Relevance Feedback and Query Expansion

Debapriyo Majumdar
Information Retrieval
Indian Statistical Institute Kolkata

Synonymy and Polysemy

Synonyms

Different words with (almost) the same meaning

Chennai and **Madras**

Car and **automobile**

Problem: the query contains **Chennai**, the document contains **Madras** $\Longrightarrow$ the document is not retrieved

Misses relevant documents (false negative) $\Longrightarrow$ decreases recall

Polysems

One word with different meanings

bank of Spain and **bank** of the river

pay the **fine** and **fine** jewellery

Problem: the query contains **bank** $\Longrightarrow$ documents containing the word bank in all senses are retrieved

Retrieves some non-relevant documents (false positive) $\Longrightarrow$ decreases precision

Importance of Recall

- Academic importance
- Not only of academic importance
  - Uncertainty about availability of information: are the returned documents relevant at all?
  - Query words may return small number of documents, none so relevant
  - Relevance is not graded, but documents missed out could be more useful to the user in practice
- What could have gone wrong?
  - Many things, for instance …
  - Some other choice of query words would have worked better
  - Searched for **aircraft**, results containing only **plane** were not returned

The gap between the user and the system

User needs some information

A retrieval system tries to bridge this gap

Assumption: the required information is present somewhere

The gap
- The retrieval system can only rely on the query words (in the simple setting)
- Wish: if the system could get another chance …

The gap between the user and the system

User needs some information

A retrieval system tries to bridge this gap

Assumption: the required information is present somewhere

If the system gets another chance

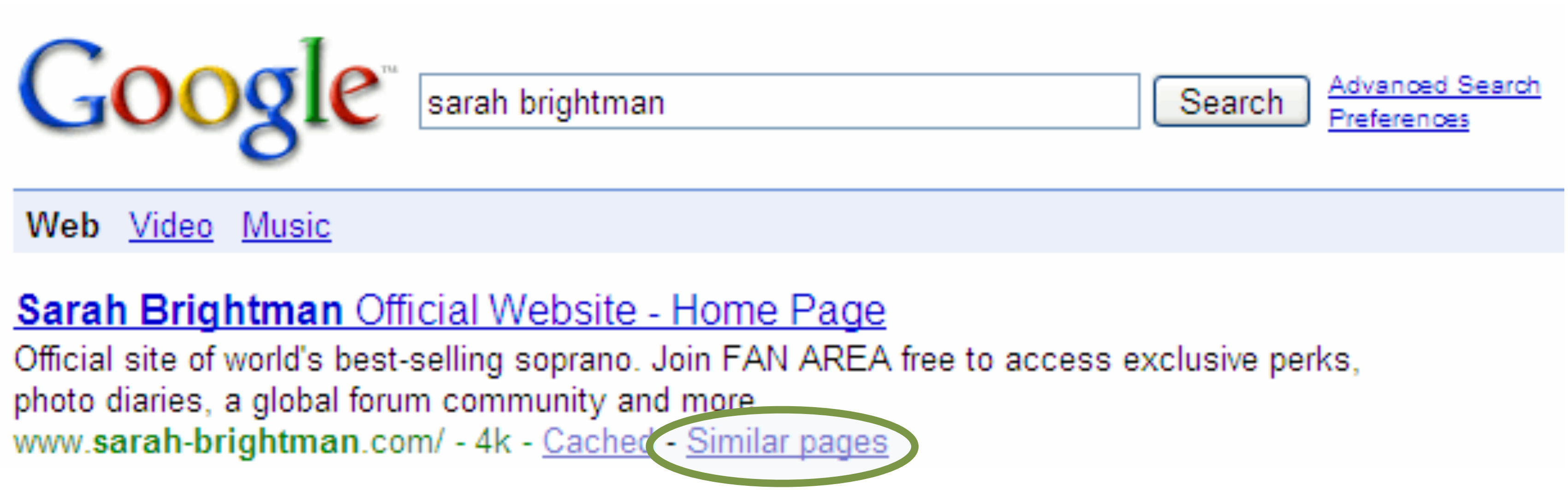
- Modify the query to fill the gap better
- Usually more query terms are added → *query expansion*
- The whole framework is called *relevance feedback*

Relevance Feedback

- User issues a query
  - Usually short and simple query
- The system returns some results
- The user marks some results 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.
  - It may be difficult to formulate a good query when you don't know the collection well, so iterate

Example: similar pages

Old time Google



- If you (the user) tell me that this result is relevant, I can give you more such relevant documents

Example 2: Initial query/results

- 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 some 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

2 1. 0.513, 07/09/91, NASA Scratches Environment Gear From Satellite Plan

1 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

8 5. 0.492, 12/02/87, Telecommunications Tale of Two Companies

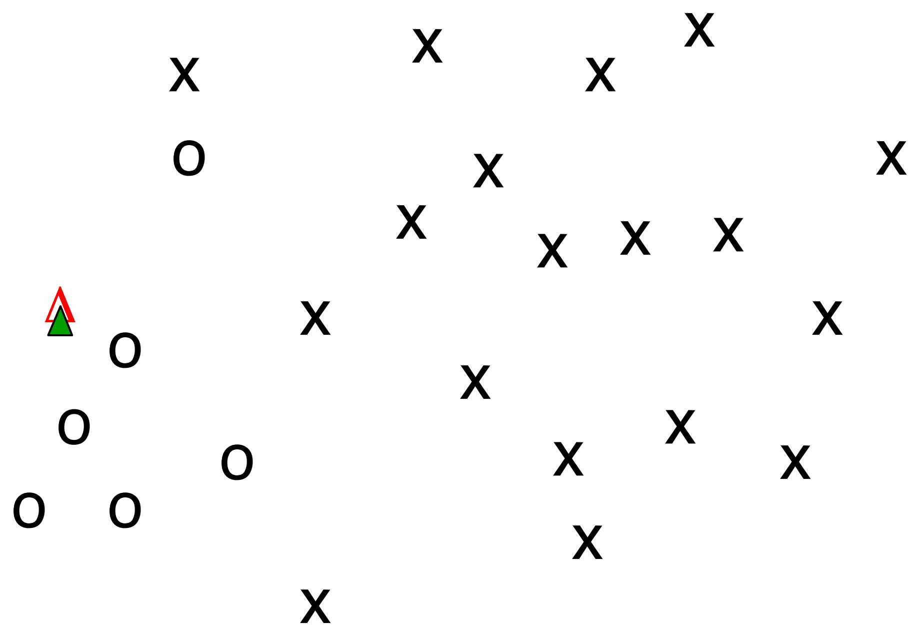
6. 0.491, 07/09/91, Soviets May Adapt Parts of SS-20 Missile For Commercial Use

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

The theoretically best query

The information need is best "realized" by the relevant and non-relevant documents



Optimal query

x non-relevant documents

o relevant documents

Key concept: 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
- Definition: Centroid

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

where C is a set of documents.

Rocchio Algorithm

- The Rocchio algorithm uses the vector space model to pick a relevance feedback query
- Rocchio seeks the query $\vec{q}_{opt}$ that maximizes

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

- Tries to separate docs marked relevant and non-relevant
- Problem: we don't know the truly relevant docs

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

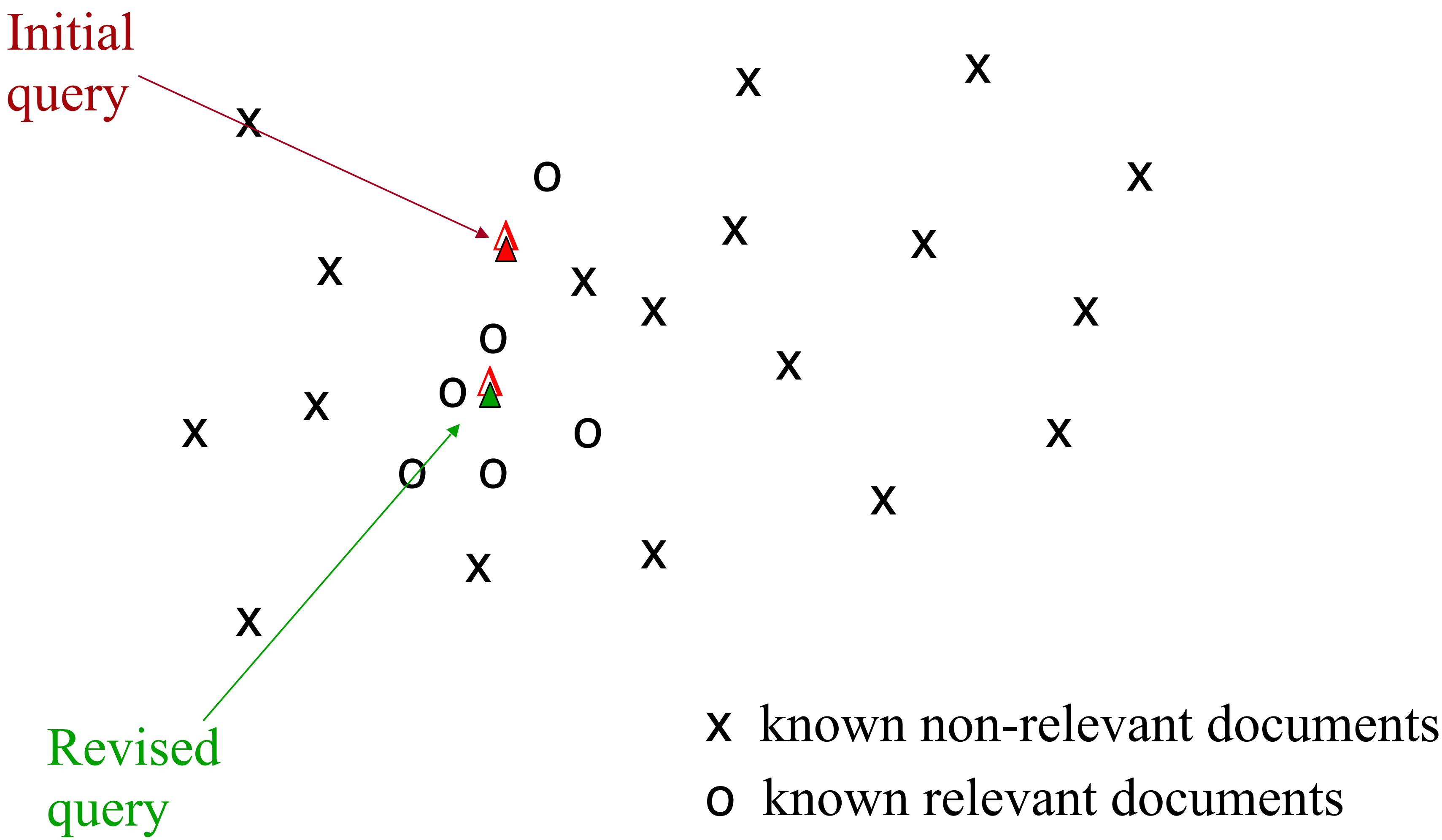
Rocchio Algorithm (SMART system)

- 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 <u>known</u> relevant doc vectors
- D_{nr} = set of <u>known</u> 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
- Tradeoff α vs. β/γ : If we have a lot of judged documents, we want a higher β/γ.
- Some weights in query vector can go negative
  - Negative term weights are ignored (set to 0)

Relevance feedback on initial query



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).

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 query
  - Mismatch of searcher’s vocabulary vs. collection vocabulary
    - car / auto

Violation of A2

- There are several relevance prototypes
- Examples:
  - Burma/Myanmar
  - Contradictory government policies
- Significantly different instances of a general concept

- Good editorial content can address this problem
  - Report on contradictory government policies

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

Evaluation of relevance feedback

- Second method – assess only the docs *not* rated by the user in the first round
  - Could make relevance feedback look worse than it really is
  - Can still assess relative performance of algorithms
- Most satisfactory – use two collections each with their own relevance assessments
  - q_0 and user feedback from first collection
  - q_m run on second collection and measured

Evaluation: Caveat

- True evaluation of usefulness must compare to other methods taking the same amount of time.
- Alternative to relevance feedback: User revises and resubmits query.
- Users may prefer revision/resubmission to having to judge relevance of documents.
- There is no clear evidence that relevance feedback is the "best use" of the user's time.

Relevance Feedback: Problems

- 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

Relevance Feedback on the Web

- Some search engines offer a similar/related pages feature (a trivial form of relevance feedback)
  - Google old version (link-based)
  - Altavista
  - Stanford WebBase
- But some don't because it's hard to explain to average user:
  - Alltheweb
  - bing
  - Yahoo
- Excite initially had true relevance feedback, but abandoned it due to lack of use.

Relevance Feedback: study on usefulness

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

Pseudo relevance feedback

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

Query Expansion

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

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.
- 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
  - There are methods to build automatic thesaurus later in the course

Thesaurus

- A thesaurus provides information on synonyms and semantically related words and phrases.
- Example:

```
physician
   syn: ||croaker, doc, doctor, MD,
medical, mediciner, medico, ||sawbones
   rel: medic, general practitioner,
surgeon,
```

Thesaurus-based Query Expansion

- For each term, *t*, in a query, expand the query with synonyms and related words of *t* from the thesaurus.
- May weight added terms less than original query terms.
- Generally increases recall.
- May significantly decrease precision, particularly with ambiguous terms.
  - "interest rate" → "interest rate fascinate evaluate"

WordNet

- A more detailed database of semantic relationships between English words.
- Developed by famous cognitive psychologist George Miller and a team at Princeton University.
- About 144,000 English words.
- Nouns, adjectives, verbs, and adverbs grouped into about 109,000 synonym sets called *synsets*.

WordNet Synset Relationships

- Antonym: front → back
- Attribute: benevolence → good (noun to adjective)
- Pertainym: alphabetical → alphabet (adjective to noun)
- Similar: unquestioning → absolute
- Cause: kill → die
- Entailment: breathe → inhale
- Holonym: chapter → text (part to whole)
- Meronym: computer → cpu (whole to part)
- Hyponym: plant → tree (specialization)
- Hypernym: apple → fruit (generalization)

WordNet Query Expansion

- Add synonyms in the same synset.
- Add hyponyms to add specialized terms.
- Add hypernyms to generalize a query.
- Add other related terms to expand query.

Statistical Thesaurus

- Existing human-developed thesauri are not easily available in all languages.
- Human thesuari are limited in the type and range of synonymy and semantic relations they represent.
- Semantically related terms can be discovered from statistical analysis of corpora.

Automatic Global Analysis

- Determine term similarity through a pre-computed statistical analysis of the complete corpus.
- Compute association matrices which quantify term correlations in terms of how frequently they co-occur.
- Expand queries with statistically most similar terms.

Association Matrix

$$
\begin{array}{c|cccc|}
 & w_1 & w_2 & w_3 \;\cdots\cdots & w_n \\
\hline
w_1 & c_{11} & c_{12} & c_{13} \cdots\cdots & c_{1n} \\
w_2 & c_{21} & & & \\
w_3 & c_{31} & & & \\
. & . & & & \\
. & . & & & \\
w_n & c_{n1} & & & \\
\hline
\end{array}
$$

c_{ij}: Correlation factor between term i and term j

$$c_{ij} = \sum_{d_k \in D} f_{ik} \times f_{jk}$$

f_{ik} : Frequency of term i in document k

Normalized Association Matrix

- Frequency based correlation factor favors more frequent terms.
- Normalize association scores:

$$s_{ij} = \frac{c_{ij}}{c_{ii} + c_{jj} - c_{ij}}$$

- Normalized score is 1 if two terms have the same frequency in all documents.

Metric Correlation Matrix

- Association correlation does not account for the proximity of terms in documents, just co-occurrence frequencies within documents.
- Metric correlations account for term proximity.

$$c_{ij} = \sum_{k_u \in V_i} \sum_{k_v \in V_j} \frac{1}{r(k_u, k_v)}$$

V_i: Set of all occurrences of term i in any document.

$r(k_u,k_v)$: Distance in words between word occurrences k_u and k_v

(∞ if k_u and k_v are occurrences in different documents).

Normalized Metric Correlation Matrix

- Normalize scores to account for term frequencies:

$$s_{ij} = \frac{c_{ij}}{|V_i| \times |V_j|}$$

Query Expansion with Correlation Matrix

- For each term i in query, expand query with the n terms, j, with the highest value of c_{ij} (s_{ij}).
- This adds semantically related terms in the "neighborhood" of the query terms.

Problems with Global Analysis

- 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.

Automatic Local Analysis

- At query time, dynamically determine similar terms based on analysis of top-ranked retrieved documents.
- Base correlation analysis on only the “local” set of retrieved documents for a specific query.
- Avoids ambiguity by determining similar (correlated) terms only within relevant documents.
  - “Apple computer” →
    “Apple computer Powerbook laptop”

Global vs. Local Analysis

- Global analysis requires intensive term correlation computation only once at system development time.
- Local analysis requires intensive term correlation computation for every query at run time (although number of terms and documents is less than in global analysis).
- But local analysis gives better results.

Global Analysis Refinements

- Only expand query with terms that are similar to *all* terms in the query.

$$\text{sim}(k_i, Q) = \sum_{k_j \in Q} c_{ij}$$

  - "fruit" not added to "Apple computer" since it is far from "computer."
  - "fruit" added to "apple pie" since "fruit" close to both "apple" and "pie."
- Use more sophisticated term weights (instead of just frequency) when computing term correlations.

Query Expansion Conclusions

- Expansion of queries with related terms can improve performance, particularly recall.
- However, must select similar terms very carefully to avoid problems, such as loss of precision.

Sources and Acknowledgements

- IR Book by Manning, Raghavan and Schuetze: http://nlp.stanford.edu/IR-book/
- Several slides are adapted from the slides by Prof. Nayak and Prof. Raghavan for their course in Stanford University