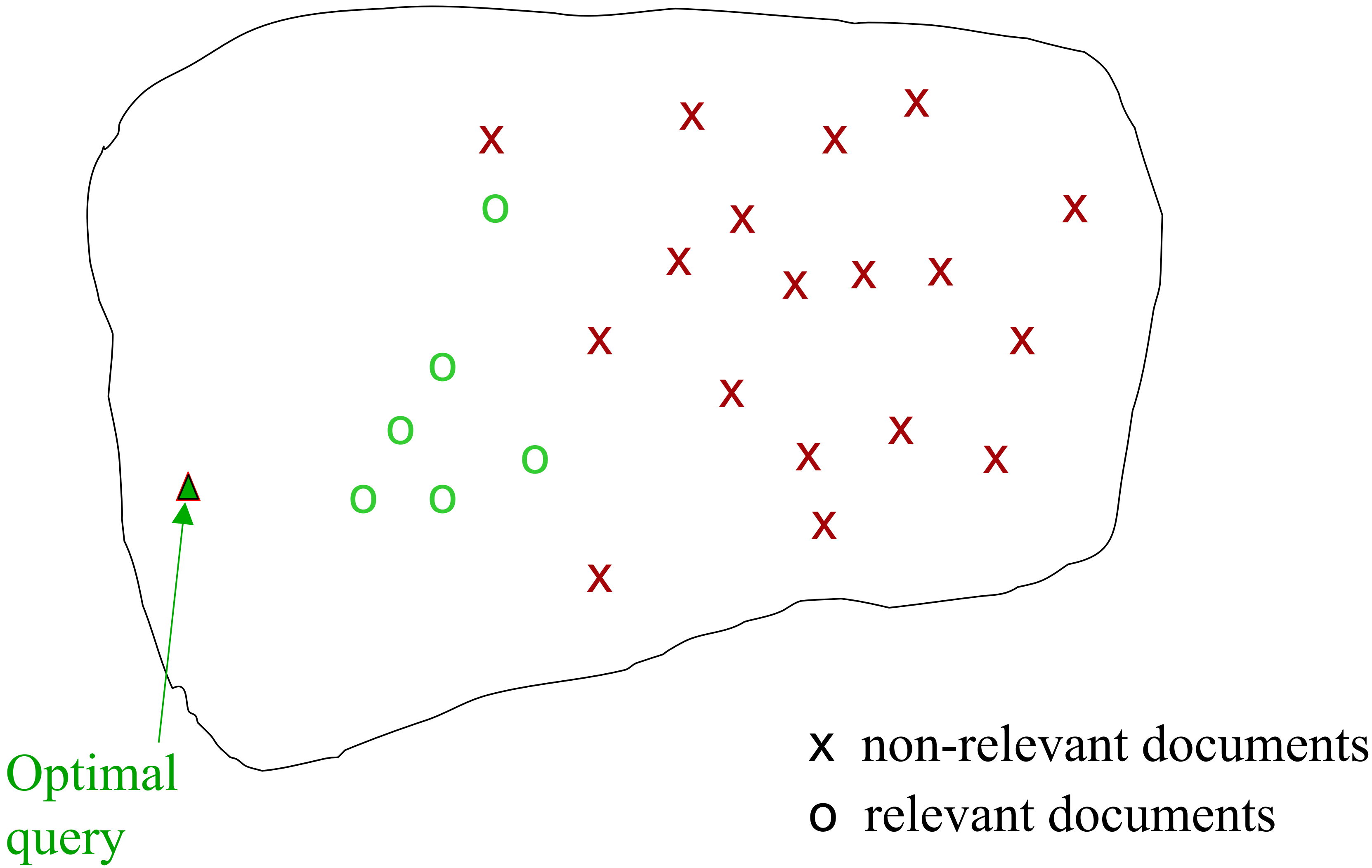
$$\vec{q}_{opt} = \arg\max_{\vec{q}} [\cos(\vec{q}, \vec{\mu}(C_r)) - \cos(\vec{q}, \vec{\mu}(C_{nr}))]$$

- Tries to separate docs marked relevant and non-relevant – the solution is:

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

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

The Theoretically Best Query



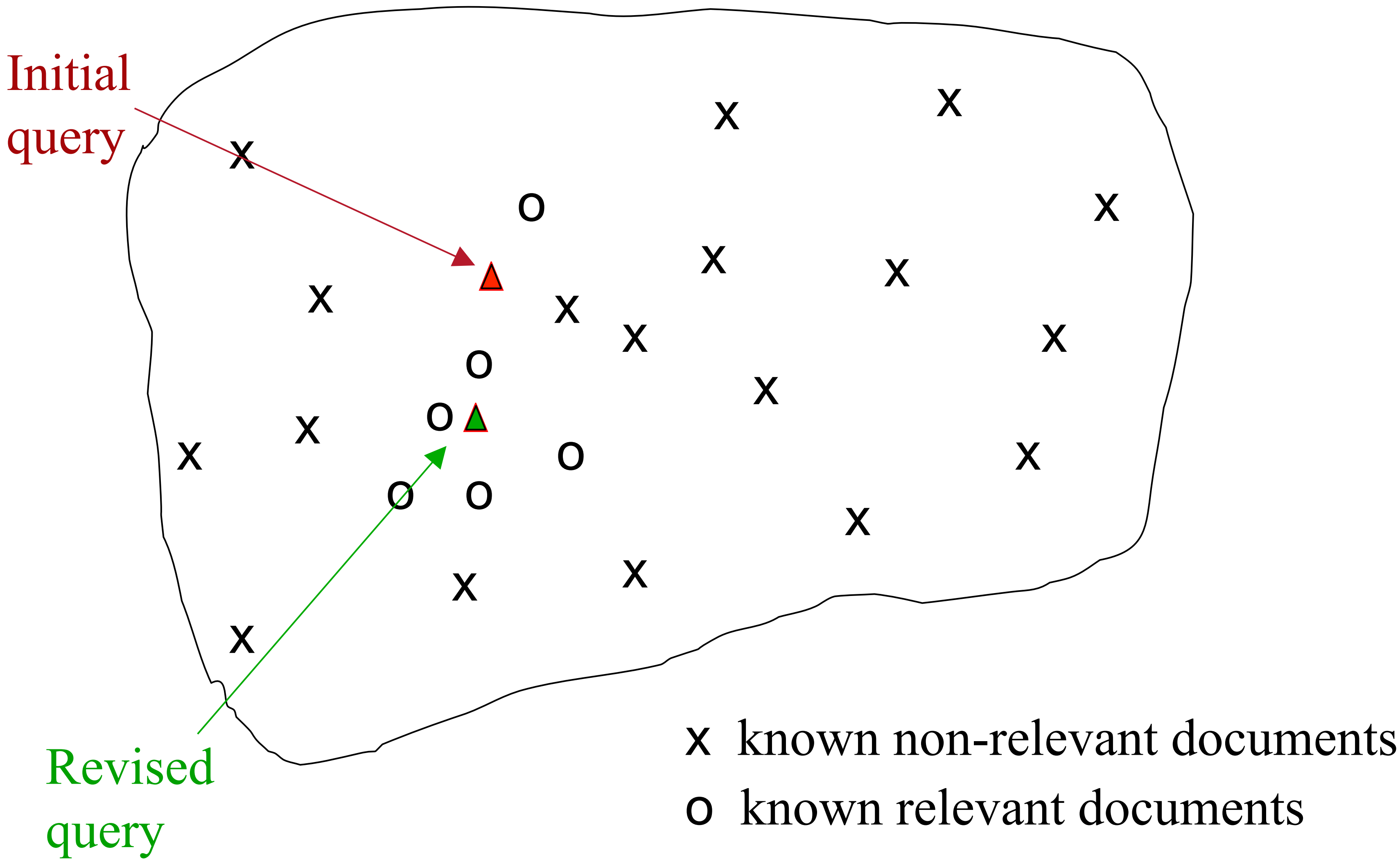
Rocchio 1971 Algorithm (SMART)

- 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
  - These are 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.

Relevance feedback on initial query



Subtleties to note

- Tradeoff α vs. β and γ: If we have a lot of judged documents, we want a higher β and γ
- Some weights in query vector can go negative:
  - Negative term weights are ignored (set to 0)
- **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 can improve recall and precision
- Relevance feedback is most useful for increasing *recall* in situations where recall is important – why?
- Users can be expected to review results and to take time to iterate – when recall is important.

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) – *there will be no relevant docs to positively feedback*
  - Cross-language information retrieval (hígado) – *documents in another language are not nearby*
  - Mismatch of searcher’s vocabulary vs. collection vocabulary
    - E.g.: searching "laptop" but in the docs is mentioned as "notebook" – no relevant docs are shown – no feedback is possible.

Violation of A2

- There are several relevance prototypes – in different clusters
- Examples:
  - Burma/Myanmar
  - Instances of a general concept, which often appear as a disjunction of more specific concepts, e.g.: felines
  - Pop stars that worked at Burger King – documents about the pop star and documents about Burger King that refer to the pop star.

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

Why?

- Users are often reluctant to provide explicit feedback
- It's often harder to understand why a particular document was retrieved after applying relevance feedback
- Information needs may change during the interaction (so what?).

Evaluation of relevance feedback strategies

- Use q_0 and compute precision and recall graph
- Use q_m and compute precision recall graph
  - 1) Assess on all documents in the collection
    - Spectacular improvements, but … it's cheating!
    - Known relevant documents ranked higher
    - Must evaluate with respect to documents not seen by user
  - 2) Use documents in residual collection (all docs 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 (i.e., split randomly the collection in two parts)
  - q_0 and user feedback from first collection
  - q_m run on second collection and measured.

Evaluation: Caveat

- True evaluation of usefulness must compare to other methods **taking the same amount of time** – or using similar user effort
- Alternative to relevance feedback: user revises and resubmits query
- Users may prefer revision/resubmission to having to judge relevance of documents
- There is no clear evidence that relevance feedback is the "best use" of the user's time.

Relevance Feedback on the Web

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

α/β/γ ??

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

Excite Relevance Feedback

Spink et al. 2000

- Only about 4% of query sessions from a user used relevance feedback option
  - Expressed as “More like this” link next to each result
- But about 70% of users only looked at first page of results and didn’t pursue things further
  - So 4% is about 1/8 of people extending search
- For people who used relevance feedback: improved results about 2/3 of the time.

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: e.g. if the top results of a query are all about a subtopic
- Several iterations can cause query drift
- Why?

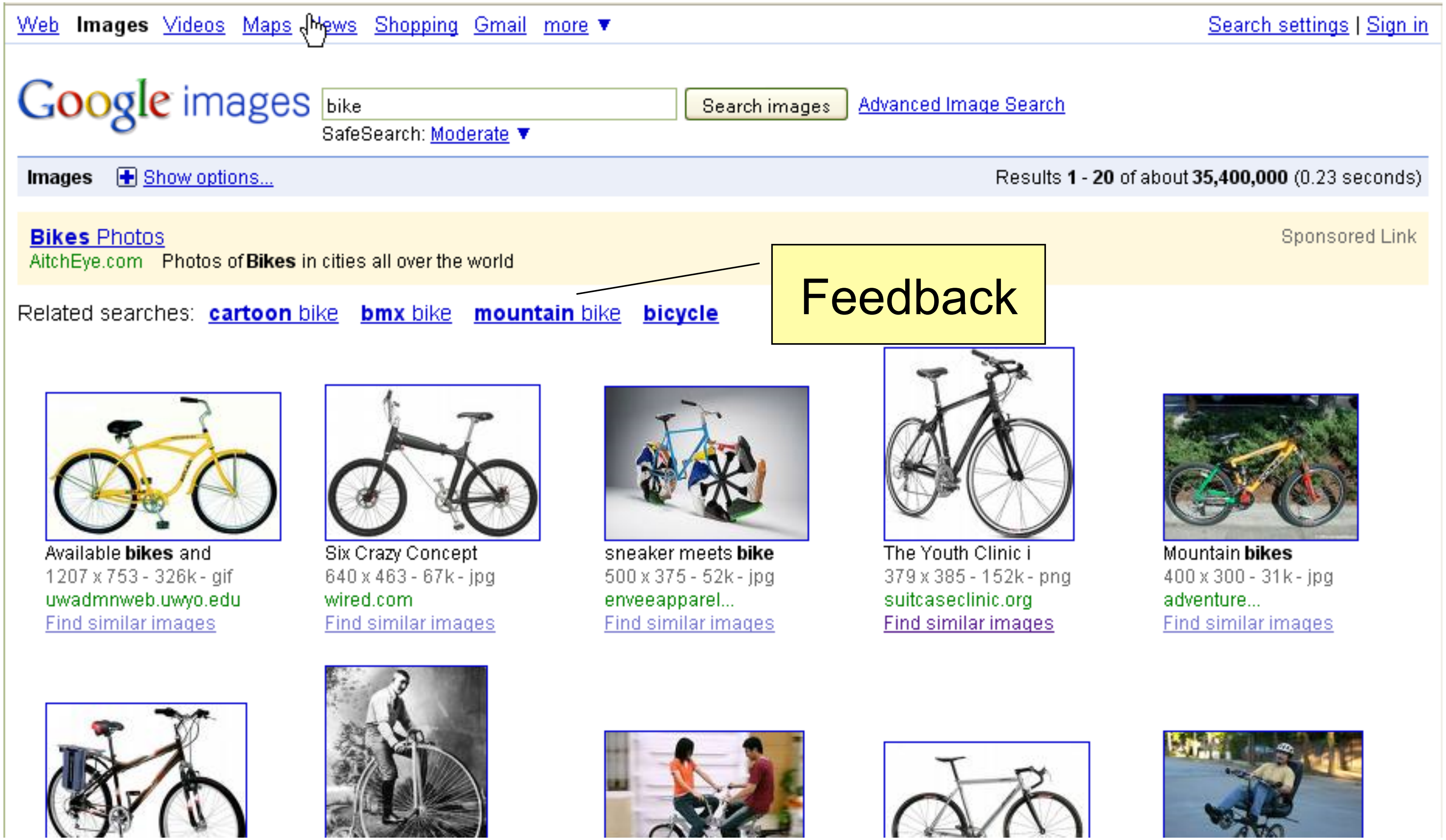
Indirect relevance feedback

- 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.

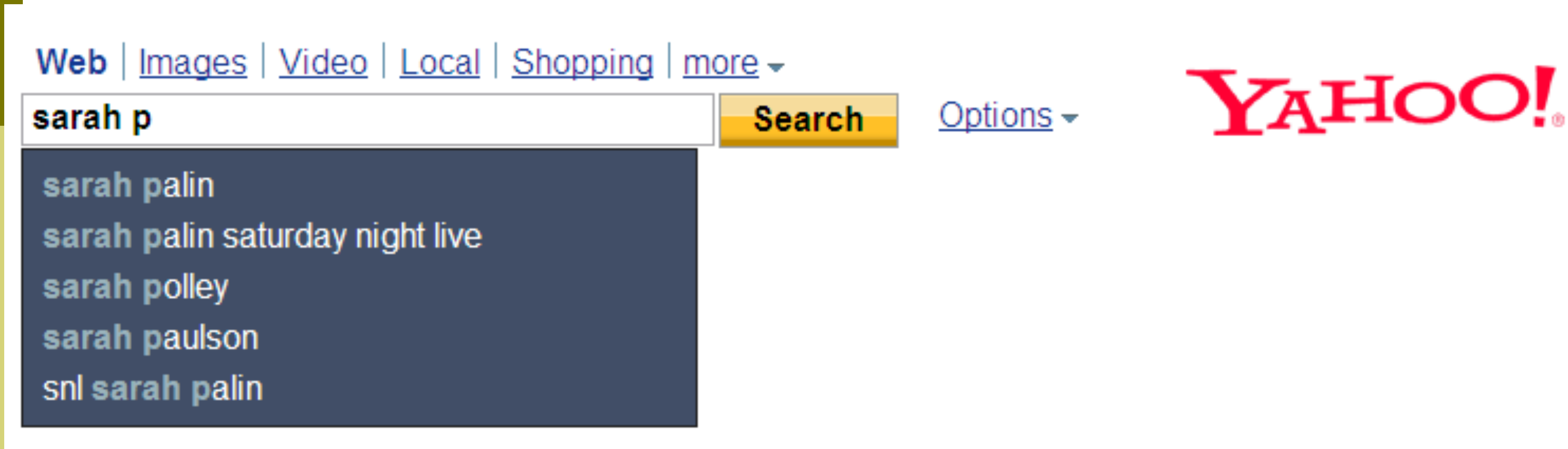
Query Expansion

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

Example: search images



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 related queries rather than just synonyms
- **Global Analysis:** static; based on 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.

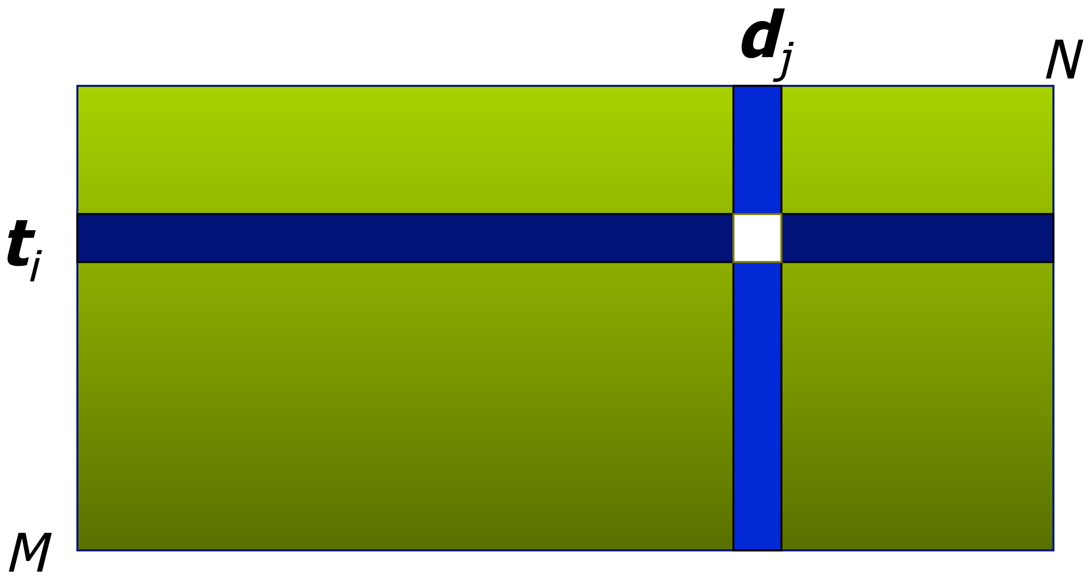
Automatic Thesaurus Generation

- 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 often co-occur
- 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 based is more robust, grammatical relations are more accurate.

Why?

Co-occurrence Thesaurus

- Simplest way to compute one is based on term-term similarities in $C = AA^T$ where A is term-document matrix.
- $w_{i,j}$ = (row normalized) weight for (t_i ,$\boldsymbol{d}_j$)



What does C contain if A is a term-doc incidence (0/1) matrix?

- For each t_i, pick terms t_j with high C_{ij} values

Automatic Thesaurus Generation Example

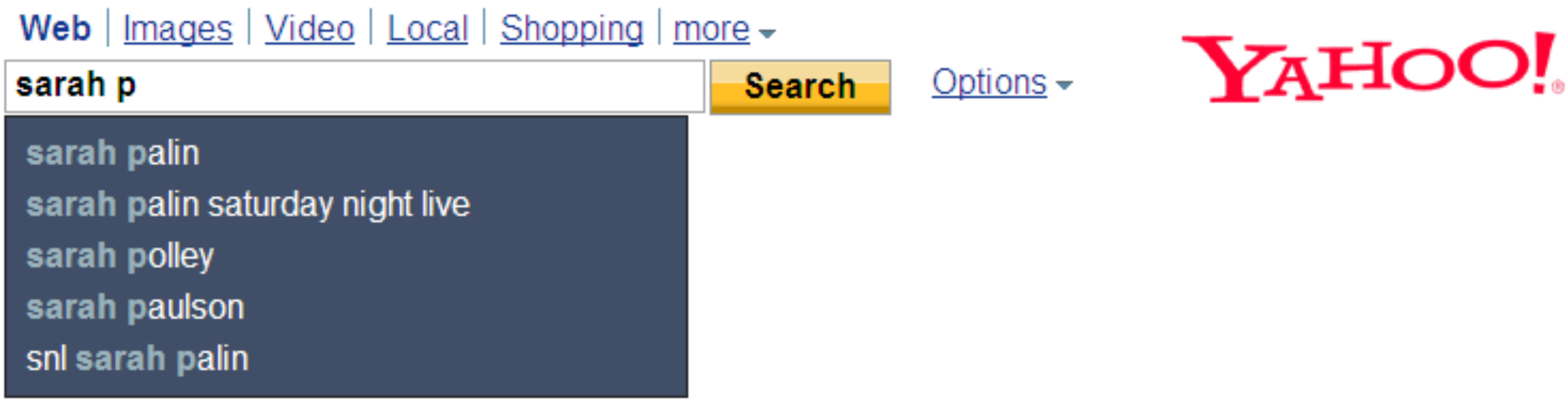
word	ten nearest neighbors
absolutely	absurd whatsoever totally exactly nothing
bottomed	dip copper drops topped slide trimmed slig
captivating	shimmer stunningly superbly plucky witty
doghouse	dog porch crawling beside downstairs gazec
Makeup	repellent lotion glossy sunscreen Skin gel p
mediating	reconciliation 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.

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 …



Resources

IIR Ch 9