Introduction to Information Retrieval

http://informationretrieval.org

IIR 9: Relevance Feedback & Query Expansion

Hinrich Schütze

Center for Information and Language Processing, University of Munich

2014-05-14

Overview

Outline

1. Recap
2. Motivation
3. Relevance feedback: Basics
4. Relevance feedback: Details
5. Query expansion

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

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

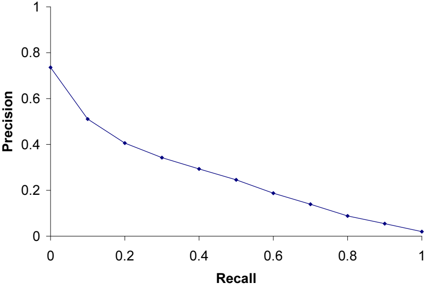
A combined measure: F

- F allows us to trade off precision against recall.
- Balanced F:

$$F_1 = \frac{2PR}{P + R}$$

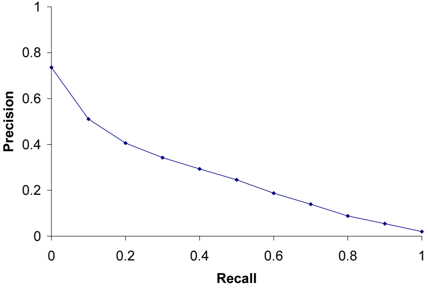
- This is a kind of soft minimum of precision and recall.

Averaged 11-point precision/recall graph



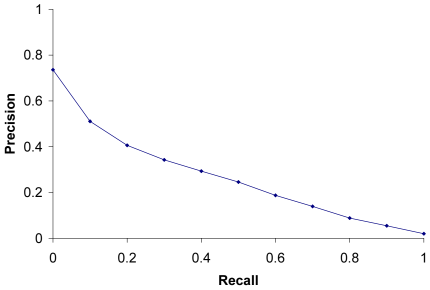
- This curve is typical of performance levels for the TREC benchmark.

Averaged 11-point precision/recall graph



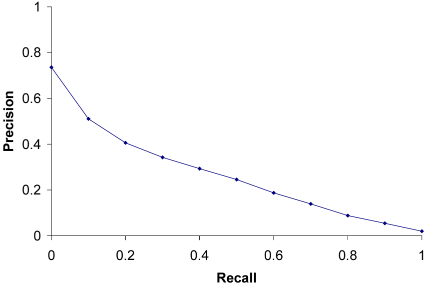
- This curve is typical of performance levels for the TREC benchmark.
- 70% chance of getting the first document right (roughly)

Averaged 11-point precision/recall graph



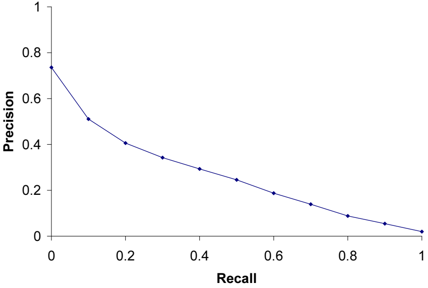
- This curve is typical of performance levels for the TREC benchmark.
- 70% chance of getting the first document right (roughly)
- When we want to look at at least 50% of all relevant documents, then for each relevant document we find, we will have to look at about two nonrelevant documents.

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.

Google dynamic summaries for [vegetarian diet running]

No Meat Athlete | **Vegetarian Running** and Fitness
www.nomeatathlete.com/
Vegetarian Running and Fitness. ... (Oh, and did I mention Rich did it all on a plant-based **diet**?) In this episode of No Meat Athlete Radio, Doug and I had the ...
Vegetarian Recipes for Athletes - Vegetarian Shirts - How to Run Long - About

Running on a **vegetarian diet** – Top tips | Freedom2Train Blog
www.freedom2train.com/blog/?p=4
Nov 8, 2012 – In this article we look to tackle the issues faced by long distance runners on a **vegetarian diet**. By its very nature, a **vegetarian diet** can lead to ...

HowStuffWorks "5 Nutrition Tips for **Vegetarian** Runners"
www.howstuffworks.com/.../**running**/.../5-nutrition-tips-for-**vegetarian**-r...
Even without meat, you can get enough fuel to keep on **running**. Stockbyte/Thinkstock ... Unfortunately, a **vegetarian diet** is not a panacea for runners. It could, for ...

Nutrition Guide for **Vegetarian** and Vegan Runners - The **Running** Bug
therunningbug.co.uk/.../nutrition-guide-for-**vegetarian**-and-vegan-runne...
Feb 28, 2012 – The **Running** Bug's guide to nutrition for vegetarian and vegan ... different types of **vegetarian diet** ranging from lacto-ovo-vegetarians who eat ...

Vegetarian Runner
www.**vegetarian**runner.com/
Vegetarian Runner - A resource center for vegetarianism and **running** and how to make sure you have proper nutrition as an athlete with a **vegetarian diet**.

Take-away today

Take-away today

- Interactive relevance feedback: improve initial retrieval results by telling the IR system which docs are relevant / nonrelevant

Take-away today

- Interactive relevance feedback: improve initial retrieval results by telling the IR system which docs are relevant / nonrelevant
- Best known relevance feedback method: Rocchio feedback

Take-away today

- Interactive relevance feedback: improve initial retrieval results by telling the IR system which docs are relevant / nonrelevant
- Best known relevance feedback method: Rocchio feedback
- Query expansion: improve retrieval results by adding synonyms / related terms to the query

Take-away today

- Interactive relevance feedback: improve initial retrieval results by telling the IR system which docs are relevant / nonrelevant
- Best known relevance feedback method: Rocchio feedback
- Query expansion: improve retrieval results by adding synonyms / related terms to the query
  - Sources for related terms: Manual thesauri, automatic thesauri, query logs

Overview

1. Recap
2. Motivation
3. Relevance feedback: Basics
4. Relevance feedback: Details
5. Query expansion

Outline

How can we improve recall in search?

- Main topic today: two ways of improving recall: relevance feedback and query expansion

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] . . .

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"

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.

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!

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:

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

Recall

- Loose definition of recall in this lecture: "increasing the number of relevant documents returned to user"

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" to "jaguar AND panthera"

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" to "jaguar AND panthera"
  - ...which eliminates some relevant documents, but increases relevant documents returned on top pages

Options for improving recall

Options for improving recall

- Local: Do a "local", on-demand analysis for a user query

Options for improving recall

- Local: Do a “local”, on-demand analysis for a user query
  - Main local method: relevance feedback

Options for improving recall

- Local: Do a "local", on-demand analysis for a user query
  - Main local method: relevance feedback
  - Part 1

Options for improving recall

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

Options for improving recall

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

Options for improving recall

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

Google used to expose query expansion in UI

- *~flights -flight*
- *~dogs -dog*
- no longer available:
  `http://searchenginewatch.com/article/2277383/Google-Kil`

Outline

Relevance feedback: Basic idea

Relevance feedback: Basic idea

- The user issues a (short, simple) query.

Relevance feedback: Basic idea

- The user issues a (short, simple) query.
- The search engine returns a set of documents.

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.

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.

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.

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.

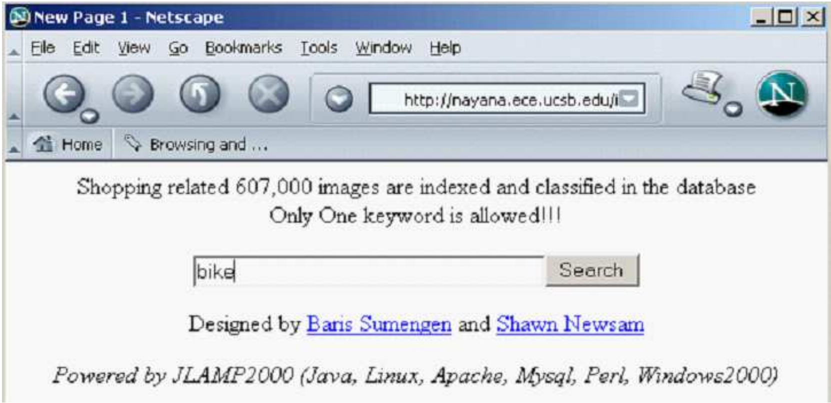
Relevance feedback: Basic idea

- 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.
- We will use the term ad hoc retrieval to refer to regular retrieval without relevance feedback.

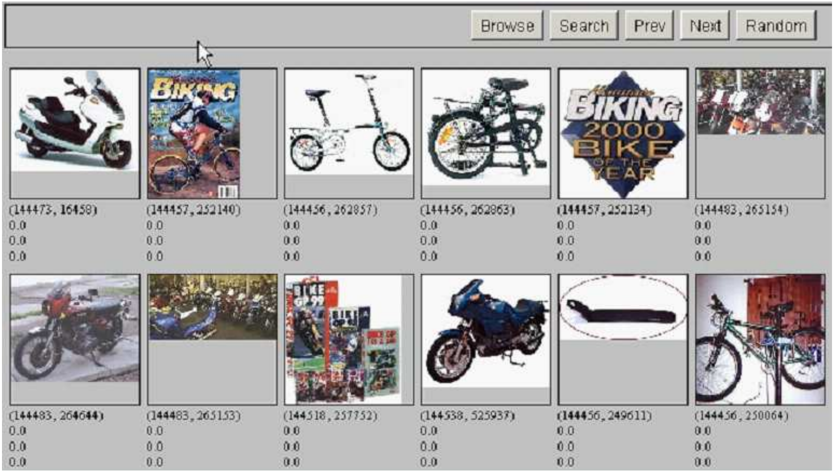
Relevance feedback: Examples

- We will now look at three different examples of relevance feedback that highlight different aspects of the process.

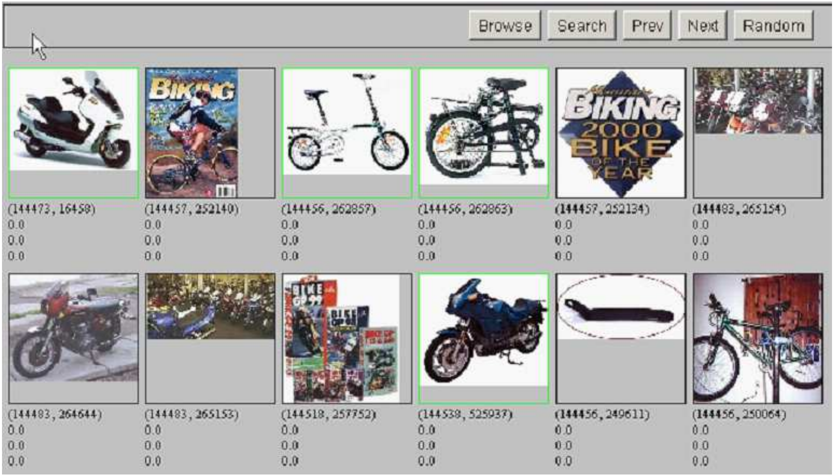
Relevance Feedback: Example 1



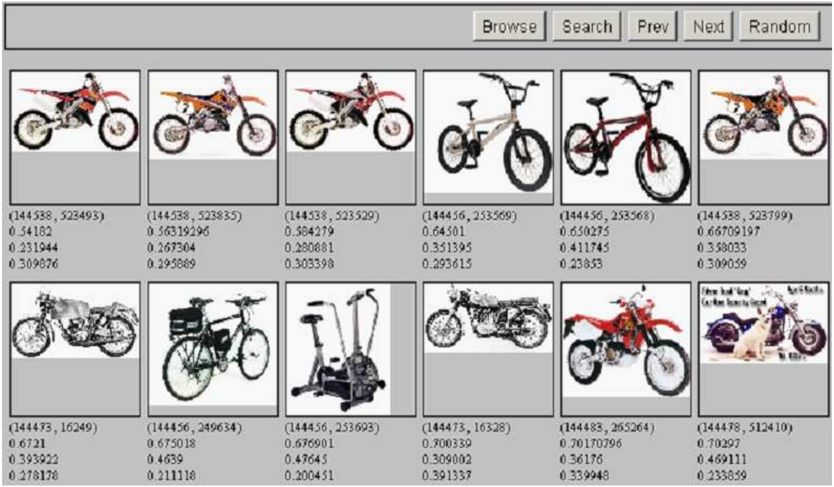
Results for initial query



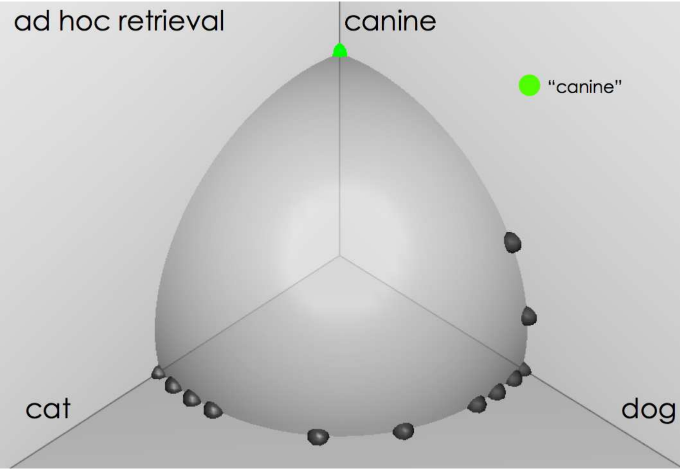
User feedback: Select what is relevant



Results after relevance feedback

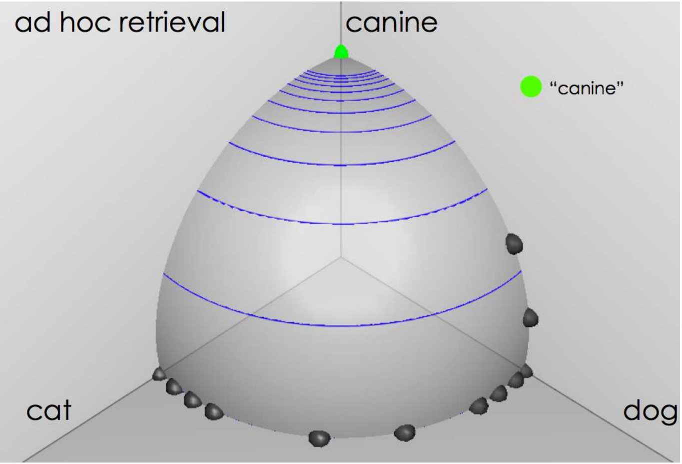


Vector space example: query "canine" (1)



source:
Fernando Díaz

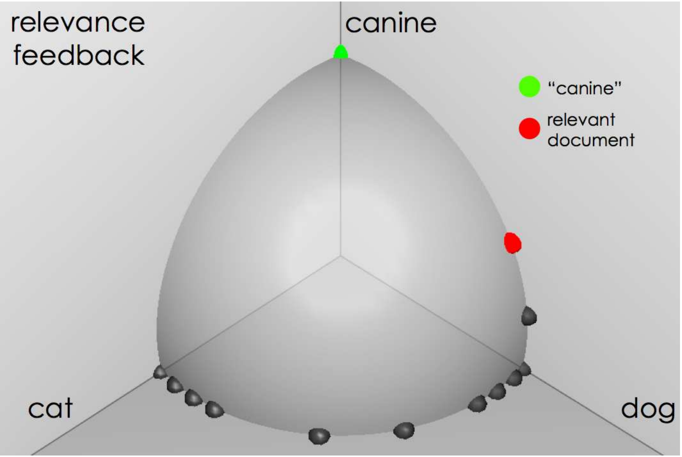
Similarity of docs to query "canine"



source:
Fernando Díaz

User feedback: Select relevant documents

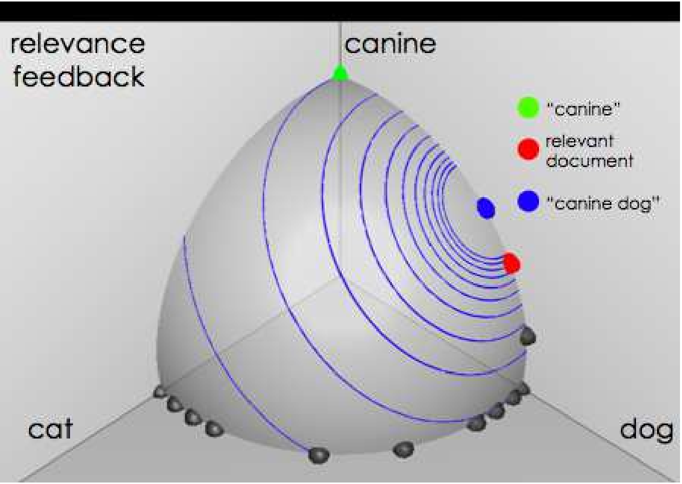
User feedback: Select relevant documents



source:
Fernando Díaz

Results after relevance feedback

Results after relevance feedback



source:
Fernando Díaz

Example 3: A real (non-image) example

Example 3: A real (non-image) example

Initial query: [new space satellite applications]

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

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 "+".

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 to original query: [new space satellite applications]

Results for expanded query (old ranks in parens)

	r		
*	1 (2)	0.513	NASA Scratches Environment Gear From Satellite Plan
*	2 (1)	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 (8)	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

Outline

Key concept for relevance feedback: Centroid

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

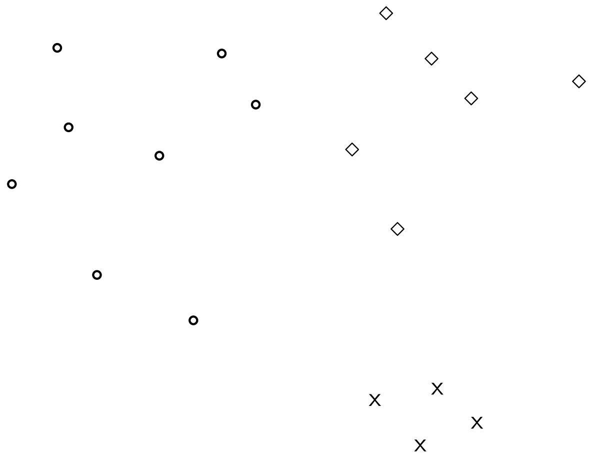
- The centroid is the center of mass of a set of points.
- Recall that we represent documents as points in a high-dimensional space.
- Thus: we can compute centroids of documents.
- Definition:

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

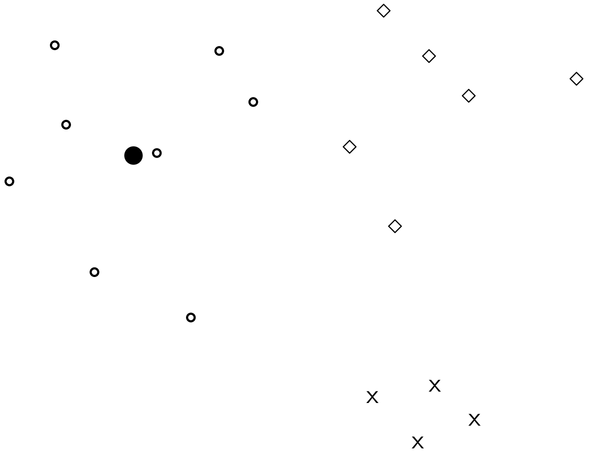
where D is a set of documents and $\vec{v}(d) = \vec{d}$ is the vector we use to represent document d.

Centroid: Examples

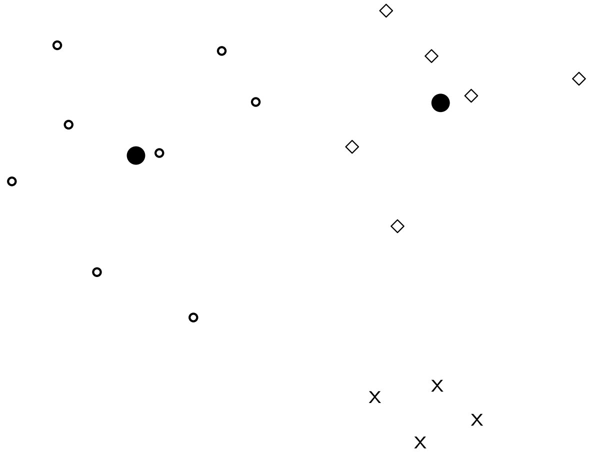
Centroid: Examples



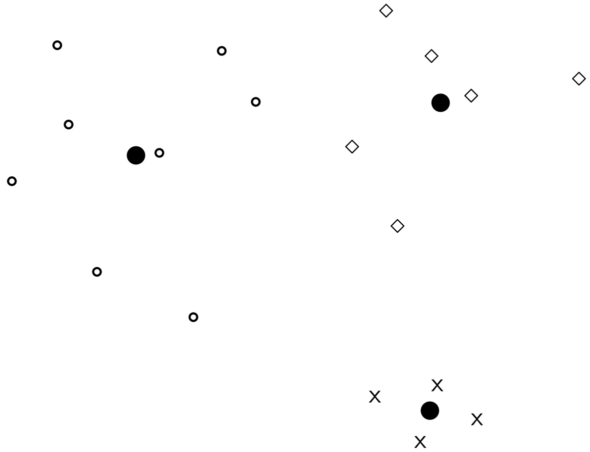
Centroid: Examples



Centroid: Examples



Centroid: Examples



Rocchio algorithm

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

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

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

Rocchio algorithm

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

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

  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.

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

Rocchio algorithm

- The optimal query vector is:

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

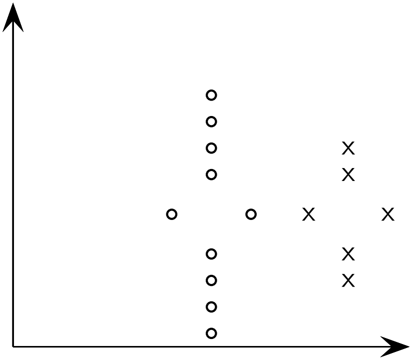
Rocchio algorithm

- The optimal query vector is:

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

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

Exercise: Compute Rocchio vector

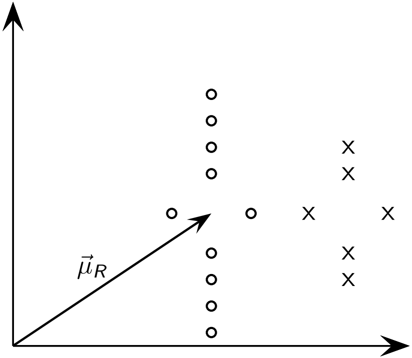


circles: relevant documents, Xs: nonrelevant documents

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

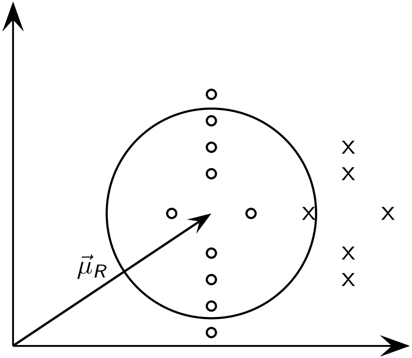
Rocchio illustrated

Rocchio illustrated



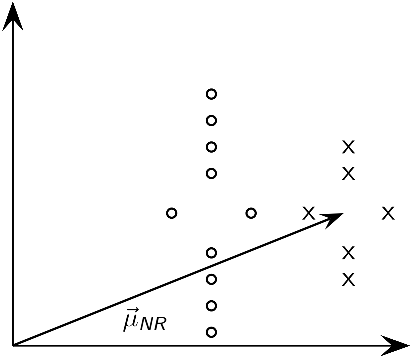
$\vec{\mu}_R$: centroid of relevant documents

Rocchio illustrated



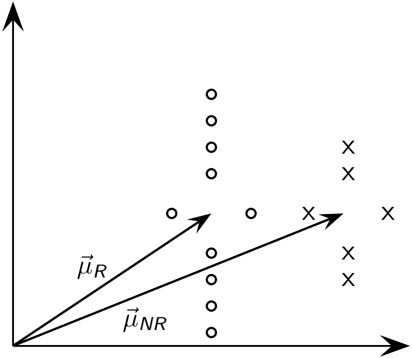
$\vec{\mu}_R$ does not separate relevant/nonrelevant.

Rocchio illustrated

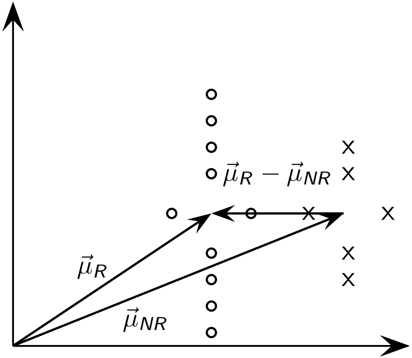


$\vec{\mu}_{NR}$: centroid of nonrelevant documents

Rocchio illustrated

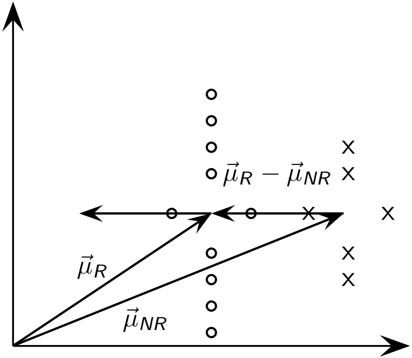


Rocchio illustrated



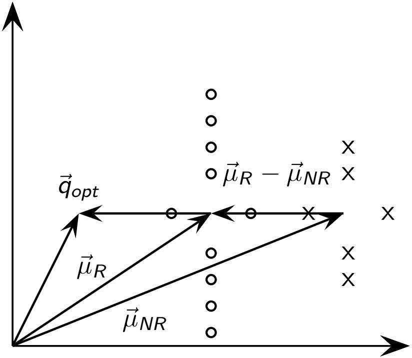
$\vec{\mu}_R - \vec{\mu}_{NR}$: difference vector

Rocchio illustrated



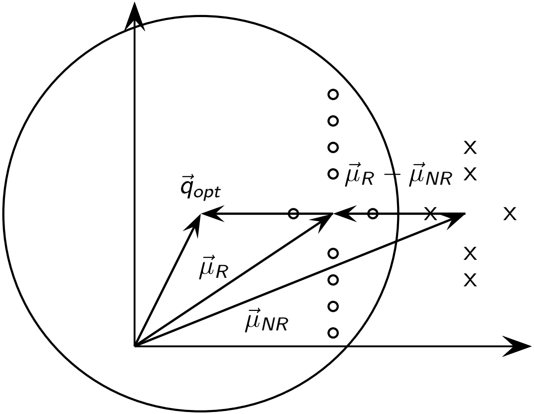
Add difference vector to $\vec{\mu}_R$ ...

Rocchio illustrated



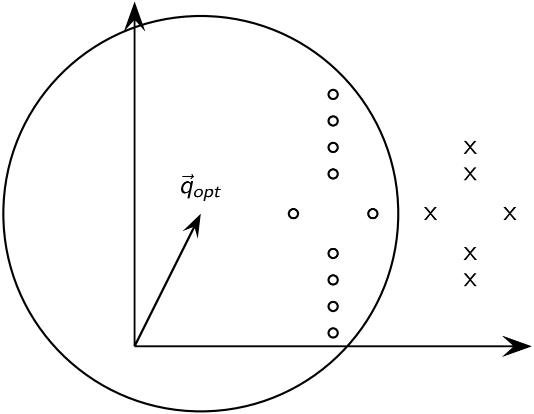
...to get $\vec{q}_{opt}$

Rocchio illustrated



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

Rocchio illustrated



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

Terminology

Terminology

- So far, we have used the name Rocchio for the theoretically better motivated original version of Rocchio.

Terminology

- So far, we have used 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 – this SMART version of Rocchio is what we will refer to from now on.

Rocchio 1971 algorithm (SMART)

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

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 nonrelevant documents.

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 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:

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

  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 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:

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

  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 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:

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

  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 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.

Positive vs. negative relevance feedback

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.
- 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.
- 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

Relevance feedback: Assumptions

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

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

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

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.

Violation of A2

- Assumption A2: Relevant documents are similar.
- Example for violation: [contradictory government policies]

Violation of A2

- Assumption A2: Relevant documents are similar.
- Example for violation: [contradictory government policies]
- Several unrelated "prototypes"

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

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

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: 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).

Relevance feedback: Evaluation

Relevance feedback: Evaluation

- Pick an evaluation measure, e.g., precision in top 10: $P@10$

Relevance feedback: Evaluation

- Pick an evaluation measure, e.g., precision in top 10: $P@10$
- Compute $P@10$ for original query q_0

Relevance feedback: Evaluation

- Pick an evaluation measure, 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 an evaluation measure, 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 an evaluation measure, 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

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

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.

Exercise

Exercise

- Do search engines use relevance feedback?
- Why?

Relevance feedback: Problems

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

Pseudo-relevance feedback

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 feedback algorithm:

Pseudo-relevance feedback

- Pseudo-relevance feedback automates the "manual" part of true relevance feedback.
- Pseudo-relevance feedback 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 feedback 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 feedback 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 feedback 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 feedback 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 feedback 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.
  - Because of query drift

Pseudo-relevance feedback

- Pseudo-relevance feedback automates the "manual" part of true relevance feedback.
- Pseudo-relevance feedback 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.
  - Because of query drift
  - If you do several iterations of pseudo-relevance feedback, then you will get query drift for a large proportion of queries.

Pseudo-relevance feedback at TREC4

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):

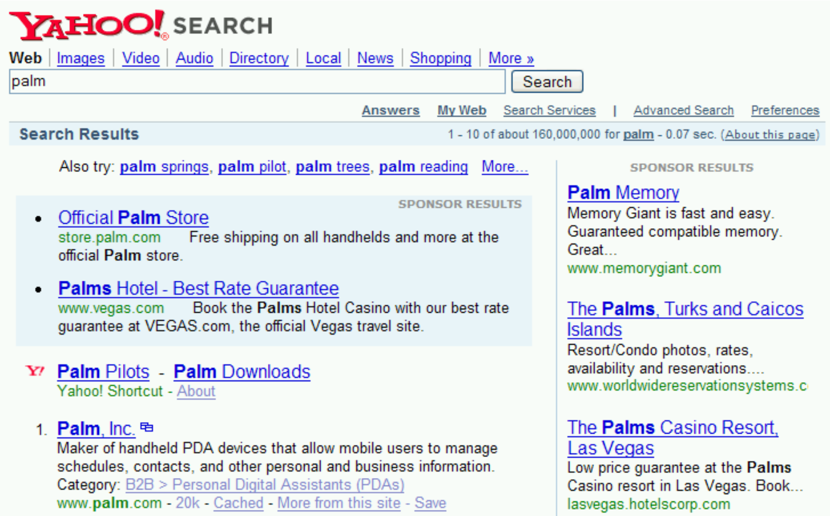
method	number of relevant documents
lnc.ltc	3210
lnc.ltc-PsRF	3634
Lnu.ltu	3709
Lnu.ltu-PsRF	4350

- 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

Query expansion: Example

Query expansion: Example



Types of user feedback

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

Query expansion

Query expansion

- Query expansion is another method for increasing recall.

Query expansion

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

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.

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" resources used for query expansion

“Global” resources used for query expansion

- A publication or database that collects (near-)synonyms is called a thesaurus.

“Global” resources used for query expansion

- A publication or database that collects (near-)synonyms is called a thesaurus.
- Manual thesaurus (maintained by editors, e.g., PubMed)

"Global" resources used for query expansion

- A publication or database that collects (near-)synonyms is called a thesaurus.
- Manual thesaurus (maintained by editors, e.g., PubMed)
- Automatically derived thesaurus (e.g., based on co-occurrence statistics)

"Global" resources used for query expansion

- A publication or database that collects (near-)synonyms is called a thesaurus.
- 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

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 $\rightarrow$ 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 $\rightarrow$ 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 $\rightarrow$ 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

Thesaurus-based query expansion

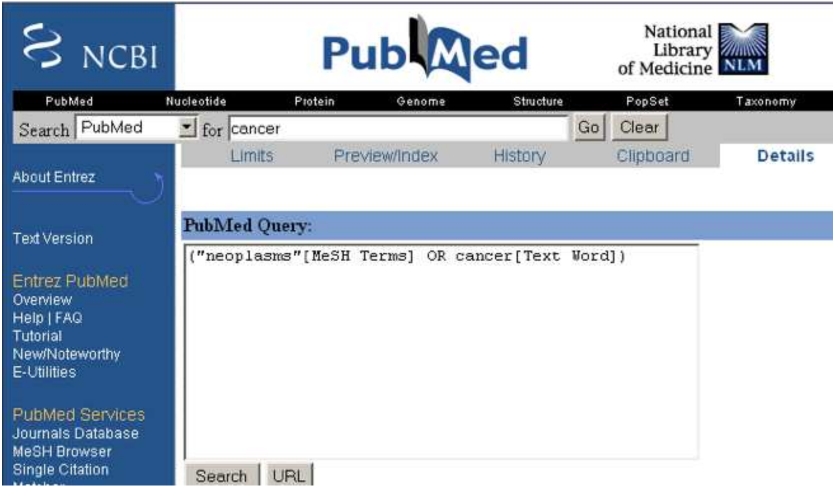
- For each term t in the query, expand the query with words the thesaurus lists as semantically related with t.
- Example from earlier: HOSPITAL $\rightarrow$ MEDICAL
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms
  - INTEREST RATE $\rightarrow$ INTEREST RATE FASCINATE
- 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
- 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.

Example for manual thesaurus: PubMed

Example for manual thesaurus: PubMed



Automatic thesaurus generation

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.
  - "car" $\approx$ "motorcycle" because both occur with "road", "gas" and "license", so they 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.
  - "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.

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.

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-occurence-based thesaurus: Examples

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

WordSpace demo on web

Query expansion at search engines

Query expansion at search engines

- Main source of query expansion at search engines: query logs

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].

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].
  - $\rightarrow$ "herbal remedies" is potential expansion of "herb".

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].
  - $\rightarrow$ "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.

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.

Take-away today

- Interactive relevance feedback: improve initial retrieval results by telling the IR system which docs are relevant / nonrelevant
- Best known relevance feedback method: Rocchio feedback
- Query expansion: improve retrieval results by adding synonyms / related terms to the query
  - Sources for related terms: Manual thesauri, automatic thesauri, query logs

Resources

- Chapter 9 of IIR
- Resources at `http://cislmu.org`
  - Salton and Buckley 1990 (original relevance feedback paper)
  - Spink, Jansen, Ozmultu 2000: Relevance feedback at Excite
  - Justin Bieber: related searches fail
  - Word Space
  - Schütze 1998: Automatic word sense discrimination (describes a simple method for automatic thesaurus generation)