Kevin Knight, http://www.isi.edu/natural-language/people/pictures/ieee-expert-1.gif

Relevance Feedback
Query Expansion

David Kauchak
cs458
Fall 2012

adapted from:
http://www.stanford.edu/class/cs276/handouts/lecture9-queryexpansion.ppt

Administrative

- computer use in class ☺
- hw3 out
- assignment 3 out later today
  - due date?

Anomalous State of Knowledge

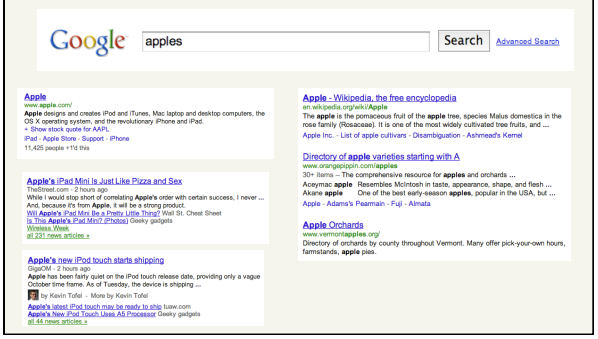
Basic paradox:

Information needs arise because the user doesn't know something

Search systems are designed to satisfy these needs, but the user needs to know what he is looking for

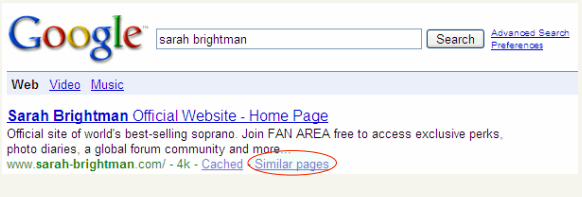
However, if the user knows what he's looking for, there may not be a need to search in the first place

What should be returned?



What is actually returned…

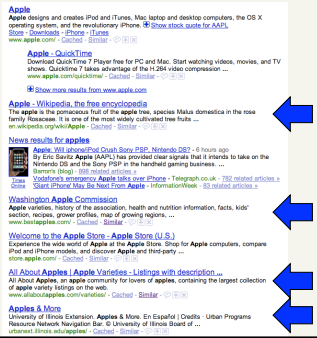
Similar pages



What did "similar pages" do?

Does this solve our problem?

Relevance feedback

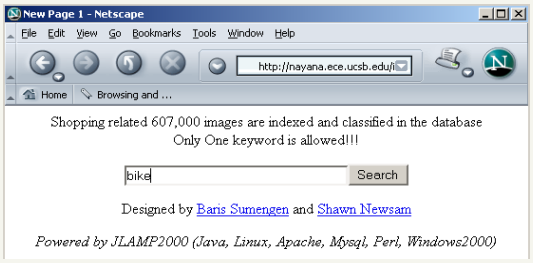


User provides feedback on relevance of documents in the initial set of results:

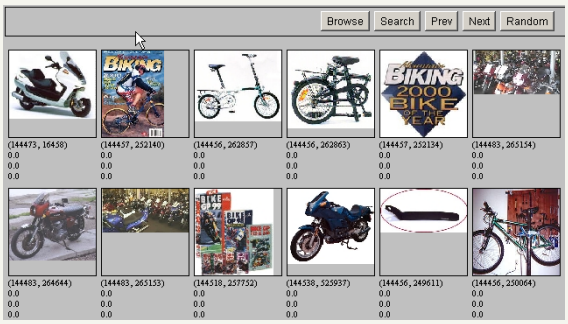
- User issues a query
- The user marks some results as relevant or non-relevant
- The system computes a better results based on the feedback
- May iterate

An example

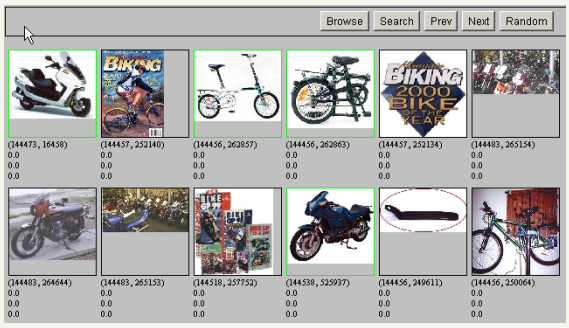
Image search engine:
http://vision.ece.ucsb.edu/multimedia/



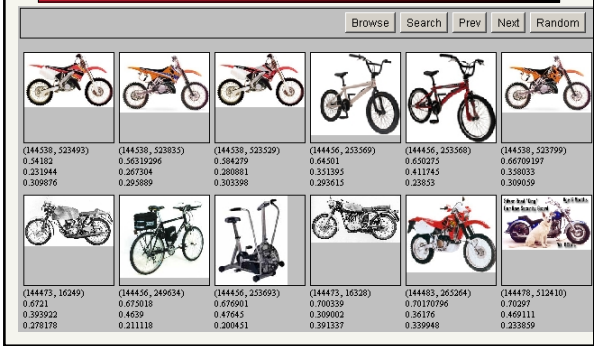
Results for initial query



Relevance Feedback



Results after Relevance Feedback



Ideas?

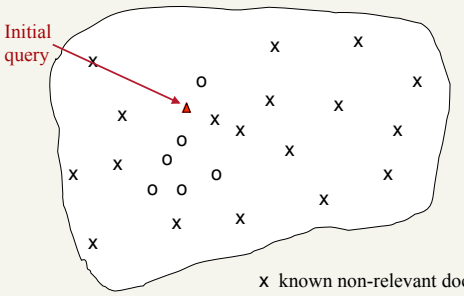
For ranked models we represent our query as a vector of weights, which we view as a point in a high dimensional space

0	4	0	8	0	0

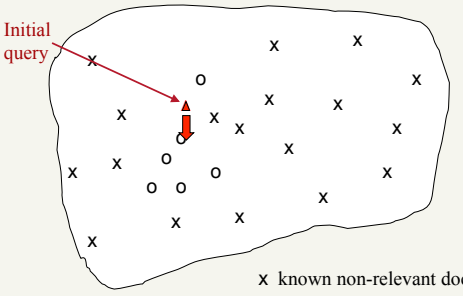
We want to bias the query towards documents that the user selected (the "relevant documents")

We want to bias the query away from documents that the user did not select (the "non-relevant documents")

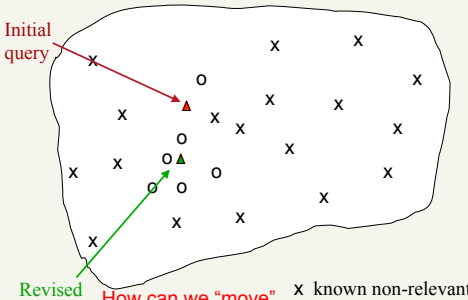
Relevance feedback



Relevance feedback



Relevance feedback on initial query



Rocchio Algorithm

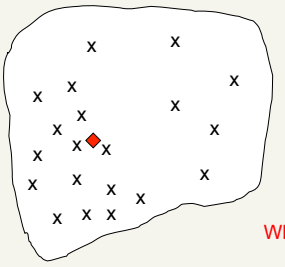
The Rocchio algorithm uses the vector space model to pick a better query

Rocchio seeks the query q_{opt} that maximizes the difference between the query similarity with the relevant set of documents (C_r) vs. the non-relevant set of documents (C_{nr})

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

Centroid

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



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

Where is the centroid?

Rocchio Algorithm

Find the new query by moving it towards the centroid of the relevant queries and away from the centroid of the non-relevant queries

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

Rocchio in action

query vector = original query vector
+ relevant vector
– non – relevant vector

Original query	0 4 0 8 0 0	
Relevant centroid	1 2 4 0 0 1	(+)
Non-relevant centroid	2 0 1 1 0 4	(-)
New query	-1 6 3 7 0 -3	?

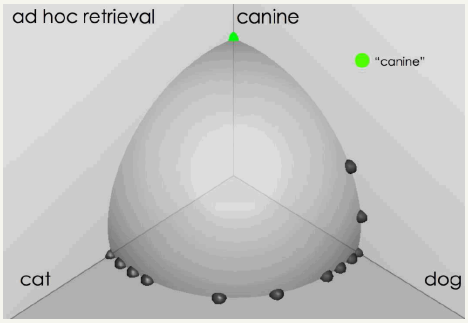
Rocchio in action

query vector = original query vector
+ relevant vector
– non – relevant vector

Original query	0 4 0 8 0 0	
Relevant centroid	1 2 4 0 0 1	(+)
Non-relevant centroid	2 0 1 1 0 4	(-)
New query	0 6 3 7 0 0	

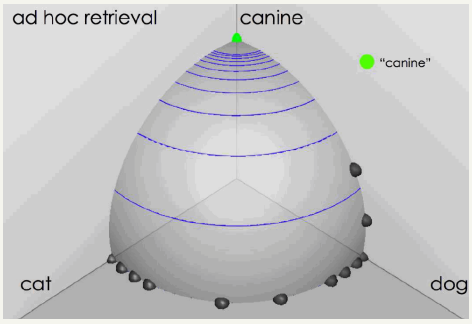
Rocchio in action

source: Fernando Diaz



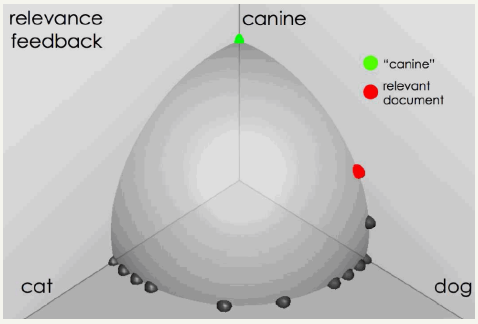
Rocchio in action

source: Fernando Diaz



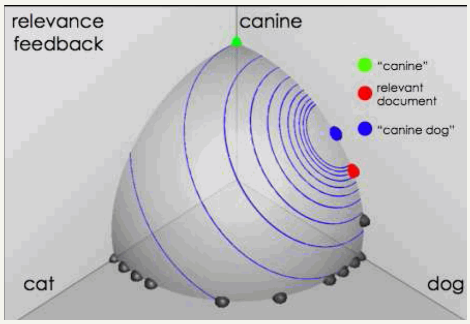
User feedback: Select what is relevant

source: Fernando Diaz



Results after relevance feedback

source: Fernando Diaz



Any problems with this?

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

C_r and C_{nr} are *all* the relevant and non-relevant documents

We get a biased sample!

Rocchio 1971 Algorithm (SMART)

Used in practice:

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

D_r = set of known relevant doc vectors
D_{nr} = set of known irrelevant doc vectors
- Different from C_r and C_{nr}

q_m = modified query vector
q_0 = original query vector
α,β,γ: weights (hand-chosen or set empirically)

New query moves toward relevant documents and away from irrelevant documents

Relevance Feedback in vector spaces

Relevance feedback can improve recall and precision

How might it improve each of these?

Which do you think it's more likely to improve?

Relevance Feedback in vector spaces

Relevance feedback can improve recall and precision

Relevance feedback is most useful for increasing *recall* in situations where recall is important

- Users can be expected to review results and to take time to iterate

Positive feedback is more valuable than negative feedback (so, set $\gamma < \beta$; e.g. $\gamma = 0.25$, $\beta = 0.75$).

Many systems only allow positive feedback (γ=0)

Another example

Initial query: *New space satellite applications*

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

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

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

Any problem with this?

Relevance Feedback: Problems

Long queries are inefficient for typical IR engine

- Long response times for user
- High cost for retrieval system
- Partial solution:
  - Only reweight certain prominent terms
    - Perhaps top 20 by term frequency

Users are often reluctant to provide explicit feedback

It's often harder to understand why a particular document was retrieved after applying relevance feedback

Will relevance feedback work?

Brittany Speers

hígado

Cosmonaut

RF assumes the user has sufficient knowledge for initial query

Misspellings - Brittany Speers

Cross-language information retrieval – hígado

Mismatch of searcher's vocabulary vs. collection vocabulary: cosmonaut/astronaut

Relevance Feedback on the Web

Some search engines offer a similar/related pages feature (this is a trivial form of relevance feedback)

- Google (used to…)
- Altavista
- Stanford WebBase

But some don't because it's hard to explain to average user:

- Google
- Alltheweb
- msn
- Yahoo
- Excite initially had true relevance feedback, but abandoned it due to lack of use

Excite Relevance Feedback

Spink et al. 2000

Only about 4% of query sessions from a user used relevance feedback option

- Expressed as "More like this" link next to each result

But about 70% of users only looked at the first page of results and didn't pursue things further

- So 4% is about 1/8 of people extending search

Relevance feedback improved results about 2/3rds of the time

Pseudo 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)

How well do you think it works?

Any concerns?

Pseudo 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

What is query drift?

- http://www.youtube.com/watch?v=i1AwFY6MuwE

Expanding the query

We would like to suggest alternative query formulations to the user with the goal of:

- increasing precision
- increasing recall

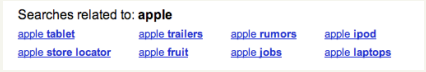
What are methods we might try to accomplish this?

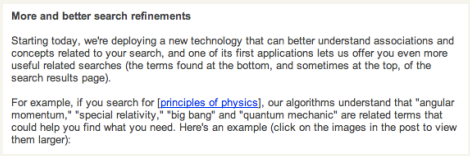
Increasing precision

Query assist:

- Generally done by query log mining
- Recommend frequent recent queries that contain partial string typed by user

Increasing precision





http://googleblog.blogspot.com/2009/03/two-new-improvements-to-google-results.html

Increasing recall: query expansion

Automatically expand the query with related terms and run through index

Spelling correction can be thought of a special case of this

cosmonaut ⇒ cosmonaut astronaut space pilot

How might we come up with these expansions?

How do we augment the user query?

Manual thesaurus
- E.g. MedLine: physician, syn: doc, doctor, MD, medico
- Wordnet

Global Analysis: (static; of all documents in collection)
- Automatically derived thesaurus
  - (co-occurrence statistics)
- Refinements based on query log mining
  - Common on the web

Local Analysis: (dynamic)
- Analysis of documents in result set

Example of manual thesaurus

Thesaurus-based query expansion

For each term, *t*, in a query, expand the query with synonyms and related words of *t* from the thesaurus
- feline → feline cat

May weight added terms less than original query terms.

May significantly decrease precision, particularly with ambiguous terms
- "interest rate" → "interest rate fascinate evaluate"

There is a high cost of manually producing a thesaurus
- And for updating it for scientific changes

Automatic thesaurus generation

Given a large collection of documents, how might we determine if two words are synonyms?

Two words are synonyms if they co-occur with similar words

I drive a car	I drive an automobile
I bought new tires for my car	I bought new tires for my automobile
can I hitch a ride with you in your car	can I hitch a ride with you in your automobile

Automatic thesaurus generation

Given a large collection of documents, how might we determine if two words are synonyms?

Two words are synonyms if they co-occur with similar words

I drive a car

I bought new tires for my car

can I hitch a ride with you in your car

I drive an automobile

I bought new tires for my automobile

can I hitch a ride with you in your automobile

Automatic Thesaurus Generation Example

word	ten nearest neighbors
absolutely	absurd whatsoever totally exactly nothing
bottomed	dip copper drops topped slide trimmed slig
captivating	shimmer stunningly superbly plucky witty
doghouse	dog porch crawling beside downstairs gazec
Makeup	repellent lotion glossy sunscreen Skin gel p
mediating	reconciliation negotiate cease conciliation p
keeping	hoping bring wiping could some would othe
lithographs	drawings Picasso Dali sculptures Gauguin
pathogens	toxins bacteria organisms bacterial parasite
senses	grasp psyche truly clumsy naive innate awl

Automatic Thesaurus Generation Discussion

Quality of associations is usually a problem

Term ambiguity may introduce irrelevant statistically correlated terms

- "Apple computer" → "Apple red fruit computer"

Since terms are highly correlated anyway, expansion may not retrieve many additional documents

Discussion

Certain query expansion techniques have thrived and many have disappeared (particularly for web search). Why? Which ones have survived?

IR: touching base