Information Retrieval

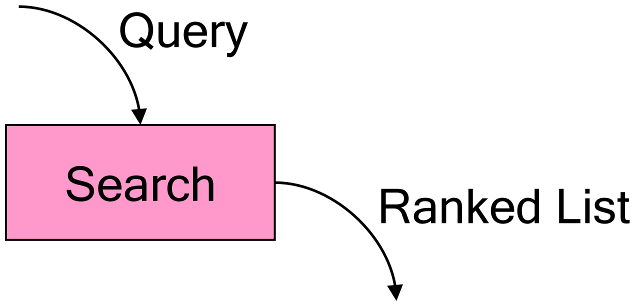
Models IV
Query Reformulation and Relevance Feedback

Qianni Zhang

Roadmap of this lecture

- Queries and information needs
- Query reformulation (refinement)
  - Relevance feedback
  - Local analysis (also called pseudo-relevance feedback)
  - Global analysis
  - Query expansion

The IR Black Box

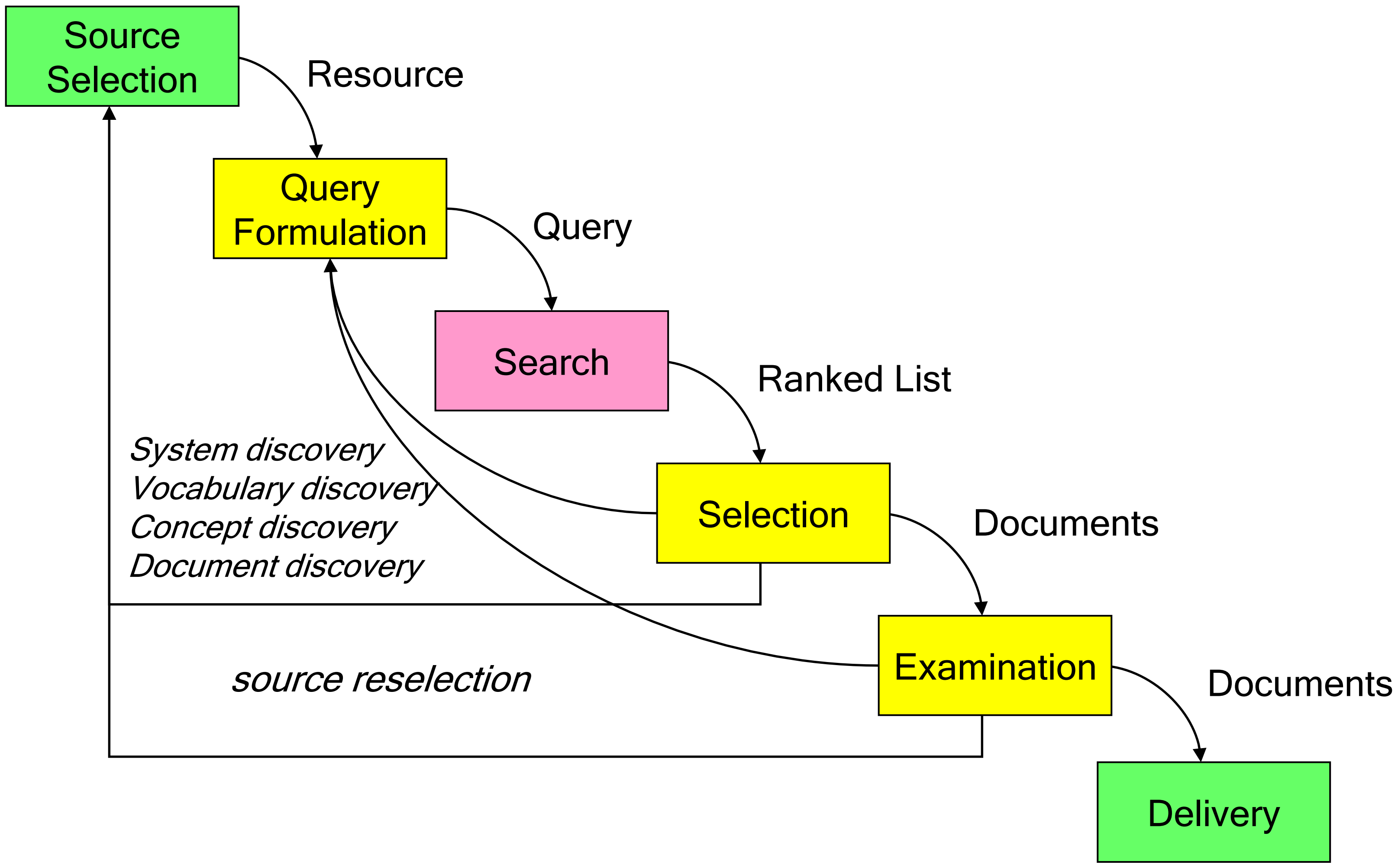


Anomalous State of Knowledge

- Basic paradox:
  - Information needs arise because the user doesn't know something: "an anomaly in his state of knowledge with respect to the problem faced"
  - Search systems are designed to satisfy these needs, but the user needs to know what he is looking for
  - However, if the user knows what he's looking for, there may not be a need to search in the first place
- Implication: computing "similarity" between queries and documents is fundamentally wrong
- How do we resolve this paradox?

Nicholas J. Belkin. (1980) Anomalous States of Knowledge as a Basis for Information Retrieval. *Canadian Journal of Information Science*, 5, 133-143.

The Information Retrieval Cycle



Different Types of Interactions

- System discovery – learning capabilities of the system
  - Playing with different types of query operators
  - “Reverse engineering” a search system

- Vocabulary discovery – learning collection-specific terms that relate to your information need
  - The literature on aerodynamics refers to *aircrafts*, but you query on *planes*
  - How do you know what terms the collection uses? Mismatch problem

Different Types of Interactions

- Concept discovery – learning the concepts that relate to your information need
  - What’s the name of the disease that Reagan had?

- Document discovery – learning about the types of documents that fulfill your information need
  - Were you looking for a news article, a column, a report,
  - or an editorial?

Introduction

Information Needs

- An information need is the underlying cause of the query that a person submits to a search engine
  - sometimes called information problem to emphasise that information need is generally related to a task
- Categorised using variety of dimensions
  - e.g., number of relevant documents being sought
  - type of information that is needed
  - type of task that led to the requirement for information
  - resources available

Introduction

Queries

- A query can represent very different information needs
  - May require different search techniques and ranking algorithms to produce the best rankings
- A query can be a poor representation of the information need
  - User may find it difficult to express the information need
  - User is encouraged to enter short queries both by the search engine interface, and by the fact that long queries don’t work

Introduction

Query Reformulation (Refinement)

- The idea is when documents are initially retrieved:
  - They should be examined for relevance information
  - Next we can improve the query to retrieve additional relevant documents
- Query reformulation:
  - Expanding original query with new terms
  - Re-weighing the terms in the (expanded) query

Introduction

Some Approaches to Query Reformulation

- Relevance feedback
  - Approaches based on feedback from users
  - E.g. Rocchio, Binary Independence Model
- Local analysis (also called pseudo-relevance feedback)
  - Approaches based on information derived from the set of initially retrieved documents (local set of documents)
- Global analysis
  - Approaches based on global information derived from the document collection or the web logs

Relevance Feedback

- Take advantage of user relevance judgments in the retrieval process:
  - User issues a (short, simple) query and gets back an initial hit list
  - User marks hits as relevant or non-relevant
  - The system computes a better representation of the information need based on this feedback
  - Single or multiple iterations (although little is typically gained after one iteration)
- Idea:
  - you may not know what you’re looking for, but you’ll know when you see it

Relevance Feedback: Outline

- Explicit feedback: users explicitly mark relevant and irrelevant documents
- Implicit feedback: system attempts to infer user intentions based on observable behavior
- Blind feedback: feedback in absence of any evidence, explicit or otherwise

Why relevance feedback?

- You may not know what you’re looking for, but you’ll know when you see it
- Query formulation may be difficult; simplify the problem through iteration
- Facilitate vocabulary and concept discovery
- Boost recall: “find me more documents like this…”

Relevance Feedback

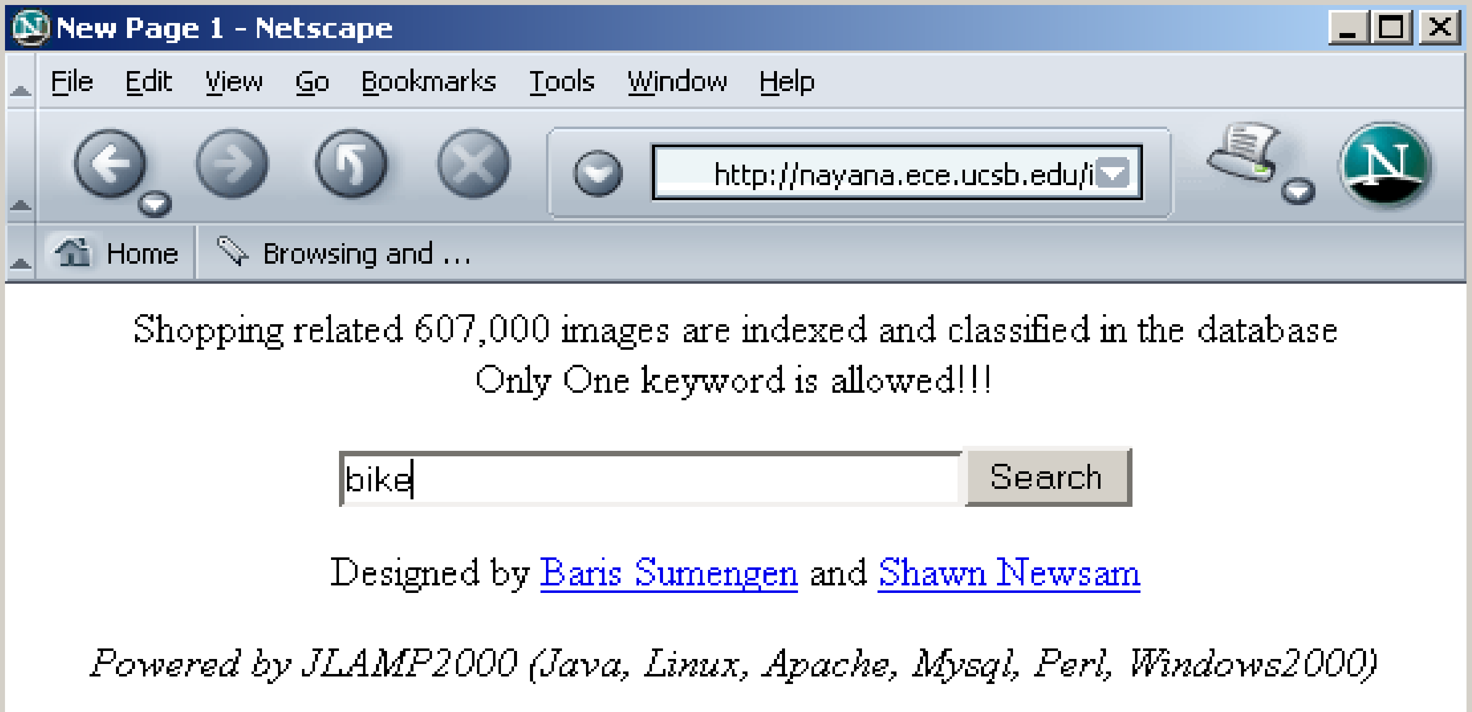
Explicit Feedback Through Clicks

- Web search engine users not only inspect the answers to their queries, they also click on them
- The clicks reflect preferences for particular answers in the context of a given query
  - They can be collected in large numbers without interfering with the user actions
  - The immediate question is whether they also reflect relevance judgments on the answers

Relevance Feedback

Example 1

Image Search Engine
http://nayana.ece.ucsb.edu/imsearch/imsearch.html



New Page 1 - Netscape
File Edit View Go Bookmarks Tools Window Help
http://nayana.ece.ucsb.edu/i
Home
Browsing and ...
Search

Shopping related 607,000 images are indexed and classified in the database
Only One keyword is allowed!!!

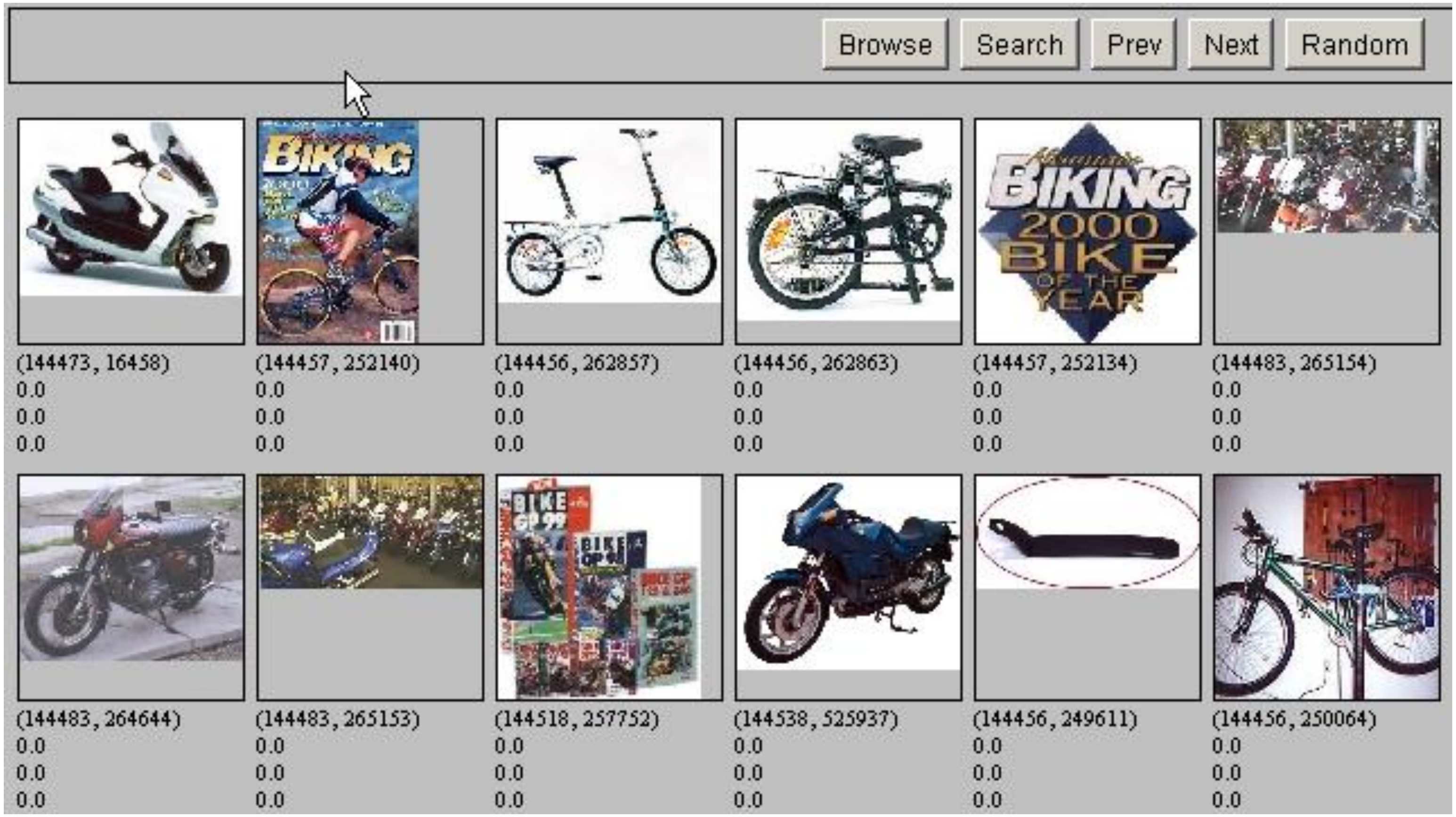
bike

Designed by Baris Sumengen and Shawn Newsam

Powered by JLAMP2000 (Java, Linux, Apache, Mysql, Perl, Windows2000)

Relevance Feedback

Example 1 - initial results

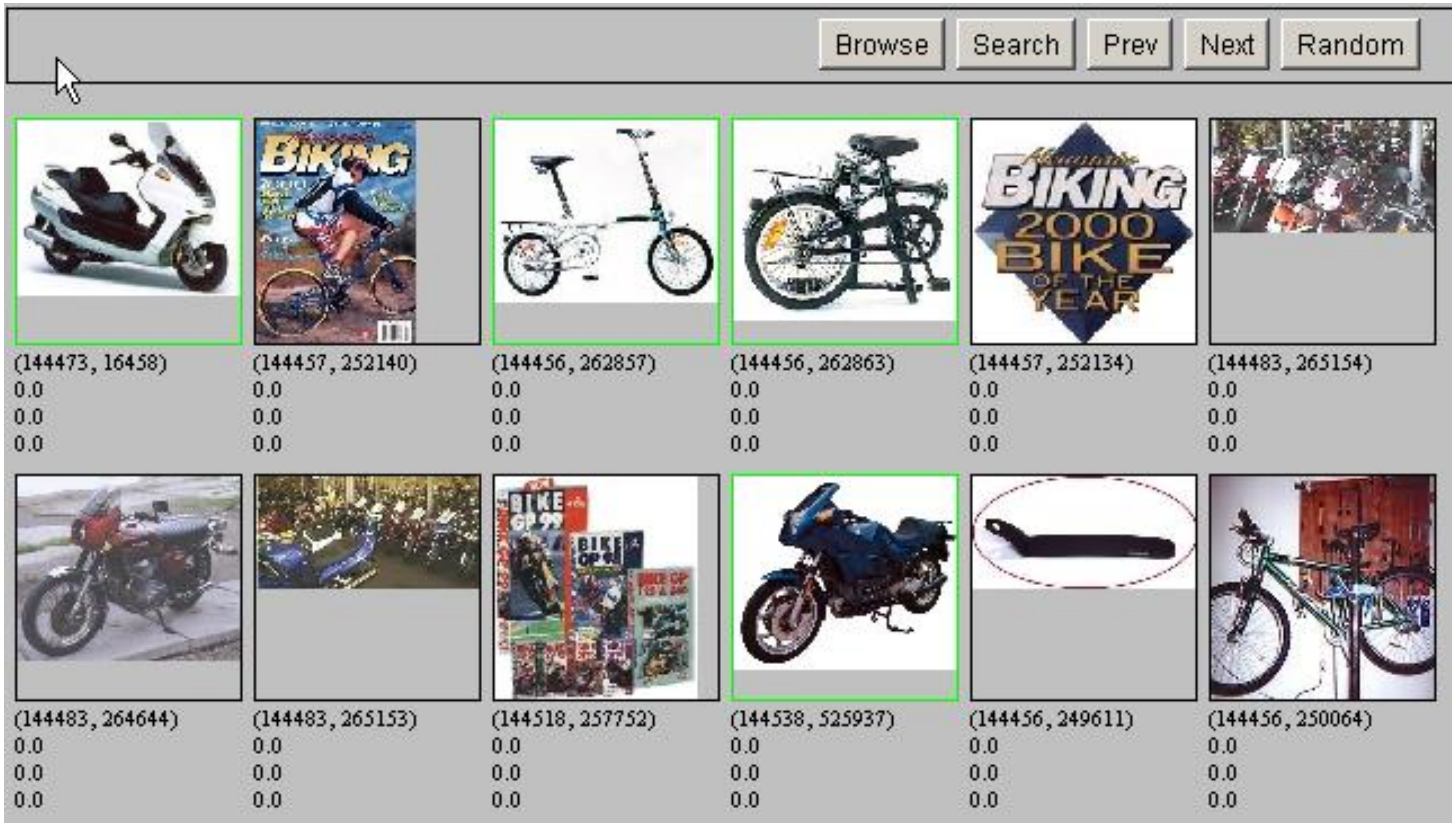


```
(144473, 16458)   (144457, 252140)   (144456, 262857)   (144456, 262863)   (144457, 252134)   (144483, 265154)
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0

(144483, 264644)  (144483, 265153)   (144518, 257752)   (144538, 525937)   (144456, 249611)   (144456, 250064)
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0
```

Relevance Feedback

Example 1 - user feedback

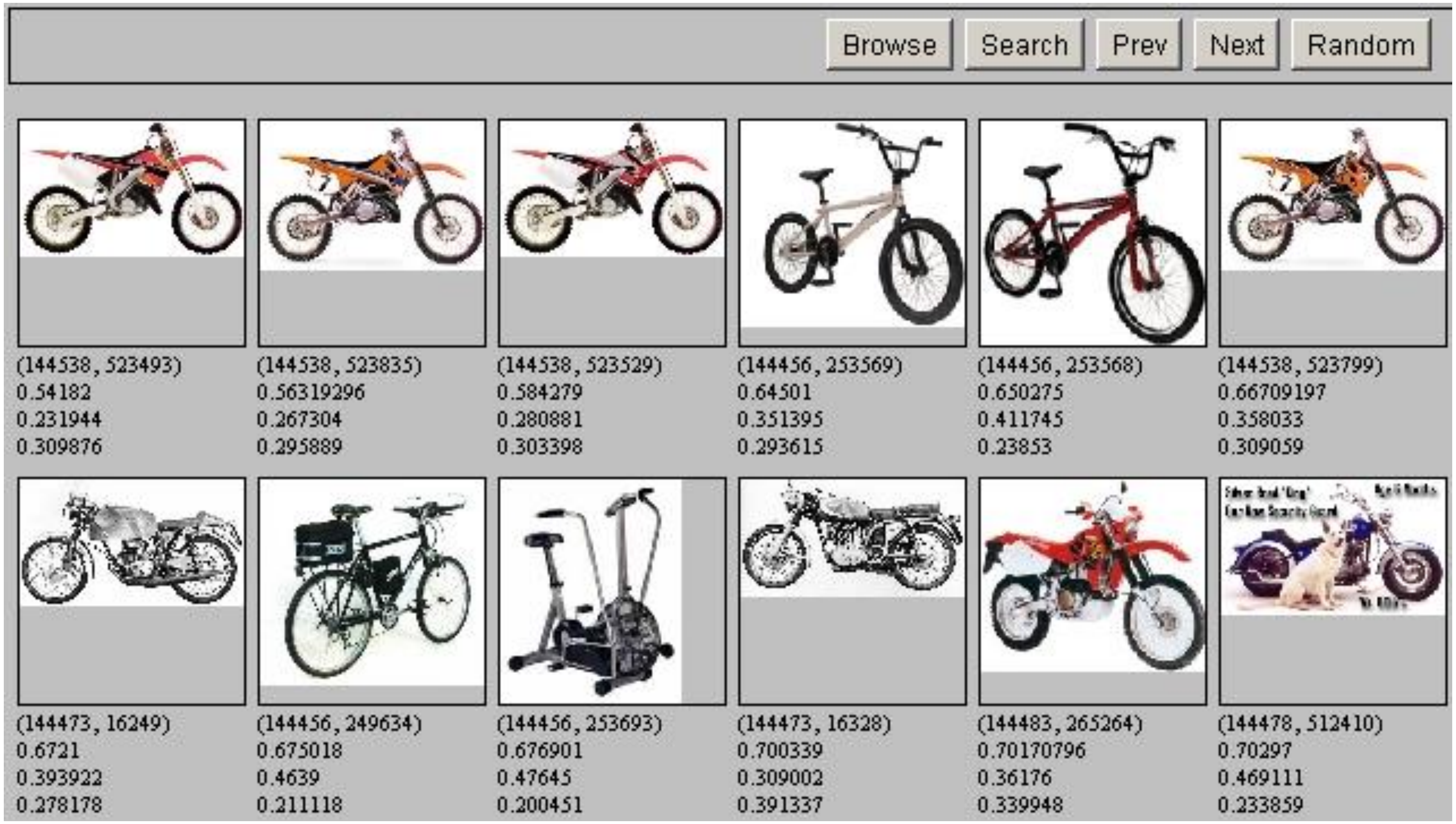


```
(144473, 16458)   (144457, 252140)   (144456, 262857)   (144456, 262863)   (144457, 252134)   (144483, 265154)
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0

(144483, 264644)  (144483, 265153)   (144518, 257752)   (144538, 525937)   (144456, 249611)   (144456, 250064)
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0
0.0               0.0                0.0                0.0                0.0                0.0
```

Relevance Feedback

Example 1 – revised results



(144538, 523493)
0.54182
0.231944
0.309876

(144538, 523835)
0.56319296
0.267304
0.295889

(144538, 523529)
0.584279
0.280881
0.303398

(144456, 253569)
0.64501
0.351395
0.293615

(144456, 253568)
0.650275
0.411745
0.23853

(144538, 523799)
0.66709197
0.358033
0.309059

(144473, 16249)
0.6721
0.393922
0.278178

(144456, 249634)
0.675018
0.4639
0.211118

(144456, 253693)
0.676901
0.47645
0.200451

(144473, 16328)
0.700339
0.309002
0.391337

(144483, 265264)
0.70170796
0.36176
0.339948

(144478, 512410)
0.70297
0.469111
0.233859

Relevance Feedback

Example 2

- 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

Relevance Feedback

Relevance feedback can be viewed as an iterative cycle:

1. User presented with a list of retrieved documents
2. User marks those which are relevant (or not relevant)
   - In practice: top 10-20 ranked documents are examined
   - Incremental: one document after the other
3. The relevance feedback algorithm selects important terms from documents assessed relevant by users
4. The relevance feedback algorithm emphasises the importance of these terms in a new query in the following ways:
   - Query expansion: add these terms to the query
   - Term re-weighing: modify the term weights in the query
   - Query expansion + term re-weighing
5. The updated query is submitted to the system
6. If the user is satisfied with the new set of retrieved documents, then the relevance feedback process stops, otherwise go to step 2

Updating Queries

- Let’s assume that there is an optimal query
  - The goal of relevance feedback is to bring the user query closer to the optimal query
- How does relevance feedback actually work?
  - Use relevance information to update query
  - Use query to retrieve new set of documents
- What exactly do we “feed back”?
  - Boost weights of terms from relevant documents
  - Add terms from relevant documents to the query
  - Note that details are not visible/understandable to the user

Relevance Feedback

Vector Space Model

- Relevance feedback was first introduced using the vector space model
- Reformulation of a query moves the query vector closer to relevant documents, closer to the *optimal query*
- The *optimal query* maximises the difference between the average vector representing the relevant documents and the average vector representing the non-relevant documents

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 feedback query
- Rocchio seeks the query $\overrightarrow{q_{opt}}$ that maximizes

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

C_r: set of relevant documents in the collection
C_{nr}: set of non-relevant documents in the collection

- Tries to separate docs marked relevant and non-relevant

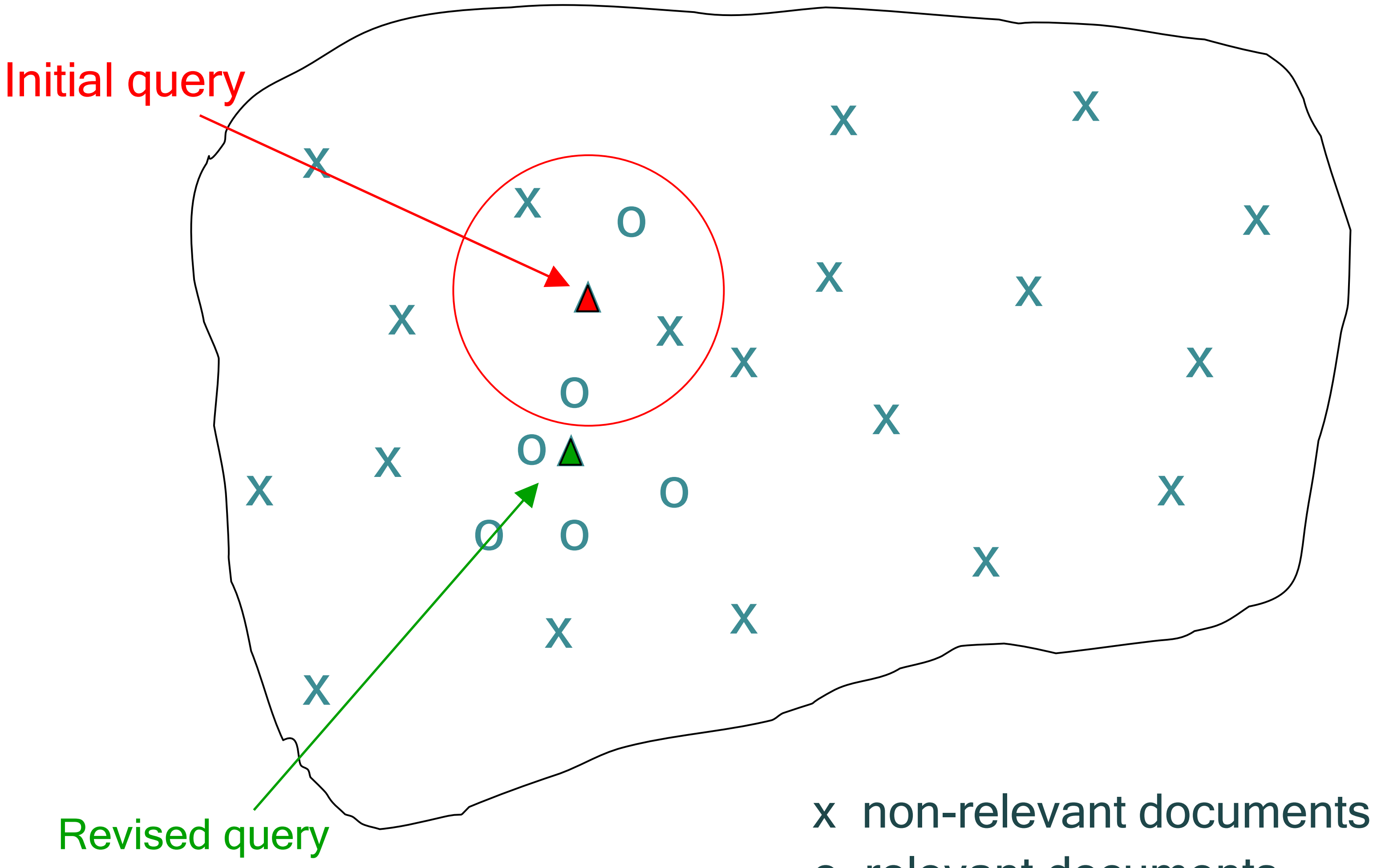
$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{d_i \in C_r} \vec{d}_i - \frac{1}{|C_{nr}|} \sum_{d_i \in C_{nr}} \vec{d}_i$$

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

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

Picture of Relevance Feedback



x non-relevant documents
o relevant documents

Relevance Feedback

Using Rocchio algorithm

- Given the limited relevance information available, the most common (and effective method) to reformulate the query is to modify the initial weights in the query vector according to the Rocchio algorithm
  - D_r: set of relevant and retrieved documents
  - D_{nr}: set of non-relevant and retrieved documents

$$\vec{q}_{next} = \alpha \cdot \vec{q}_{prev} + \beta \cdot \frac{1}{|D_r|} \sum_{d_i \in D_r} \vec{d}_i - \gamma \cdot \frac{1}{|D_{nr}|} \sum_{d_i \in D_{nr}} \vec{d}_i$$

- The factors *α, β, γ* control the effect of previous query, relevant documents and non-relevant documents on the new query

Relevance Feedback

Rocchio Cont’d

$$\vec{q}_{next} = \alpha \cdot \vec{q}_{prev} + \beta \cdot \frac{1}{|D_r|} \sum_{d_i \in D_r} \vec{d}_i - \gamma \cdot \frac{1}{|D_{nr}|} \sum_{d_i \in D_{nr}} \vec{d}_i$$

- The formula modifies the query term weights by adding a component based on the average weight in the relevant documents and subtracting a component base on the average weight in the non-relevant documents
- Usually information in relevant documents more important than in non-relevant documents ($\gamma << \beta$)
- Positive relevance feedback ($\gamma = 0$) is when we only extract information from documents assessed relevant
- α emphasises the importance of the original query ($\vec{q}_{prev}$)

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).

Why?

Relevance Feedback

Rocchio Cont’d

$$\vec{q}_{next} = \alpha \cdot \vec{q}_{prev} + \beta \cdot \frac{1}{|D_r|} \sum_{d_i \in D_r} \vec{d}_i - \gamma \cdot \frac{1}{|D_{nr}|} \sum_{d_i \in D_{nr}} \vec{d}_i$$

- $\alpha = 1$
- Terms forming the reformulated query ($\vec{q}_{prev}$) are those:
  - in the original query,
  - that appear in more relevant documents than non-relevant documents
  - that appear in more than half of the relevant documents
- Negative weights ignored
- New query
  - Moves toward relevant documents
  - Away from irrelevant documents

Rocchio in Pictures

$$
\begin{aligned}
\text{query vector} = {} & \alpha \cdot \text{original query vector} \\
& + \beta \cdot \text{positive feedback vector} \\
& - \gamma \cdot \text{negative feedback vector}
\end{aligned}
$$

Typically, $\gamma < \beta$

Original query	0	4	0	8	0	0	$\alpha = 1.0$	0	4	0	8	0	0	
Positive Feedback	2	4	8	0	0	2	$\beta = 0.5$	1	2	4	0	0	1	(+)
Negative feedback	8	0	4	4	0	16	$\gamma = 0.25$	2	0	1	1	0	4	(-)
							New query	-1	6	3	7	0	-3	

Evaluation of relevance feedback strategies

- 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 strategies

- 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
  - user feedback from first collection
  - run on second collection and measured

Evaluation: Caveat

- True evaluation of usefulness must compare to other methods taking the same amount of time.
- 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: Assumptions

A1: User has sufficient knowledge for a reasonable 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
- Not enough relevant documents are retrieved in the initial query
- Examples:
  - Misspellings (Brittany Speers)
  - Cross-language information retrieval
  - Mismatch of searcher's vocabulary vs. collection vocabulary (e.g., cosmonaut/astronaut)

Relevance Prototypes

- Relevance feedback assumes that relevance prototypes are "well-behaved"
  - All relevant documents are clustered together
  - Different clusters of relevant documents, but they have significant vocabulary overlap
- In other words,
  - Term distribution in relevant documents will be similar
  - Term distribution in non-relevant documents will be different from those in relevant documents

Violation of A2

- There are several clusters of relevant documents
- 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

- 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

Implicit Feedback

- Users are often reluctant to provide relevance judgments
  - Some searches are precision-oriented
  - They’re lazy!
- Can we gather feedback without requiring the user to do anything?
- Idea: gather feedback from observed user behavior

Implicit Feedback

- The user is not assessing relevance for the benefit of the IR system, but only satisfying their own needs and
- The user is not necessarily informed that their behavior (selected documents) will be used as relevance feedback

Discussion Point

- How might user behaviors provide clues for relevance feedback?

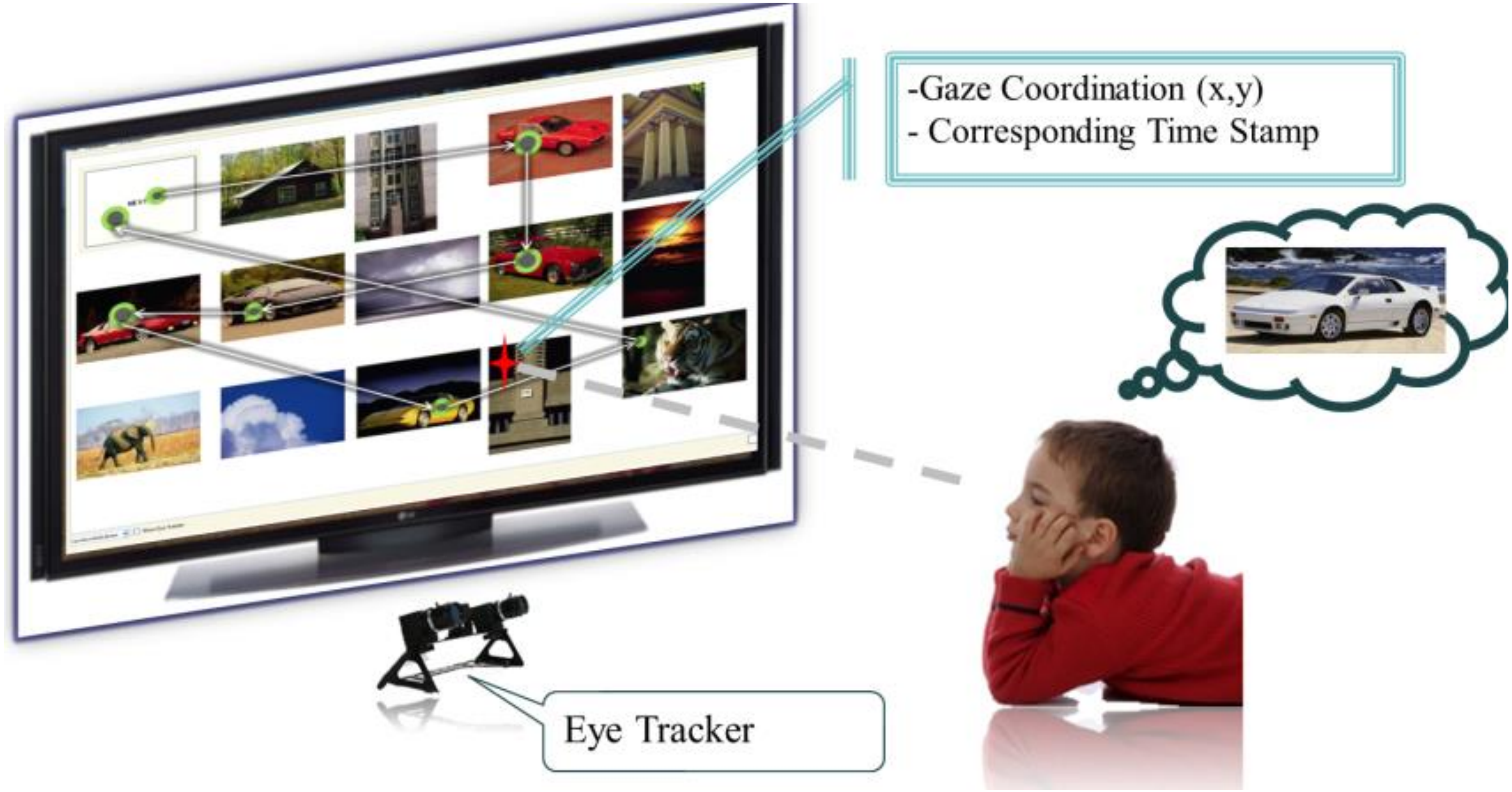
Implicit Feedback

Eye Tracking

- Click-through data provides limited information on the user behaviour
- One approach to complement information on the user behaviour is to use eye tracking devices
  - Such commercially available devices can be used to determine the area of the screen the user is focused in
  - The approach allows correctly detecting the area of the screen of interest to the user in 60-90% of the cases
  - Further, the cases for which the method does not work can be determined

Implicit Feedback

Eye Tracking



Blind Relevance Feedback

- Also called “pseudo relevance feedback”
- Motivation: it’s difficult to elicit relevance judgments from users
    - Can we automate this process?
- Idea: take top *n* documents, and simply *assume* that they are relevant
- Perform relevance feedback as before
- If the initial hit list is reasonable, system should pick up good query terms

- Does it work?

Cornell SMART system published in (Buckley et al.1995)

The Complete Landscape

- Explicit, implicit, blind feedback: it’s all about manipulating terms
- Dimensions of query expansion
  - “Local” vs. “global”
  - User involvement vs. no user involvement

Local vs. Global

- “Local” methods
  - Only considers documents that have been retrieved by an initial query
  - Query specific
  - Computations must be performed on the fly
- “Global” methods
  - Takes entire document collection into account
  - Does not depend on the query
  - Thesauri can be computed off-line (for faster access)

Local Analysis

Pseudo-Relevance Feedback

- Blind relevance feedback (discussed before)
- Pseudo-relevance feedback automates the “manual” part of true relevance feedback.
- Works very well on average
- But can go horribly wrong for some queries.
- Several iterations can cause query drift.

Global Analysis

Global Analysis for Query Reformulation

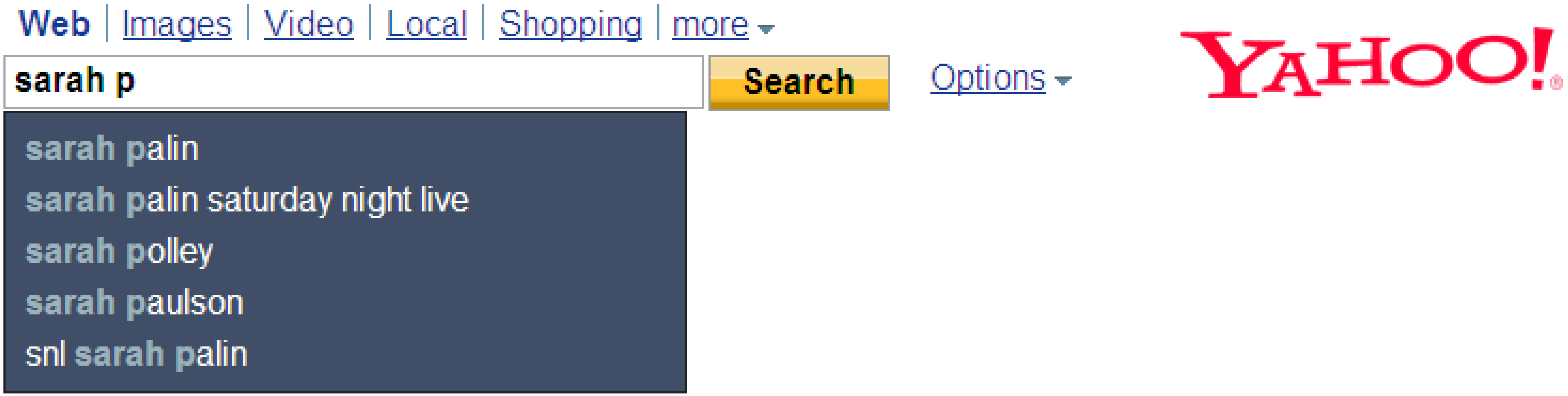
- Expand query using information from the whole set of documents in collection
- No user assistance (i.e. automatic)
- Make use of of a global thesaurus that is based on the document collection (not effective)
- Two issues:
  - Approach to build thesaurus (e.g. term co-occurrence using term association Mmeasures)
  - Approach to select terms for query expansion (e.g. the top 20 terms ranked according to IDF value)
- BUT ALSO session analysis (queries used in same sessions as analysed from logs) for query recommendation/suggestion

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

Query assist



sarah p

sarah palin
sarah palin saturday night live
sarah polley
sarah paulson
snl sarah palin

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

Query Expansion

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

Global Methods

- Controlled vocabulary
  - For example, MeSH terms
  - MeSH (Medical Subject Headings) is the (U.S.) National Library of Medicine's controlled vocabulary thesaurus and is used for indexing articles for MEDLINE.
  - It is a set of terms naming descriptors in a hierarchical structure that enables you to search at various levels of specificity.
- Manual thesaurus
  - For example, WordNet
  - WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.
  - Synsets are interlinked by means of conceptual-semantic and lexical relations.
  - The resulting network of meaningfully related words and concepts can be navigated with the browser.
- Automatically derived thesaurus
  - For example, based on co-occurrence statistics

Query Expansion

Manual thesauri

- A thesaurus may contain information about lexical semantic relations:
  - Synonyms: similar words
    e.g., violin → fiddle
  - Hypernyms: more general words
    e.g., violin → instrument
  - Hyponyms: more specific words
    e.g., violin → Stradivari
  - Meronyms: parts
    e.g., violin → strings

Query Expansion

Using Manual Thesauri

- 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

Query Expansion

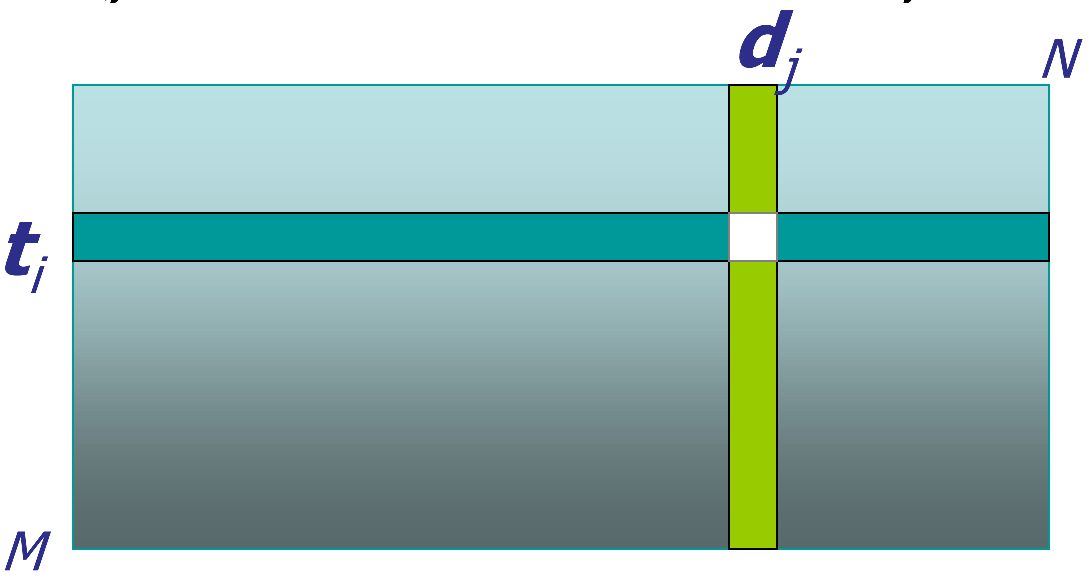
Automatic Thesaurus Generation

- Attempt to generate a thesaurus automatically by analyzing the collection of documents
- Fundamental notion: similarity between two words
- Two possible approaches
  - Co-occurrence statistics (co-occurring words are more likely to be similar)
  - Shallow analysis of grammatical relations
    - Entities that are grown, cooked, eaten, and digested are more likely to be food items.
- Co-occurrence based is more robust, grammatical relations are more accurate.

Query Expansion

Co-occurrence Thesaurus

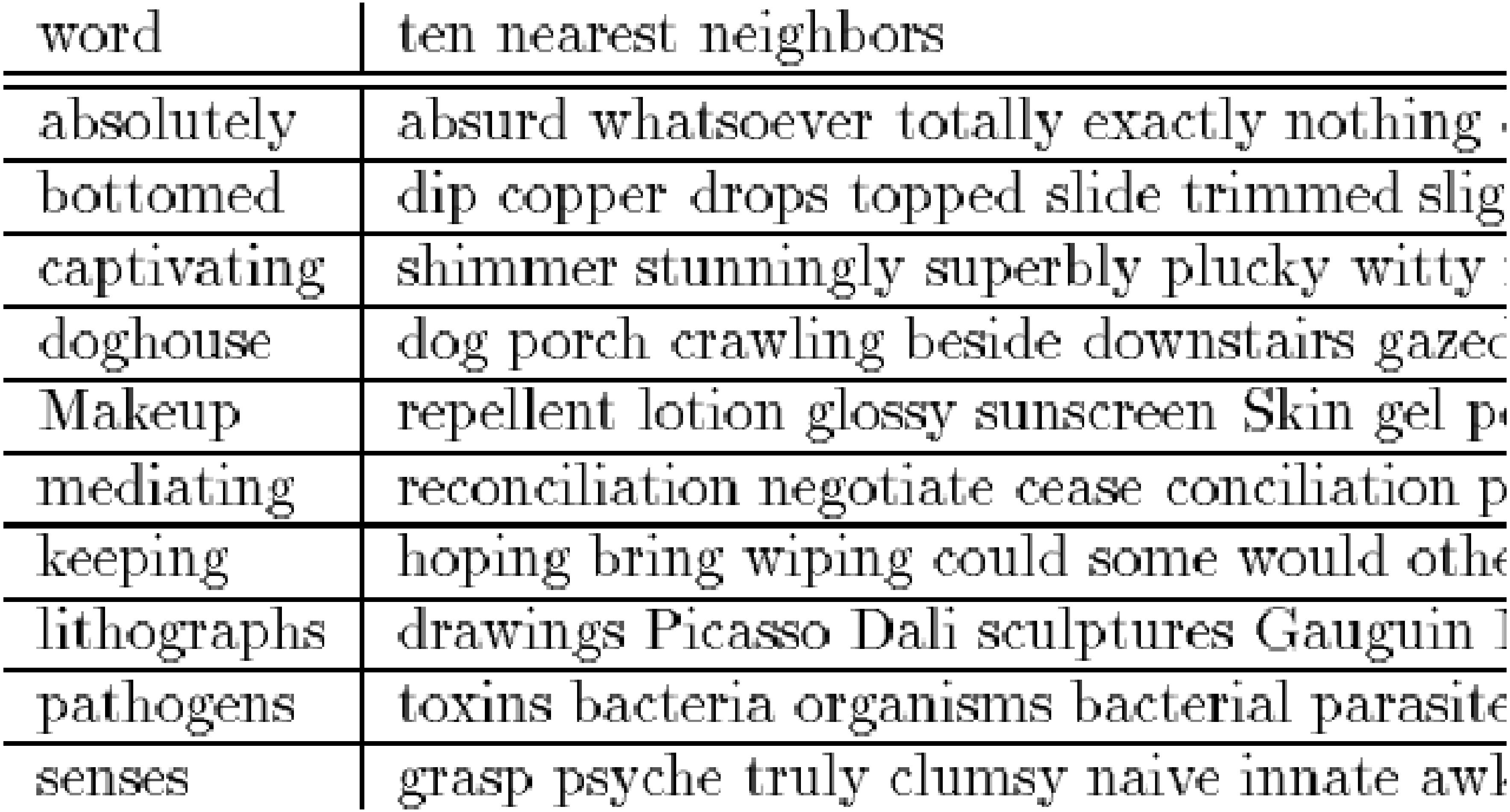
- Simplest way to compute one is based on term-term similarities in $C = AA^T$ where A is term-document matrix.
- $w_{i,j}$ = (normalized) weight for (t_i ,$\mathbf{d}_j$)



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

Query Expansion

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 gaze
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

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.

Other Approaches

Query Suggestion

- Semi-automatic
- The problem of selecting the terms for query expansion can be "reformulated" as a problem of finding similar queries rather than expansion terms
- Query logs
  - Best source of information about queries and related terms short pieces of text and click data
  - e.g., most frequent words in queries containing "tropical fish" from MSN log: stores, pictures, live, sale, types, clipart, blue, freshwater, aquarium, supplies
- Query suggestion based on finding similar queries
  - group-based on click data

Other approaches

Other approaches to Query Reformulation

- Query-based stemming
- Spell-checking and suggestions, soundex code, edit distance, etc.
- Context and personalisation User Models
- ……