

# Relevance Feedback & Other Query Expansion Techniques

(Thesaurus, Semantic Network)

(CS429)  
Nazli Goharian  
nazli@ir.iit.edu

Slides are *mostly* based on Information Retrieval Algorithms and Heuristics, Grossman, Frieder

© Goharian, Grossman, Frieder, 2002, 2009

1

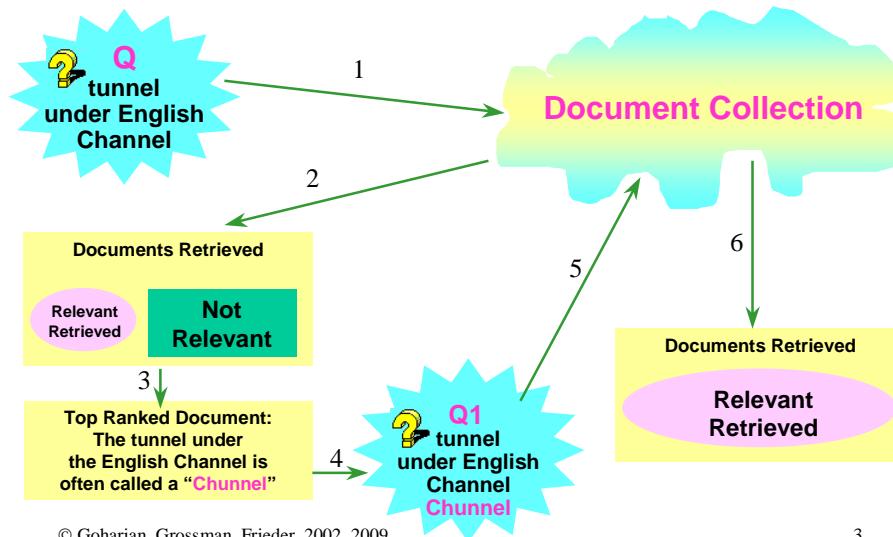
## Relevance Feedback

- The modification of the search process to improve the effectiveness of an IR system by incorporating information obtained from prior relevance judgments.
- Basic idea is to do an initial query, get feedback from the user (or automatically) as to what documents are relevant and then add term from known relevant document(s) to the query.

© Goharian, Grossman, Frieder, 2002, 2009

2

## Relevance Feedback Example



3

## Feedback Mechanisms

- Automatic (pseudo/ Blind)
  - The “good” terms from the “good”, top ranked documents, are selected by the system and added to the users query.
- Semi-automatic
  - User provides feedback as to which documents are relevant (via clicked document or selecting a set of documents); the “good” terms from those documents are added to the query.
  - Similarly terms can be shown to the user to pick from.
  - Suggesting new queries to the user based on:
    - Query log
    - Clicked document (limited to one document)

© Goharian, Grossman, Frieder, 2002, 2009

4

## Pseudo Relevance Feedback Algorithm

- Identify “good” ( $N$  top-ranked) documents.
- Identify all terms from the  $N$  top-ranked documents.
- Select the “good” ( $T$  top) feedback terms.
- Merge the feedback terms with the original query.
- Identify the top-ranked documents for the modified queries through relevance ranking.

© Goharian, Grossman, Frieder, 2002, 2009

5

## Sort Orders

- Methods to select the “good” terms:
  - $n$
  - $n*idf$  (a reasonable measure)
  - $f*idf$
  - $cf*n$
  - .....
- where:
  - $n$ : is number of documents in relevant set having term  $t$
  - $f$ : is frequency of term  $t$  in relevant set
  - $cf$ : collection frequency of term  $t$

© Goharian, Grossman, Frieder, 2002, 2009

6

## Example of Sort Orders

- Top 3 documents
  - d1: A, B, B, C, D
  - d2: C, D, E, E, A, A
  - d3: A, A, A
  - Assume *idf* of A, B, C is 1 and D, E is 2.

Term	n	f	n*idf	f*idf
A	3	6	3	6
B	1	2	1	2
C	2	2	2	2
D	2	2	4	4
E	1	2	2	4

© Goharian, Grossman, Frieder, 2002, 2009

7

## Original Rocchio Vector Space Relevance Feedback [1965]

- Step 1: Run the query.
- Step 2: Show the user the results.
- Step 3: Based on the user feedback:
  - add new terms to query or increase the query term weights.
  - Remove terms or decrease the term weights.
- *Objective* => *increase the query accuracy.*

© Goharian, Grossman, Frieder, 2002, 2009

8

## Rocchio Vector Space Relevance Feedback

$$Q' = \alpha Q + \beta \sum_{i=1}^{n_1} R_i - \gamma \sum_{i=1}^{n_2} S_i$$

- Q: original query vector
- R: set of relevant document vectors
- S: set of non-relevant document vectors
- $\alpha, \beta, \gamma$ : constants (Rocchio weights)
- Q': new query vector

© Goharian, Grossman, Frieder, 2002, 2009

9

## Variations in Vector Model

$$Q' = \alpha Q + \beta \sum_{i=1}^{n_1} R_i - \gamma \sum_{i=1}^{n_2} S_i$$

Options:

$$\alpha = 1, \beta = \frac{1}{|R|}, \gamma = \frac{1}{|S|}$$

$$\alpha = \beta = \gamma = 1$$

- Use only first  $n$  documents from R and S
- Use only first document of S
- Do not use S ( $\gamma=0$ )

© Goharian, Grossman, Frieder, 2002, 2009

10

## Example

D1: A, B, C    => < 1 1 1 >    Relevant  
D2: A, C        => < 1 0 1 >    Relevant  
D3: C            => < 0 0 1 >    Non-Relevant  
Q: A             => < 1 0 0 >

$$\frac{1}{|R|} \sum R : \langle 1, \frac{1}{2}, 1 \rangle$$

$$\frac{1}{|S|} \sum S : \langle 0, 0, 1 \rangle$$

$$Q' = \langle 2, \frac{1}{2}, 0 \rangle$$

© Goharian, Grossman, Frieder, 2002, 2009

11

## Implementing Relevance Feedback

- First obtain top documents, do this with the usual inverted index
- Now we need the top terms from the top  $X$  documents.
- Two choices
  - Retrieve the top  $x$  documents and scan them in memory for the top terms.
  - Use a separate doc-term structure that contains for each document, the terms that will contain that document.

© Goharian, Grossman, Frieder, 2002, 2009

12

## Relevance Feedback in Probabilistic Model

- Need training data for  $R$  and  $r$  (unlikely).
- Some other strategy like VSM can be used for the initial pass to get the top  $n$  docs, as the relevant docs
  - $R$  can be estimated as the total relevant docs found in top  $n$
  - $r$  is then estimated based on these documents
- As discussed in the *Probabilistic Model Section*,  $R$  and  $r$  can be set to zero. Query can be expanded using the expanded *Probabilistic Model* term weighting.

© Goharian, Grossman, Frieder, 2002, 2009

13

## Relevance Feedback in Probabilistic Model [Wu & Salton 1981]

### Four approaches:

1. Re-weight: Generate query  $Q'$  using weights after 1<sup>st</sup> pass (based on new  $R$  and  $r$ )
2. Re-weight: Combine old and new weight
3. Expand original query to include all new terms found in relevant documents
4. Expand original query with re-weighting

© Goharian, Grossman, Frieder, 2002, 2009

14

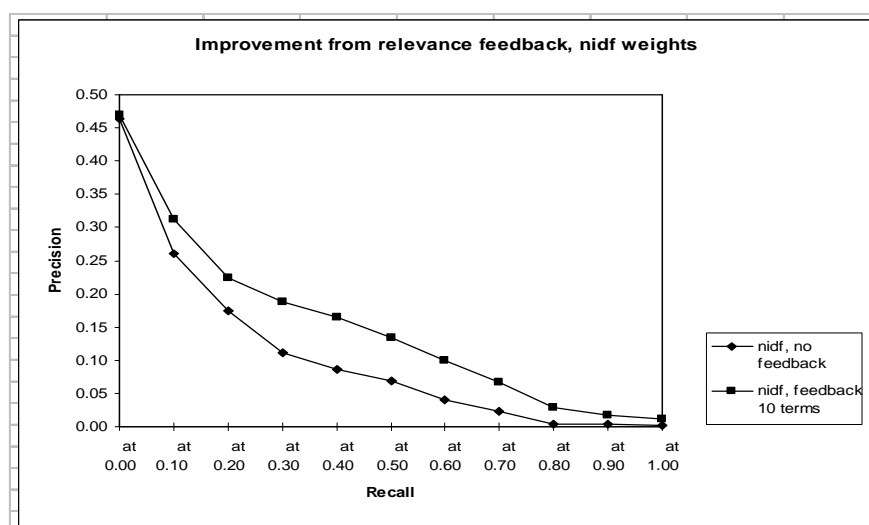
## Relevance Feedback Modifications

- Various techniques can be used to improve the relevance feedback process.
  - Number of Top-Ranked Documents
  - Number of Feedback Terms
  - Feedback Term Selection Techniques
  - Iterations
  - Term Weighting
  - Phrase versus single term
  - Document Clustering
  - Relevance Feedback Thresholding
  - Term Frequency Cutoff Points
  - Query Expansion Using a Thesaurus

© Goharian, Grossman, Frieder, 2002, 2009

15

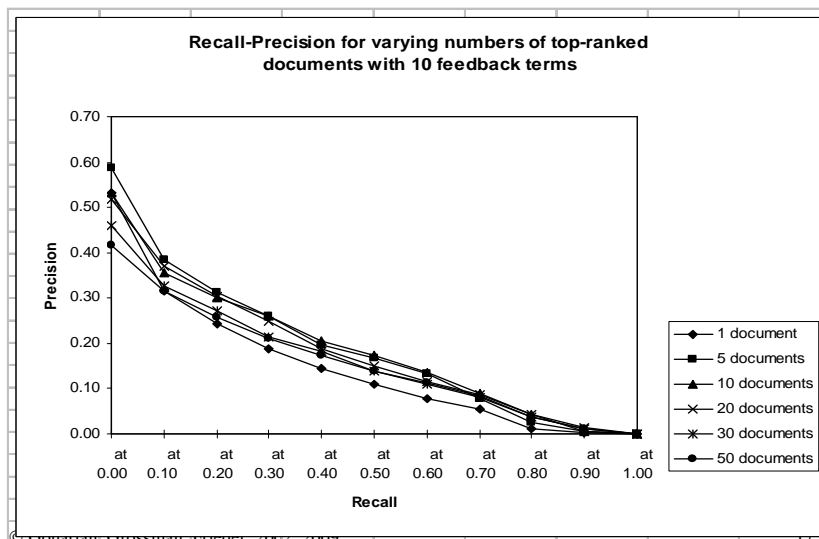
## Relevance Feedback Justification



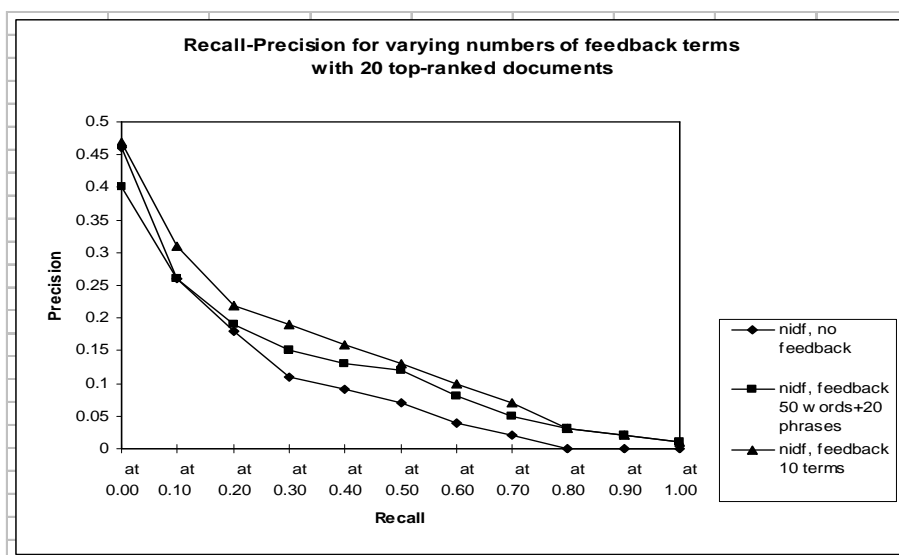
© Goharian, Grossman, Frieder, 2002, 2009

16

## Number of Top-Ranked Documents



## Number of Feedback Terms



## Summary of Relevance Feedback

- Pro
  - Relevance feedback usually improves average precision by increasing the number of good terms in the query (generally 10-15% improvement)
- Con
  - More computational work
  - Easy to decrease Precision (one horrible word can undo the good caused by lots of good words).

## Thesauri

- It is intuitive to use thesauri to *expand* a query to enhance the accuracy.
- A query about “dogs” might well be expanded to include “canine” if a thesauri was consulted.
- Only problem is that you can easily add a “bad” word. A synonym for “dog” might well be “pet” and then the query would be too generic.

## Manual vs. Automatic

- Manual
  - use a readily available machine-readable form of a thesauri (e.g. Roget's, etc.).
- Automatic
  - build a thesaurus automatically in a language independent fashion
  - Notion is that an algorithm that could build a thesaurus automatically could be used on many different languages.

## Thesaurus Generation with Term Co-occurrence

- Thesaurus is generated by finding similar terms.
- terms that *co-occur* with each other over a threshold are considered *similar*.
- Term-Term similarity matrix is created, having SC between every term  $t_i$  with  $t_j$

## Term Co-occurrence (example)

- Term Vectors (term-doc mapping):

$$t_1 \quad \langle 1 \ 1 \rangle$$

$$t_2 \quad \langle 0 \ 1 \rangle$$

$$SC(t_1, t_2) = \langle 1 \ 1 \rangle \cdot \langle 0 \ 1 \rangle = 1 \quad \text{dot product}$$

$$SC(t_1, t_2) = SC(t_2, t_1) \quad \text{symmetric coefficient}$$

## Expanding Query using Term Co-occurrence

- For a given term  $t_i$ , the top  $t$  similar terms are picked.
- These words can now be used for query expansion.

## Problems with Term co-occurrence

- A very frequent term will co-occur with everything
- Very general terms will co-occur with other general terms (*hairy* will co-occur with *furry*)

## Semantic Networks

- Attempt to resolve the *mismatch* problem
- Instead of matching query terms and document terms, measures the *semantic distance*
- Premise: Terms that share the same meaning are closer (smaller distance) to each other in semantic network

See publicly available tool, WordNet ([www.cogsci.princeton.edu/~wn](http://www.cogsci.princeton.edu/~wn))

## Semantic Networks

- Builds a network that for each word shows its relationships to other words. (recent efforts, 2004, to incorporate phrases).
- For *dog* and *canine* a *synonym* arc would exist.
- To expand a query, find the word in the semantic network and follow the various arcs to other related words.
- Different *distance measures* can be used to compute the distance from one word in the network to another.

© Goharian, Grossman, Frieder, 2002, 2009

27

## Types of Links in Wordnet

- Synonyms
  - dog, canine
- Antonyms (opposite)
  - night, day
- Hyponyms (is-a)
  - dog, mammal
- Meronyms (part-of)
  - roof, house
- Entailment (one entails the other)
  - buy, pay
- Troponyms (two words related by entailment must occur at the same time)
  - limp, walk

© Goharian, Grossman, Frieder, 2002, 2009

28

## Summary

- Pros
  - Thesauri and Semantic Networks (WordNet) can be used to find good words for users “more like this”
- Cons
  - Little improvement has been found with automatic techniques to expand query without user intervention
  - Manual thesauri and WordNet are language dependent