

Information Retrieval

- Query expansion

Jian-Yun Nie

Recap of previous lectures

- Indexing
- Traditional IR models
- Lemur toolkit
 - How to run it
 - How to modify it (for query expansion)
- Evaluating a search engine
 - Benchmarks
 - Measures

This lecture

- Improving results
 - For high recall. E.g., searching for *aircraft* didn't match with *plane*; nor *thermodynamic* with *heat*
- Options for improving results...
 - Focus on relevance feedback
 - The complete landscape
 - Global methods
 - Query expansion
 - Thesauri
 - Automatic thesaurus generation
 - Local methods
 - Relevance feedback
 - Pseudo relevance feedback

Review of traditional IR models

- Document and query are represented by a set of terms, organized in some way (vector, probabilistic model, ...)
- Preprocessing on words: stemming to create the same term for related words
- Each indexing term is considered to represent a unique meaning

Assumptions

1. Different terms are assumed to represent different meanings

- phone vs. telephone
- Information retrieval vs. search engine
- Consequence: silence – relevant documents are not retrieved

2. A term is assumed to represent only one meaning

- table: furniture, data structure, ...
- office: a work place, an organization, software,
- Consequence: noise: irrelevant documents are retrieved

Possible ways to deal with the first problem

- Different terms may represent the same meaning
 - Create a semantic representation
 - Each term is mapped to a concept
 - Two terms representing the same meaning are mapped to the same concept
 - Problem: requires extensive semantic resources – not feasible at large scale now (may be done in specialized area such as medicine)
 - Using relationships between terms in retrieval
 - Term b means the same thing as term a ($b \rightarrow a$)
 - Query a :
 - Match documents containing a
 - Match documents containing b
 - Equivalent to consider a query $b \vee a$ (Query expansion)
 - If b is only *related* to a (not the same meaning), one may want to decrease the weight of b in the query

Possible ways to deal with the second problem

- A term may mean different things (ambiguity)
 - Semantic representation
 - A term is mapped to different concepts depending on what it means
 - Term disambiguation
 - Often difficult to do, and the experiments using word sense disambiguation has not proven to be effective
 - Use compound terms/phrases instead of single terms
 - Office update
 - Office address
 - Q: Does this help in practice?
- We will come back on this later

Query expansion

- Goal: extend the initial query by adding related terms
- E.g. phone number → phone number, telephone
- Why is it necessary to expand query?
 - Queries are short: 2-3 words
 - They do not include all the words that may describe the information need
 - They only describe some of the aspects of the information need
 - Can we automatically complete the query so as to arrive at a better and more complete description of the information need?
- Key problems
 - Recognize term relationships: phone→telephone
 - Determine how strongly the new term is related to the query
 - Combine the new terms with the initial query

Term relationships

- Various relationships between terms:
 - Syntactic:
 - ADJ-NN (e.g. beautiful campus)
 - Lexical
 - NN → ADJ (e.g. computation → computational)
 - Semantic:
 - Synonymy: computer ↔ electronic computer
 - Hypernymy: computer → machine
 - Hyponymy: machine → computer
 - Related to: program → computer
 - ...
- What relationships are useful for IR?

Term relationships

- Useful relationships for IR
 - When one asks for documents on a , a document on b can also be relevant
 - $a \rightarrow b$ is useful
- Usually, relationships are defined between terms
 - Assumption: a query expanded by related terms is a related query, and the documents matching the expanded query also match the initial query

How to determine term relationships?

- Thesaurus
 - A thesaurus contains a set of manually defined relations between terms
 - Synonymy, hypernymy, hyponymy, meronymy, holonymy, ...
- Term co-occurrences in documents
 - Two terms that co-occur often are related
- Relevance feedback
 - Terms extracted from the relevant documents are related to the query

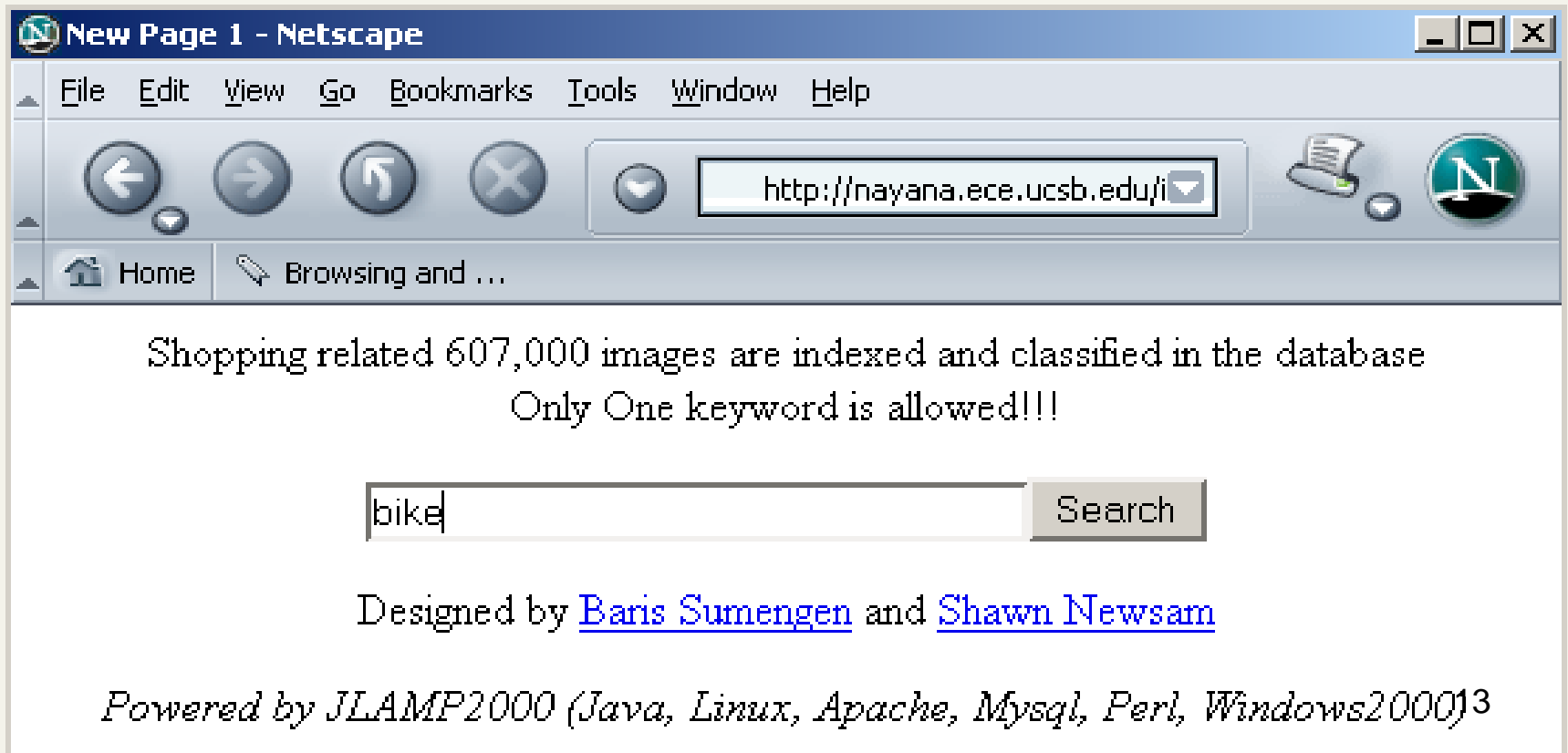
Relevance Feedback

- Relevance feedback: user feedback on relevance of docs in initial set of results
 - User issues a (short, simple) query
 - The **user** marks returned documents as relevant or non-relevant.
 - The **system** computes a better representation of the information need based on feedback.
 - Relevance feedback can go through one or more **iterations**.
- Idea: it may be difficult to formulate a good query when you don't know the collection well, so iterate

Relevance Feedback: Example

- Image search engine

<http://nayana.ece.ucsb.edu/imsearch/imsearch.html>



The screenshot shows a Netscape browser window titled "New Page 1 - Netscape". The address bar contains the URL "http://nayana.ece.ucsb.edu/i". The main content area displays the following text:













Shopping related 607,000 images are indexed and classified in the database
Only One keyword is allowed!!!

Below the text is a search input field containing the word "bike" and a "Search" button.

At the bottom of the page, it says "Designed by [Baris Sumengen](#) and [Shawn Newsam](#)" and "Powered by JLAMP2000 (Java, Linux, Apache, Mysql, Perl, Windows2000)3".













Results for Initial Query

Browse Search Prev Next Random

					
(144473, 16458) 0.0 0.0 0.0	(144457, 252140) 0.0 0.0 0.0	(144456, 262857) 0.0 0.0 0.0	(144456, 262863) 0.0 0.0 0.0	(144457, 252134) 0.0 0.0 0.0	(144483, 265154) 0.0 0.0 0.0
					
(144483, 264644) 0.0 0.0 0.0	(144483, 265153) 0.0 0.0 0.0	(144518, 257752) 0.0 0.0 0.0	(144538, 525937) 0.0 0.0 0.0	(144456, 249611) 0.0 0.0 0.0	(144456, 250064) 0.0 0.0 0.0

Relevance Feedback

Navigation buttons: [Browse](#) [Search](#) [Prev](#) [Next](#) [Random](#)

					
(144473, 16458) 0.0 0.0 0.0	(144457, 252140) 0.0 0.0 0.0	(144456, 262857) 0.0 0.0 0.0	(144456, 262863) 0.0 0.0 0.0	(144457, 252134) 0.0 0.0 0.0	(144483, 265154) 0.0 0.0 0.0
					
(144483, 264644) 0.0 0.0 0.0	(144483, 265153) 0.0 0.0 0.0	(144518, 257752) 0.0 0.0 0.0	(144538, 525937) 0.0 0.0 0.0	(144456, 249611) 0.0 0.0 0.0	(144456, 250064) 0.0 0.0 0.0

Results after Relevance Feedback

[Browse](#)[Search](#)[Prev](#)[Next](#)[Random](#)

(144538, 523493)
0.54182
0.231944
0.309876



(144538, 523835)
0.56319296
0.267304
0.295889



(144538, 523529)
0.584279
0.280881
0.303398



(144456, 253569)
0.64501
0.351395
0.293615



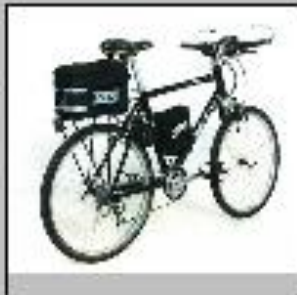
(144456, 253568)
0.650275
0.411745
0.23853



(144538, 523799)
0.66709197
0.358033
0.309059



(144473, 16249)
0.6721
0.393922
0.278178



(144456, 249634)
0.675018
0.4639
0.211118



(144456, 253693)
0.676901
0.47645
0.200451



(144473, 16328)
0.700339
0.309002
0.391337



(144483, 265264)
0.70170796
0.36176
0.339948



(144478, 512410)
0.70297
0.4691146
0.233859

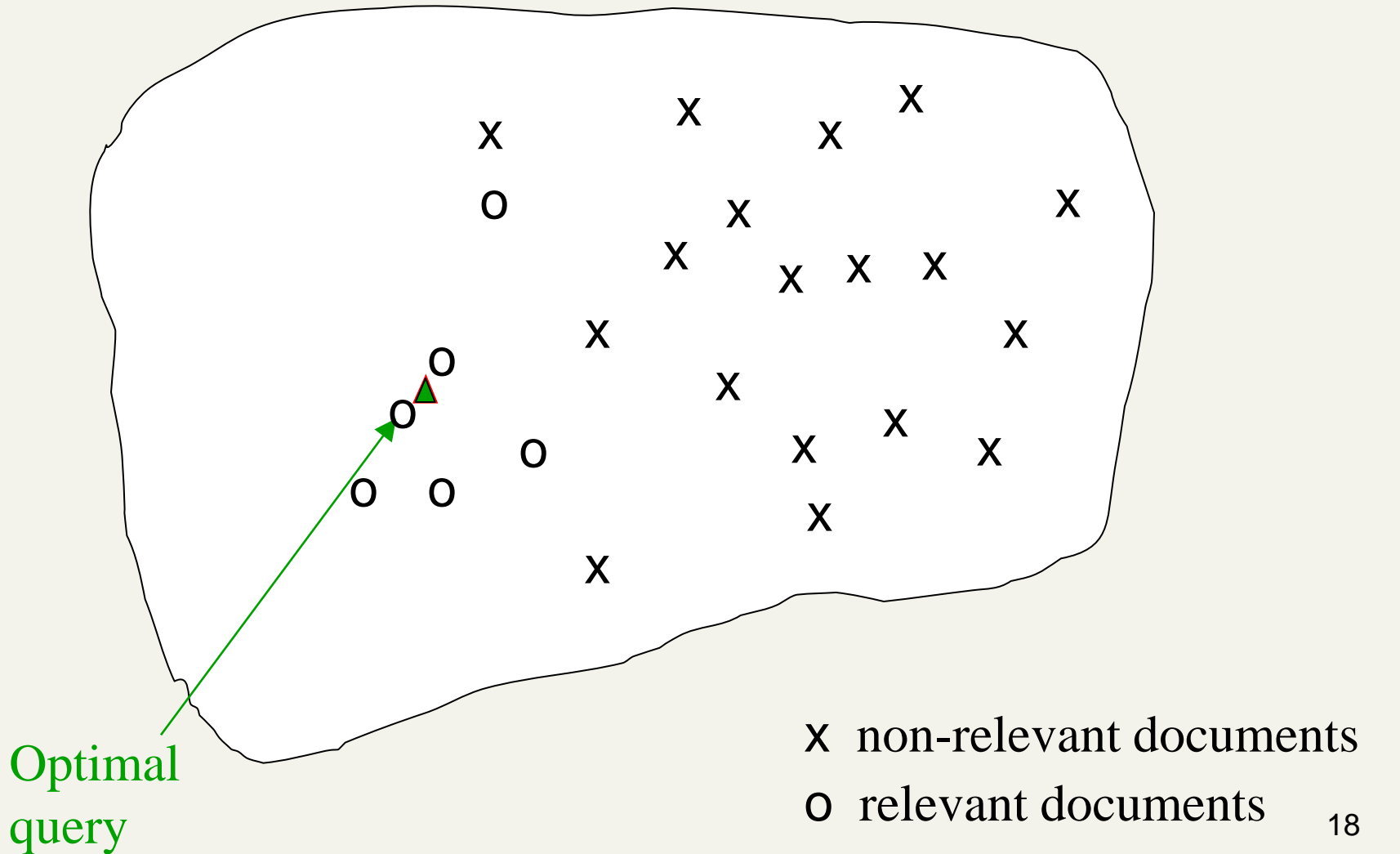
Rocchio Algorithm

- The Rocchio algorithm incorporates relevance feedback information into the vector space model.
- Want to maximize $sim(Q, C_r) - sim(Q, C_{nr})$
- The optimal query vector for separating relevant and non-relevant documents (with cosine sim.):

$$\vec{Q}_{opt} = \frac{1}{|C_r|} \sum_{\vec{d}_j \in C_r} \vec{d}_j - \frac{1}{N - |C_r|} \sum_{\vec{d}_j \notin C_r} \vec{d}_j$$

- Q_{opt} = optimal query; C_r = set of rel. doc vectors; N = collection size
- Unrealistic: we don't know relevant documents.

The Theoretically Best Query



Rocchio 1971 Algorithm (SMART)

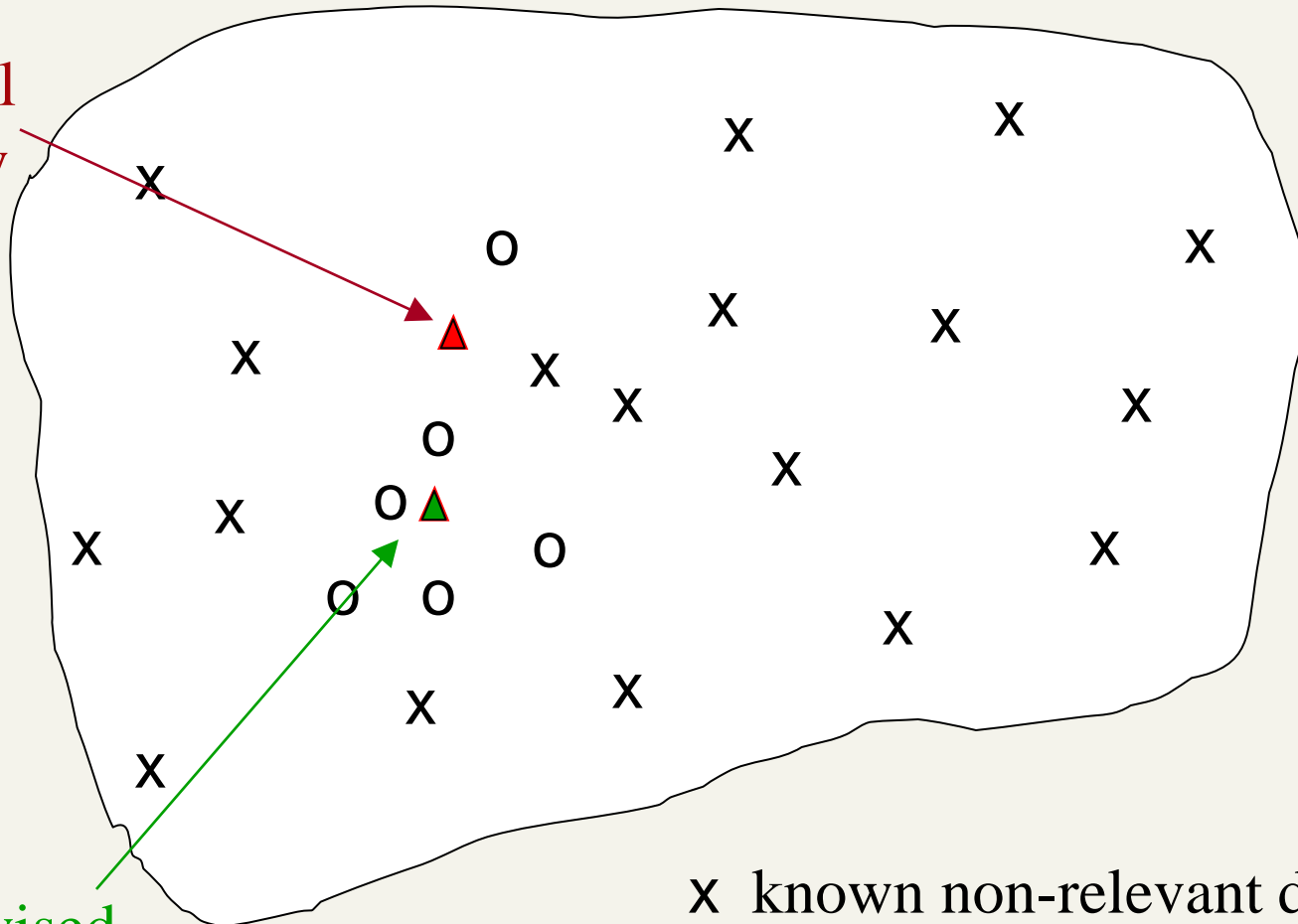
- Used in practice:

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

- q_m = modified query vector; q_0 = original query vector; α, β, γ : weights (hand-chosen or set empirically); D_r = set of known relevant doc vectors; D_{nr} = set of known irrelevant doc vectors
- New query moves toward relevant documents and away from irrelevant documents
- Tradeoff α vs. β/γ : If we have a lot of judged documents, we want a higher β/γ .
- Term weight can go negative
 - Negative term weights are ignored

Relevance feedback on initial query

Initial query



Revised query

x known non-relevant documents

o known relevant documents

Relevance Feedback in vector spaces

- We can modify the query based on relevance feedback and apply standard vector space model.
- Use only the docs that were marked.
- Relevance feedback can improve recall and precision
- Relevance feedback is *believed* to be 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 vs Negative Feedback

- 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 ($\gamma=0$).



Probabilistic relevance feedback

- Rather than reweighting in a vector space...
- If user has told us some relevant and irrelevant documents, then we can proceed to build a classifier, such as a Naive Bayes model:
 - $P(t_k|R) = |\mathbf{D}_{rk}| / |\mathbf{D}_r|$
 - $P(t_k|NR) = (N_k - |\mathbf{D}_{rk}|) / (N - |\mathbf{D}_r|)$
 - t_k = term in document; \mathbf{D}_{rk} = known relevant doc containing t_k ; N_k = total number of docs containing t_k
- Cf. classification
 - This is effectively another way of changing the query term weights
 - But note: the above proposal preserves no memory of the original weights

Relevance Feedback: Assumptions

- A1: User has sufficient knowledge for initial query.
- A2: Relevance prototypes are “well-behaved”.
 - Term distribution in relevant documents will be similar
 - Term distribution in non-relevant documents will be different from those in relevant documents
 - Either: All relevant documents are tightly clustered around a single prototype.
 - Or: There are different prototypes, but they have significant vocabulary overlap.
 - Similarities between relevant and irrelevant documents are small

Violation of A1

- User does not have sufficient initial knowledge.
- Examples:
 - Misspellings (Brittany Speers).
 - Cross-language information retrieval (hígado).
 - Mismatch of searcher's vocabulary vs. collection vocabulary
 - Cosmonaut/astronaut

Violation of A2

- There are several relevance prototypes.
- Examples:
 - Burma/Myanmar
 - Contradictory government policies
 - Pop stars that worked at Burger King
- Often: instances of a general concept
- Good editorial content can address problem
 - Report on contradictory government policies

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 apply relevance feedback



Relevance Feedback Example: Initial Query and Top 8 Results

- 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

Note: want high recall

Relevance Feedback Example: Expanded Query

- 2.074 new
- 30.816 satellite
- 5.991 nasa
- 4.196 launch
- 3.516 instrument
- 3.004 bundespost
- 2.790 rocket
- 2.003 broadcast
- 0.836 oil
- 15.106 space
- 5.660 application
- 5.196 eos
- 3.972 aster
- 3.446 arianespace
- 2.806 ss
- 2.053 scientist
- 1.172 earth
- 0.646 measure

Top 8 Results After Relevance Feedback

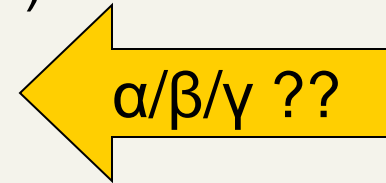
- + 1. 0.513, 07/09/91, NASA Scratches Environment Gear From Satellite Plan
- + 2. 0.500, 08/13/91, NASA Hasn't Scrapped Imaging Spectrometer
- 3. 0.493, 08/07/89, When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own
- 4. 0.493, 07/31/89, NASA Uses 'Warm ' Superconductors 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

Evaluation of relevance feedback strategies

- Use q_0 and compute precision and recall graph
- Use q_m and compute precision recall graph
 - Assess on all documents in the collection
 - Spectacular improvements, but ... it's cheating!
 - Partly due to known relevant documents ranked higher
 - Must evaluate with respect to documents not seen by user
 - Use documents in residual collection (set of documents minus those assessed relevant)
 - Measures usually then lower than for original query
 - But a more realistic evaluation
 - Relative performance can be validly compared
- Empirically, one round of relevance feedback is often very useful. Two rounds is sometimes marginally useful.

Relevance Feedback on the Web

- Some search engines offer a similar/related pages feature (this is a trivial form of relevance feedback)
 - Google (link-based)
 - Altavista
 - Stanford WebBase
- But some don't because it's hard to explain to average user:
 - 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 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 2/3 of the time

Other Uses of Relevance Feedback

- Following a changing information need
- Maintaining an information filter (e.g., for a news feed)
- Active learning
 - [Deciding which examples it is most useful to know the class of to reduce annotation costs]

Relevance Feedback Summary

- Relevance feedback has been shown to be very effective at improving relevance of results.
 - Requires enough judged documents, otherwise it's unstable (≥ 5 recommended)
 - Requires queries for which the set of relevant documents is medium to large
- Full relevance feedback is painful for the user.
- Full relevance feedback is not very efficient in most IR systems.
- Other types of interactive retrieval may improve relevance by as much with less work.

Pseudo-Relevance Feedback

- As true relevance feedback is hard to obtain, we assume that the top-ranked documents in the initial retrieval results are *relevant*
- Then we use the same query modification process to create a new query
- Notice that this subset of documents are not all relevant. However, they are often *more* relevant than the documents at lower ranks. So, they still capture some characteristics of relevance.

Pseudo-Relevance Feedback in VSM

- Vector space model

$$\vec{q}_m = a\vec{q}_0 + b \frac{1}{|D_F|} \sum_{\vec{d}_j \in D_F} \vec{d}_j$$

D_F = set of top-ranked documents

- Usually, we also select the k strongest terms from top-ranked documents, i.e. only keep the k terms in $\frac{1}{|D_F|} \sum_{\vec{d}_j \in D_F} \vec{d}_j$ and let the other terms to be 0.

- A typical figure: use top 10-20 documents, and the 20-100 strongest terms.
- There are experiments with massive expansion, e.g. using 500 terms from 100 top documents. But this is unrealistic.

Pseudo-Relevance Feedback in LM

- KL divergence:

$$\text{Score}(Q, D) = \sum_{t_i \in Q} P(t_i | q_Q) \log P(t_i | q_D)$$

Query expansion = a new $P(t_i | q_Q)$

$$\text{Document model: } P(t_i | q_D) = \frac{tf(t_i, D) + mP(t_i | C)}{|D| + m}, \text{ } m\text{-pseudo-count}$$

Expanding query model

$$P(q_i | q_Q) = \lambda P_{ML}(q_i | q_Q) + (1 - \lambda) P_F(q_i | q_Q)$$

$P_{ML}(t_j | q_Q)$: Max.Likelihood unigram model (not smoothed)

$P_F(t_j | q_Q)$: Feedback model

$$\text{Score}(Q, D) = \sum_{q_i \in V} P(q_i | q_Q) \cdot \log P(q_i | q_D)$$

$$= \sum_{q_i \in V} [\lambda P_{ML}(q_i | q_Q) + (1 - \lambda) P_F(q_i | q_Q)] \cdot \log P(q_i | q_D)$$

$$= \lambda \sum_{q_i \in Q} P_{ML}(q_i | q_Q) \cdot \log P(q_i | q_D) + (1 - \lambda) \sum_{q_i \in V} P_F(q_i | q_Q) \cdot \log P(q_i | q_D)$$

Classical
LM

Feedback
model

Estimating the feedback model

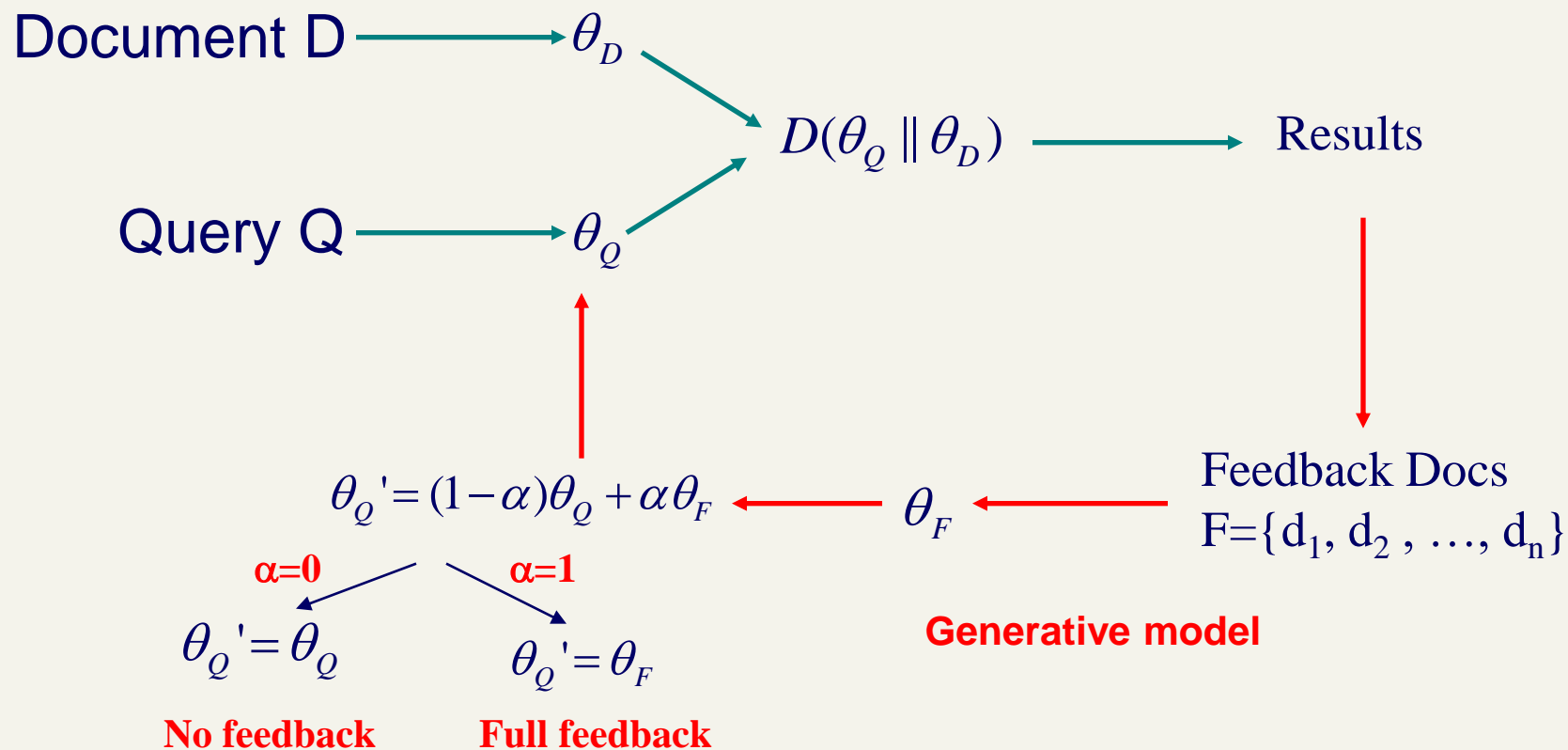
- Relevance Model (Lavrenko and Croft, 2001)
 - Viewing top-ranked documents as relevance samples

$$P(w|q_q) = (1 - l)P(w|q_o) + l P(w|q_R)$$

$$P(w|q_R) \propto \sum_{D \in F} P(w|D)P(D|Q)$$

- Mixture Model (Zhai and Lafferty, 2001)
 - Generating top-ranked documents from two sources: topic model and general model (collection)
 - Fitting topic model with *EM* algorithm by maximizing the likelihood of top-ranked documents

Feedback as Model Interpolation



Mixture model – estimating feedback model by EM

- Principle of Expectation Maximization (EM): Given a set of documents, determine a component model so that the global model can maximize the likelihood of the set of documents

- Global model (documents' likelihood) to maximize

$$\log p(F | \theta) =$$

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

- E-step
$$t^{(n)}(w) = \frac{(1 - \lambda) p_\lambda^{(n)}(w | \theta_F)}{(1 - \lambda) p_\lambda^{(n)}(w | \theta_F) + \lambda p(w | C)}$$

- M-step
$$p_\lambda^{(n+1)}(w | \theta_F) = \frac{\sum_{j=1}^n c(w; d_j) t^{(n)}(w)}{\sum_i \sum_{j=1}^n c(w_i; d_j) t^{(n)}(w_i)}$$

Results with mixture feedback model

Collection		Simple LM	Mixture FB	Improv.	Div. Min.	Improv.
AP88-89	AvgPr	0.210	0.296	+ 41%	0.295	+ 40%
	InitPr	0.617	0.591	- 4%	0.617	+ 0%
	Recall	3067/ 4805	3888/ 4805	+ 27%	3665/ 4805	+ 19%
TREC8	AvgPr	0.256	0.282	+ 10%	0.269	+ 5%
	InitPr	0.729	0.707	- 3%	0.705	- 3%
	Recall	2853/ 4728	3160/ 4728	+ 11%	3129/ 4728	+ 10%
WEB	AvgPr	0.281	0.306	+ 9%	0.312	+ 11%
	InitPr	0.742	0.732	- 1%	0.728	- 2%
	Recall	1755/ 2279	1758/ 2279	+ 0%	1798/ 2279	+ 2%

Divergence minimization is another model based on feedback documents (see Zhai and Lafferty 2001)

The complete landscape

- Local methods
 - Relevance feedback
 - Pseudo relevance feedback
 - Variant: use passage retrieval, and top-retrieved passages for feedback (better than document feedback)
(Why?)
- Global methods
 - Query expansion/reformulation
 - Thesauri (e.g. WordNet, HowNet)
 - Automatic thesaurus generation

Query Expansion

- In relevance feedback, users give additional input (relevant/non-relevant) on **documents**, which is used to extract additional terms for query
- In query expansion, users give additional input (good/bad search term) on **words or phrases**.

Query Expansion: Example

YOU ARE HERE > [Home](#) > [My InfoSpace](#) > [Meta-Search](#) > Web Search Results

Web Search Results

Your Search

Select: ▼

[Yellow Pages](#) [White Pages](#) [Classifieds](#)

Are you looking for?

[Jacksonville Jaguars](#)

[Jaquar Car](#)

[Black Jaguar](#)

[Jaquar Xk8](#)

[Wild Jaguars](#)

[Jaquare](#)

[Jaguar Accessories](#)

[Jaguar Automobile](#)

Also: see www.altavista.com, www.teoma.com

Controlled Vocabulary

The screenshot displays the PubMed search interface. At the top left is the NCBI logo, and at the top center is the PubMed logo. On the top right is the National Library of Medicine (NLM) logo. Below the logos is a navigation bar with tabs for PubMed, Nucleotide, Protein, Genome, Structure, PopSet, and Taxonomy. The search bar contains the text "PubMed" in a dropdown menu, followed by "for cancer". To the right of the search bar are "Go" and "Clear" buttons. Below the search bar is a row of links: "Limits", "Preview/Index", "History", "Clipboard", and "Details". On the left side, there is a vertical menu with links for "About Entrez", "Text Version", "Entrez PubMed", "Overview", "Help | FAQ", "Tutorial", "New/Noteworthy", "E-Utilities", "PubMed Services", "Journals Database", "MeSH Browser", "Single Citation", and "MetaBox". The main content area shows the "PubMed Query:" section with the following query:

```
("neoplasms"[MeSH Terms] OR cancer[Text Word])
```

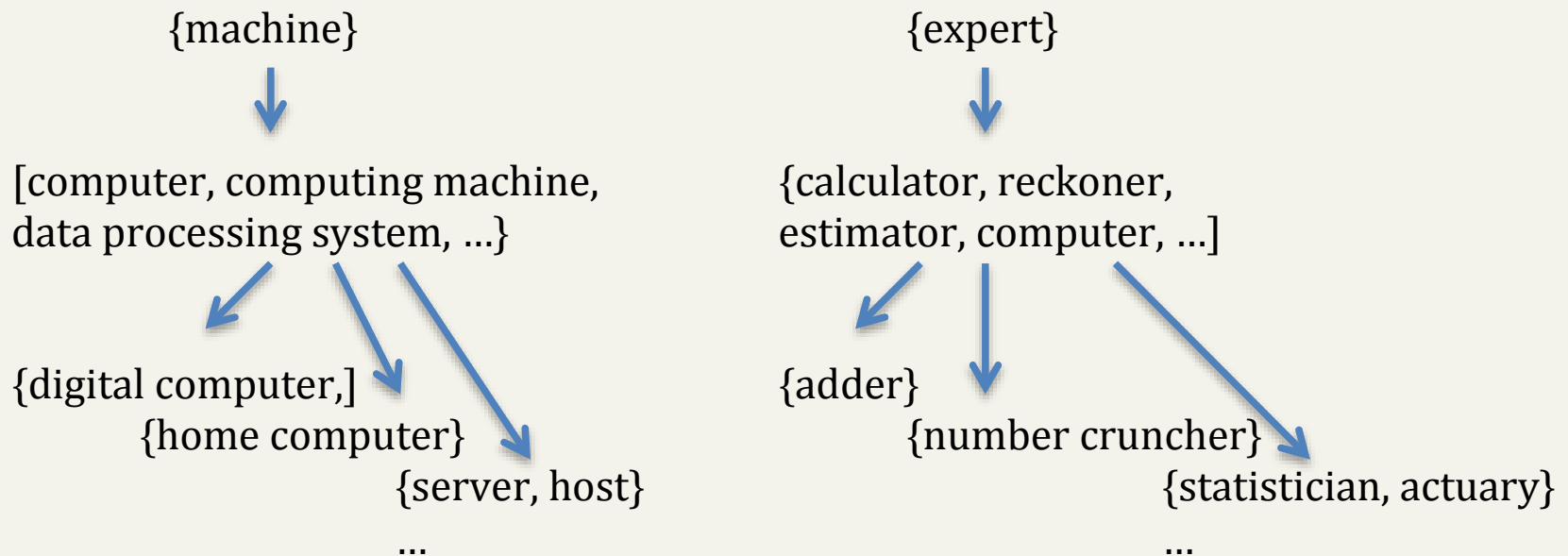
 At the bottom of the search bar area, there are "Search" and "URL" buttons.

Types of Query Expansion

- Global Analysis: Thesaurus-based
 - Controlled vocabulary
 - Maintained by editors (e.g., medline)
 - Manual thesaurus
 - E.g. MedLine: physician, syn: doc, doctor, MD, medico
 - Automatically derived thesaurus
 - (co-occurrence statistics)
 - Refinements based on query log mining
 - Common on the web
- Local Analysis (better than Global analysis – see Xu and Croft):
 - Analysis of documents in result set

QE based on thesauri

- Thesaurus
 - Relations between terms
 - E.g. WordNet (<http://wordnetweb.princeton.edu/perl/webwn>)



Thesaurus-based Query Expansion

- This doesn't require user input
- 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.
- Generally increases recall.
- Widely used in many science/engineering fields
- 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

Problems using Wordnet

- Strength: the stored relations are manually validated
- However:
 - Coverage: not all the terms are included
 - Usefulness: not all the related terms are useful for IR (e.g. computer → computing machine)
 - Ambiguity: computer → machine? Or → expert?
 - Lack of weighting: strong and weak relations are not distinguished
- Experiments
 - (Voorhees 1994, 1995): Wordnet does not help
 - Automatically adding related terms decreases effectiveness
 - Even adding correct related synsets does not help
 - Others: some improvements using appropriate weighting (according to collection statistics)

Automatic extraction of term relationships

- Co-occurrence: two terms occur at the same time within the same text (fragment)
- Assumption: The more two terms co-occur, the stronger is the relationship between them.
- Two aspects to consider:
 - The context in which co-occurrences are considered: document, paragraph, sentence, passage, **text window** (e.g. 10 words)...
 - Calculation of the strength:
 - $f(a \rightarrow b) = \# \text{co-occ}(a, b) / \# \text{occ}(a)$ or $\# \text{co-occ}(a, b) / \sum_c \# \text{co-occ}(a, c)$
 - $f(a \rightarrow b) = |A \cap B| / |A \cup B|$ $A = \text{set of contexts of } a$
 - ...

Automatic Thesaurus Generation

Example

word	ten nearest neighbors
absolutely	absurd whatsoever totally exactly nothing
bottomed	dip copper drops topped slide trimmed slight
captivating	shimmer stunningly superbly plucky witty
doghouse	dog porch crawling beside downstairs gazed
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 l
pathogens	toxins bacteria organisms bacterial parasite
senses	grasp psyche truly clumsy naive innate awl

Analysis of co-occurrence relations

- Coverage: large
- Nature: no semantic nature can be determined, but usually suggest that the two terms are used in the same contexts, to describe the same topics
- Problems
 - Noise: semantically unrelated terms can co-occur
 - Silence: related terms may not co-occur often (tyre-tire)
 - Co-occ. are domain dependent: usually cannot use the results obtained from one area to another area.
- Strength
 - Weighting
 - Co-occurring terms can correspond to some related topics (may be relevant)
 - Better effectiveness than thesauri (Wordnet)

How to integrate term relationships?

Vector space model

■ Voorhees (1993, 1994)

- The expansion terms form another query vector
- Similarities with the original vector and the expansion vector are interpolated

■ Qiu and Frei (1995)

- Determine a relation of the term to the whole query
 - Sum up its relations to all the query terms

$$\text{Sim}_{qt}(q,t) = \sum_{t_i \in q} q_i \cdot \text{SIM}(t_i,t)$$

q_i : weight of t_i in q

- Expect to reduce noise and ambiguity:
 - terms related to one query term may not be strongly related to the query
 - Q: Is this true?

How to integrate term relationships?

- In LM, based on translation model (Berger and Lafferty 1999):

$$p_a(q | \mathbf{d}) = ap(q | D) + (1 - a) p(q | \mathbf{d})$$

$$= ap(q | D) + (1 - a) \sum_{w \in \mathbf{d}} l(w | \mathbf{d}) t(q | w)$$

- Recall KL-divergence model:

$$\text{Score}(Q, D) = \sum_{t_i \in Q} P(t_i | Q) \log P(t_i | D)$$

Query expansion = a new $P(t_i | Q)$

$$P(t_i | Q) = l \sum_{t_j \in Q} P(t_i | t_j) P(t_j | Q) + (1 - l) P(t_i | Q)$$

- Similar to Qiu and Frei 1995 (see discussions in Bai et al. 2006)

An example (from Bai et al. 06)

Coll.	UM	UQE	BQE
AP	0.2767	0.2891* (+4%)	0.3280** (+19%)
SJM	0.2017	0.2175** (+8%)	0.2456** (+22%)
WSJ	0.2373	0.2390 (+1%)	0.2564 (+8%)
FR	0.1966	0.2057 (+5%)	0.2331 (+19%)

UM- classical unigram LM
 UQE- unigram query expansion
 BQE- bigram query expansion

$$P'(t_i | Q) = \lambda \sum_{t_j, t_k \hat{=} Q} P(t_i | t_j t_k) P(t_j | Q) + (1 - \lambda) P(t_i | Q)$$

Query Reformulation: Vocabulary Tools

- Feedback
 - Information about stop lists, stemming, etc.
 - Numbers of hits on each term or phrase
- Suggestions
 - Thesaurus
 - Controlled vocabulary
 - Browse lists of terms in the inverted index
 - User queries (query log analysis) (see Baidu, Google, Bing, ...)
- Users use suggestions, but not feedback. **Why?**

Query Expansion: Summary

- Query expansion is often effective in increasing recall.
 - Not always with general thesauri
 - Fairly successful for subject-specific collections
- Usually, query expansion is considered as a means to increase recall, and it hurts precision. **This is not entirely true. Why?**
- Overall, not as useful as relevance feedback; *may* be as good as pseudo-relevance feedback

Query expansion v.s. document expansion

$$P(t_i | Q) = P_{ML}(t_i | D)$$



Document expansion

$$P(t_i | D) = \dot{\alpha}_{t_j} P(t_i | t_j) P(t_j | D)$$

$$P(t_i | D) = \int \dot{\alpha}_{t_j} P(t_i | t_j) P(t_j | D) + (1 - \int) P_{ML}(t_i | D)$$

$$P(t_i | D) = P_{ML}(t_i | Q)$$



Query expansion

$$P(t_i | Q) = \dot{\alpha}_{t_j} P(t_i | t_j) P(t_j | Q)$$

$$P(t_i | Q) = \int \dot{\alpha}_{t_j} P(t_i | t_j) P(t_j | Q) + (1 - \int) P_{ML}(t_j | Q)$$

Query expansion vs. document expansion (Cao et al. 2007)

- It is observed that when the same resource (e.g. term co-occ. statistics) are used for query expansion and document expansion, query expansion is more effective.
- Why?

Performance of General Model								
Coll.		UM	DE	%UM	QE	GM	%UM	%QE
AP	MAP	0.1925	0.2128	+11.06**	0.2580	0.2629	+22.96**	+2.02
	Ret.	3289	3523		3994	4064		
WSJ	MAP.	0.2466	0.2597	+5.02*	0.2860	0.2891	+11.62**	+1.08
	Ret.	1659	1706		1794	1845		
SJM	MAP.	0.2045	0.2142	+5.37	0.2522	0.2584	+19.91**	+2.46
	Ret.	1417	1572		1621	1742		

* and ** mean statistical significance at level of $p < 0.05$ and $p < 0.01$.

Questions for discussion (next lecture)

- Why is it necessary to perform query expansion? Why isn't it so successful in practice even though it may increase retrieval effectiveness?
- Compare and contrast different ways to do query expansion
- What is the difference between global query expansion (or global context analysis) and local query expansion (local context analysis)? Why did some experiments show better performances with local query expansion?
- Are the strong terms extracted from feedback documents all useful? If not, how to select?
- Term relationships extracted are used in a context-independent way (i.e. one term is considered related to another whatever the query). What are the possible problems this may bring? What are your solutions?
- Do you have a way to make query expansion useable by end users?
- We talked about query expansion using feedback documents, thesauri and co-occurrence statistics. What other methods/resources can you think of to do query expansion?
- Does query expansion allows us to move IR from term matching to sense matching?

Resources

MG Ch. 4.7

MIR Ch. 5.2 – 5.4

Yonggang Qiu , Hans-Peter Frei, Concept based query expansion. *SIGIR* 16: 161–169, 1993.

Schuetze: Automatic Word Sense Discrimination, Computational Linguistics, 1998.

Singhal, Mitra, Buckley: Learning routing queries in a query zone, ACM SIGIR, 1997.

Buckley, Singhal, Mitra, Salton, New retrieval approaches using SMART: TREC4, NIST, 1996.

Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288-297, 1990.

Bai, J. Nie, J.-Y., Bouchard,H., Cao, G. Using Query Contexts in Information Retrieval, *SIGIR*, pp. 15-22, 2007.

Cao G. Nie, J.Y., Bai, J., Using Markov Chains to Exploit Word Relationships in Information Retrieval, *RIAO*, 2007

G. Cao, J.Y. Nie, J. Gao, S. Robertson, Selecting Good Expansion Terms for Pseudo-Relevance Feedback, *ACM-SIGIR*, 2008, pp. 243-250

Resources

Harman, D. (1992): Relevance feedback revisited. *SIGIR 15*: 1-10

Chris Buckley, Gerard Salton, and James Allan.

The effect of adding relevance information in a relevance feedback environment.

In *SIGIR 17*, pages 292-300, Dublin, Ireland, 1994.

Xu, J., Croft, W.B. (1996): Query Expansion Using Local and Global Document Analysis, in *SIGIR 19*: 4-11.

Spink, A., Jansen, J. and Ozmultu, H.C. (2000) "Use of query reformulation and relevance feedback by Excite users." *Internet Research: Electronic Networking Applications and Policy*.

http://ist.psu.edu/faculty_pages/jjansen/academic/pubs/internetresearch2000.pdf

Zhai, C., Lafferty, J., Model-based feedback in the language modeling approach to information retrieval, CIKM 2001.

Berger, A., Lafferty, J., Information retrieval as statistical translation, *SIGIR 1999*, pp. 222-229.