Query Modification, Relevance Feedback

- Lecture for SI650 / EECS549 Information Retrieval
- October 22, 2025

Slides adapted from David Juergen

CRLT Reviews

- In the next few slides, I will use red to share quotes from CRLT summaries and blue to share individual quotes.
- Echo Checks: Out of 10 responses, 5 were positive and only 1 was purely negative.
  - "The biggest strengths of the echo check were that it gave students an opportunity to demonstrate understanding and was straightforward to prepare for and complete."
  - "The biggest concern about the echo check centered on memory (3 responses). Students worried that if there was too much time between when they completed the assignment and when they completed the echo check, they would forget their code."

CRLT Reviews

- Strengths identified:
  - "Practical assignments and hands-on experience builds student interest and helps students apply what they learn"
  - "Feedback and responsiveness of the instructor and GSIs opens opportunities for learning through cycles of feedback."
  - "Lecture topics and specific models provide necessary information."
  - "Groups also listed time to be creative, learning in collaboration with peers, and transparent instructions as useful aspects of the course."

CRLT Reviews

- *Areas for growth:*
  - "Slower pace would help students take in information they're learning."
  - "for exam preparation, except for coding, i don't know how to prepare"
    - We will have exam prep last week of classes
  - "Divide up homeworks into smaller, more manageable assignments.
    - Homeworks moving forward are much smaller in scale
  - "Regarding math equations, it may be more helpful to introduce the purpose, intuition, and goal behind certain formulas before introducing the formula itself, so students and [sic] gather a strong intuition behind a certain topic."
    - Attempted to change order for slides today but please ask questions, both during class but also remember that I have office hours and would love to help you
  - "Introducing more practice questions/class polls (ungraded) that will help students stay engaged during lectures"
    - Added some polls today and will add some more future weeks. Also note that discussion sections are mainly designed to get you to practice and ask questions as well

CRLT Reviews

- *Ideas from you for you:*
  - *Ask questions, attend office hours, when struggling reach out early*
  - *Learning in groups*
  - *Starting the homeworks early*
  - *…*

Now, Feedback (not your feedback but feedback in IR ☺)

How can we improve recall in search?

- Main topic today: two ways of improving recall: relevance feedback and query expansion
- As an example consider query q: “aircraft” . . .
  - . . . and document d containing “plane”, but not containing “aircraft”
- A simple IR system will not return d for q.
- Even if d is the most relevant document for q!
- We want to change this:
  - Return relevant documents even if there is no term match with the (original) query

Recall

- Loose definition of recall in this lecture: “increasing the number of relevant documents returned to user”
- This may actually decrease recall on some measures, e.g., when expanding “jaguar” with “panthera”
- . . .which eliminates some relevant documents, but increases relevant documents returned on top pages

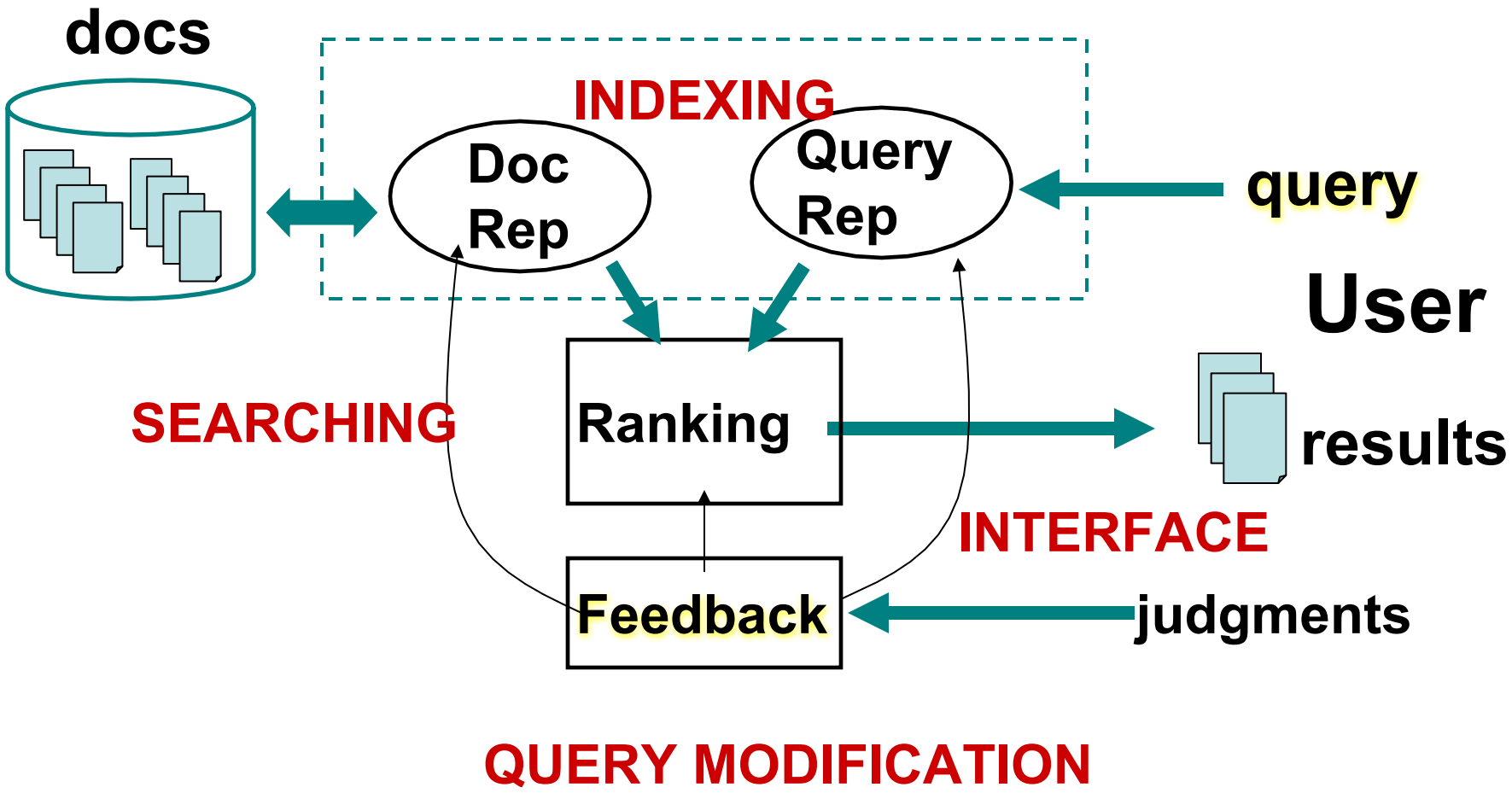
Options for improving recall

- Global: Do a global analysis once (e.g., of collection) to produce thesaurus
  - Use thesaurus for query expansion
  - Part 1
- Local: Do a “local”, on-demand analysis for a user query
  - Main local method: relevance feedback
  - Part 2

Lecture Plan

- Query modification
- Relevance feedback and pseudo feedback
- Feedback in vector space models
- Feedback in language models
- Deep learning

A Typical IR System Architecture

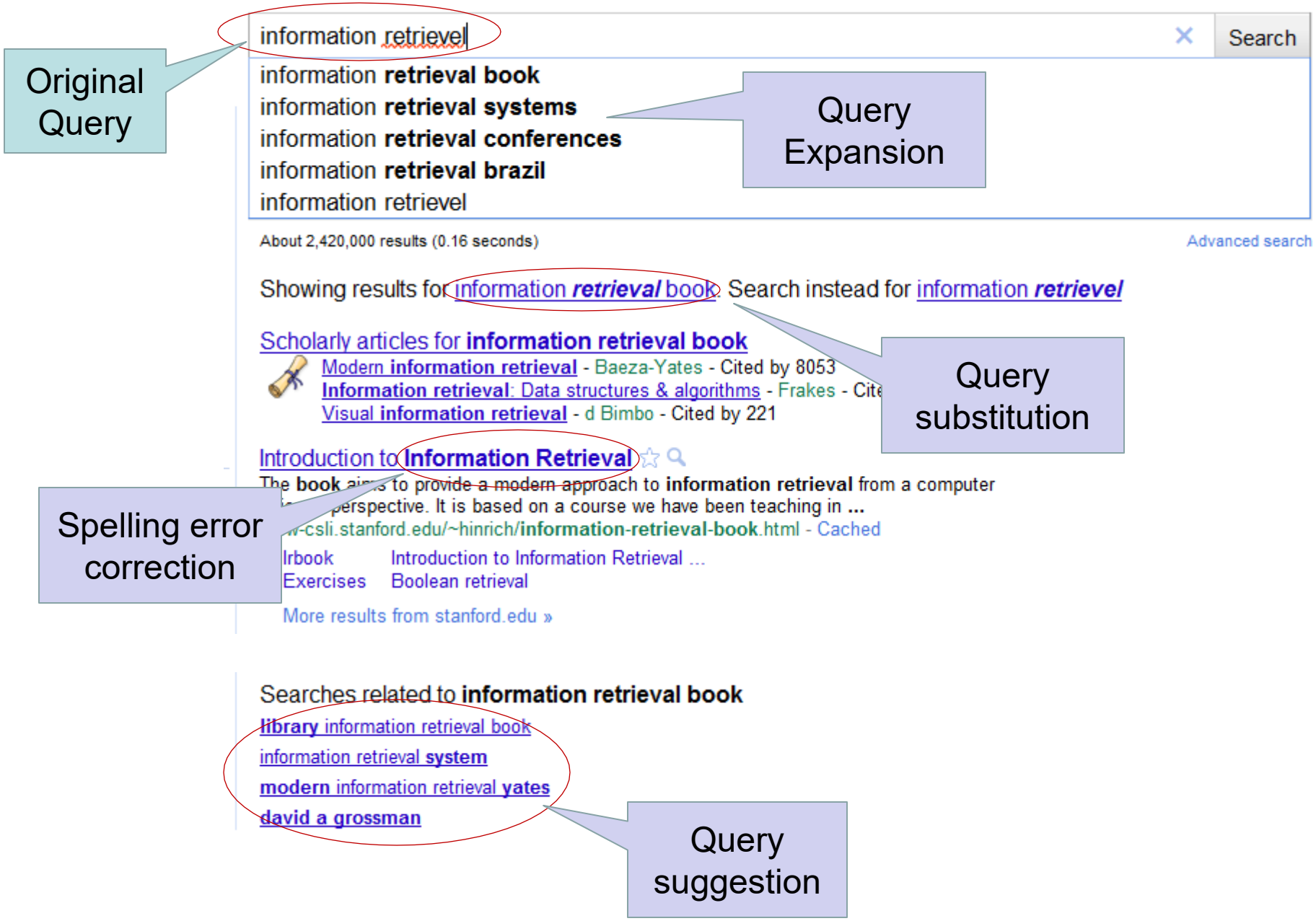


Query Modification

- Also known as query reformulation, query substitution, …
- Problem: initial query may not be the most appropriate to satisfy a given information need
- Idea: modify the original query so that it gets closer to the right documents in the vector space

Types of Query Modification

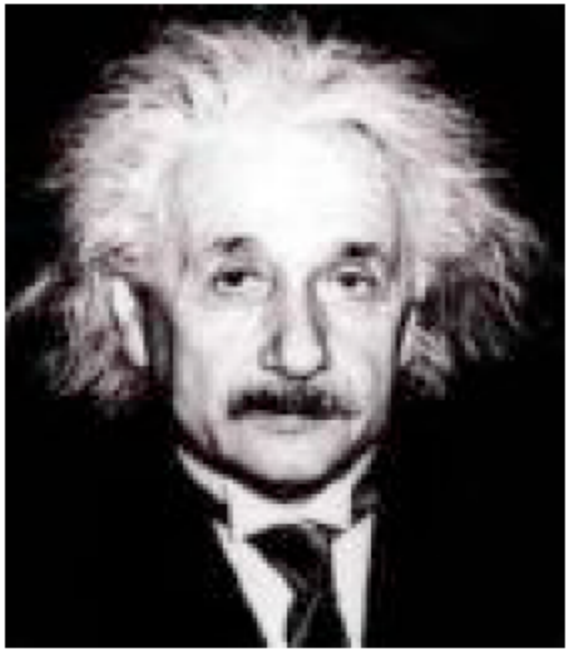
- Morphological:
  - Spelling check
- Semantic:
  - Query expansion
  - Query substitution
  - Query suggestion



Spelling Error Correction

- Roughly 10-15% of the queries sent to search engines contain errors. (Cucerzan and Brill 2004)
- Traditional techniques rely on dictionary match, combined with
  - Common keyboard mistakes
  - Phonetic/cognitive mistakes
  - Context mistakes
  - Cognitive mistakes
- Modern techniques rely on query log analysis + string similarity

Query Reformulation – Spelling Correction



[Cucerzan and Brill, 2004]

albert einstein	4834
albert einstien	525
albert einstine	149
albert einsten	27
albert einsteins	25
albert einstain	11
albert einstin	10
albert eintein	9
albeart einstein	6
aolbert einstein	6
alber einstein	4
albert einseint	3
albert einsteirn	3
albert einsterin	3
albert eintien	3
alberto einstein	3
albrecht einstein	3
alvert einstein	3

Query Recommendation/Suggestion

- Recommend alternative queries to the user. Alternative queries could be totally different from the original query.
- Usually done with query log analysis
- We will revisit this later in the semester..

Query Expansion

- Query expansion is another method for increasing recall.
- We use “global query expansion” to refer to “global methods for query reformulation”.
- In global query expansion, the query is modified based on some global resource, i.e. a resource that is not query-dependent.
- Main information we use: (near-)synonymy
- A publication or database that collects (near-)synonyms is called a thesaurus.
- We will look at two types of thesauri: manually created and automatically created.

Query Expansion

- Refining the information need of search by adding new terms to query
- Sometimes also remove terms …
- Traditional methods:
  - E.g., thesaurus-based expansion
- Corpus-based methods
  - Mining related terms from large scale corpus
- Feedback: most effective method in IR
- Query log based methods (later)

Types of user feedback

- User gives feedback on documents.
  - More common in relevance feedback
- User gives feedback on words or phrases.
  - More common in query expansion

Types of query expansion

- Manually constructed thesaurus (maintained by editors, e.g., Unified Medical Language System)
- Automatically derived thesaurus (e.g., based on co-occurrence statistics of terms)
- Query-equivalence based on query log mining (common on the web as in the “palm” example)

Thesaurus-based query expansion

- For each term t in the query, expand the query with words the thesaurus lists as semantically related with t.
  - Example: HOSPITAL → MEDICAL
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms:
  INTEREST RATE → INTEREST RATE FASCINATE
- Widely used in specialized search for science & engineering
- It's very expensive to create a manual thesaurus and to maintain it over time.
- A manual thesaurus has an effect roughly equivalent to annotation with a controlled vocabulary

Automatic thesaurus generation

- Attempt to generate a thesaurus automatically by analyzing the distribution of words in documents
- Fundamental notion: similarity between two words
- Definition 1: Two words are similar if they co-occur with similar words.
  - “car” ≈ “motorcycle” because both occur with “road”, “gas” and “license”, so they must be similar.
- 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 is more robust, grammatical relations are more accurate.

Co-occurrence-based thesaurus construction

- Statistically measure whether two words co-occur frequently (relative to their global frequencies).
- PMI is point-wise mutual information
- The goal is to account for co-occurrence by

$$P_{corpus}(w_1, w_2) = \frac{freq(w_1, w_2)}{N} \quad P_{corpus}(w) = \frac{freq(w)}{N}$$

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{corpus}(w_1) P_{corpus}(w_2)}$$

Example

Lexical semantics (Hyponymy/ontology):

Book: publication, product, fact, dramatic composition, record
Computer: machine, expert, calculator, reckoner, figurer
Fruit: reproductive structure, consequence, product, bear
Politician: leader, schemer
Newspaper: press, publisher, product, paper, newsprint

Corpus-based method (distributional clustering):

Book: autobiography, essay, biography, memoirs, novels
Computer: adobe, computing, computers, developed, hardware
Fruit: leafy, canned, fruits, flowers, grapes
Politician: activist, campaigner, politicians, intellectuals, journalist
Newspaper: daily, globe, newspapers, newsday, paper

Co-occurrence-based thesaurus: Examples using PMI

petroleum	oil:0.032 gas:0.029 crude:0.029 barrels:0.028 exploration:0.027 barrel:0.026 opec:0.026 refining:0.026 gasoline:0.026 fuel:0.025 natural:0.025 exporting:0.025
drug	trafficking:0.029 cocaine:0.028 narcotics:0.027 fda:0.026 police:0.026 abuse:0.026 marijuana:0.025 crime:0.025 colombian:0.025 arrested:0.025 addicts:0.024
insurance	insurers:0.028 premiums:0.028 lloyds:0.026 reinsurance:0.026 underwriting:0.025 pension:0.025 mortgage:0.025 credit:0.025 investors:0.024 claims:0.024 benefits:0.024
forest	timber:0.028 trees:0.027 land:0.027 forestry:0.026 environmental:0.026 species:0.026 wildlife:0.026 habitat:0.025 tree:0.025 mountain:0.025 river:0.025 lake:0.025
robotics	robots:0.032 automation:0.029 technology:0.028 engineering:0.026 systems:0.026 sensors:0.025 welding:0.025 computer:0.025 manufacturing:0.025 automated:0.025

$$PMI(w_1, w_2) = log_2 \frac{P_{corpus}(w_1, w_2)}{P_{corpus}(w_1) P_{corpus}(w_2)}$$

$$P_{corpus}(w_1, w_2) = \frac{freq(w_1, w_2)}{N} \qquad P_{corpus}(w) = \frac{freq(w)}{N}$$

Query expansion at search engines

- Main source of query expansion at search engines: query logs
- Example 1: After issuing the query [herbs], users frequently search for [herbal remedies].
  - → "herbal remedies" is potential expansion of "herb".
- Example 2: Users searching for [flower pix] frequently click on the URL photobucket.com/flower. Users searching for [flower clipart] frequently click on the same URL.
  - → "flower clipart" and "flower pix" are potential expansions of each other.

Relevance Feedback

Relevance feedback: Basic idea

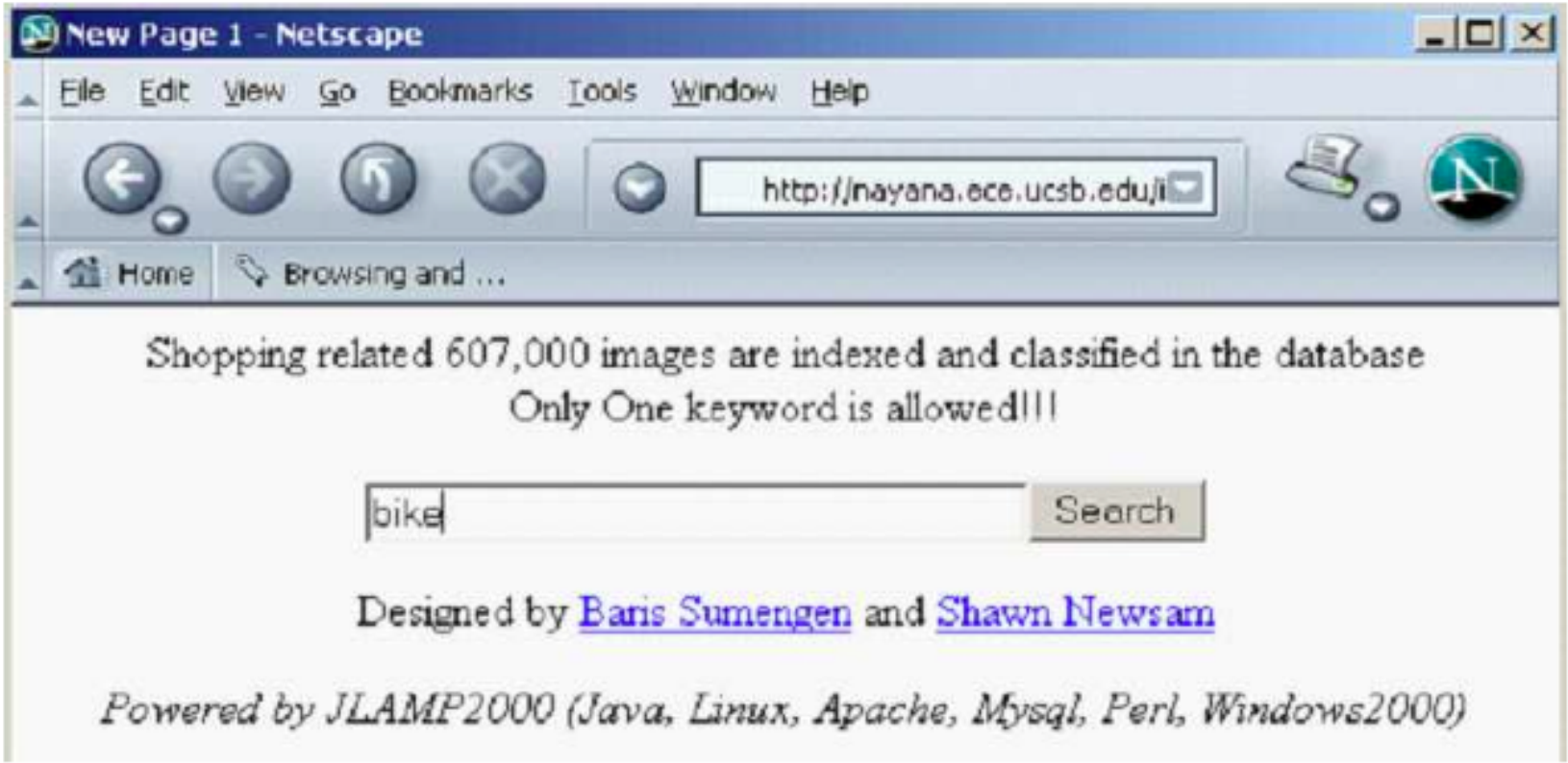
- The user issues a (short, simple) query.
- The search engine returns a set of documents.
- User marks some docs as relevant, some as nonrelevant.
- Search engine computes a new representation of the information need. Hope: better than the initial query.
- Search engine runs new query and returns new results.
- New results have (hopefully) better recall.

Relevance feedback

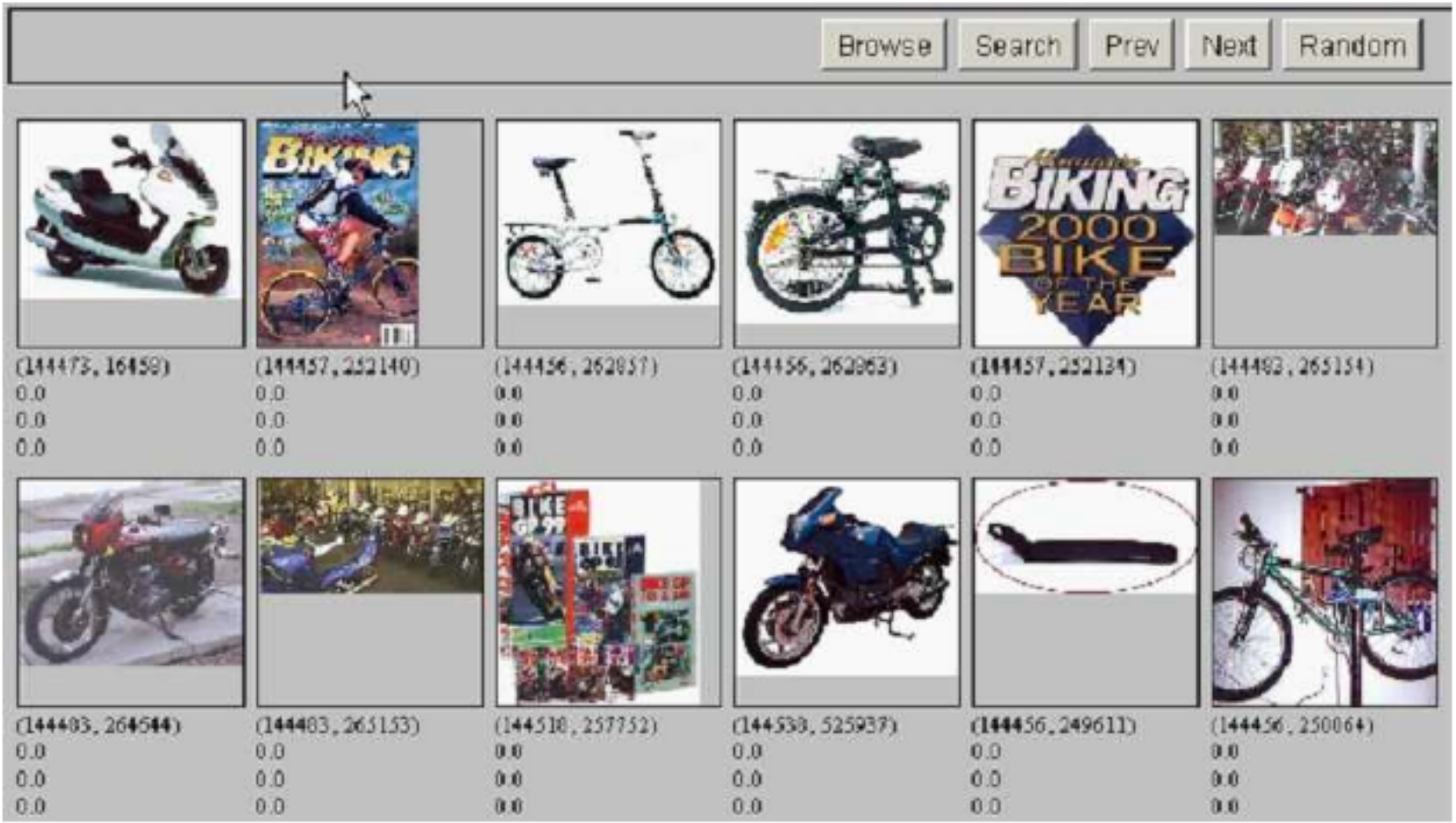
- We can iterate this: several rounds of relevance feedback.
- We will use the term ad hoc retrieval to refer to regular retrieval without relevance feedback.
- We will now look at three different examples of relevance feedback that highlight different aspects of the process.

Ancient example next…

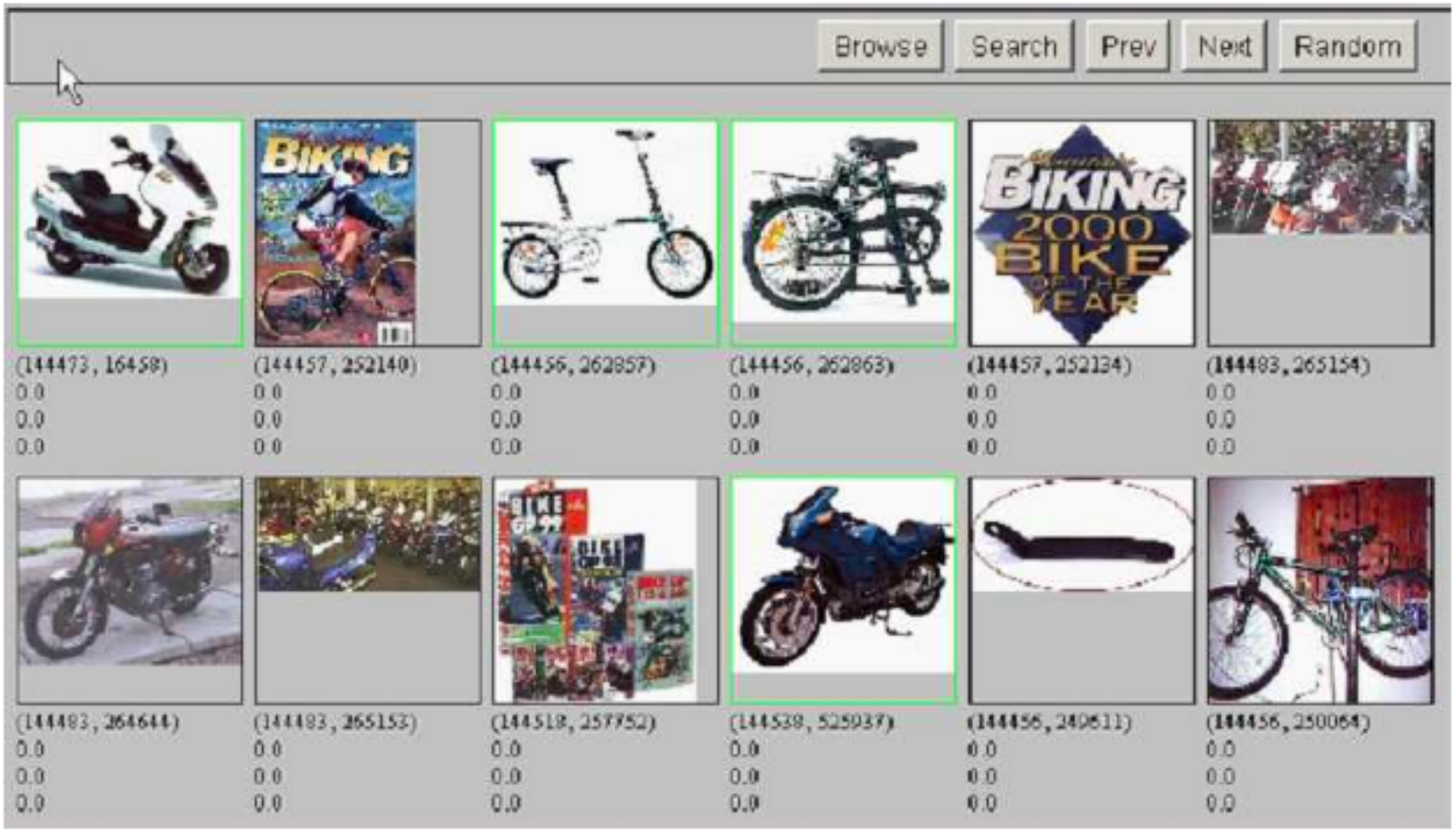
Relevance feedback: Example 1



Results for initial query



User feedback: Select what is relevant



Results after relevance feedback

Relevance feedback: Basic idea

- The user issues a (short, simple) query.
- The search engine returns a set of documents.
- User marks some docs as relevant, some as non-relevant.
- Search engine computes a new representation of the information need. Hope: better than the initial query.
- Search engine runs new query and returns new results.
- New results have (hopefully) better recall.

Relevance feedback

- We can iterate this: several rounds of relevance feedback.
- We will use the term ad hoc retrieval to refer to regular retrieval without relevance feedback.
- We will now look at three different examples of relevance feedback that highlight different aspects of the process.

Example 3: A real (non-image) example

Initial query:

[new space satellite applications] Results for initial query: (r = rank)

	r		
+	1	0.539	NASA Hasn't Scrapped Imaging Spectrometer
+	2	0.533	NASA Scratches Environment Gear From Satellite Plan
	3	0.528	Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes
	4	0.526	A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget
	5	0.525	Scientist Who Exposed Global Warming Proposes Satellites for Climate Research
	6	0.524	Report Provides Support for the Critics Of Using Big Satellites to Study Climate
	7	0.516	Arianespace Receives Satellite Launch Pact From Telesat Canada
+	8	0.509	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

Compare term weights to original

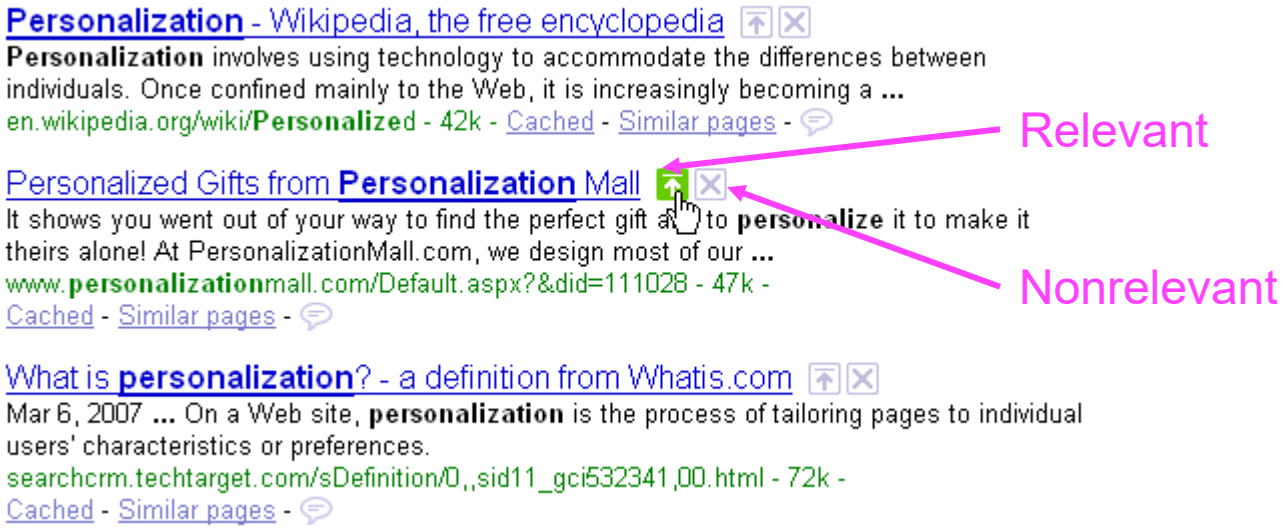
query: [new space satellite applications]

Results for expanded query

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

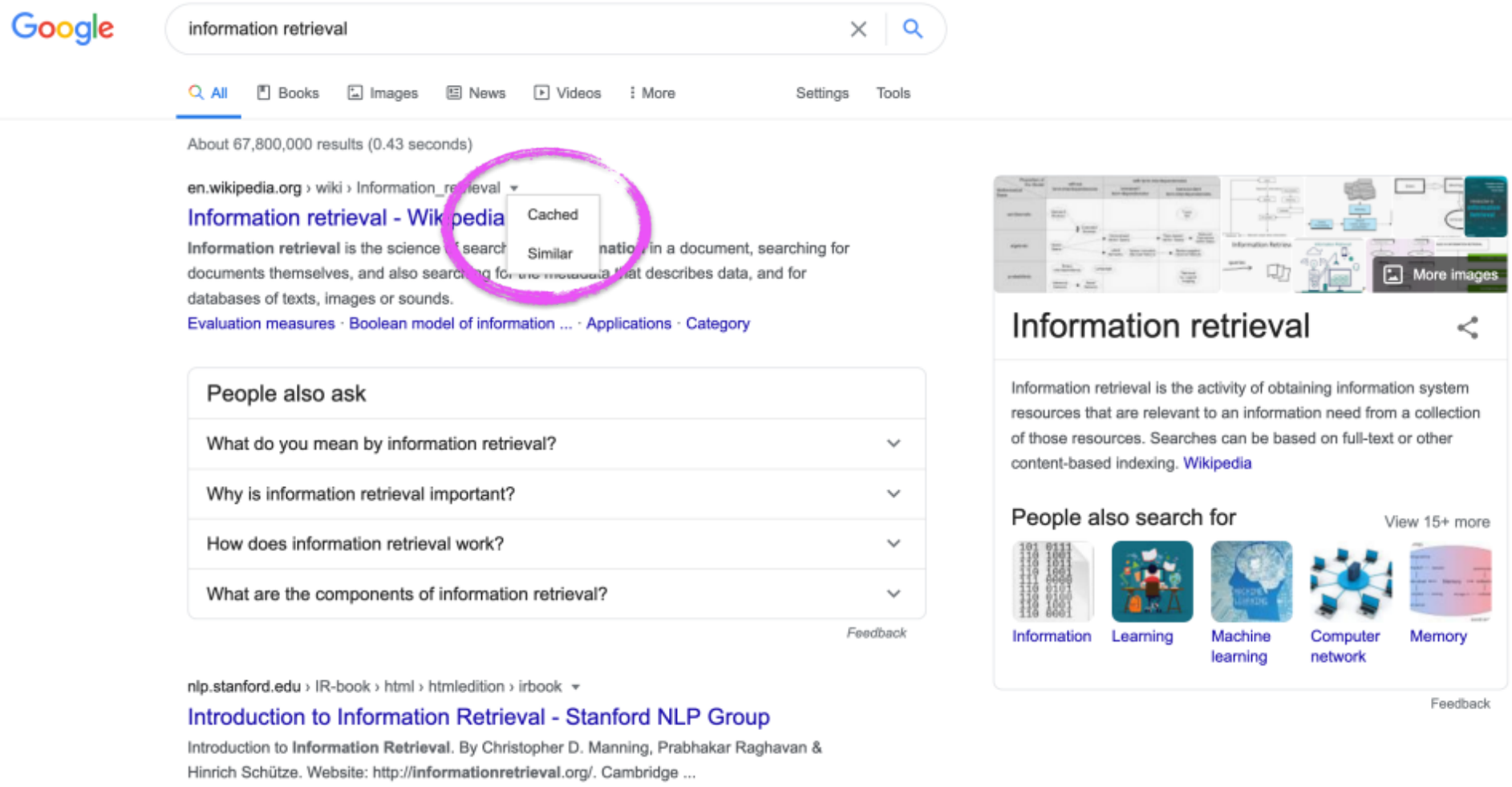
Relevance feedback in real systems

- Google used to provide such functions

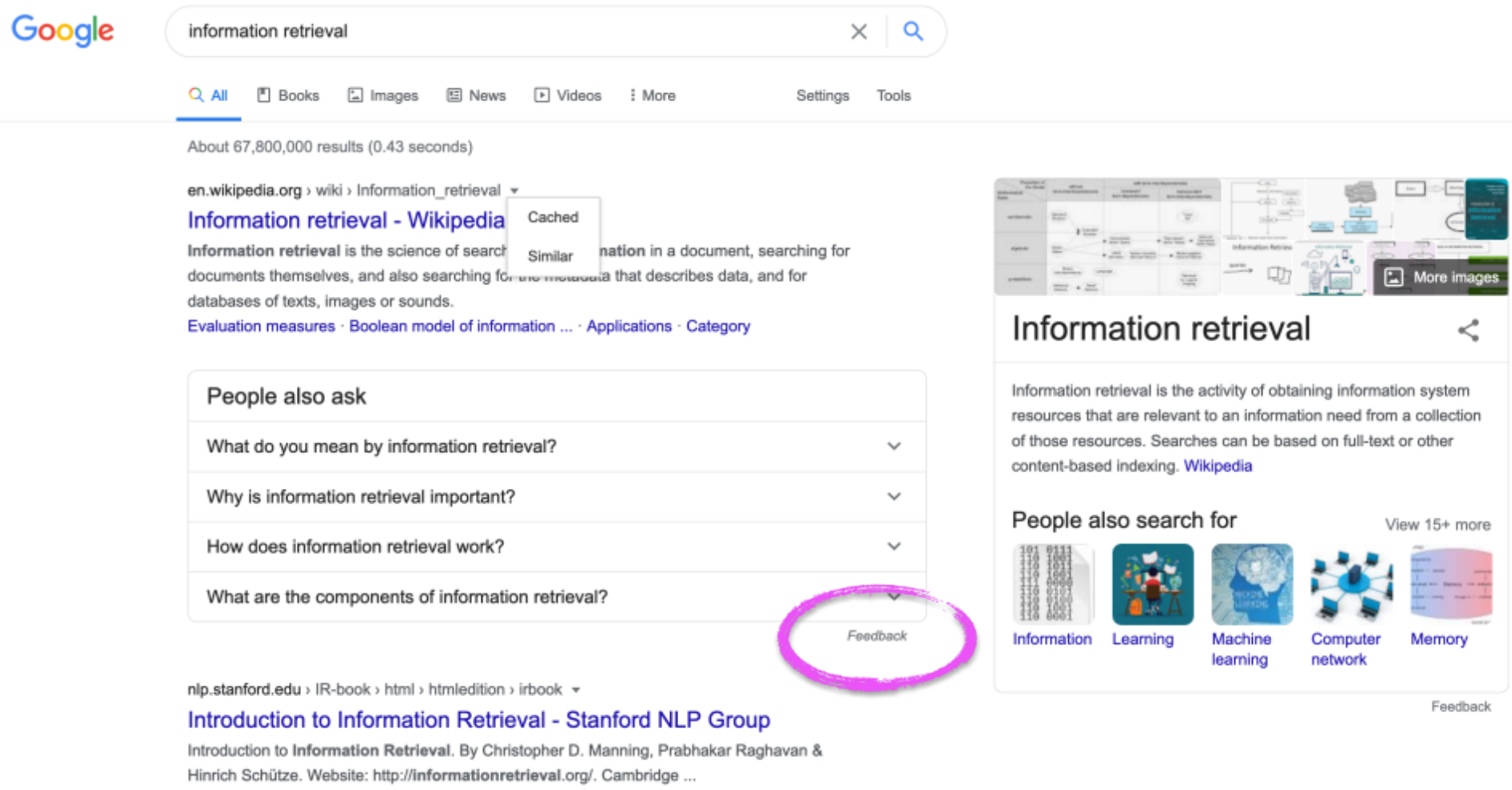


  - Guess why it got abandoned?

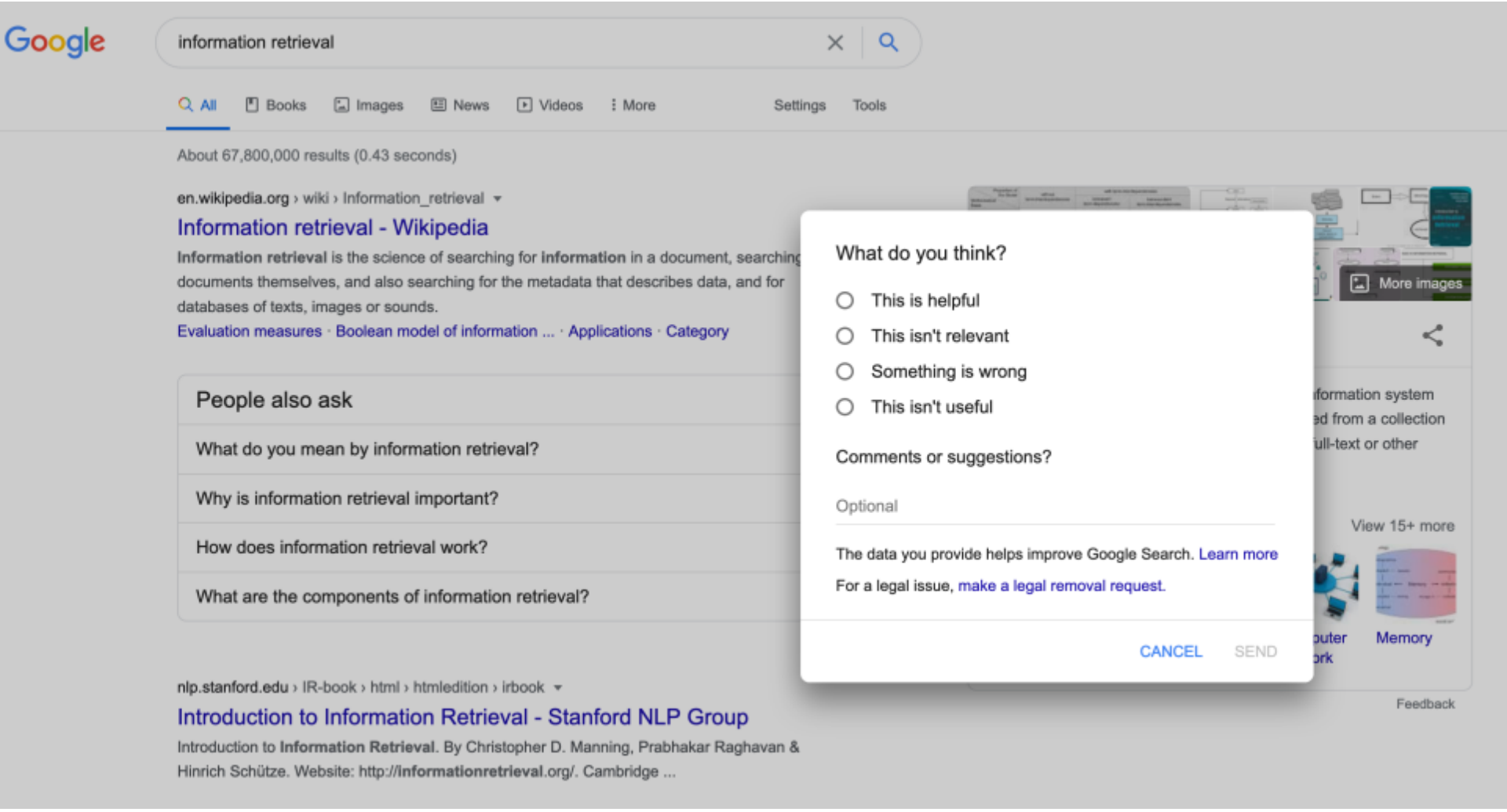
Evolution of feedback



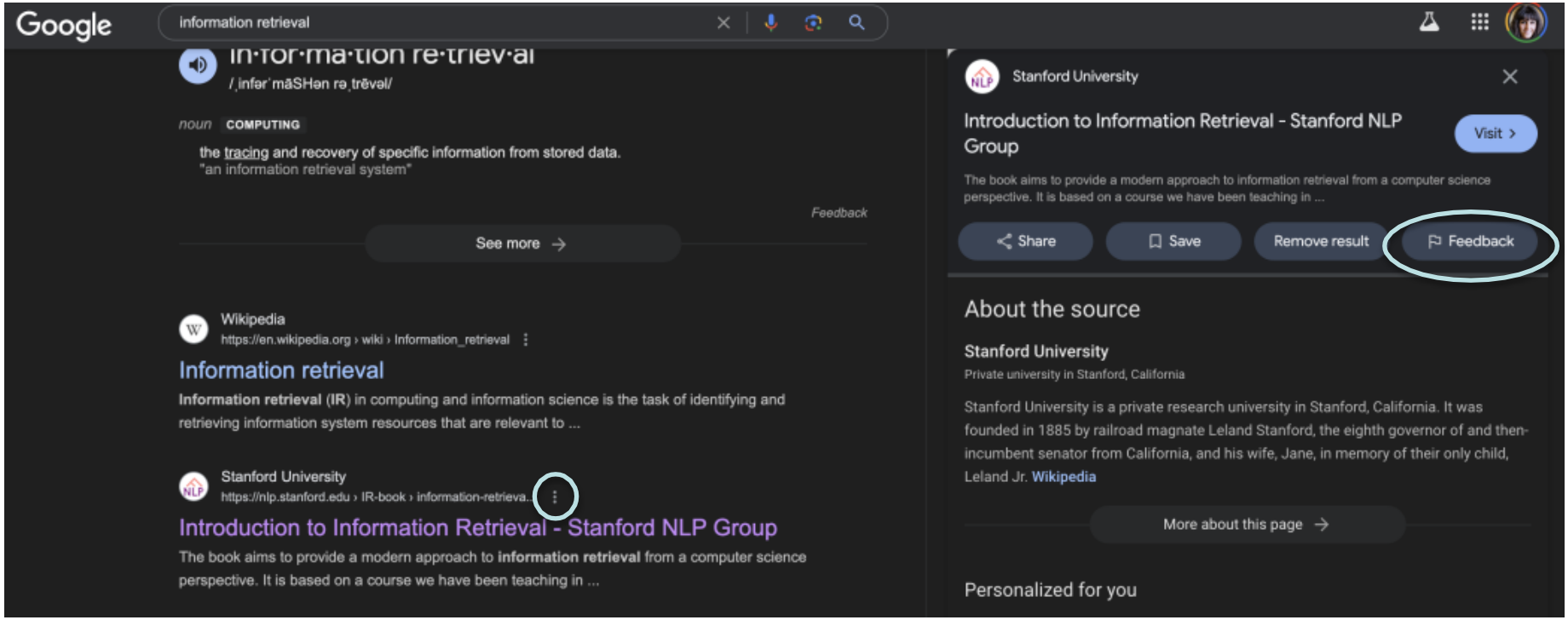
Evolution of Feedback



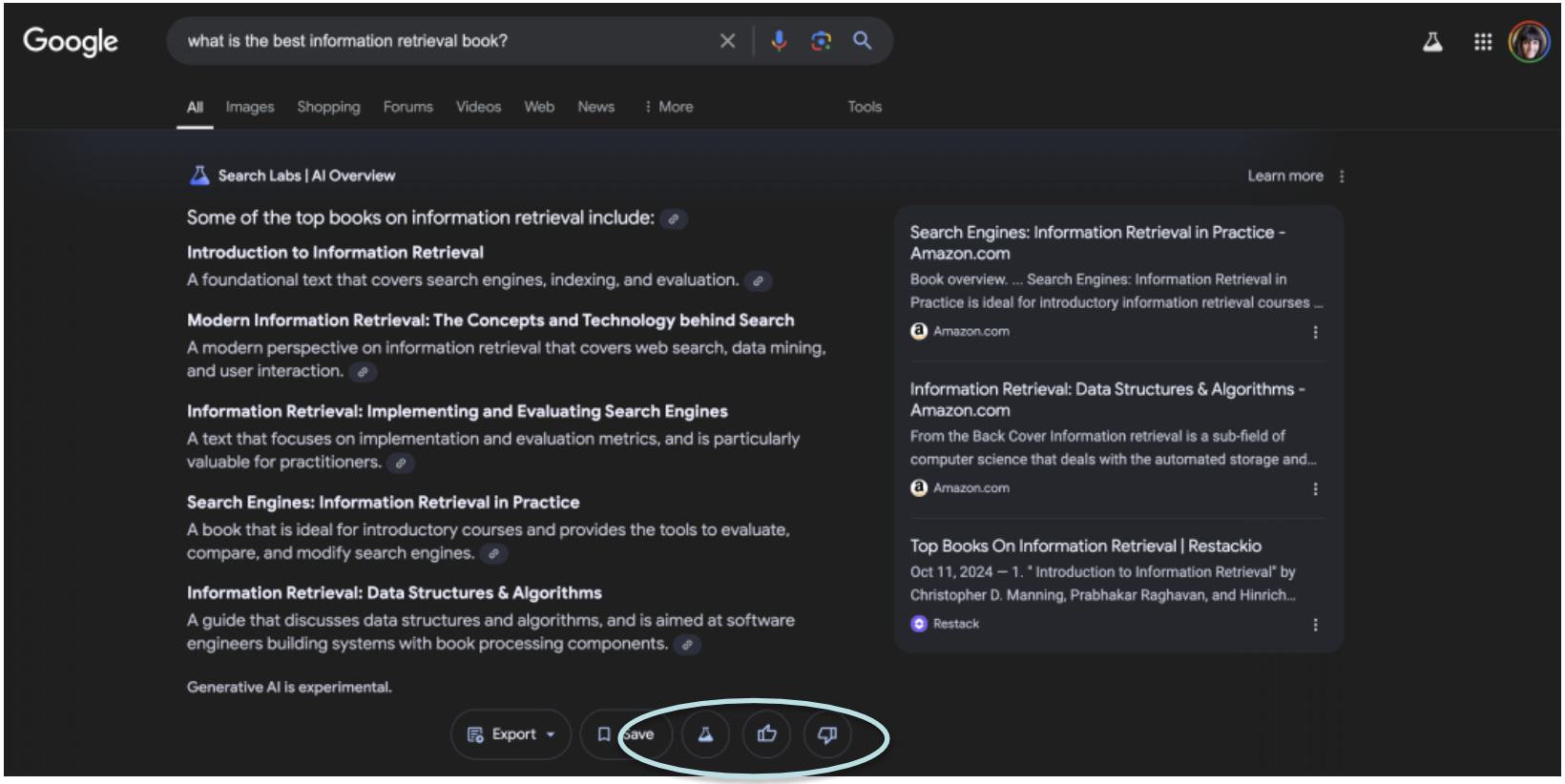
Evolution of feedback



Evolution of feedback



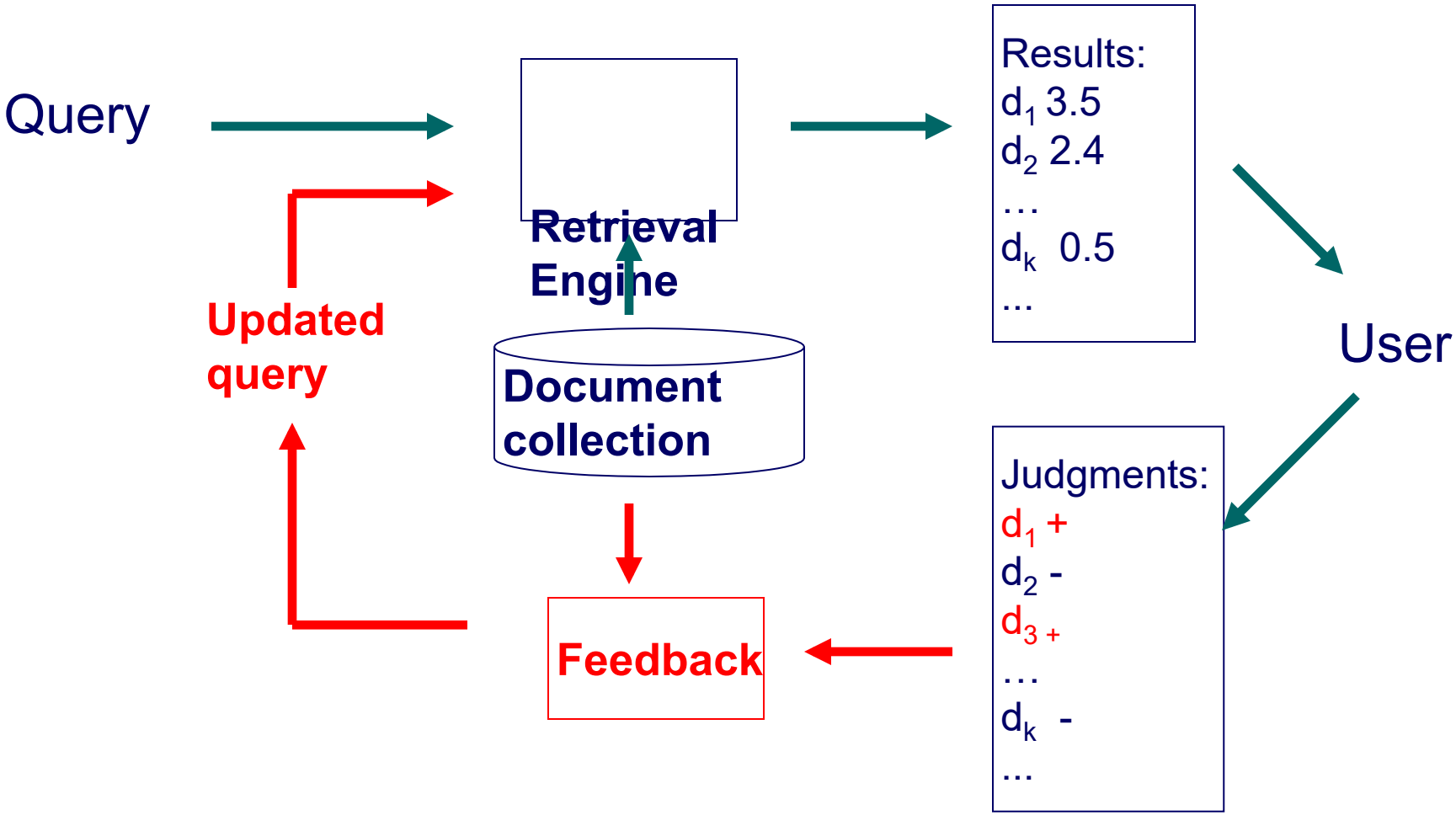
Evolution of feedback



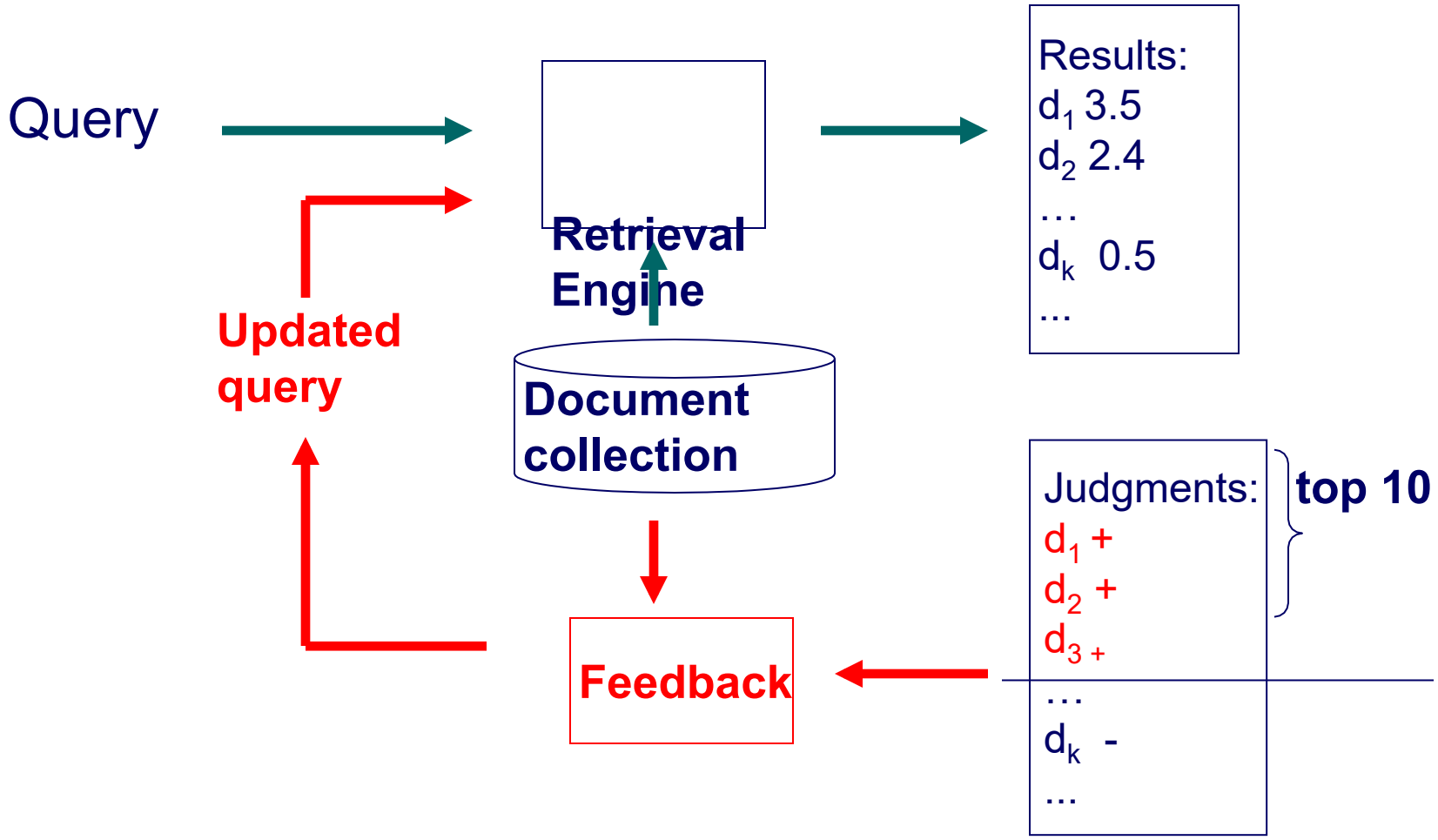
Pseudo-relevance feedback

- 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.

Relevance Feedback



Pseudo/Blind/Automatic Feedback



Intuition in Feedback

- Query expansion: feedback can help discover related query terms
  - Query = “information retrieval”
  - Relevant or pseudo-relevant docs may would likely share words related to “information retrieval”, e.g., “search engine”, “search”, “user”, “query”, etc.
  - These words generally have higher frequency in these relevant or pseudo-relevant documents than in the whole collection
  - They can be used to expand the original query to increase recall and sometimes also precision

Overview of Feedback Techniques (Cont.)

- Feedback as query expansion: traditional IR
  - Step 1: Term selection
  - Step 2: Query expansion
  - Step 3: Query term re-weighting
- Traditional IR is still robust (Rocchio), but ML approaches can potentially be more accurate

Example of Feedback

- Q = “safety minivans”
- D_1 = “car safety minivans tests injury statistics” - relevant
- D_2 = “liability tests safety” - relevant
- D_3 = “car passengers injury reviews” – not relevant
- What should the updated query look like?

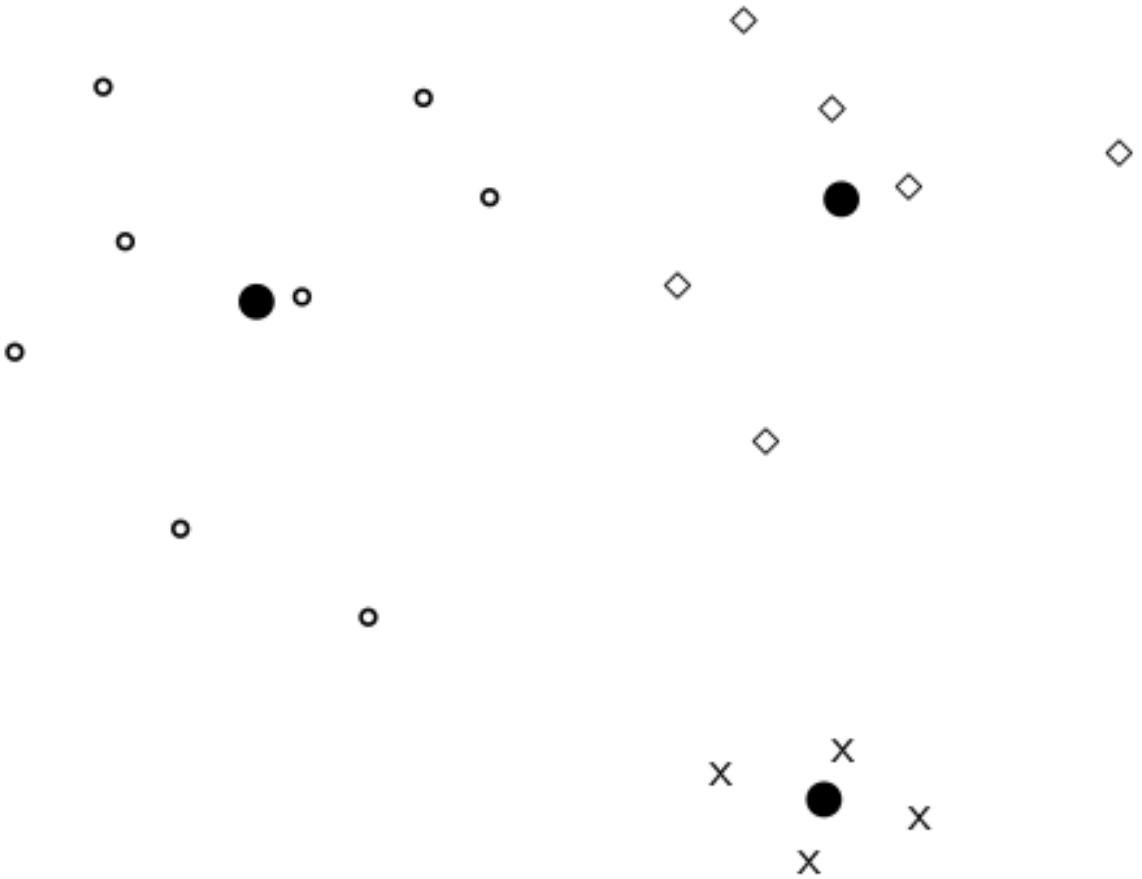
Relevance Feedback in Vector Space Model

- Basic setting: Learn from examples
  - Positive examples: docs known to be relevant
  - Negative examples: docs known to be non-relevant
  - How do you learn from this to improve performance?
- General method: Query modification
  - Adding new (weighted) terms
  - Adjusting weights of old terms
  - Doing both
- The most well-known and effective approach is Rocchio [Rocchio 1971]

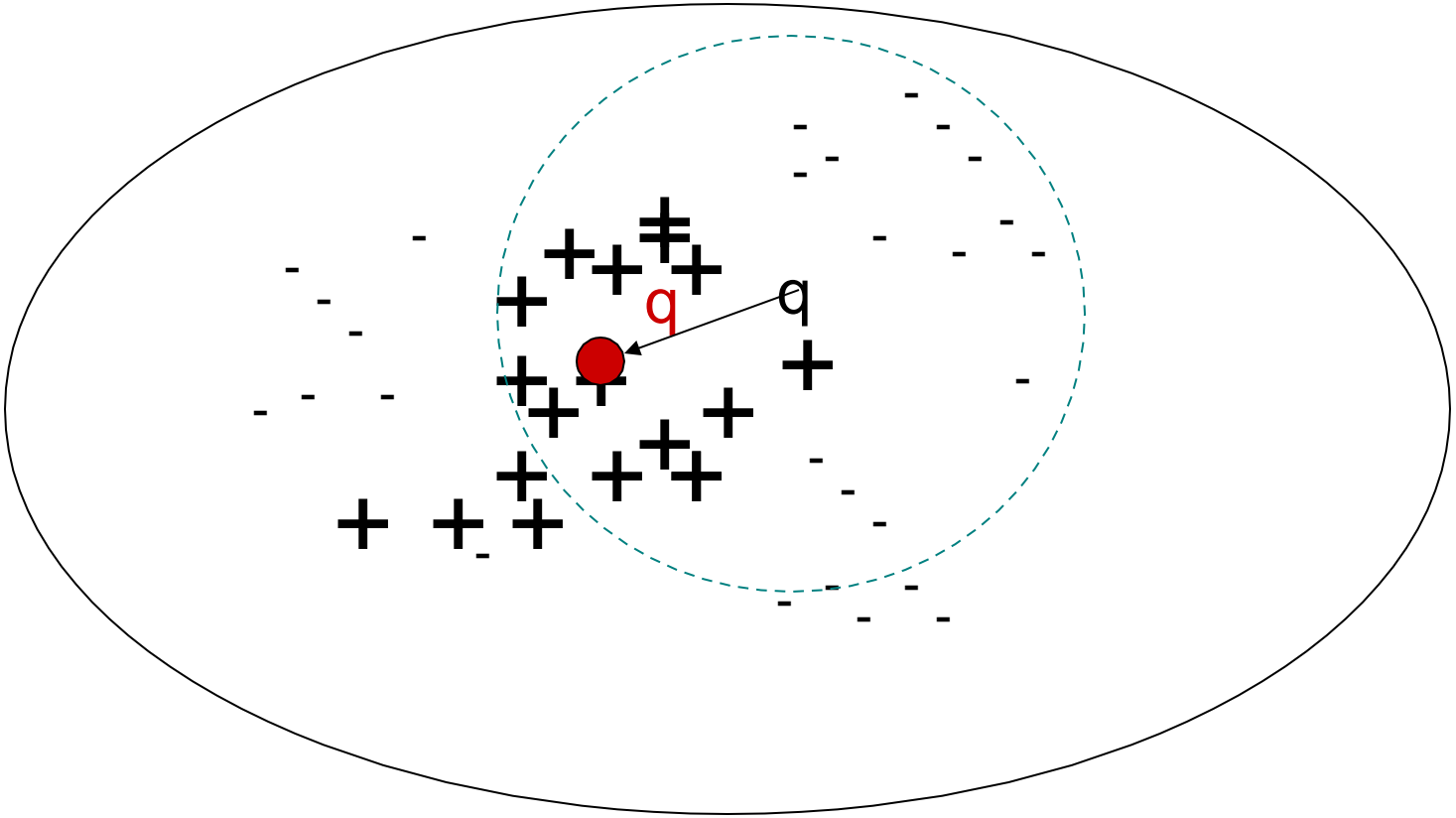
Key concept for relevance feedback: Centroid

- The centroid is the center of mass of a set of points.
- Recall that we represent documents as points in a high-dimensional space.
- Thus: we can compute centroids of documents.
- Centroid: average over all the doc vector reps
- Definition: $\vec{\mu}(D) = \frac{1}{|D|} \sum_{d \in D} \vec{v}(d)$
- where D is a set of documents and $\vec{v}(d) = \vec{d}$ is the vector we use to represent document d.

Centroid: Example



Rocchio Feedback: Illustration



Rocchio Feedback: Intuition

- Query is represented as a vector (VSM)
- Query can be updated by adding document vectors to query vectors
- If a document is labeled as relevant – add it to the query vector
- If a document is labeled as irrelevant – subtract it from the query vector
- Weighting needed.

Rocchio's algorithm

- The Rocchio' algorithm implements relevance feedback in the vector space model.
- High level optimization: maximize similarity to relevant docs and minimize similarity to non-relevant docs
- Rocchio' chooses the query $\vec{q}_{opt}$ that maximizes $\vec{q}_{opt} = \underset{\vec{q}}{argmax}\left[sim(\vec{q}, \mu(D_r)) - sim(\vec{q}, \mu(D_{nr}))\right]$
- D_r : set of relevant docs; D_{nr}: set of non-relevant docs
- Intent: $\vec{q}_{opt}$ is the vector that separates relevant and non-relevant docs maximally.
- Making some additional assumptions, we can rewrite $\vec{q}_{opt}$ as: $\vec{q}_{opt} = \mu(D_r) + [\mu(D_r) - \mu(D_{nr})]$

Rocchio' algorithm

- The optimal query vector is:

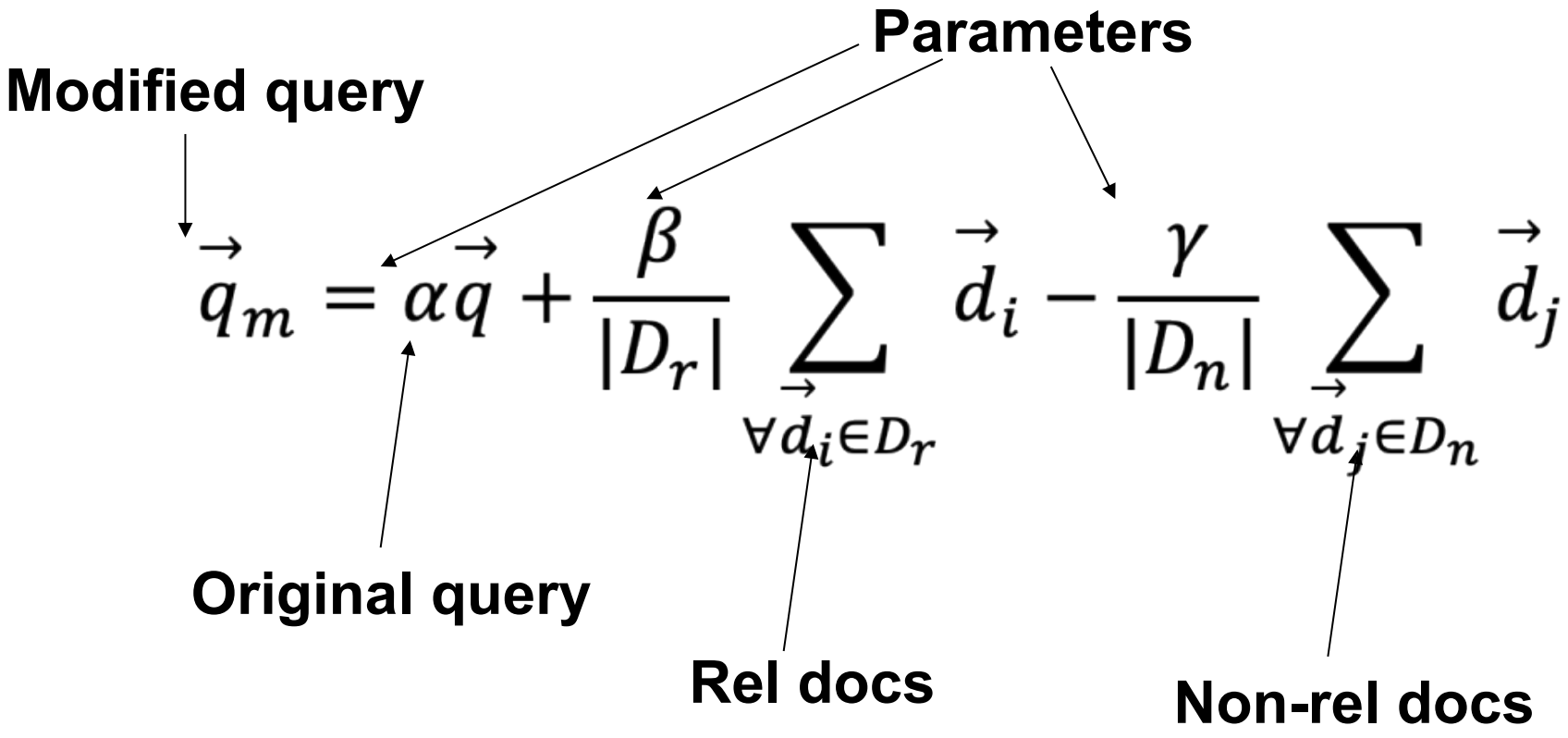
$$\vec{q}_{opt} = \mu(D_r) + [\mu(D_r) - \mu(D_{nr})]$$

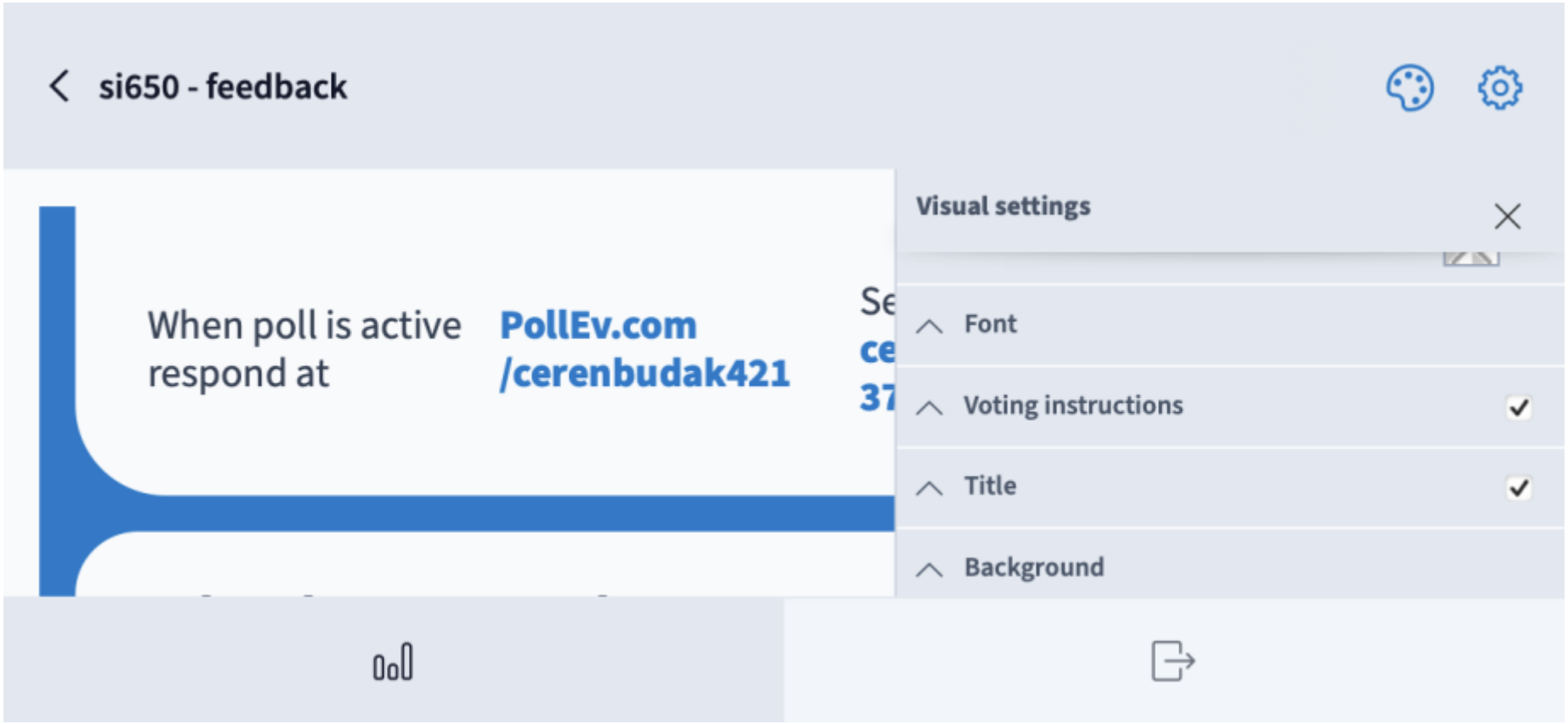
$$\vec{q}_{opt} = \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j + \left[\frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_n r} \vec{d}_j \right]$$

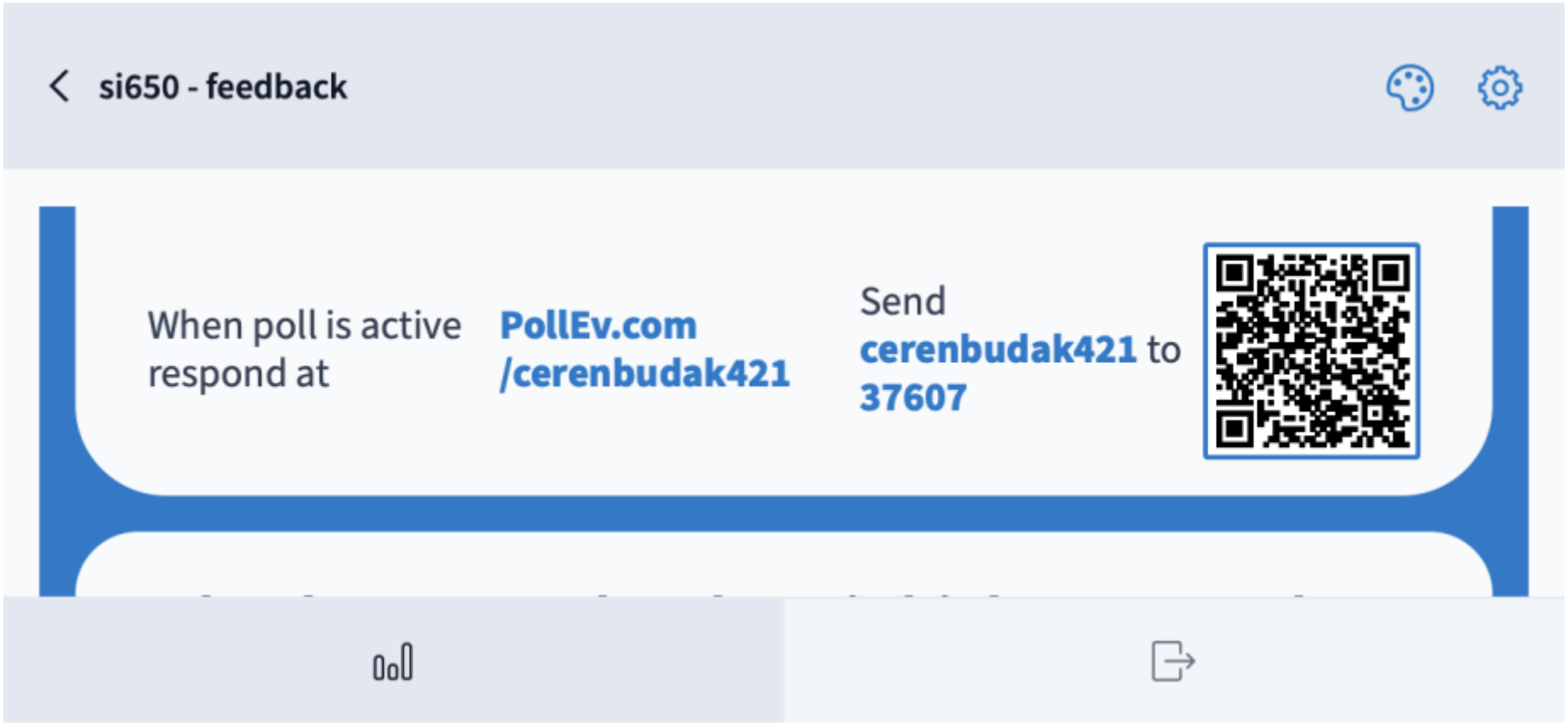
- We move the centroid of the relevant documents by the difference between the two centroids.

Formula for Rocchio feedback

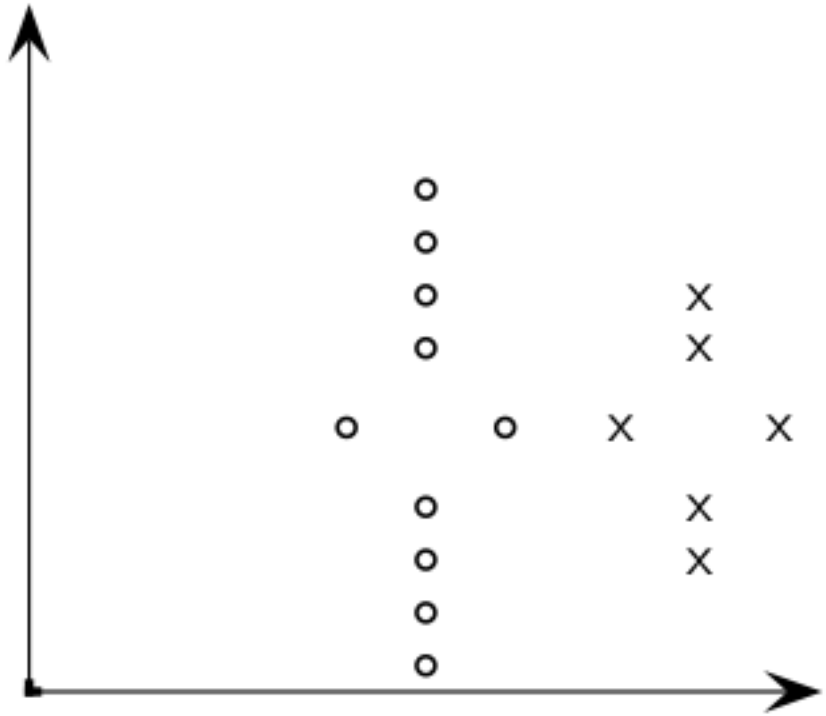
- Standard operation in vector space





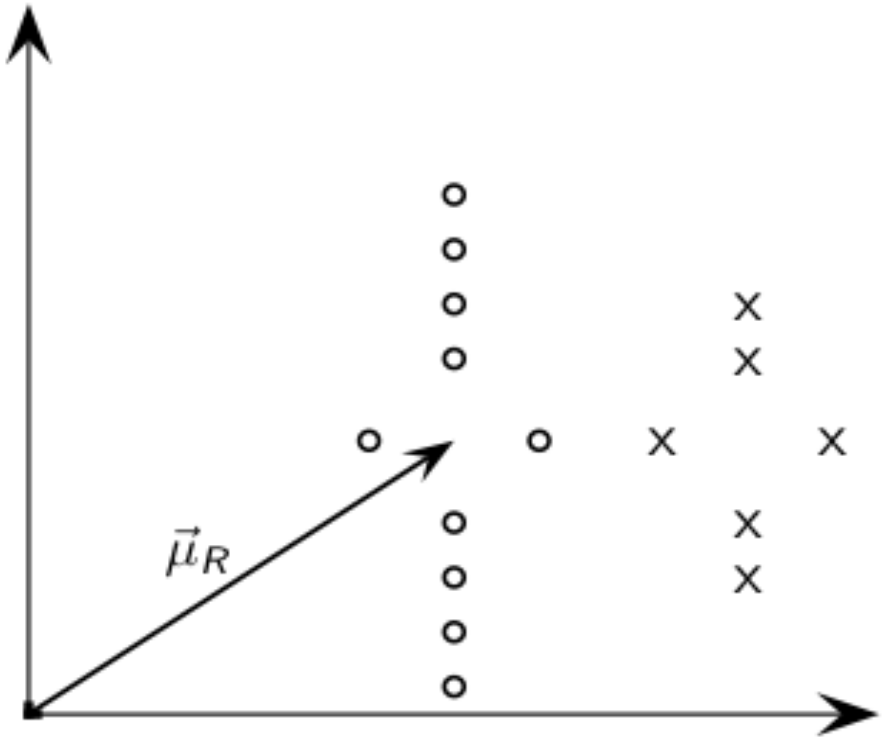


Exercise: Compute Rocchio' vector



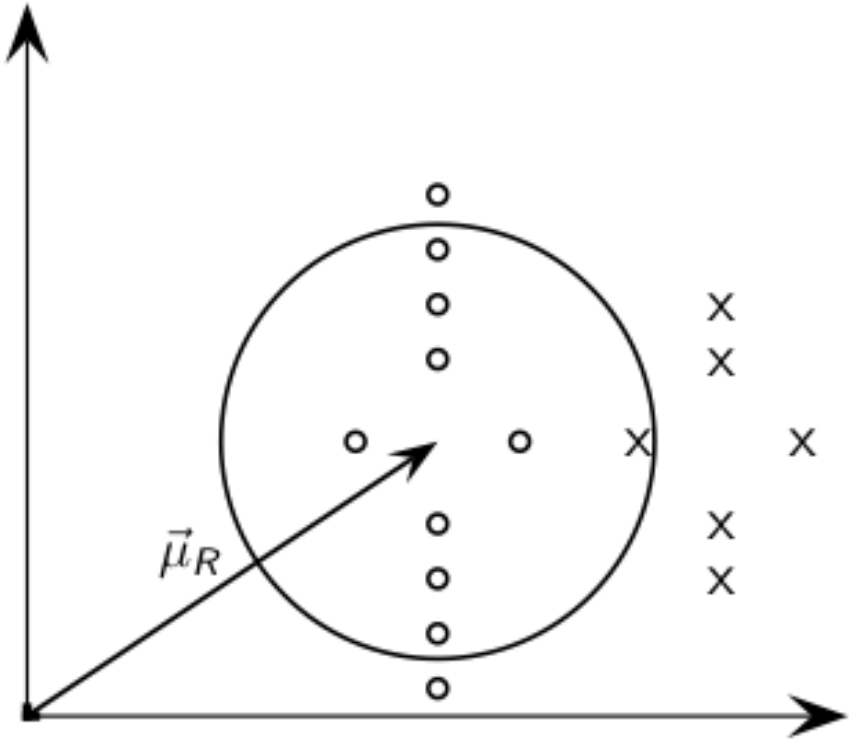
circles: relevant documents, Xs: non-relevant documents

Rocchio' illustrated



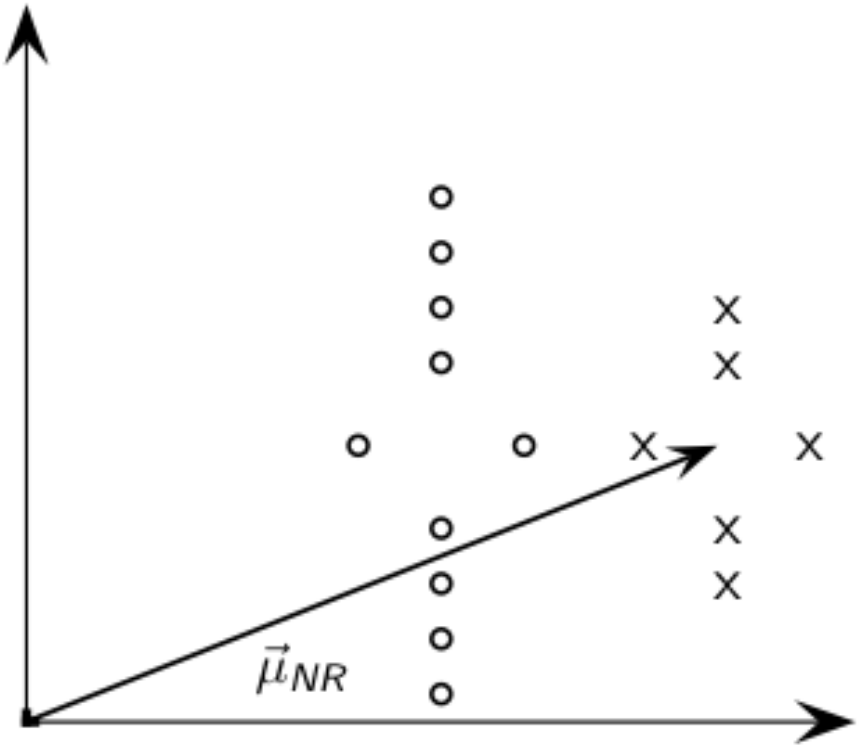
$\vec{\mu}_R$: centroid of relevant documents

Rocchio' illustrated



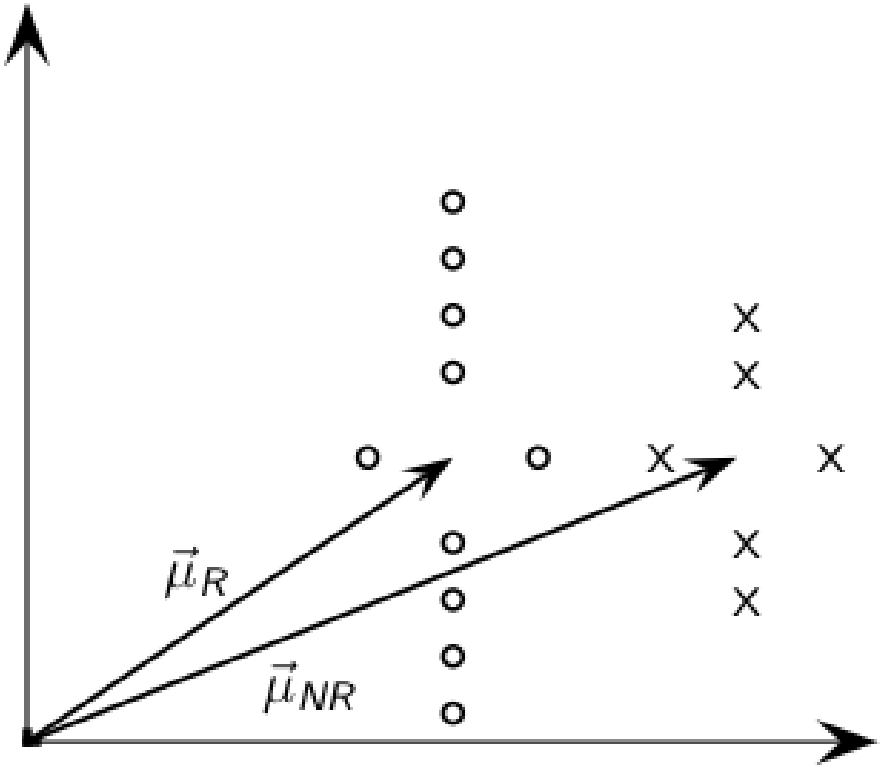
$\vec{\mu}_R$ does not separate relevant / nonrelevant.

Rocchio' illustrated

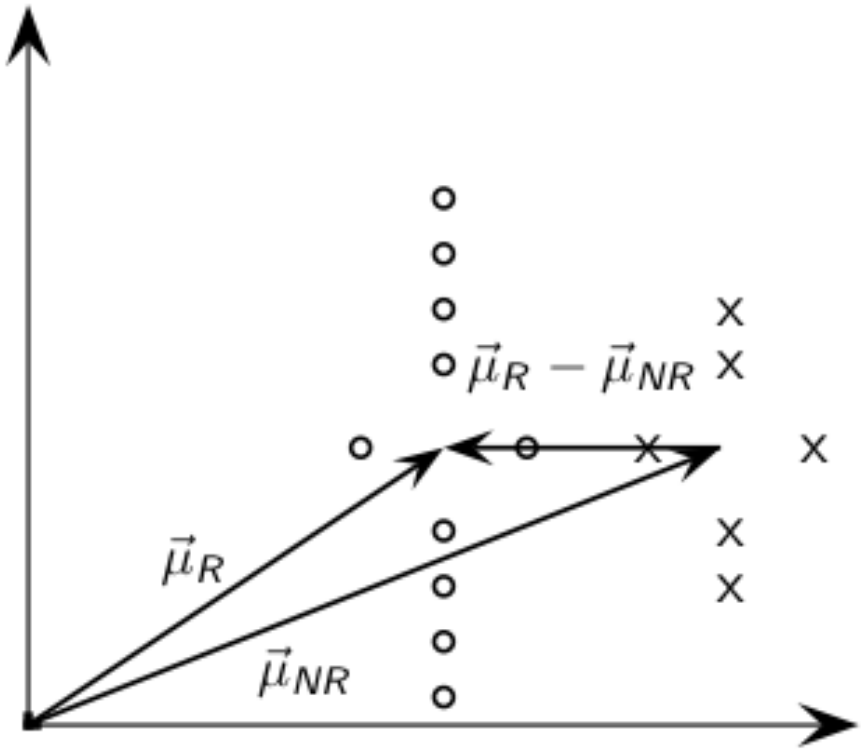


$\vec{\mu}_{NR}$: centroid of non-relevant documents.

Rocchio' illustrated

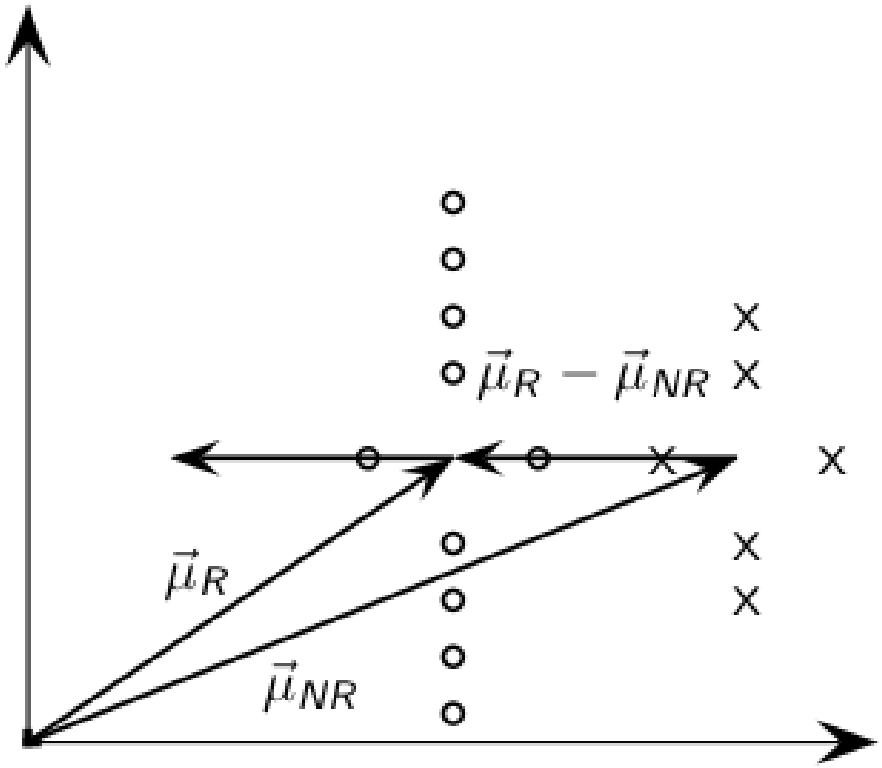


Rocchio' illustrated



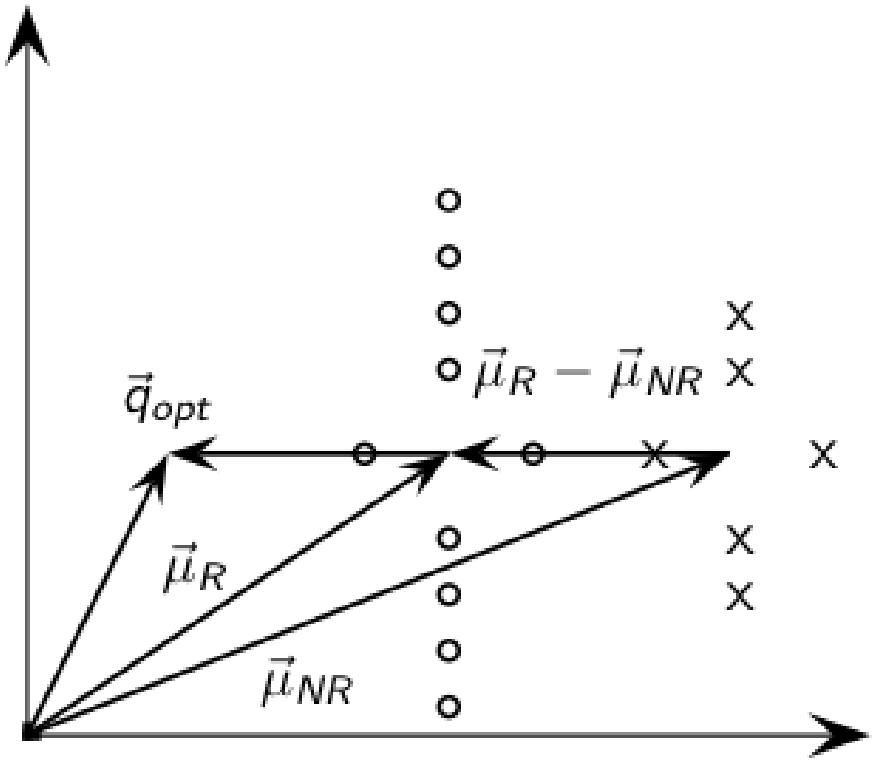
$\vec{\mu}_R$ - $\vec{\mu}_{NR}$: difference vector

Rocchio' illustrated



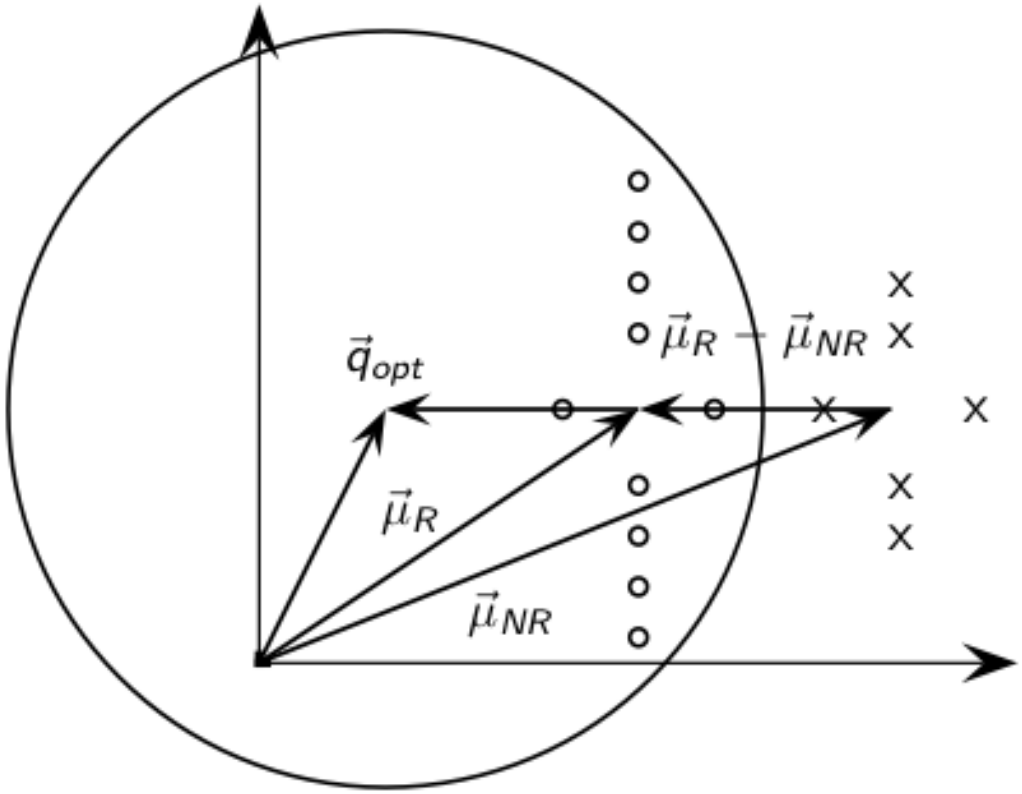
Add difference vector to $\vec{\mu}_R$ …

Rocchio' illustrated



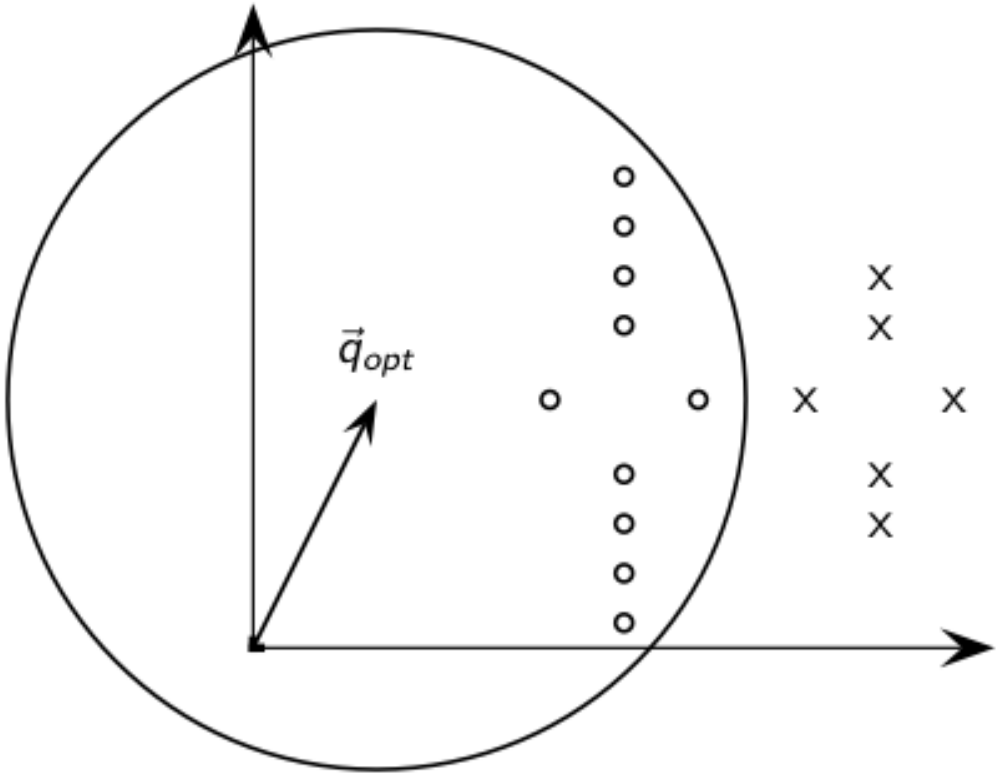
… to get $\vec{q}_{opt}$

Rocchio' illustrated



$\vec{q}_{opt}$ separates relevant / non-relevant perfectly.

Rocchio' illustrated



$\vec{q}_{opt}$ separates relevant / non-relevant perfectly.

Terminology

- We use the name Rocchio' for the theoretically better motivated original version of Rocchio.
- The implementation that is actually used in most cases is the SMART implementation—we use the name Rocchio (without prime) for that.

Rocchio 1971 algorithm (SMART)

- What's used in practice:

$$\vec{q}_m = \alpha\vec{q}_0 + \beta\mu(D_r) - \gamma\mu(D_{nr})$$

  q_m: modified query vector; q_0: original query vector; D_r and D_{nr}: sets of known relevant and non-relevant documents respectively; α, β, and γ: weights
- New query moves towards relevant documents and away from non -relevant documents.
- Tradeoff α vs. β/γ: If we have a lot of judged documents, we want a higher β/γ.
- Set negative term weights to 0.
- "Negative weight" for a term doesn't make sense in the vector space model.

Rocchio in Practice

- Negative (non-relevant) examples are not very important (why?)
- Often project the vector onto a lower dimension (i.e., consider only a small number of words that have high weights in the centroid vector) (efficiency concern)
- Avoid “training bias” (keep relatively high weight on the original query weights) (why?)
- Can be used for relevance feedback and pseudo feedback
- Usually robust and effective

Feedback in Language Models

Question:

- How to exploit language modeling to perform natural and effective feedback?

Answer: **Introduce a query model & treat feedback as query model updating**

Feedback in language models

- Recap of language model
  - Rank documents based on *query likelihood*
  - *Bag of words language model would lead to:*

$$\log p(q \mid d) = \sum_{w_i \in q} \log p(w_i \mid d)$$

$$where,\ \ q = w_1 w_2 \ldots w_n$$

Document language model

  - Difficulty
    - Documents are given, i.e., $p(w|d)$ is fixed

Kullback-Leibler (KL) Divergence

- Important concept in information theory
  - Expected number of extra bits to code a sample from distribution P using a code based on distribution Q, rather than using a code based on P.
  - P: true distribution
  - Q: hypothetical distribution

- Measure the distance of two probabilistic distributions

$$D(P\|Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} = \sum_{x} P(x) \left[\log P(x) - \log Q(x)\right]$$

outcome

sample space

Retrieval based on KL-Divergence

- Similarity can be thought of as (0 – Distance), where distance is computed using KL divergence
- So similarity between query and document language models:

$$sim(Q,D) \propto -D(\theta_Q \| \theta_D)$$

$$= \sum_w p(w|\theta_Q)\log p(w|\theta_D) - \sum_w p(w|\theta_Q)\log p(w|\theta_Q)$$

query entropy (ignored for ranking)

- Retrieval ~ Estimation of θ_D and θ_Q (details skipped here, see the notes for a short video description)

$$sim(q,d) \propto \sum_{w \in d, p(w|\theta_Q)>0} [p(w|\hat{\theta}_Q)\log \frac{p_{seen}(w|d)}{\alpha_d p(w|C)}] + \log \alpha_d$$

- Special case: $p(w|\theta_Q)=c(w, q)/|q|$ recovers "query likelihood"

Feedback in language models

- Approach
  - Introduce a probabilistic query model
  - Ranking: measure distance between query model and document model
  - Feedback: query model update

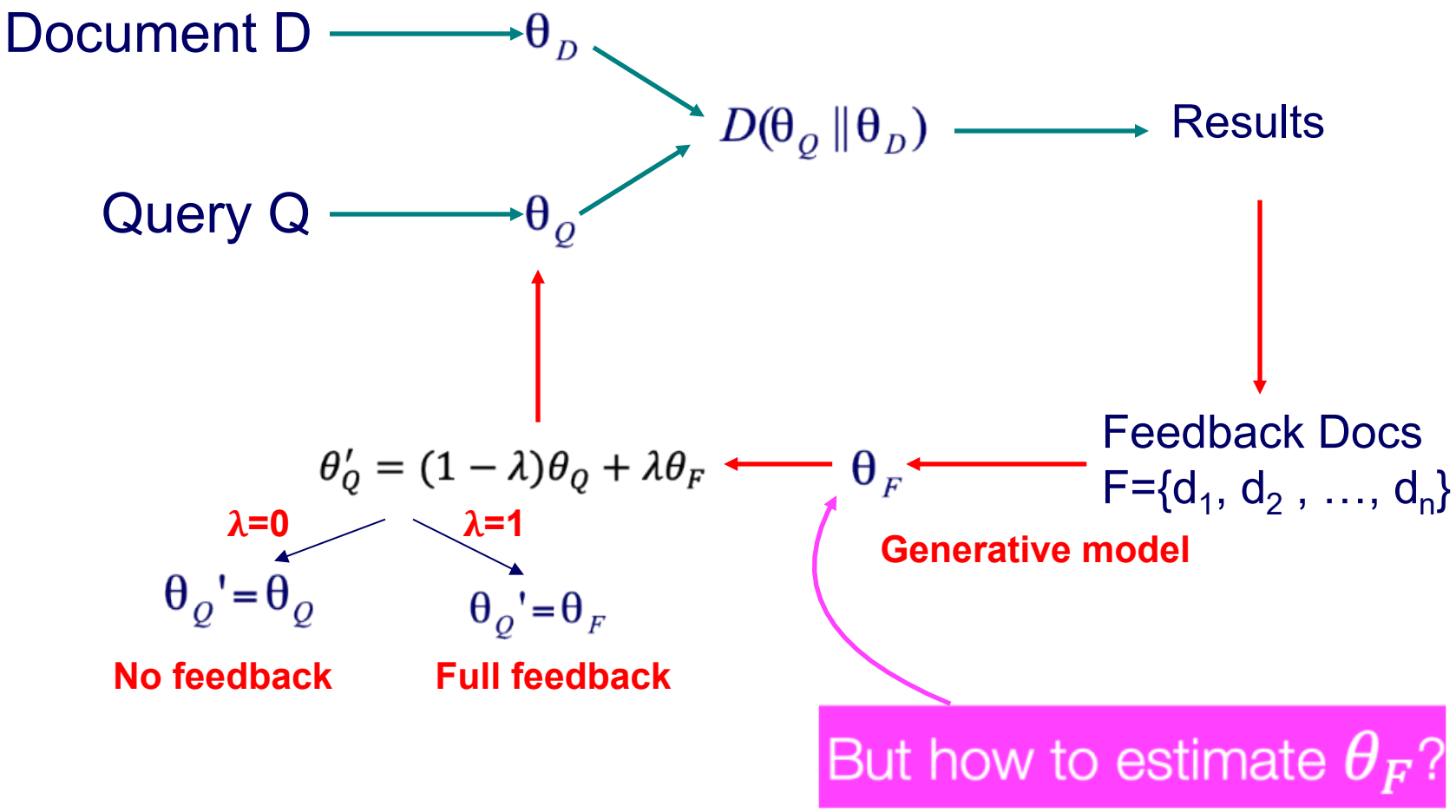
Q: Back to vector space model?
A: Kind of, but in different perspective.

Feedback in language models

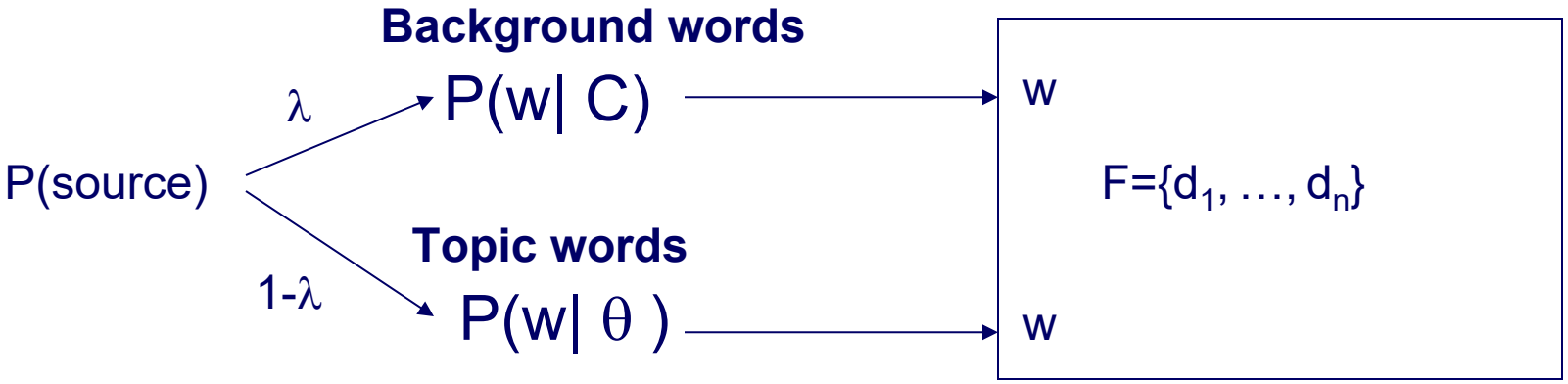
$$sim(q,d) \approx \sum_{w \in d, p(w|\theta_Q)>0} [p(w|\hat{\theta}_Q) \log \frac{p_{seen}(w|d)}{\alpha_d p(w|C)}] + \log \alpha_d$$

- The query is too sparse to estimate a robust language model
- Update $\hat{\theta}_Q$ based on relevant (or pseudo-relevant) documents
- Learn a language model from relevant documents then interpolate it with the original query (smoothing)

Feedback as Model Interpolation



Generative Mixture Model



$$\log p(F|\theta) = \sum_i \sum_w c(w;d_i) \log[(1-\lambda)p(w|\theta) + \lambda p(w|C)]$$

λ = Noise in feedback documents

How to Estimate θ_F?

Known
Background
$p(w|C)$

the 0.2
a 0.1
we 0.01
to 0.02
…
text 0.0001
mining 0.00005
…

Unknown
query topic
$p(w|\theta_F)$=?

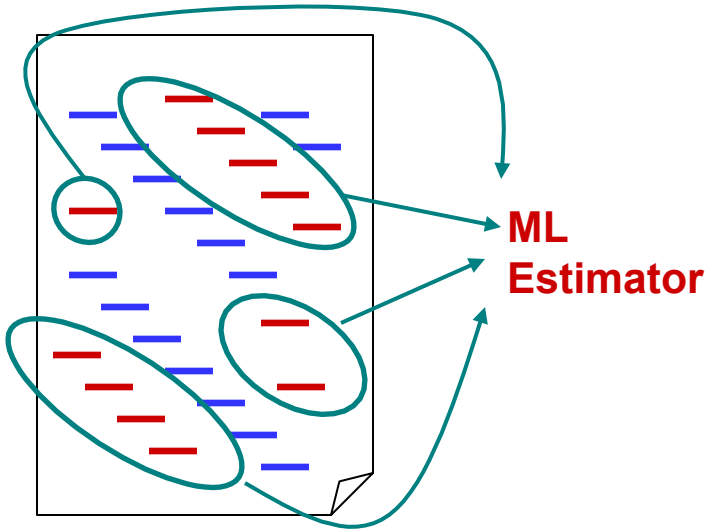
"Text mining"

…
text =?
mining =?
association =?
word =?
…

λ=0.7

λ=0.3

Observed
Doc(s)



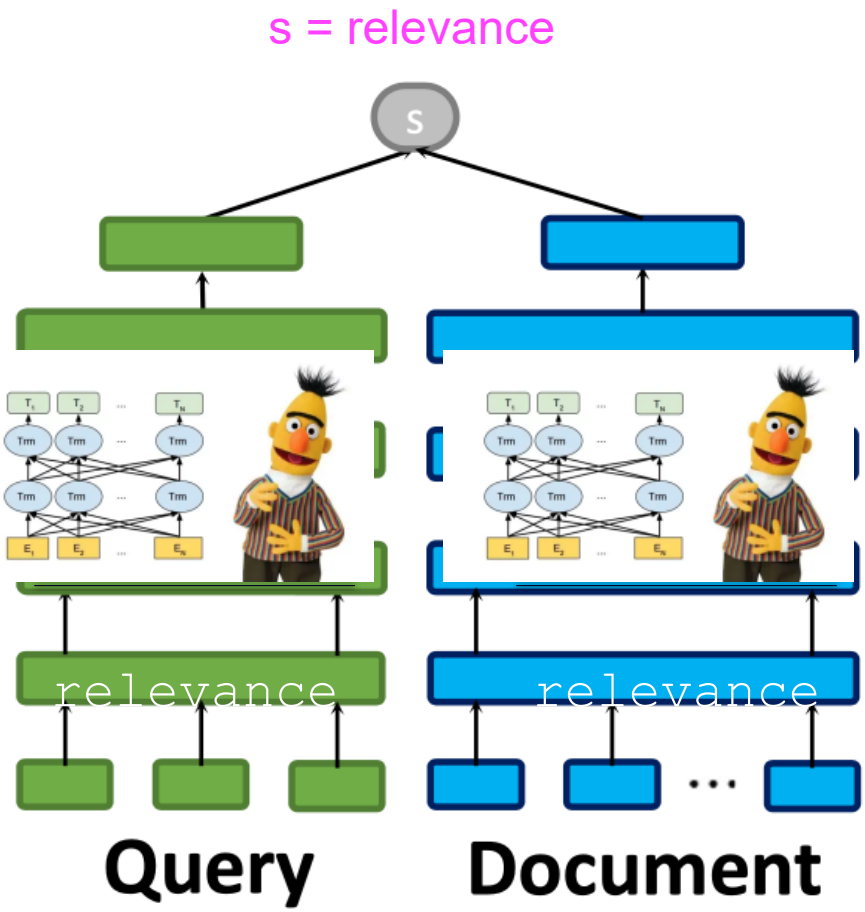
We can do this if
we know
the identity of each word ...

Discussion sessions

- Discussion sessions will cover how we can learn how you can do this.

Feedback in Deep Learning IR Systems

Bi-encoder retrieval for IR

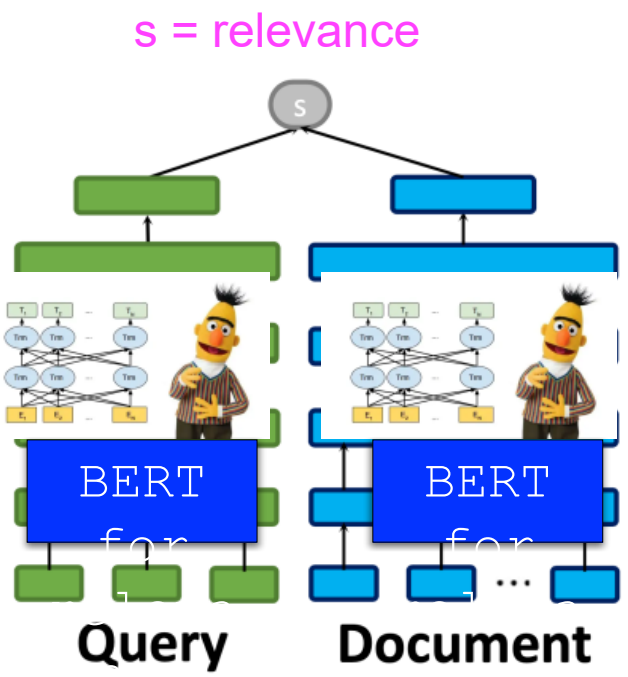


How might we incorporate pseudo feedback in neural models?

- Use Rocchio feedback!
  - Remember that dense representations are also vectors.

$$\vec{q}_m = \alpha\vec{q} + \frac{\beta}{|D_r|}\sum_{\forall \vec{d}_i \in D_r} \vec{d}_i - \frac{\gamma}{|D_n|}\sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

- Fancier models are possible



Concerns and Issues with Relevance Feedback

Relevance feedback: Assumptions

- When can relevance feedback enhance recall?
- Assumption A1: The user knows the terms in the collection well enough for an initial query.
- Assumption A2: Relevant documents contain similar terms (so I can “hop” from one relevant document to a different one when giving relevance feedback).

Violation of A1

- Assumption A1: The user knows the terms in the collection well enough for an initial query.
- Violation: Mismatch of searcher's vocabulary and collection vocabulary
- Example: cosmonaut / astronaut

Violation of A2

- Assumption A2: Relevant documents are similar.
- Example for violation: [contradictory government policies]
- Several unrelated “prototypes”
  - Subsidies for tobacco farmers vs. anti-smoking campaigns
  - Aid for developing countries vs. high tariffs on imports from developing countries
- Relevance feedback on tobacco docs will not help with finding docs on developing countries.

Relevance feedback: Evaluation

- Pick one of the evaluation measures from earlier lectures, e.g., precision in top 10: P@10
- Compute P@10 for original query q0
- Compute P@10 for modified relevance feedback query q1
- In most cases: q1 is spectacularly better than q0!
- Is this a fair evaluation?

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: Problems

- Relevance feedback is expensive.
- Relevance feedback creates long modified queries.
- Long queries are expensive to process.
- Users are reluctant to provide explicit feedback.
- It's often hard to understand why a particular document was retrieved after applying relevance feedback.
- The search engine Excite had full relevance feedback at one point, but abandoned it later.

What Should You Know

- Types of Query modification
- Feedback as an approach to query modification
- The difference between relevance feedback and pseudo-feedback
- Rocchio Feedback
- The KL-divergence retrieval formula as a generalization of the query likelihood method
- Feedback in Deep Learning IR Systems
- Some concerns and issues with relevance feedback