

Monsoon 2020

10 - Relevance Feedback

I n f o r m a t i o n

R e t r i e v a l

by

Dr. Rajendra Prasath



Indian Institute of Information Technology, Sri City, Chittoor
Sri City – 517 646, Andhra Pradesh, India

✧ Topics Covered So Far

- ✧ Bi-Word Index
- ✧ Wild Card Queries
- ✧ Permuterm Index
- ✧ K-gram Index ($k = 2$ □ Bigram Index)
- ✧ Spell Correction
- ✧ Term Weighting
- ✧ Vector Space Models

Now: Relevance Feedback

Recap: Overview

- ✧ Why Ranked Retrieval?
- ✧ Term Frequency
- ✧ Term Weighting
- ✧ TF-IDF Weighting
- ✧ The Vector Space Model

Scoring as the basis of ranked retrieval

- ✧ Rank documents such that more relevant documents higher than less relevant document
- ✧ How do we do follow?
 - ✧ Accomplish a ranking of the documents in the collection with respect to a query?
- ✧ Assign a score to each query-document pair, say in $[0, 1]$
- ✧ This score measures how well document and query “**match**”

Query – Docs matching scores

- ✧ How do we compute the score of a query-document pair?
- ✧ Let's start with a one-term query.
- ✧ If the query term does not occur in the document: score should be 0.
- ✧ **Term Frequency:**
 - ✧ The more matching of the query term in the document ☐ higher the score

Measure of Closeness of Vectors

- ✦ **Measure the closeness between two vectors**
- ✦ Two texts are semantically related if they share some vocabulary
 - ✦ More Vocabulary they share, the stronger is the relationship
- ✦ This implies that the measure of closeness increases with the number of words matches between two texts
- ✦ If matching terms are important then vectors should be considered closer to each other

Modern Vector Space Models

- ✧ The length of the sub-vector in dimension - i is used to represent the importance or the weigh of word – i in a text
- ✧ Words that are absent in a text get a weight – 0 (zero)
- ✧ Apply **Vector Inner Product** measure between two vectors:
- ✧ This vector inner product increases:
 - ✧ # words match between two texts
 - ✧ Importance of the matching terms

Basics

- ✧ The user issues a (short, simple) query.
- ✧ The search engine returns a set of documents.
- ✧ User marks some docs as relevant (possibly 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 (and possibly also better precision).
- ✧ Limited form of RF - “more like this” or “findsimilar”

Relevance Basics

- ✧ Developed in the late 60s or early 70s.
 - ✧ It was developed using the VSM as its basis.
 - ✧ Therefore, we represent documents as points in a high-dimensional term space.
 - ✧ Uses centroids to calculate the center of a set of documents
-
- ✧ **Improving Recall**
 - Local: Do a “local”, on-demand analysis for a user query
 - Main local method: 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.

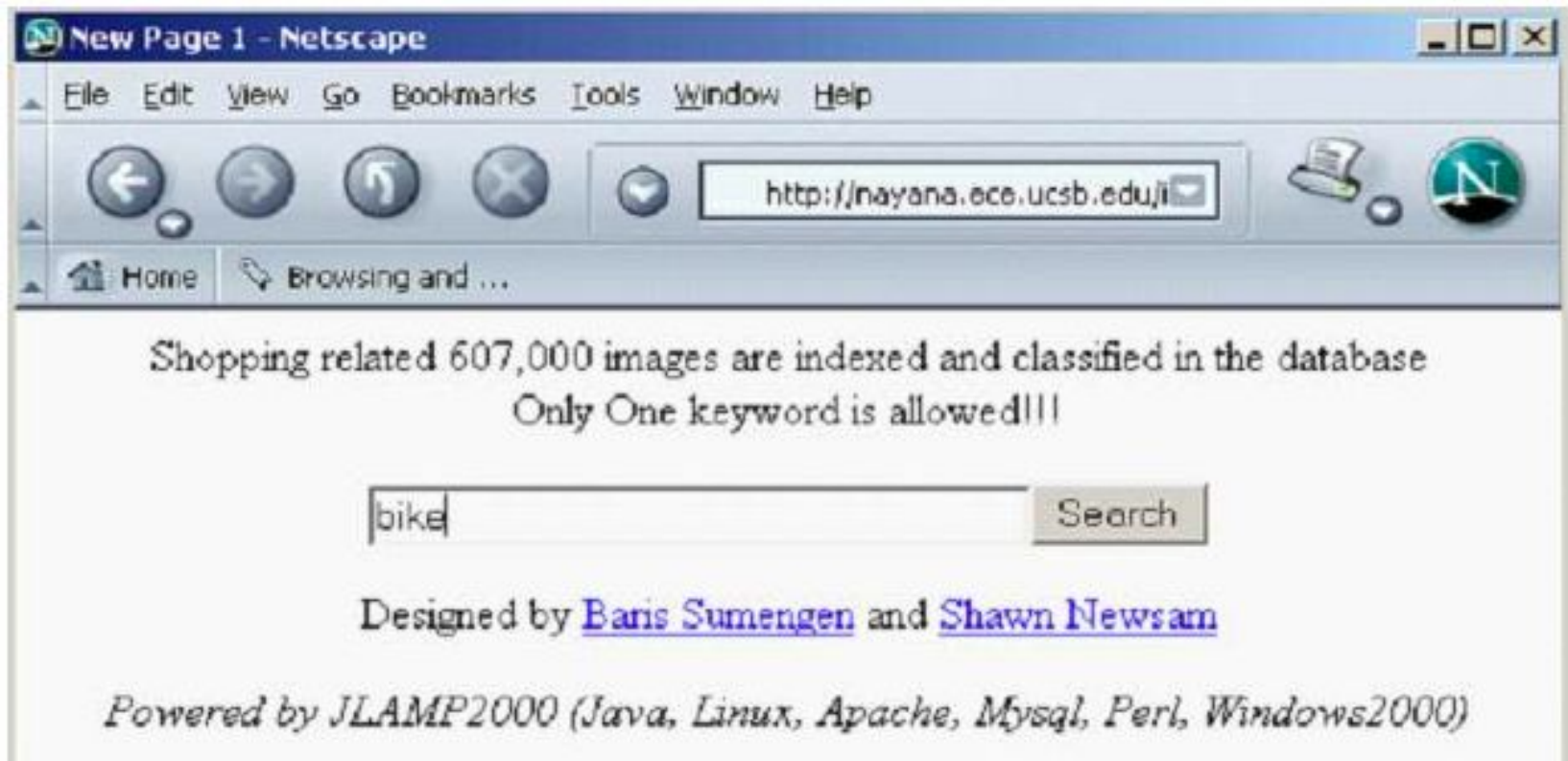


Relevance feedback

- 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.















Relevance feedback: Example



Results for initial query

Results for initial query







Buttons: Browse Search Prev Next Random

					
(144473, 16459)	(144457, 252140)	(144456, 262037)	(144456, 262063)	(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
					
(144403, 264544)	(144483, 265153)	(144510, 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







User feedback: Select what is relevant

Interface for user feedback on bike-related images. The interface includes a navigation bar with buttons: Browse, Search, Prev, Next, and Random. Below the navigation bar, there are two rows of image thumbnails, each with associated metadata.

Row 1:

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













Row 2:

-  (144483, 264644)
0.0
0.0
0.0
-  (144483, 265153)
0.0
0.0
0.0
-  (144518, 257752)
0.0
0.0
0.0
-  (144538, 525937)
0.0
0.0
0.0
-  (144456, 240611)
0.0
0.0
0.0
-  (144456, 250064)
0.0
0.0
0.0

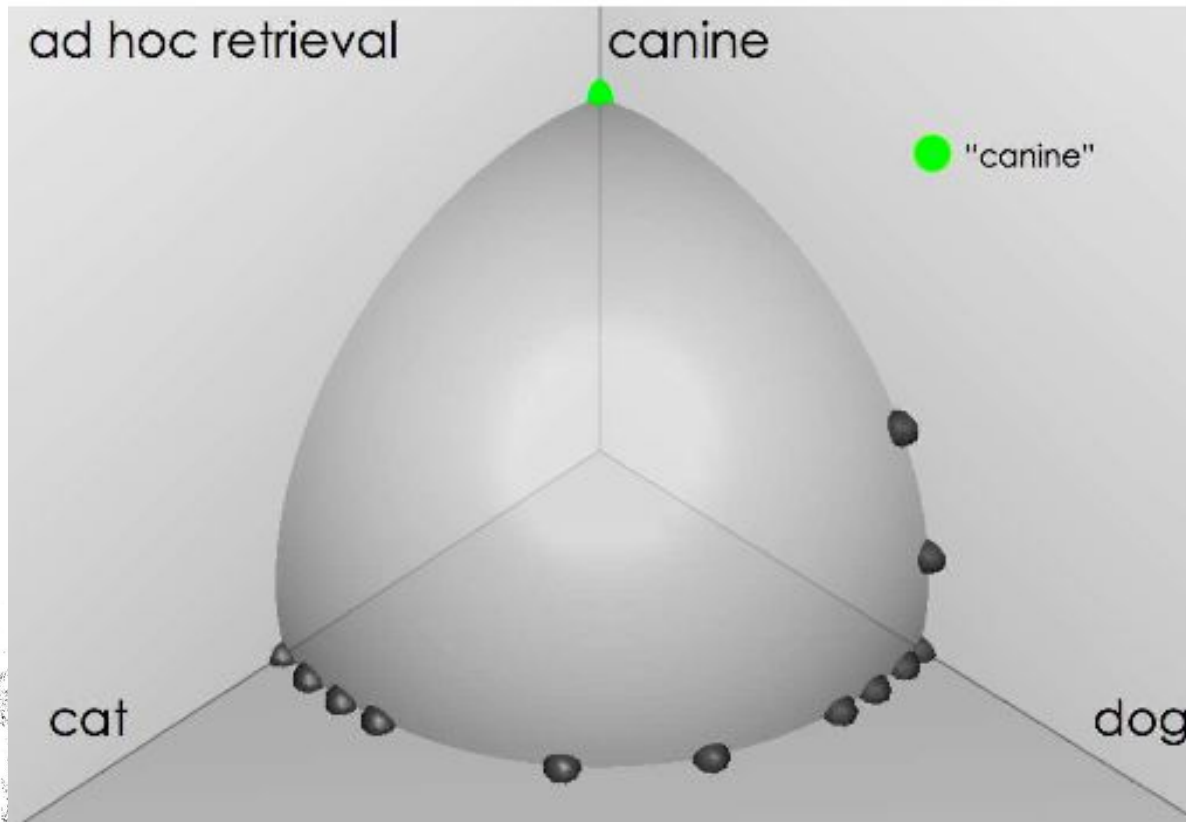
Results after relevance feedback

Interface showing results after relevance feedback, displaying a grid of images and associated numerical data.

Navigation buttons: Browse, Search, Prev, Next, Random

					
(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

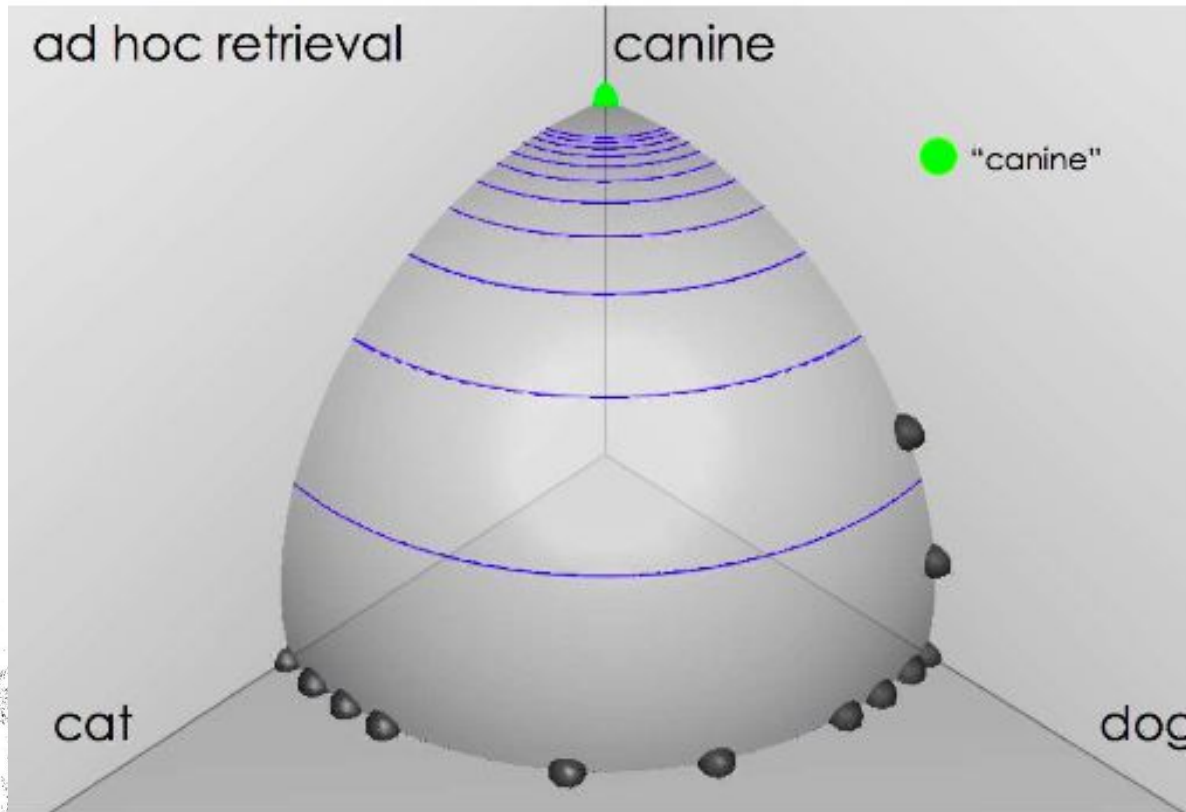
Vector space example: query “canine”



Source:

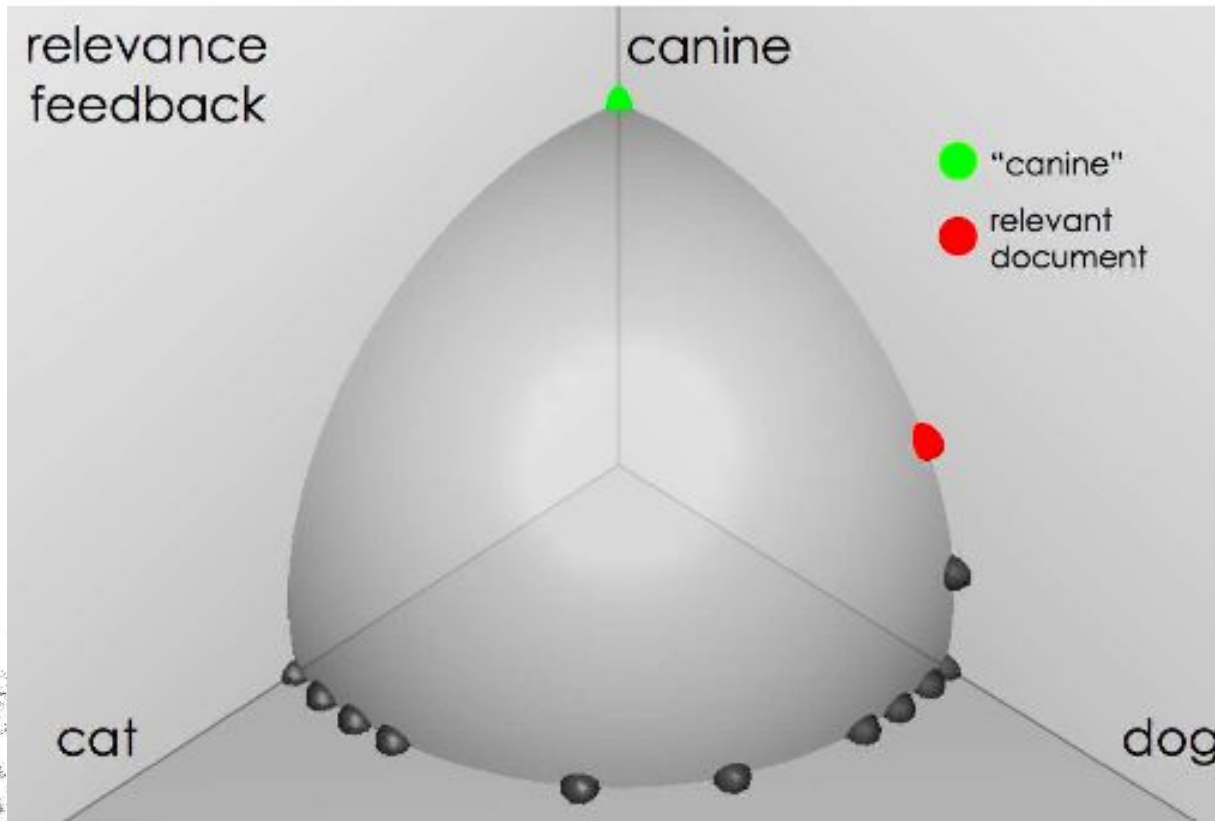
Fernando Díaz

Similarity of docs to query “canine”



Source:
Fernando Díaz

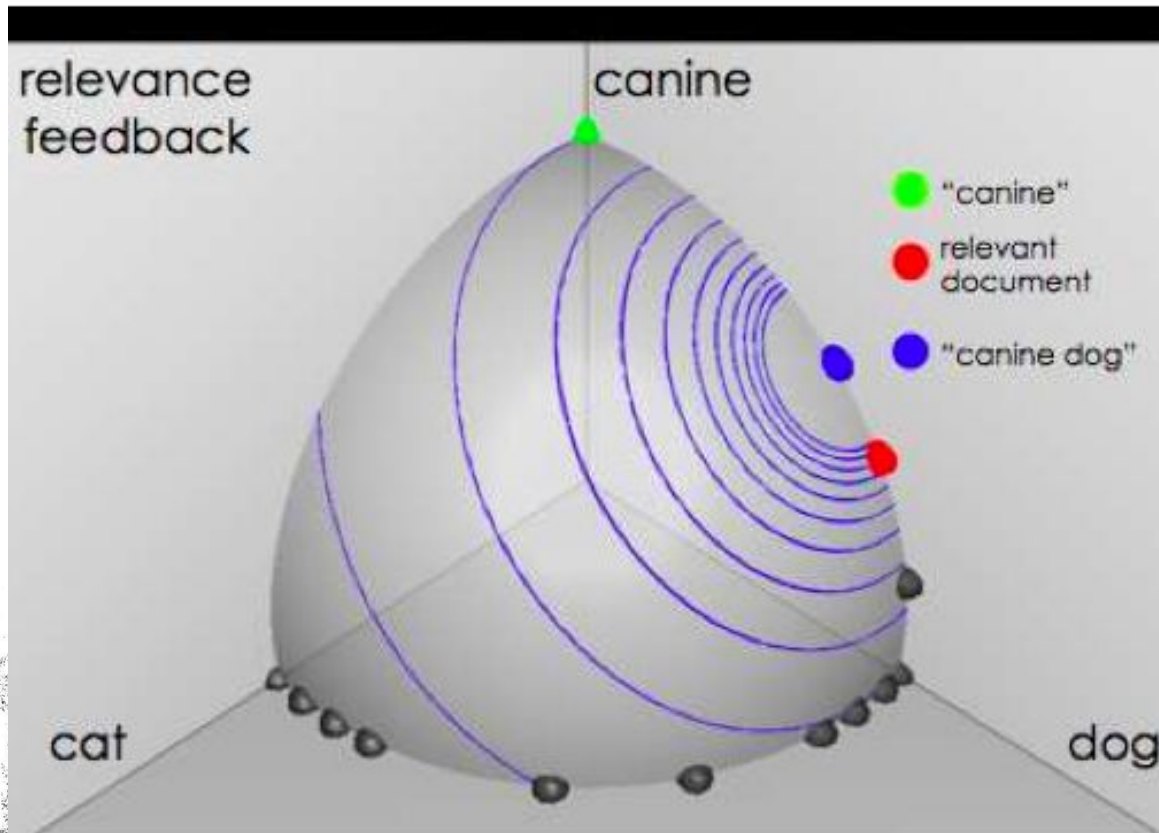
User feedback: Select relevant documents



Source:

Fernando Díaz

Results after relevance feedback



Source:

Fernando Díaz

Example: A real 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 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

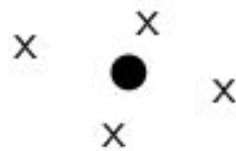
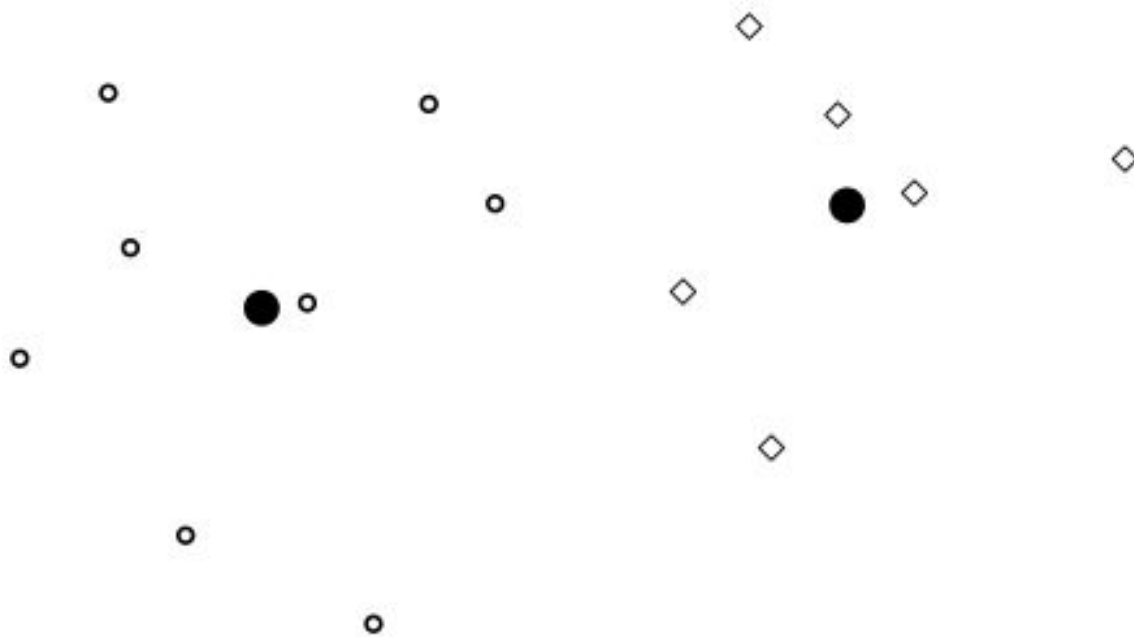
Key concept for relevance feedback: Centroid

- 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.
- 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' algorithm

- The Rocchio' algorithm implements relevance feedback in the vector space model.
- Rocchio' chooses the query \vec{q}_{opt} that maximizes

$$\vec{q}_{opt} = \arg \max_{\vec{q}} [\text{sim}(\vec{q}, \mu(D_r)) - \text{sim}(\vec{q}, \mu(D_{nr}))]$$

D_r : set of relevant docs; D_{nr} : set of nonrelevant docs

- Intent: \vec{q}_{opt} is the vector that separates relevant and nonrelevant 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

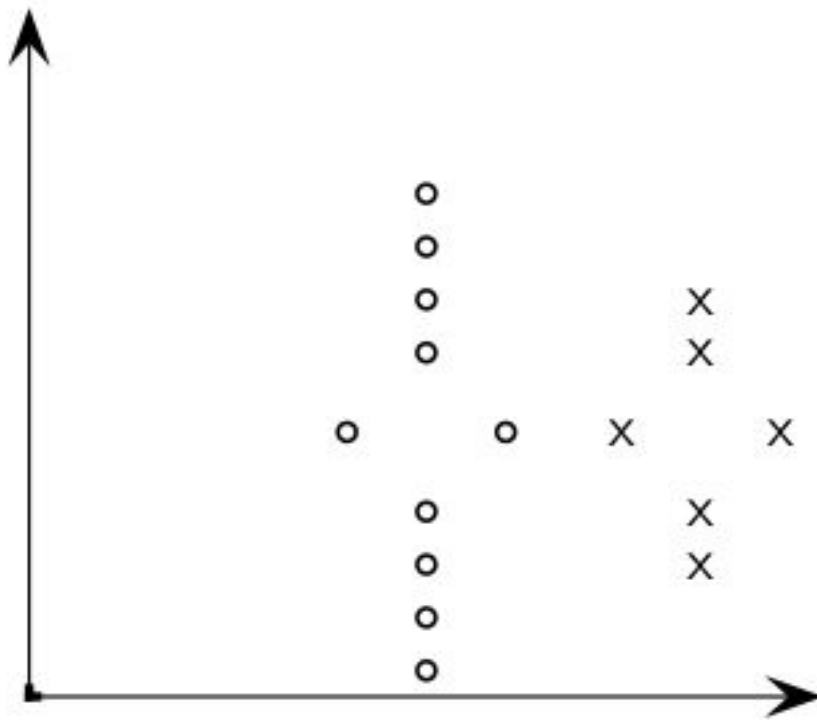
- The optimal query vector is:

$$\begin{aligned}\vec{q}_{opt} &= \mu(D_r) + [\mu(D_r) - \mu(D_{nr})] \\ &= \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j + \left[\frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j \right]\end{aligned}$$

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

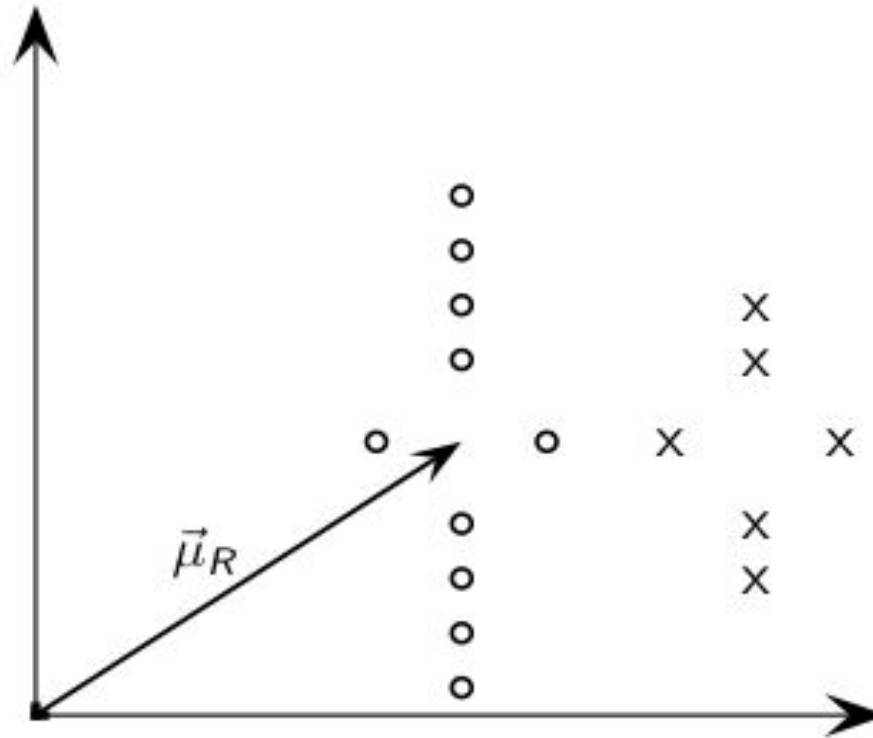


Exercise: Compute Rocchio' vector



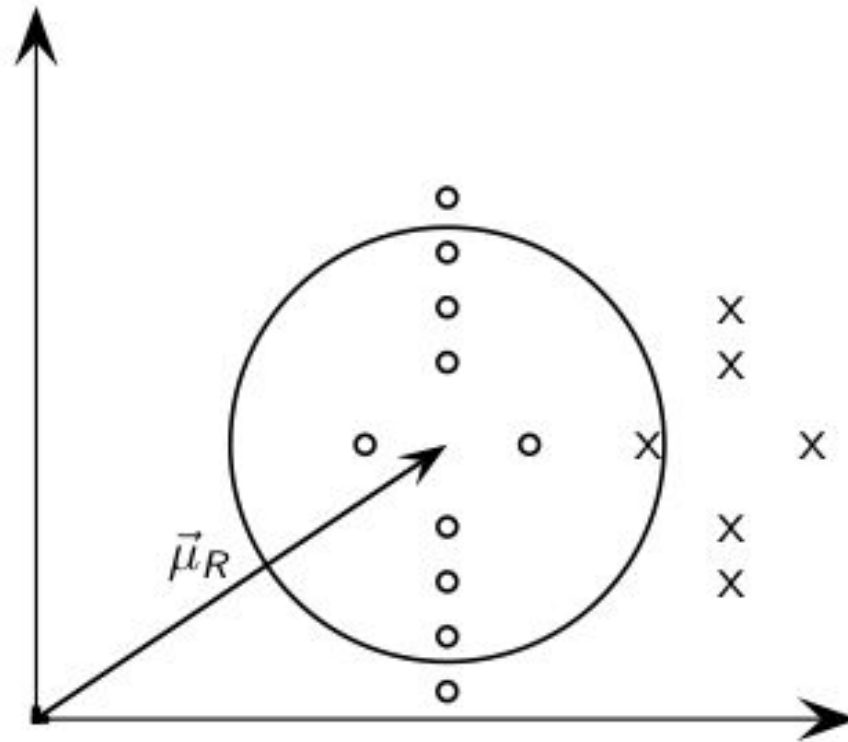
circles: relevant documents, Xs: nonrelevant documents

Rocchio' illustrated



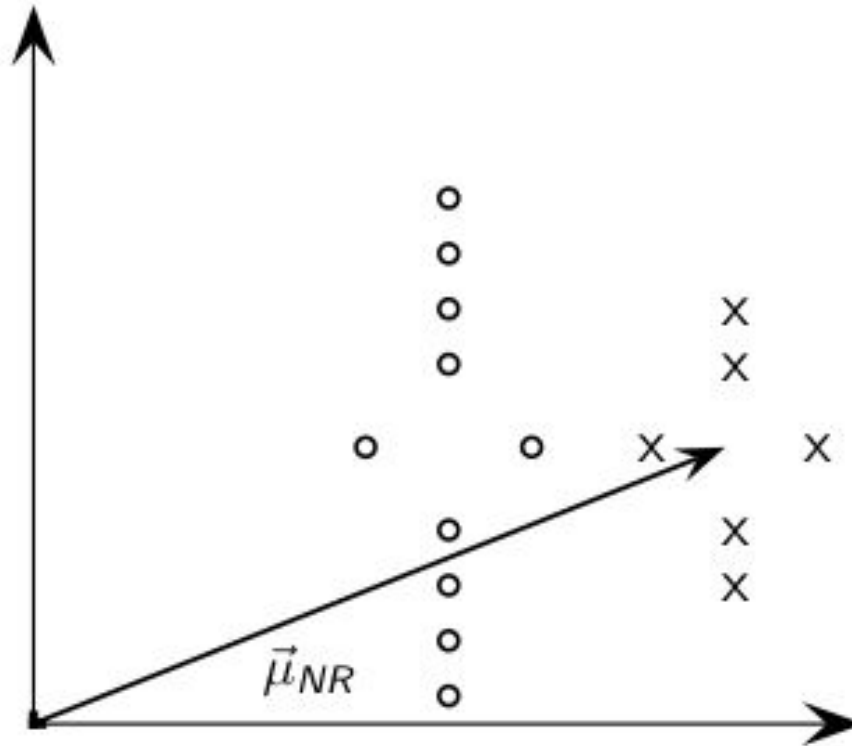
$\vec{\mu}_R$: centroid of relevant documents

Rocchio' illustrated



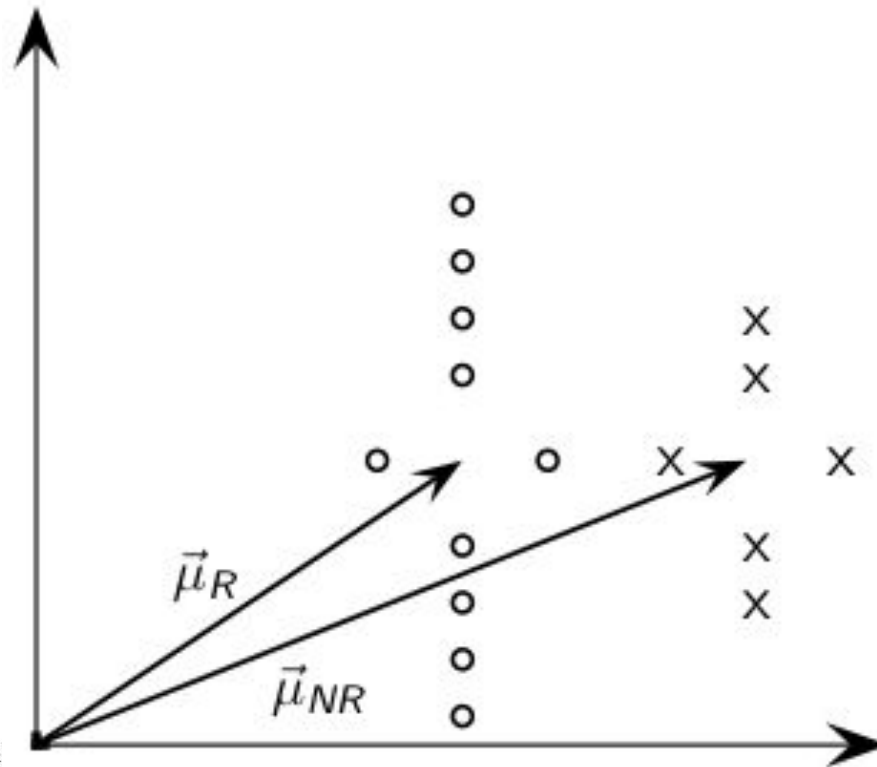
$\vec{\mu}_R$ does not separate relevant / nonrelevant.

Rocchio' illustrated

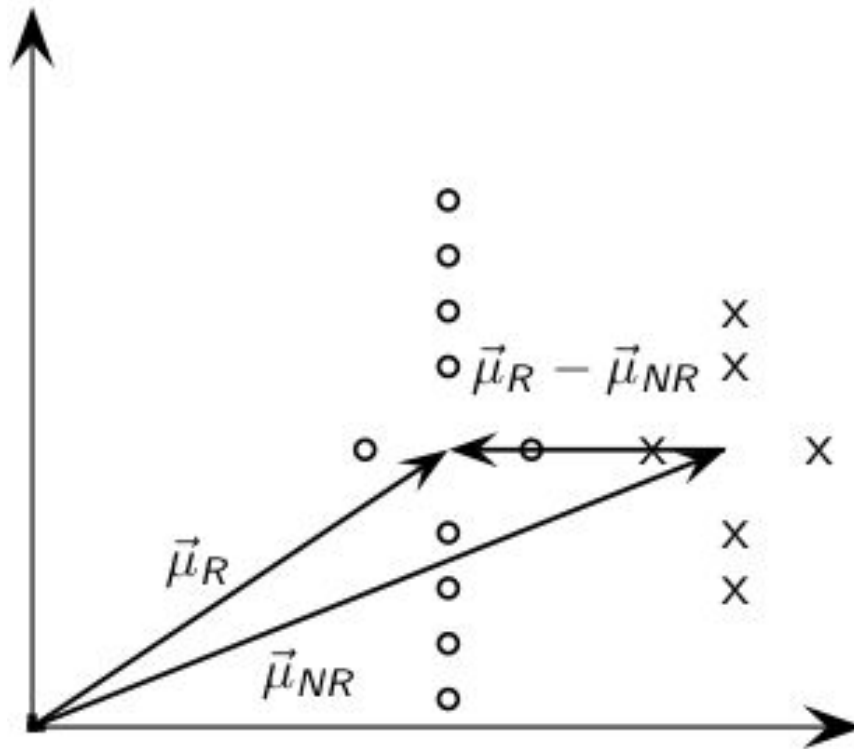


$\vec{\mu}_{NR}$: centroid of nonrelevant documents.

Rocchio' illustrated

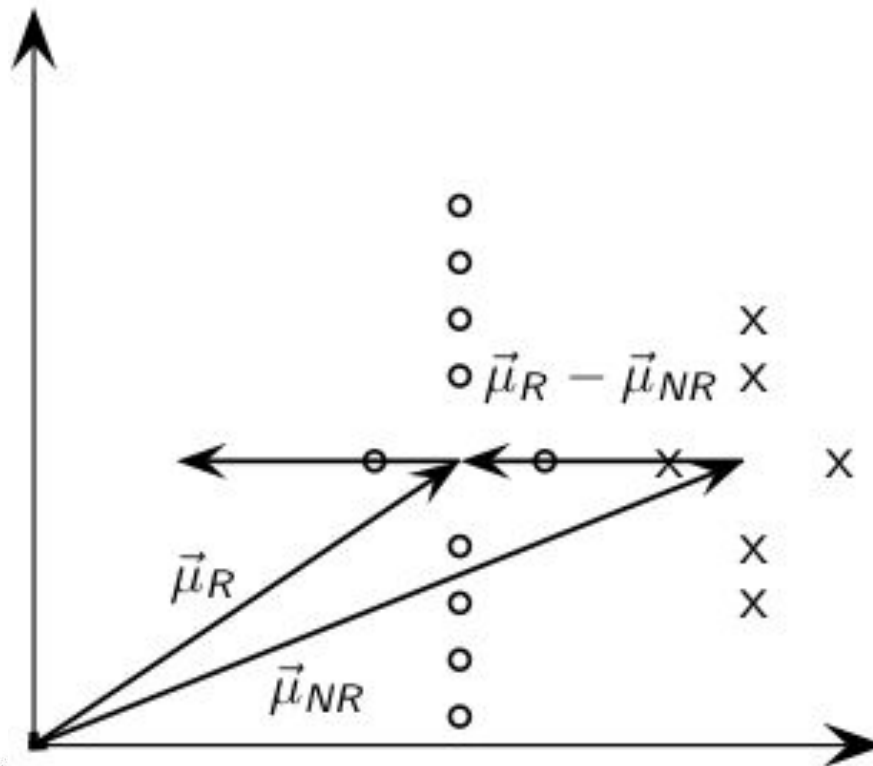


Rocchio' illustrated



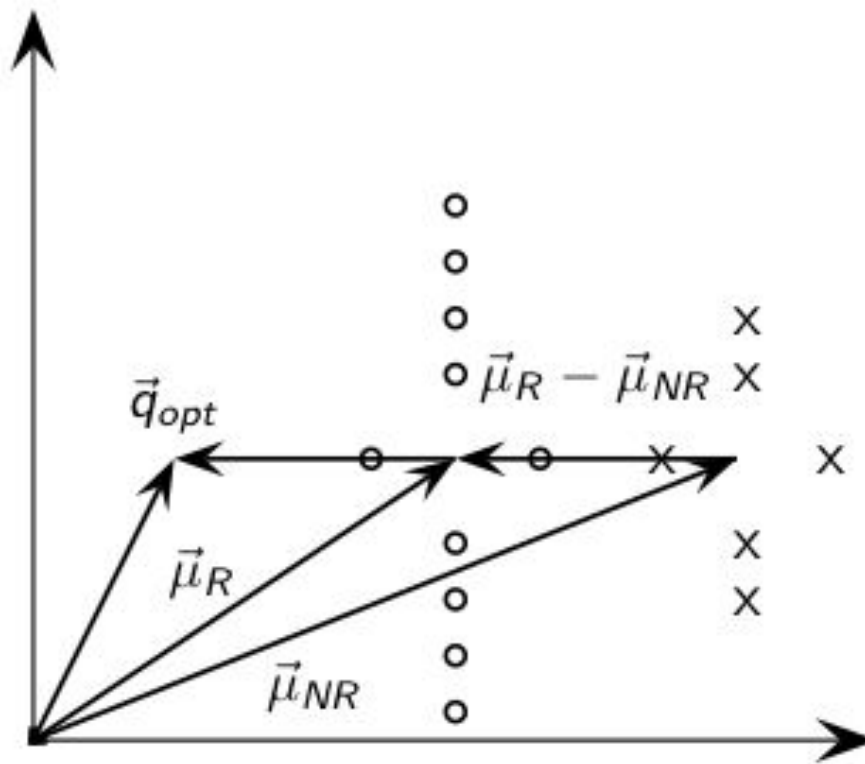
$\vec{\mu}_R - \vec{\mu}_{NR}$: difference vector

Rocchio' illustrated



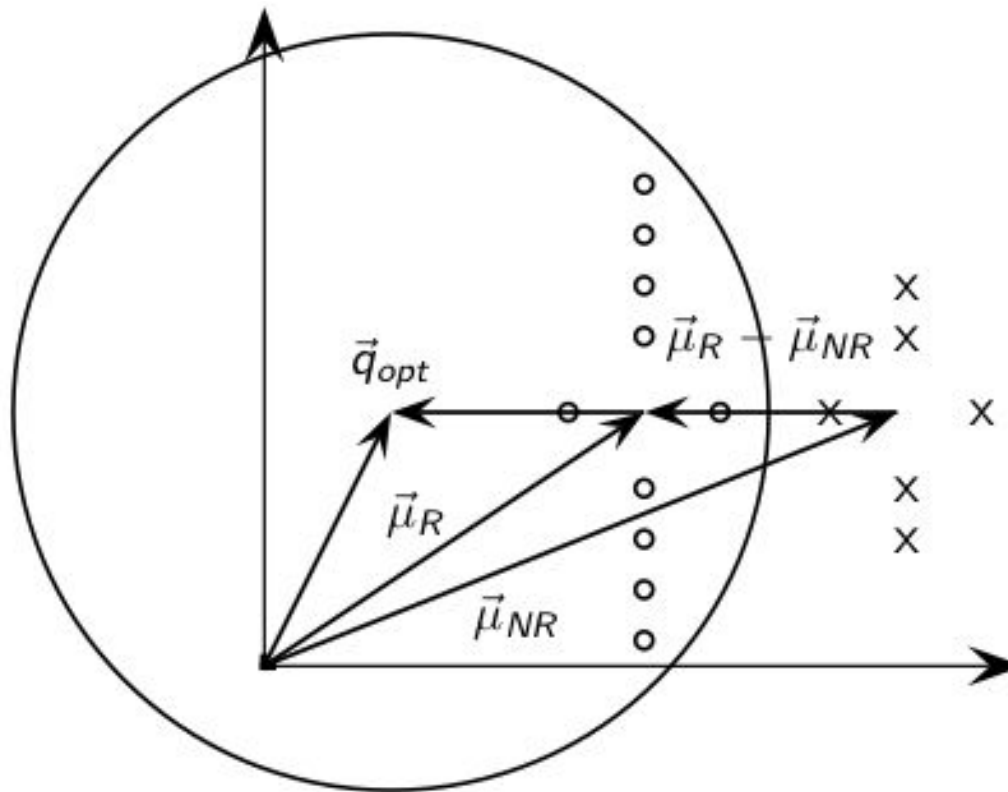
Add difference vector to $\vec{\mu}_R$...

Rocchio' illustrated



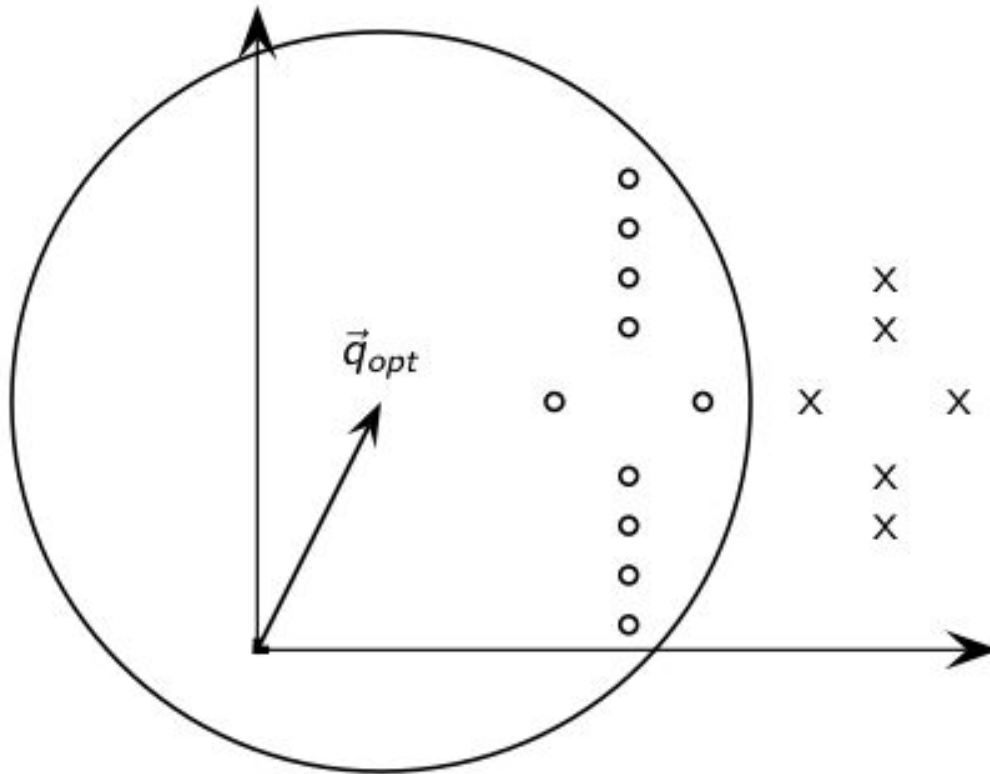
... to get \vec{q}_{opt}

Rocchio' illustrated



\vec{q}_{opt} separates relevant / nonrelevant perfectly.

Rocchio' illustrated



\vec{q}_{opt} separates relevant / nonrelevant perfectly.

Rocchio 1971 algorithm (SMART)

Used in practice:

$$\begin{aligned}\vec{q}_m &= \alpha \vec{q}_0 + \beta \mu(D_r) - \gamma \mu(D_{nr}) \\ &= \alpha \vec{q}_0 + \beta \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j\end{aligned}$$

q_m : modified query vector; q_0 : original query vector; D_r and D_{nr} : sets of known relevant and nonrelevant 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.

Positive vs. negative relevance feedback

- Positive feedback is more valuable than negative feedback.
- For example, set $\beta = 0.75$, $\gamma = 0.25$ to give higher weight to positive feedback.
- Many systems only allow positive 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
e.g., precision in top 10: $P@10$
- Compute $P@10$ for original query q_0
- Compute $P@10$ for modified relevance feedback query q_1
- In most cases: q_1 is spectacularly better than q_0 !
- Is this a fair evaluation?



Relevance feedback: Evaluation

- Fair evaluation must be on “residual” collection: docs not yet judged by user.
- Studies have shown that relevance feedback is successful when evaluated this way.
- Empirically, one round of relevance feedback is often very useful. Two rounds are marginally useful.



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.



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*.

Pseudo Relevance Feedback

- ✧ Uses Feedback from user activities
 - Web Click through data
 - Items searched from Search History
 - Profile information
 - and so on



Summary

In this class, we focused on:

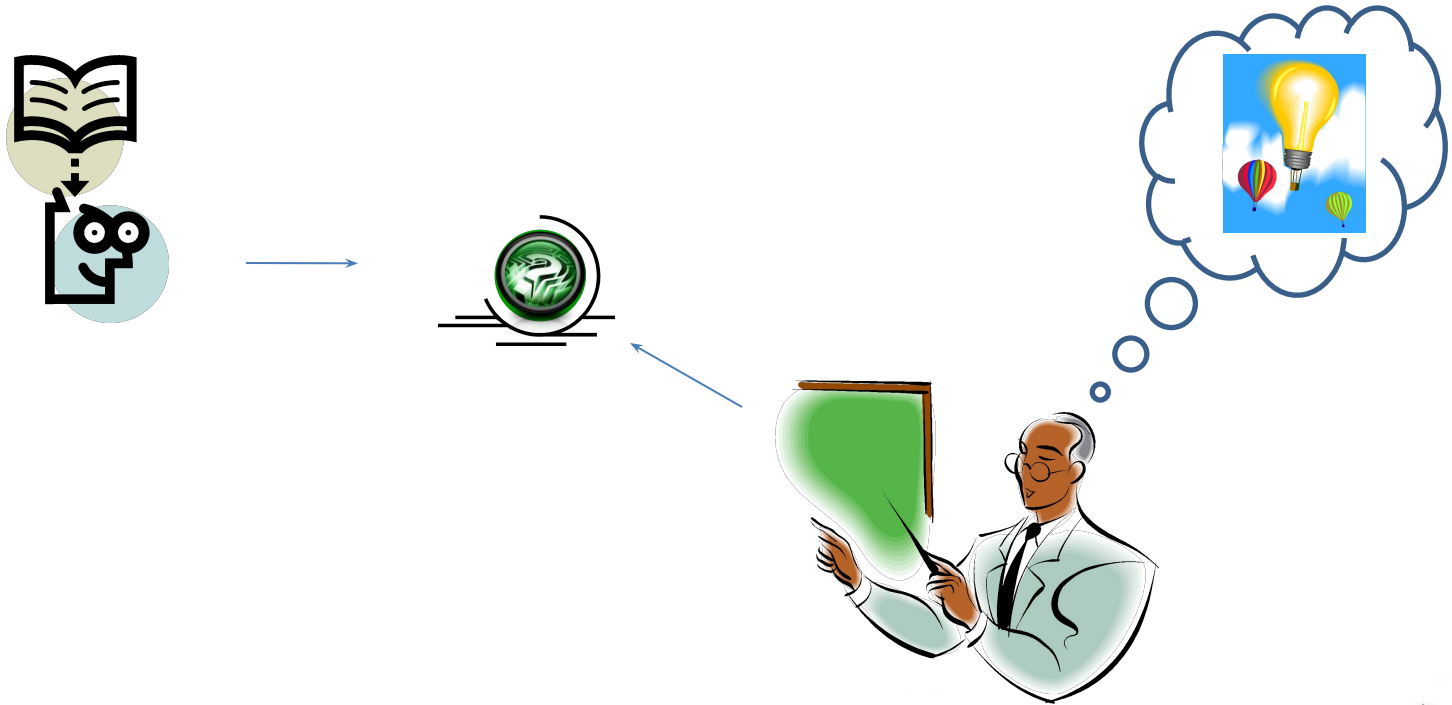
- (a) Words / Terms / Lexical Units
- (b) Preparing Term – Document matrix
- (c) Boolean Retrieval
- (d) Inverted Index Construction
 - i. Computational Cost
 - ii. Managing Bigger Collections
 - iii. How much storage is required?
 - iv. Boolean Queries: Exact match

Acknowledgements

Thanks to ALL RESEARCHERS:

1. Introduction to Information Retrieval Manning, Raghavan and Schütze, Cambridge University Press, 2008.
2. Search Engines Information Retrieval in Practice W. Bruce Croft, D. Metzler, T. Strohman, Pearson, 2009.
3. Information Retrieval Implementing and Evaluating Search Engines Stefan Büttcher, Charles L. A. Clarke and Gordon V. Cormack, MIT Press, 2010.
4. Modern Information Retrieval Baeza-Yates and Ribeiro-Neto, Addison Wesley, 1999.
5. Many Authors who contributed to SIGIR / WWW / KDD / ECIR / CIKM / WSDM and other top tier conferences
6. Prof. Mandar Mitra, Indian Statistical Institute, Kolkata (<https://www.isical.ac.in/~mandar/>)

Thanks ...



... Questions ???