Information Retrieval and Web Search

Cornelia Caragea

Computer Science
University of Illinois at Chicago

Credits for slides: Mooney

Relevance Feedback and Query Expansion

Required Reading

- "Information Retrieval" textbook
  - Chapter 9: Relevance Feedback and Query Expansion

Introduction

- An information need may be expressed using different keywords (*synonymy*)

  → impact on recall
  → examples: ship vs boat, aircraft vs airplane

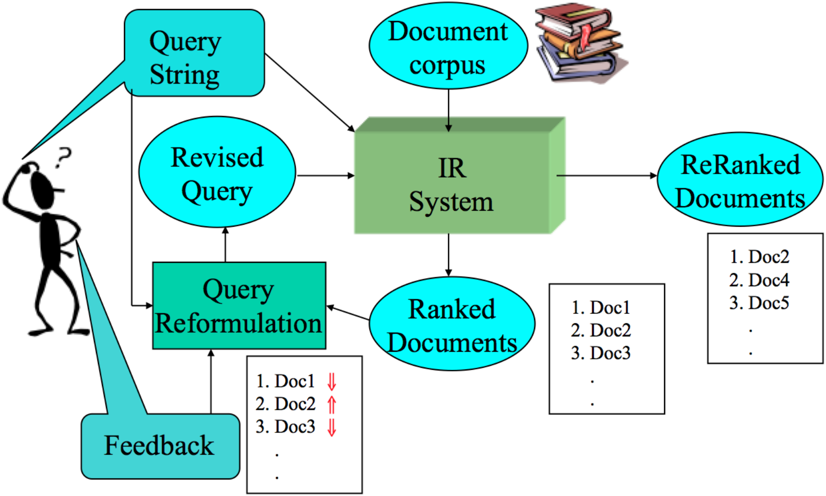
- A search for *aircraft* should ideally match *plane* only for references to an *airplane*, and not for *woodworking plane*.
- Solutions: refining queries manually *or* expanding queries (semi) automatically

- Semi-automatic query expansion:
  - based on the retrieved documents and the query (ex: **Relevance Feedback**)
  - independent of the query and results (ex: **thesaurus, spelling corrections**)

Relevance Feedback (RF)

- Involves the user in the retrieval process to improve the final result set
- After the initial retrieval results are presented, allow the user to provide feedback on the relevance of one or more of the retrieved documents
- Use this feedback information to reformulate the query
- Produce new results based on reformulated query
- RF allows for a more interactive, multi-pass process

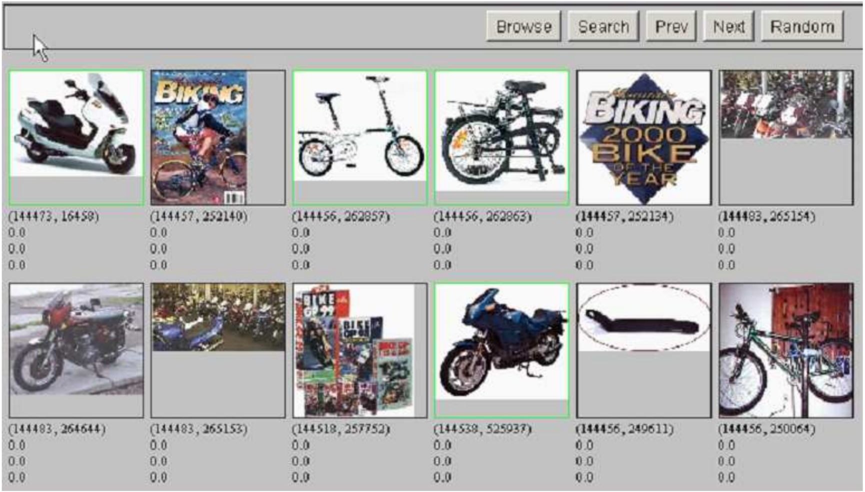
Relevance Feedback Architecture



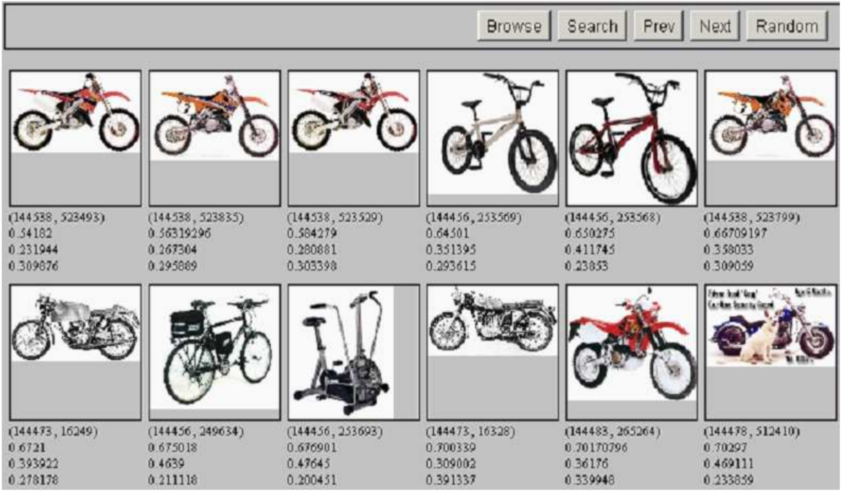
Why Relevance Feedback?

- Defining good queries is difficult when the collection is (partly) unknown
- It is easy to judge particular documents
- RF allows to deal with situations where the information needs of a user evolve with the checking of the retrieved documents

Relevance Feedback Searching Over Images



Relevance Feedback Searching Over Images



Query Reformulation

- Revise the query to account for feedback:
  - **Query Expansion:** Add new terms to the query from the relevant documents.
  - **Term Reweighting:** Increase weight of terms in relevant documents and decrease weight of terms in irrelevant documents.

Query Reformulation on Text Documents

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

Query Reformulation on Text Documents

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

Query Reformulation - Example

```
* 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
```

The Rocchio Algorithm for Relevance Feedback

- Rocchio is the classic algorithm for implementing relevance feedback.
    - Incorporates relevance feedback information into the vector space model.

Query Reformulation for the Vector Space Retrieval

- Change the query vector using vector algebra
- Find a query vector, $\vec{q}$ that maximizes similarity with relevant documents while minimizing similarity with non-relevant documents
  - **Add** the vectors for the **relevant** documents to the query vector
  - **Subtract** the vectors for the **irrelevant** docs from the query vector
  - This adds both positively and negatively weighted terms to the query as well as reweighting the initial terms

Optimal Query

- If C_r is the set of relevant documents and C_{nr} is the set of non-relevant documents, we want to find:

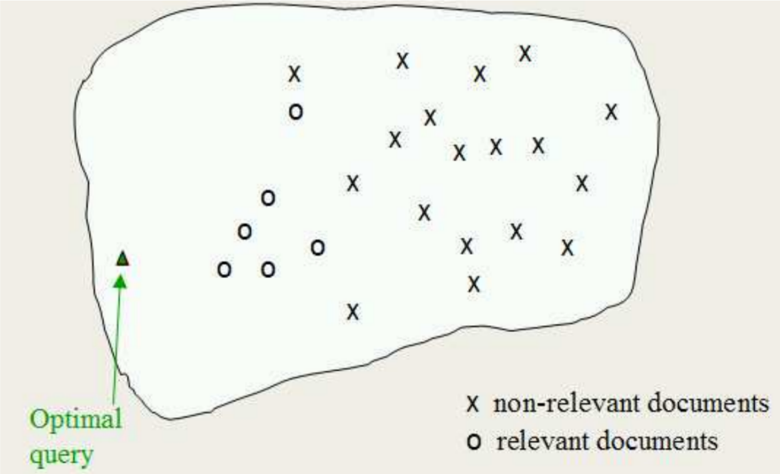
$$\vec{q}_{opt} = argmax_{\vec{q}}[sim(\vec{q}, C_r) - sim(\vec{q}, C_{nr})]$$

- Under cosine similarity, the optimal query vector for separating the relevant and non-relevant documents is:

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

- The optimal query is the vector difference between the centroids of the relevant and non-relevant documents

Optimal Query



The optimal query for separating relevant and non-relevant documents

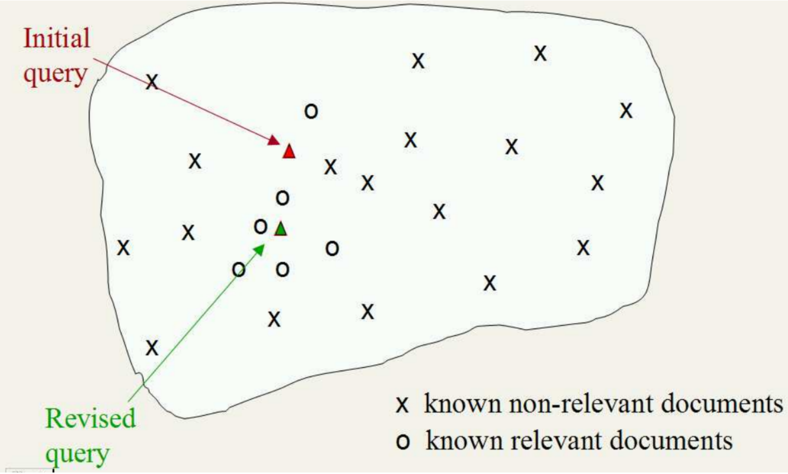
Standard Rocchio Method

- Since all relevant documents are generally unknown, just use the **known** relevant (D_r) and irrelevant (D_n) sets of documents and include the initial query q_0

$$\vec{q_m} = \alpha \vec{q_0} + \beta \frac{1}{|D_r|} \sum_{\vec{d_j} \in D_r} \vec{d_j} - \gamma \frac{1}{|D_{nr}|} \sum_{\vec{d_j} \in D_{nr}} \vec{d_j}$$

- α : Tunable weight for initial query
- β : Tunable weight for relevant documents
- γ : Tunable weight for irrelevant documents

Standard Rocchio Method



Evaluating Relevance Feedback

- By construction, the reformulated query will rank explicitly-marked relevant documents higher and explicitly-marked irrelevant documents lower
- The method should not get credit for improvement on these documents, since it was told their relevance
- In machine learning, this error is called "testing on the training data"
- Evaluation should focus on generalizing to other un-rated documents

Fair Evaluation of Relevance Feedback

- Remove from the corpus any documents for which feedback was provided
- Measure recall/precision performance on the remaining *residual collection*
- Compared to complete corpus, specific recall/precision numbers may decrease since relevant documents were removed
- However, **relative** performance on the residual collection provides fair data on the effectiveness of relevance feedback

Pseudo Feedback

- Users sometimes are reluctant to provide explicit feedback

- Use relevance feedback methods without explicit user input
- Just assume the top m retrieved documents are relevant, and use them to reformulate the query
- Allows for query expansion that includes terms that are correlated with the query terms

Thesaurus

- A thesaurus provides information on synonyms and semantically related words and phrases
- Example: physician
  - syn: doc, doctor, MD, medical, mediciner, medico
  - rel: medic, general practitioner, surgeon, anesthetist

Thesaurus-based Query Expansion

- For each term t in a query, expand the query with synonyms and related words of t from the thesaurus
- We may want to weigh added terms less than the original query terms
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms
  - "interest rate" $\rightarrow$ "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 Query Expansion

- Add synonyms in the same synset
  - “ship” and “boat”
- Add hyponyms to add specialized terms
  - “plant” and “tree”
- Add hypernyms to generalize a query
  - “apple” and “fruit”
- 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

Problems with Global Analysis

- Term ambiguity may introduce irrelevant statistically correlated terms
  - "Apple computer" $\rightarrow$ "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)
- Generally, local analysis gives better results

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