Introduction to Information Retrieval

http://informationretrieval.org

IIR 9: Relevance Feedback & Query Expansion

Hinrich Schütze

Institute for Natural Language Processing, Universität Stuttgart

2008.06.03

Overview

Plan for this lecture

- First: Recap

Plan for this lecture

- First: Recap
- Main topic today: How can we improve recall in search?

Plan for this lecture

- First: Recap
- Main topic today: How can we improve recall in search?
  - "aircraft" in query doesn't match with "plane" in document

Plan for this lecture

- First: Recap
- Main topic today: How can we improve recall in search?
  - "aircraft" in query doesn't match with "plane" in document
  - "heat" in query doesn't match with "thermodynamics" in document

Plan for this lecture

- First: Recap
- Main topic today: How can we improve recall in search?
  - "aircraft" in query doesn't match with "plane" in document
  - "heat" in query doesn't match with "thermodynamics" in document
- Options for improving recall

Plan for this lecture

- First: Recap
- Main topic today: How can we improve recall in search?
  - "aircraft" in query doesn't match with "plane" in document
  - "heat" in query doesn't match with "thermodynamics" in document
- Options for improving recall
  - Local methods: Do a "local", on-demand analysis for a user query

Plan for this lecture

- First: Recap
- Main topic today: How can we improve recall in search?
  - "aircraft" in query doesn't match with "plane" in document
  - "heat" in query doesn't match with "thermodynamics" in document
- Options for improving recall
  - Local methods: Do a "local", on-demand analysis for a user query
  - Main local method: relevance feedback

Plan for this lecture

- First: Recap
- Main topic today: How can we improve recall in search?
  - "aircraft" in query doesn't match with "plane" in document
  - "heat" in query doesn't match with "thermodynamics" in document
- Options for improving recall
  - Local methods: Do a "local", on-demand analysis for a user query
  - Main local method: relevance feedback
  - Global methods: Do a global analysis once (e.g., of collection) to produce thesaurus

Plan for this lecture

- First: Recap
- Main topic today: How can we improve recall in search?
  - "aircraft" in query doesn't match with "plane" in document
  - "heat" in query doesn't match with "thermodynamics" in document
- Options for improving recall
  - Local methods: Do a "local", on-demand analysis for a user query
  - Main local method: relevance feedback
  - Global methods: Do a global analysis once (e.g., of collection) to produce thesaurus
  - Use thesaurus for query expansion

Google example query:
~hospital -hospital -hospitals

Overview

1. Recap
2. Relevance feedback: Basics
3. Relevance feedback: Details
4. Global query expansion

Outline

Relevance

- We will evaluate the quality of an information retrieval system and, in particular, its ranking algorithm with respect to relevance.
- A document is relevant if it gives the user the information she was looking for.
- To evaluate relevance, we need an evaluation benchmark with three elements:
  - A benchmark document collection
  - A benchmark suite of queries
  - An assessment of the relevance of each query-document pair

Relevance: query vs. information need

- The notion of "relevance to the query" is very problematic.
- Information need i: You are looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.
- Query q: WINE AND RED AND WHITE AND HEART AND ATTACK
- Consider document d': *He then launched into the heart of his speech and attacked the wine industry lobby for downplaying the role of red and white wine in drunk driving.*
- d' is relevant to the query q, but d' is not relevant to the information need i.
- User happiness/satisfaction (i.e., how well our ranking algorithm works) can only be measured by relevance to information needs, not by relevance to queries.

Precision and recall

- Precision (P) is the fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved})$$

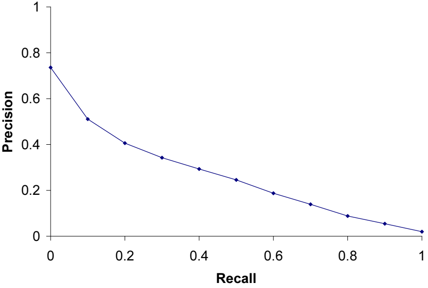
- Recall (R) is the fraction of relevant documents that are retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant})$$

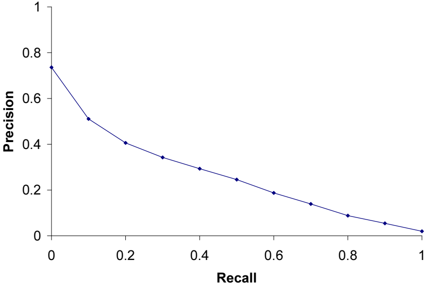
A combined measure: F

- F allows us to trade off precision against recall.
- Balanced F:
$$F_1 = \frac{2PR}{P + R}$$
- This is a kind of soft minimum of precision and recall.

Averaged 11-point precision/recall graph

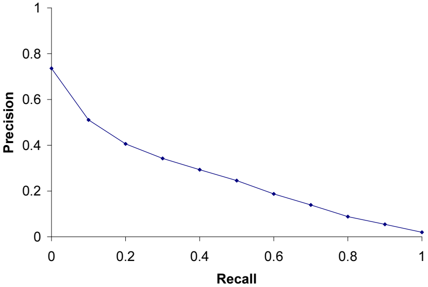


Averaged 11-point precision/recall graph



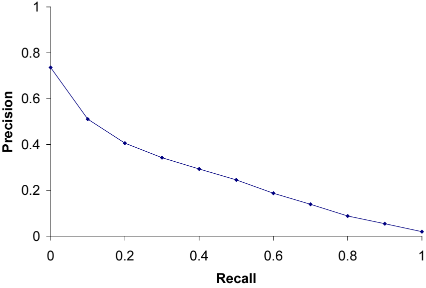
- This curve is typical of performance levels for the TREC benchmark.

Averaged 11-point precision/recall graph



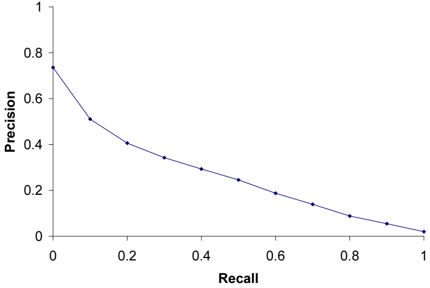
- This curve is typical of performance levels for the TREC benchmark.
- 70% chance of getting the first document right (roughly)

Averaged 11-point precision/recall graph



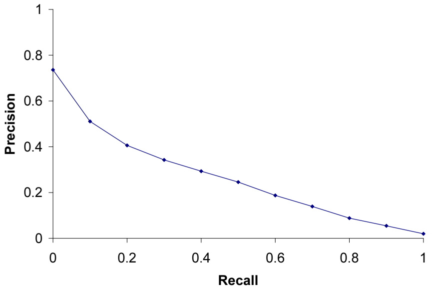
- This curve is typical of performance levels for the TREC benchmark.
- 70% chance of getting the first document right (roughly)
- When we want to look at at least 50% of all relevant documents, then for each relevant document we find, we will have to look at about two nonrelevant documents.

Averaged 11-point precision/recall graph



- This curve is typical of performance levels for the TREC benchmark.
- 70% chance of getting the first document right (roughly)
- When we want to look at at least 50% of all relevant documents, then for each relevant document we find, we will have to look at about two nonrelevant documents.
- That's not very good.

Averaged 11-point precision/recall graph



- This curve is typical of performance levels for the TREC benchmark.
- 70% chance of getting the first document right (roughly)
- When we want to look at at least 50% of all relevant documents, then for each relevant document we find, we will have to look at about two nonrelevant documents.
- That's not very good.
- High-recall retrieval is an unsolved problem.

Outline

Relevance feedback: Basic idea

- User issues a (short, simple) query

Relevance feedback: Basic idea

- User issues a (short, simple) query
- Search engine returns set of docs

Relevance feedback: Basic idea

- User issues a (short, simple) query
- Search engine returns set of docs
- User marks some docs as relevant, some as nonrelevant

Relevance feedback: Basic idea

- User issues a (short, simple) query
- Search engine returns set of docs
- User marks some docs as relevant, some as nonrelevant
- Search engine computes a new representation of information need – better than initial query

Relevance feedback: Basic idea

- User issues a (short, simple) query
- Search engine returns set of docs
- User marks some docs as relevant, some as nonrelevant
- Search engine computes a new representation of information need – better than initial query
- Search engine runs new query and returns new results

Relevance feedback: Basic idea

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

Relevance feedback: Basic idea

- User issues a (short, simple) query
- Search engine returns set of docs
- User marks some docs as relevant, some as nonrelevant
- Search engine computes a new representation of information need – better than initial query
- Search engine runs new query and returns new results
- New results have (hopefully) better recall.
- We can iterate this.

Relevance feedback: Basic idea

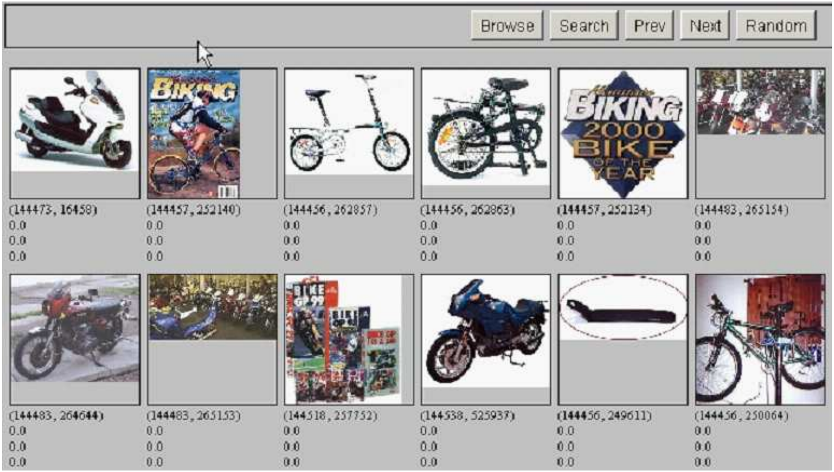
- User issues a (short, simple) query
- Search engine returns set of docs
- User marks some docs as relevant, some as nonrelevant
- Search engine computes a new representation of information need – better than initial query
- Search engine runs new query and returns new results
- New results have (hopefully) better recall.
- We can iterate this.
- We will use the term *ad hoc retrieval* to refer to regular retrieval without relevance feedback.

Relevance feedback: Basic idea

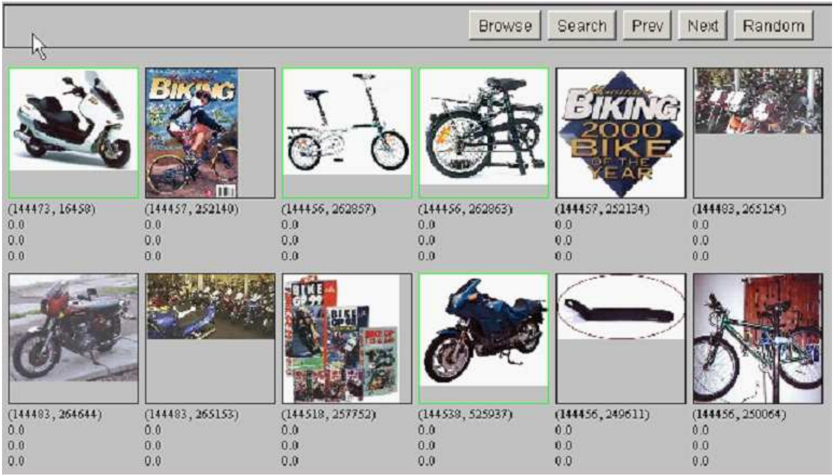
- User issues a (short, simple) query
- Search engine returns set of docs
- User marks some docs as relevant, some as nonrelevant
- Search engine computes a new representation of information need – better than initial query
- Search engine runs new query and returns new results
- New results have (hopefully) better recall.
- We can iterate this.
- 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

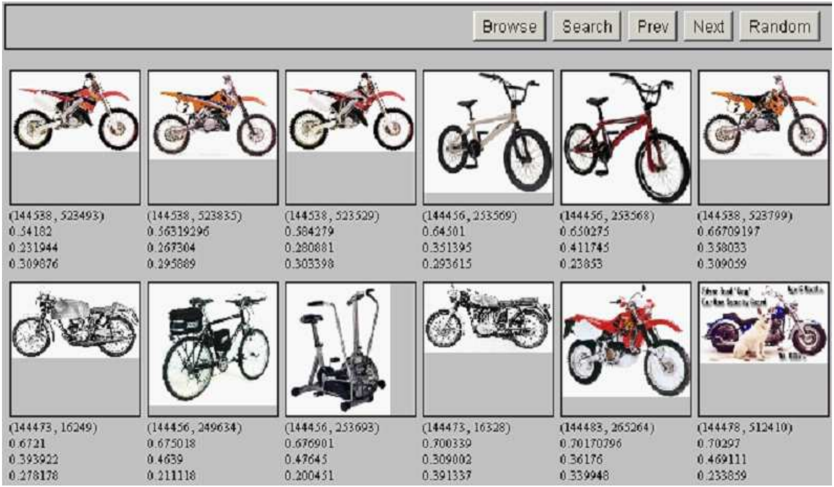
Results for initial query



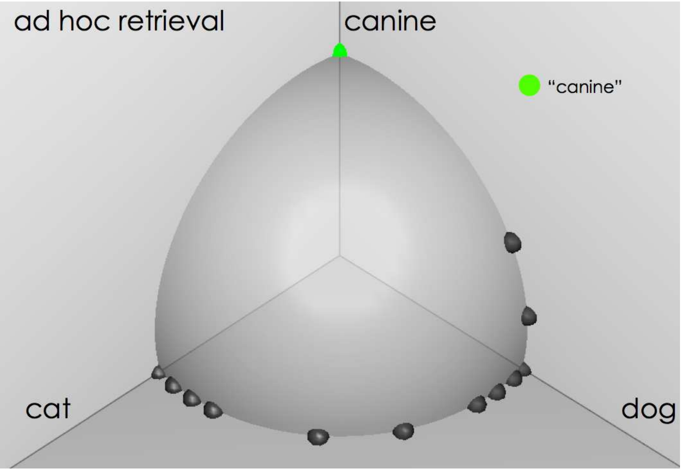
User feedback: Select what is relevant



Results after relevance feedback

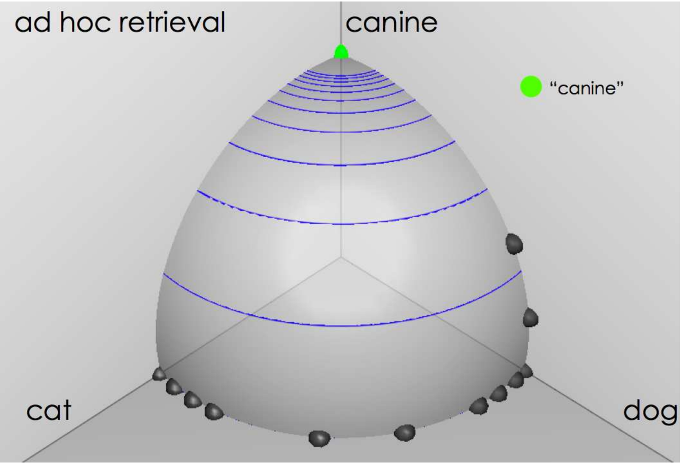


Ad hoc retrieval for query "canine" (1)



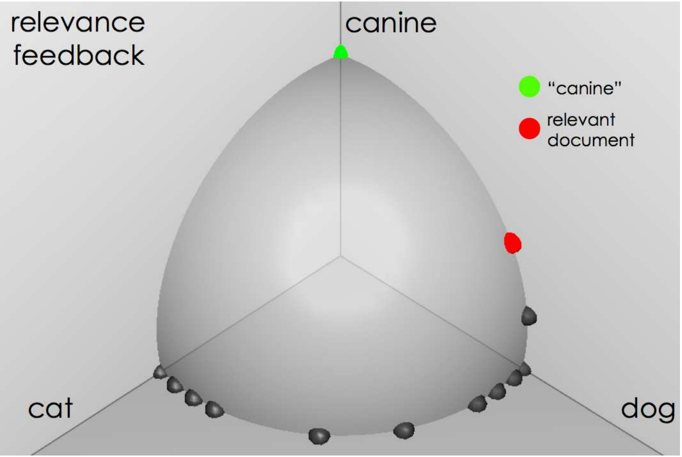
source:
Fernando Díaz

Ad hoc retrieval for query "canine" (2)



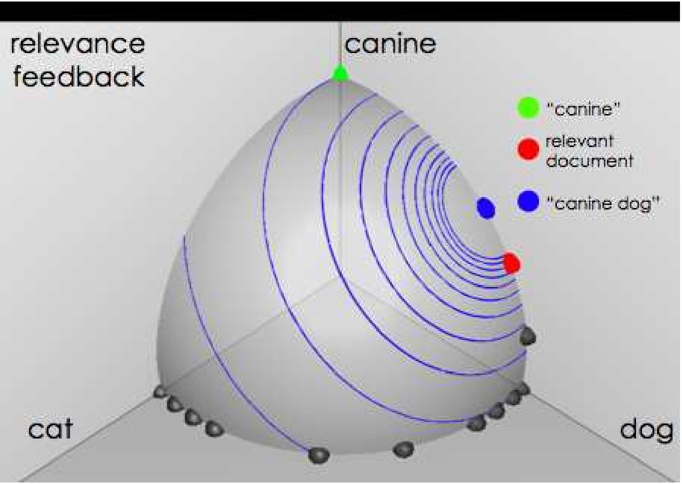
source:
Fernando Díaz

User feedback: Select what is relevant



source:
Fernando Díaz

Results after relevance feedback



source:
Fernando Díaz

Results for initial query

Initial query: New space satellite applications

Results for initial query

Initial query: New space satellite applications

Results for initial query:

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

Results for initial query

Initial query: New space satellite applications

Results for initial query:

1. 0.539, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
2. 0.533, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
3. 0.528, 04/04/90, Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes
4. 0.526, 09/09/91, A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget
5. 0.525, 07/24/90, Scientist Who Exposed Global Warming Proposes Satellites for Climate Research
6. 0.524, 08/22/90, Report Provides Support for the Critics Of Using Big Satellites to Study Climate
7. 0.516, 04/13/87, Arianespace Receives Satellite Launch Pact From Telesat Canada
8. 0.509, 12/02/87, Telecommunications Tale of Two Companies

User then marks relevant documents with "+".

Results for initial query

Initial query: New space satellite applications

Results for initial query:

\+ 1. 0.539, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer

\+ 2. 0.533, 07/09/91, NASA Scratches Environment Gear From Satellite Plan

3. 0.528, 04/04/90, Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes

4. 0.526, 09/09/91, A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget

5. 0.525, 07/24/90, Scientist Who Exposed Global Warming Proposes Satellites for Climate Research

6. 0.524, 08/22/90, Report Provides Support for the Critics Of Using Big Satellites to Study Climate

7. 0.516, 04/13/87, Arianespace Receives Satellite Launch Pact From Telesat Canada

\+ 8. 0.509, 12/02/87, Telecommunications Tale of Two Companies

User then marks relevant documents with "+".

Results for initial query

Initial query: New space satellite applications

Results for initial query:

\+ 1. 0.539, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer

\+ 2. 0.533, 07/09/91, NASA Scratches Environment Gear From Satellite Plan

3. 0.528, 04/04/90, Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes

4. 0.526, 09/09/91, A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget

5. 0.525, 07/24/90, Scientist Who Exposed Global Warming Proposes Satellites for Climate Research

6. 0.524, 08/22/90, Report Provides Support for the Critics Of Using Big Satellites to Study Climate

7. 0.516, 04/13/87, Arianespace Receives Satellite Launch Pact From Telesat Canada

\+ 8. 0.509, 12/02/87, Telecommunications Tale of Two Companies

User then marks relevant documents with "+".

Expanded query after relevance feedback

2.074	new	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

Results for expanded query

* * 1. 0.513, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
* * 2. 0.500, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
* 3. 0.493, 08/07/89, When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
* 4. 0.493, 07/31/89, NASA Uses 'Warm' Superconductors For Fast Circuit
* * 5. 0.492, 12/02/87, Telecommunications Tale of Two Companies
* 6. 0.491, 07/09/91, Soviets May Adapt Parts of SS-20 Missile For Commercial Use
* 7. 0.490, 07/12/88, Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers
* 8. 0.490, 06/14/90, Rescue of Satellite By Space Agency To Cost $90 Million

Outline

Key concept for relevance feedback: Centroid

- The centroid is the center of mass of a set of points.

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.

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.

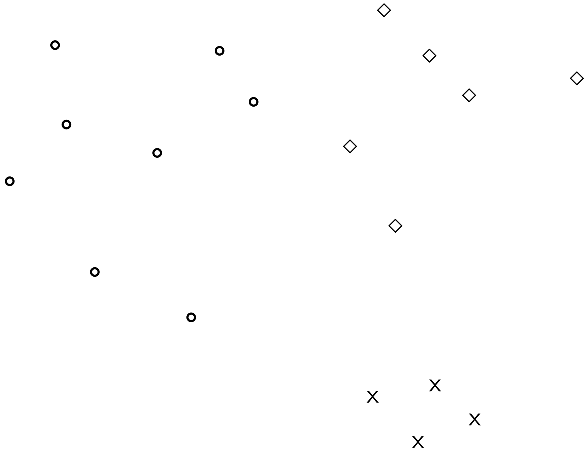
Key concept for relevance feedback: Centroid

- 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:

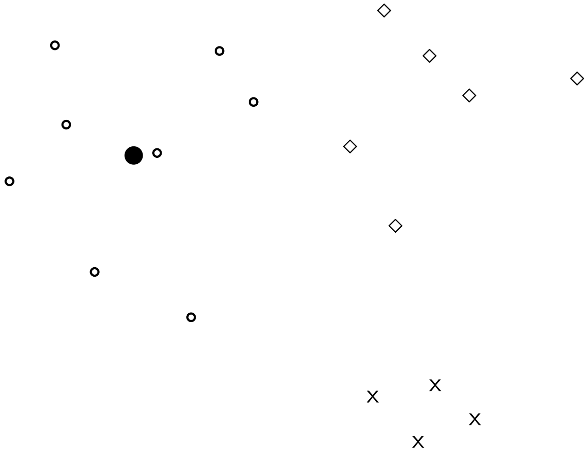
$$\vec{\mu}(D) = \frac{1}{|D|} \sum_{d \in D} \vec{v}(d)$$

where D is a set of documents and $\vec{v}(d) = \vec{d}$ is the vector we use to represent the document d.

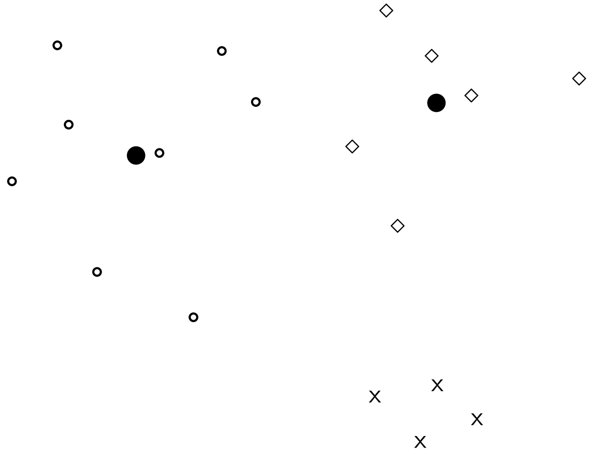
Centroid: Examples



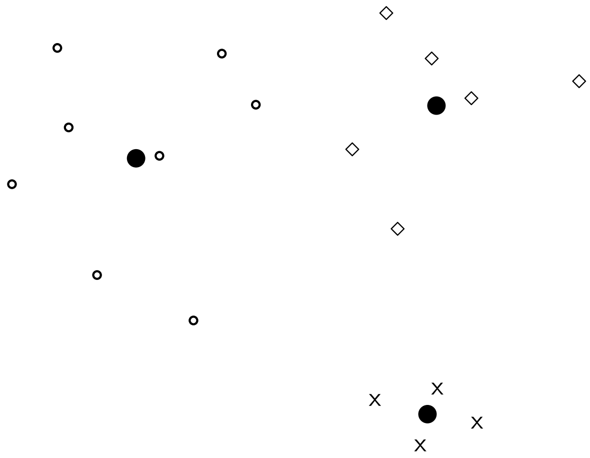
Centroid: Examples



Centroid: Examples



Centroid: Examples



Rocchio algorithm

- The Rocchio algorithm implements relevance feedback in the vector space model.

Rocchio algorithm

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

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

Rocchio algorithm

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

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

- Closely related to maximum separation between relevant and nonrelevant docs

Rocchio algorithm

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

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

- Closely related to maximum separation between relevant and nonrelevant docs
- This optimal query vector is:

$$\vec{q}_{opt} = \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j + [\frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j]$$

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

Rocchio algorithm

- The optimal query vector is:

$$\vec{q}_{opt} = \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j + [\frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j]$$

Rocchio algorithm

- The optimal query vector is:

$$\vec{q}_{opt} = \frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j + [\frac{1}{|D_r|} \sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|} \sum_{\vec{d}_j \in D_{nr}} \vec{d}_j]$$

- q-opt = centroid-rel - (centroid-rel - centroid-nonrel)

Rocchio algorithm

- The optimal query vector is:

$$\vec{q}_{opt} = \frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r} \vec{d}_j + [\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}} \vec{d}_j]$$

- q-opt = centroid-rel - (centroid-rel - centroid-nonrel)
- We move the centroid of the relevant documents by the difference between the two centroids.

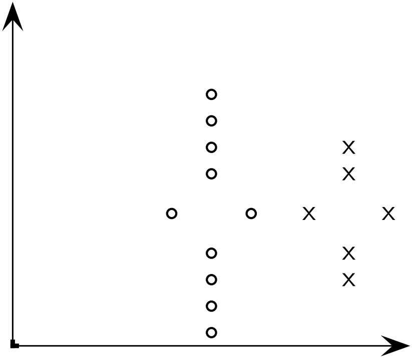
Rocchio algorithm

- The optimal query vector is:

$$\vec{q}_{opt} = \frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r} \vec{d}_j + [\frac{1}{|D_r|}\sum_{\vec{d}_j \in D_r} \vec{d}_j - \frac{1}{|D_{nr}|}\sum_{\vec{d}_j \in D_{nr}} \vec{d}_j]$$

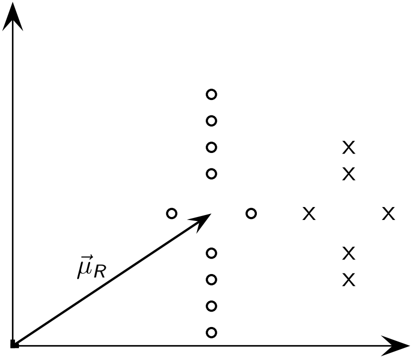
- q-opt = centroid-rel - (centroid-rel - centroid-nonrel)
- We move the centroid of the relevant documents by the difference between the two centroids.
- We had to assume $|\vec{\mu}_r| = |\vec{\mu}_{nr}| = 1$ for this derivation.

Rocchio illustrated



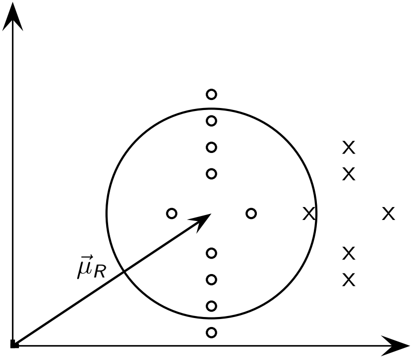
circles: relevant documents, Xs: nonrelevant documents

Rocchio illustrated



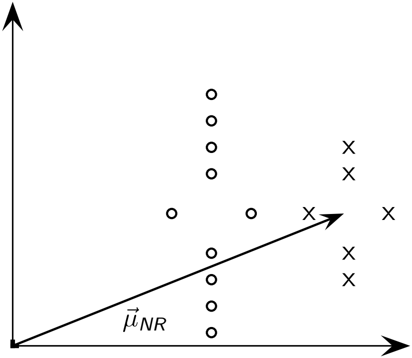
$\vec{\mu}_R$: centroid of relevant documents

Rocchio illustrated



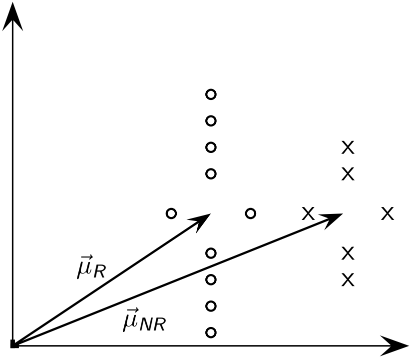
$\vec{\mu}_R$ does not separate relevant/nonrelevant.

Rocchio illustrated

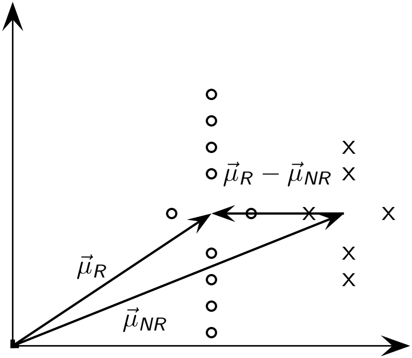


$\vec{\mu}_{NR}$: centroid of nonrelevant documents

Rocchio illustrated

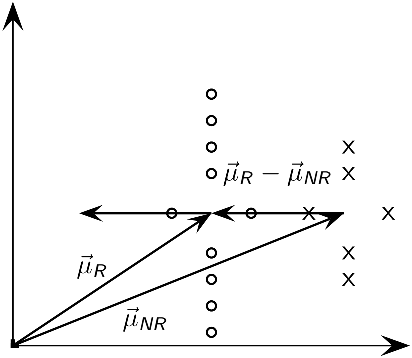


Rocchio illustrated



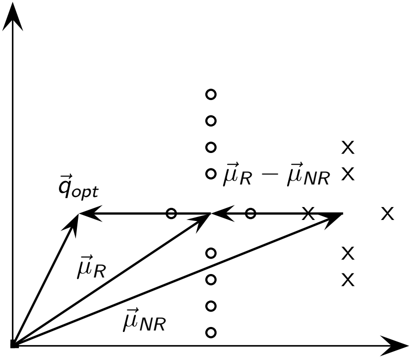
$\vec{\mu}_R - \vec{\mu}_{NR}$: difference vector

Rocchio illustrated



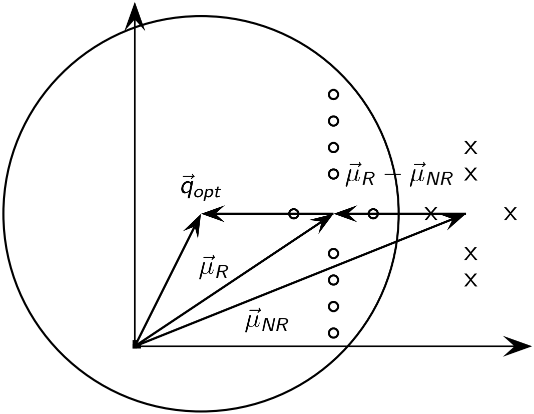
Add difference vector to $\vec{\mu}_R$ . . .

Rocchio illustrated



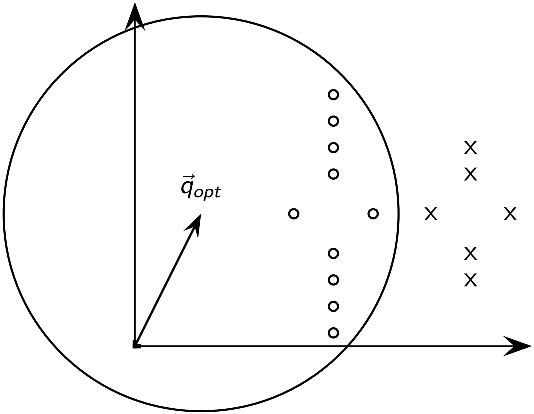
... to get $\vec{q}_{opt}$

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:

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

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 attached to each term

Rocchio 1971 algorithm (SMART)

- Used in practice:

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

  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 attached to each term
- New query moves towards relevant documents and away from nonrelevant documents.

Rocchio 1971 algorithm (SMART)

- Used in practice:

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

  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 attached to each term
- New query moves towards relevant documents and away from nonrelevant documents.
- Tradeoff α vs. β/γ: If we have a lot of judged documents, we want a higher β/γ.

Rocchio 1971 algorithm (SMART)

- Used in practice:

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

  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 attached to each term
- New query moves towards relevant documents and away from nonrelevant 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.

Rocchio 1971 algorithm (SMART)

- Used in practice:

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

  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 attached to each term
- New query moves towards relevant documents and away from nonrelevant 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.

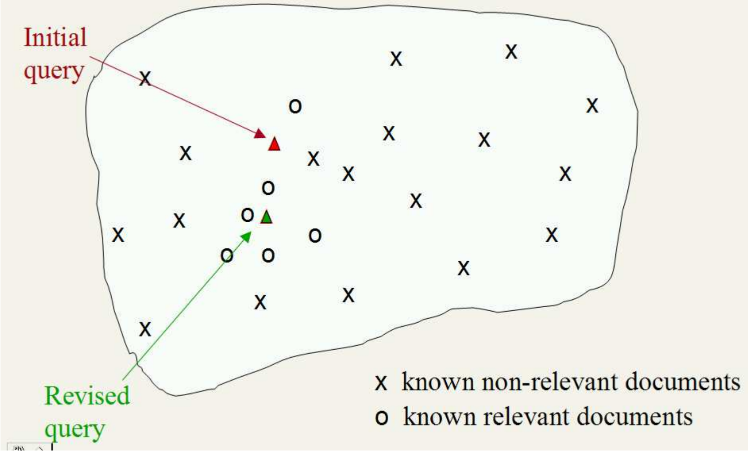
Rocchio 1971 algorithm (SMART)

- Used in practice:

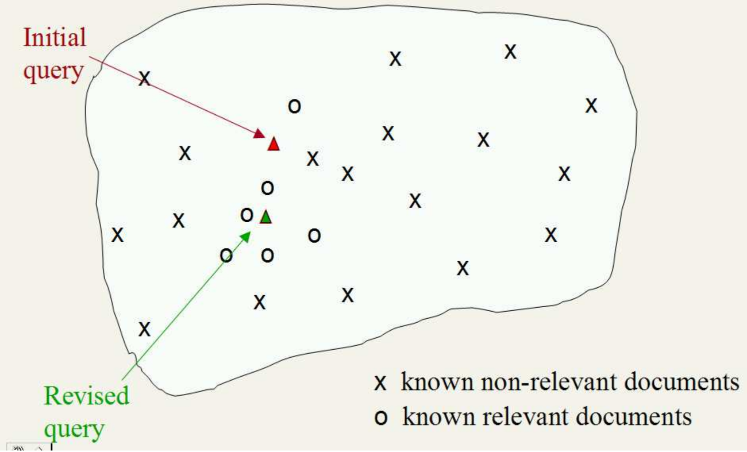
$$\vec{q}_m = \alpha\vec{q}_0 + \beta\frac{1}{|D_r|}\sum_{\vec{d}_j\in D_r}\vec{d}_j - \gamma\frac{1}{|D_{nr}|}\sum_{\vec{d}_j\in D_{nr}}\vec{d}_j$$

  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 attached to each term
- New query moves towards relevant documents and away from nonrelevant 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 relevance feedback illustrated



Rocchio relevance feedback illustrated



- Questions?

Positive vs. negative relevance feedback

- Positive feedback is more valuable than negative feedback.

Positive vs. negative relevance feedback

- Positive feedback is more valuable than negative feedback.
- Why?

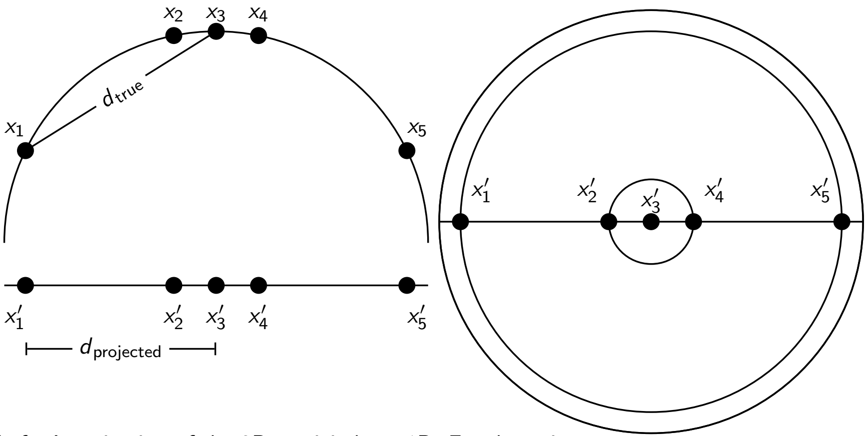
Positive vs. negative relevance feedback

- Positive feedback is more valuable than negative feedback.
- Why?
- For example, set $\beta = 0.75$, $\gamma = 0.25$ to give higher weight to positive feedback.

Positive vs. negative relevance feedback

- Positive feedback is more valuable than negative feedback.
- Why?
- For example, set $\beta = 0.75$, $\gamma = 0.25$ to give higher weight to positive feedback.
- Many systems only allow positive feedback.

Aside: 2D/3D graphs can be misleading



Left: A projection of the 2D semicircle to 1D. For the points x_1, x_2, x_3, x_4, x_5 at x coordinates $-0.9, -0.2, 0, 0.2, 0.9$ the distance $|x_2x_3| \approx 0.201$ only differs by 0.5% from $|x'_2x'_3| = 0.2$; but $|x_1x_3|/|x'_1x'_3| = d_{\text{true}}/d_{\text{projected}} \approx 1.06/0.9 \approx 1.18$ is an example of a large distortion (18%) when projecting a large area. *Right:* The corresponding projection of the 3D hemisphere to 2D.

Relevance feedback: Assumptions

- When can relevance feedback enhance recall?

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.

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

- Violation of assumption A1: The user knows the terms in the collection well enough for an initial query.

Violation of A1

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

Violation of A1

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

Violation of A2

- Violation of A2: Relevant documents are not similar.

Violation of A2

- Violation of A2: Relevant documents are not similar.
- Example query: contradictory government policies

Violation of A2

- Violation of A2: Relevant documents are not similar.
- Example query: contradictory government policies
- Why is relevance feedback unlikely to increase recall substantially for this query?

Violation of A2

- Violation of A2: Relevant documents are not similar.
- Example query: contradictory government policies
- Why is relevance feedback unlikely to increase recall substantially for this query?
- Several unrelated "prototypes"

Violation of A2

- Violation of A2: Relevant documents are not similar.
- Example query: contradictory government policies
- Why is relevance feedback unlikely to increase recall substantially for this query?
- Several unrelated "prototypes"
  - Subsidies for tobacco farmers vs. anti-smoking campaigns

Violation of A2

- Violation of A2: Relevant documents are not similar.
- Example query: contradictory government policies
- Why is relevance feedback unlikely to increase recall substantially for this query?
- Several unrelated "prototypes"
  - Subsidies for tobacco farmers vs. anti-smoking campaigns
  - Aid for developing countries vs. high tariffs on imports from developing countries

Violation of A2

- Violation of A2: Relevant documents are not similar.
- Example query: contradictory government policies
- Why is relevance feedback unlikely to increase recall substantially for this query?
- 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 last lecture, e.g., precision in top 10: $P@10$

Relevance feedback: Evaluation

- Pick one of the evaluation measures from last lecture, e.g., precision in top 10: $P@10$
- Compute $P@10$ for original query q_0

Relevance feedback: Evaluation

- Pick one of the evaluation measures from last lecture, 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

Relevance feedback: Evaluation

- Pick one of the evaluation measures from last lecture, 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!

Relevance feedback: Evaluation

- Pick one of the evaluation measures from last lecture, 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.

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.

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.

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.

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.

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.

Do search engines use relevance feedback?

“similar pages” at Google

Relevance feedback: Problems

- Relevance feedback is expensive.

Relevance feedback: Problems

- Relevance feedback is expensive.
  - Relevance feedback creates long modified queries.

Relevance feedback: Problems

- Relevance feedback is expensive.
  - Relevance feedback creates long modified queries.
  - Long queries are expensive to process.

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.

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.

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.
- Excite had full relevance feedback at one point, but abandoned it later.

Other use of relevance feedback

- Maintaining a standing query

Other use of relevance feedback

- Maintaining a standing query
- Example: "multicore computer chips"

Other use of relevance feedback

- Maintaining a standing query
- Example: "multicore computer chips"
- I want to receive each morning a list of news articles published in the previous 24 hours on "multicore computer chips".

Other use of relevance feedback

- Maintaining a standing query
- Example: "multicore computer chips"
- I want to receive each morning a list of news articles published in the previous 24 hours on "multicore computer chips".
- Relevance feedback can be used to refine this standing query over time.

Other use of relevance feedback

- Maintaining a standing query
- Example: "multicore computer chips"
- I want to receive each morning a list of news articles published in the previous 24 hours on "multicore computer chips".
- Relevance feedback can be used to refine this standing query over time.
- Many spam filters offer a similar functionality.

Other use of relevance feedback

- Maintaining a standing query
- Example: "multicore computer chips"
- I want to receive each morning a list of news articles published in the previous 24 hours on "multicore computer chips".
- Relevance feedback can be used to refine this standing query over time.
- Many spam filters offer a similar functionality.
- For standing queries, relevance feedback is more practical than in web search.

Other use of relevance feedback

- Maintaining a standing query
- Example: “multicore computer chips”
- I want to receive each morning a list of news articles published in the previous 24 hours on “multicore computer chips”.
- Relevance feedback can be used to refine this standing query over time.
- Many spam filters offer a similar functionality.
- For standing queries, relevance feedback is more practical than in web search.
- We’ll revisit this issue in IIR 13.

Pseudo-relevance feedback

- Pseudo-relevance feedback automates the “manual” part of true relevance feedback.

Pseudo-relevance feedback

- Pseudo-relevance feedback automates the "manual" part of true relevance feedback.
- Pseudo-relevance algorithm:

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

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.

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)

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

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.

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

- Pseudo-relevance feedback automates the “manual” part of true relevance feedback.
- Pseudo-relevance algorithm:
  - Retrieve a ranked list of hits for the user’s query
  - Assume that the top k documents are relevant.
  - Do relevance feedback (e.g., Rocchio)
- Works very well on average
- But can go horribly wrong for some queries.
- Several iterations can cause *query drift*.
- Why?

Pseudo-relevance feedback at TREC4

- Cornell SMART system

Pseudo-relevance feedback at TREC4

- Cornell SMART system
- Results show number of relevant documents out of top 100 for 50 queries (so total number of documents is 5000):

method	number of relevant documents
lnc.ltc	3210
lnc.ltc-PsRF	3634
Lnu.ltu	3709
Lnu.ltu-PsRF	4350

Pseudo-relevance feedback at TREC4

- Cornell SMART system
- Results show number of relevant documents out of top 100 for 50 queries (so total number of documents is 5000):

method	number of relevant documents
lnc.ltc	3210
lnc.ltc-PsRF	3634
Lnu.ltu	3709
Lnu.ltu-PsRF	4350

- Results contrast two length normalization schemes (L vs. l) and pseudo-relevance feedback (PsRF).

Pseudo-relevance feedback at TREC4

- Cornell SMART system
- Results show number of relevant documents out of top 100 for 50 queries (so total number of documents is 5000):

method	number of relevant documents
lnc.ltc	3210
lnc.ltc-PsRF	3634
Lnu.ltu	3709
Lnu.ltu-PsRF	4350

- Results contrast two length normalization schemes (L vs. l) and pseudo-relevance feedback (PsRF).
- The pseudo-relevance feedback method used added only 20 terms to the query. (Rocchio will add many more.)

Pseudo-relevance feedback at TREC4

- Cornell SMART system
- Results show number of relevant documents out of top 100 for 50 queries (so total number of documents is 5000):

method	number of relevant documents
lnc.ltc	3210
lnc.ltc-PsRF	3634
Lnu.ltu	3709
Lnu.ltu-PsRF	4350

- Results contrast two length normalization schemes (L vs. l) and pseudo-relevance feedback (PsRF).
- The pseudo-relevance feedback method used added only 20 terms to the query. (Rocchio will add many more.)
- This demonstrates that pseudo-relevance feedback is effective on average.

Outline

1. Recap
2. Relevance feedback: Basics
3. Relevance feedback: Details
4. Global query expansion

Global query expansion

- Query expansion is another method for increasing recall.

Global query expansion

- Query expansion is another method for increasing recall.
- We use "global query expansion" to refer to "global methods for query reformulation".

Global 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.

Global 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

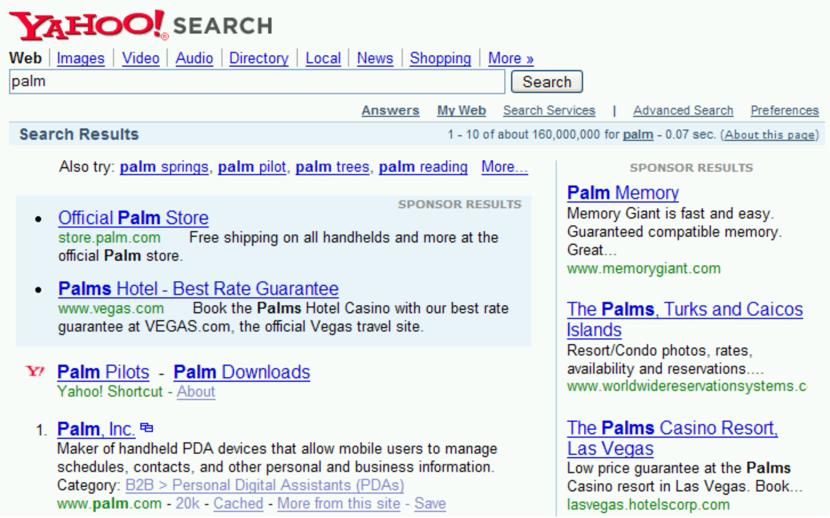
Global query expansion

- 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.

Global 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.

"Global" query expansion: Example



Types of user feedback

- User gives feedback on documents.

Types of user feedback

- User gives feedback on documents.
  - More common in relevance feedback

Types of user feedback

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

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 user feedback

- User gives feedback on documents.
    - More common in relevance feedback
- User gives feedback on words or phrases.
    - More common in query expansion
- Relevance feedback can also be thought of as a type of query expansion.

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
- Relevance feedback can also be thought of as a type of query expansion.
- We add terms to the query.

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
- Relevance feedback can also be thought of as a type of query expansion.
- We add terms to the query.
- The terms added in relevance feedback are based on “local” information in the result list.

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
- Relevance feedback can also be thought of as a type of query expansion.
- We add terms to the query.
- The terms added in relevance feedback are based on “local” information in the result list.
- The terms added in query expansion are often based on “global” information that is not query-specific.

Types of query expansion

- Manual thesaurus (maintained by editors, e.g., PubMed)

Types of query expansion

- Manual thesaurus (maintained by editors, e.g., PubMed)
- Automatically derived thesaurus (e.g., based on co-occurrence statistics)

Types of query expansion

- Manual thesaurus (maintained by editors, e.g., PubMed)
- Automatically derived thesaurus (e.g., based on co-occurrence statistics)
- 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.

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 from earlier: HOSPITAL $\longrightarrow$ MEDICAL

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 from earlier: HOSPITAL $\longrightarrow$ MEDICAL
- Generally increases recall

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 from earlier: HOSPITAL $\longrightarrow$ MEDICAL
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms

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 from earlier: HOSPITAL → MEDICAL
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms
    - INTEREST RATE → INTEREST RATE FASCINATE EVALUATE

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 from earlier: HOSPITAL → MEDICAL
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms
  - INTEREST RATE → INTEREST RATE FASCINATE EVALUATE
- Widely used in specialized search engines for science and engineering

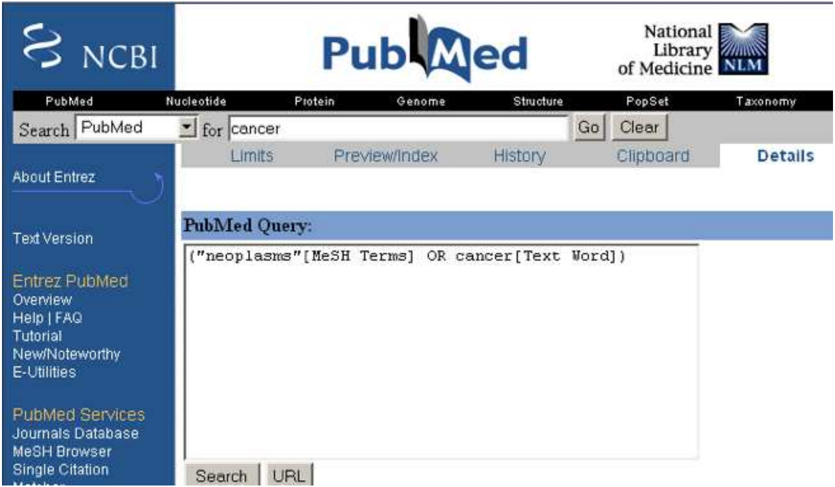
Thesaurus-based query expansion

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

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 from earlier: HOSPITAL → MEDICAL
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms
  - INTEREST RATE → INTEREST RATE FASCINATE EVALUATE
- Widely used in specialized search engines for science and engineering
- It's very expensive to create a manual thesaurus and to maintain it over time.
- A manual thesaurus is roughly equivalent to annotation with a *controlled vocabulary*.

Example for manual thesaurus: PubMed



Automatic thesaurus generation

- Attempt to generate a thesaurus automatically by analyzing the distribution of words in documents

Automatic thesaurus generation

- Attempt to generate a thesaurus automatically by analyzing the distribution of words in documents
- Fundamental notion: similarity between two words

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.

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.
- Definition 2: Two words are similar if they occur in a given grammatical relation with the same words.

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.
- 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.

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.
- 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-occurence-based thesaurus: Examples

Word	Nearest neighbors
absolutely	absurd, whatsoever, totally, exactly, nothing
bottomed	dip, copper, drops, topped, slide, trimmed
captivating	shimmer, stunningly, superbly, plucky, witty
doghouse	dog, porch, crawling, beside, downstairs
makeup	repellent, lotion, glossy, sunscreen, skin, gel
mediating	reconciliation, negotiate, case, conciliation
keeping	hoping, bring, wiping, could, some, would
lithographs	drawings, Picasso, Dali, sculptures, Gauguin
pathogens	toxins, bacteria, organisms, bacterial, parasite
senses	grasp, psyche, truly, clumsy, naive, innate

Summary

- Relevance feedback and query expansion increase recall.

Summary

- Relevance feedback and query expansion increase recall.
- In many cases, precision is decreased, often significantly.

Summary

- Relevance feedback and query expansion increase recall.
- In many cases, precision is decreased, often significantly.
- Log-based query modification (which is more complex than simple query expansion) is more common on the web than relevance feedback.

Resources

- Chapter 9 of IIR

Resources

- Chapter 9 of IIR
- Resources at `http://ifnlp.org/ir`

Resources

- Chapter 9 of IIR
- Resources at `http://ifnlp.org/ir`
- Salton and Buckley 1990 (original relevance feedback paper)

Resources

- Chapter 9 of IIR
- Resources at `http://ifnlp.org/ir`
- Salton and Buckley 1990 (original relevance feedback paper)
- Spink, Jansen, Ozmultu 2000: Relevance feedback at Excite

Resources

- Chapter 9 of IIR
- Resources at `http://ifnlp.org/ir`
- Salton and Buckley 1990 (original relevance feedback paper)
- Spink, Jansen, Ozmultu 2000: Relevance feedback at Excite
- Schütze 1998: Automatic word sense discrimination (describes a simple method for automatic thesuarus generation)