Information Retrieval
Techniques for Relevance Feedback
Introduction

- An information need may be expressed using different keywords (*synonymy*)
  - impact on recall
  - examples: ship vs boat, aircraft vs airplane

- Solutions: refining queries manually *or* expanding queries (semi) automatically

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

Feedback given by the user about the relevance of the documents in the initial set of results.
About Relevance Feedback (continued)

- Based on the idea that:
  (i) defining good queries is difficult when the collection is (partly) unknown
  (ii) judging particular documents is easy

- Allows to deal with situations where the user’s information needs evolve with the checking of the retrieved documents

- Example: image search engine
  http://nayana.ece.ucsb.edu/imsearch/imsearch.html
Relevance feedback example 1

Instructions:

**Browse:** If the first page displayed doesn't include any interesting images, click browse to see the next page.

**Search:** Once you find some initial images you are interested, click on them to select and press search.

**Iterate:** After the search results are displayed, select/unselect more relevant images and click search.

The system is based on relevance feedback and it learns while you select more images and iterate.
Relevance feedback example 1

Instructions:
- **Browse**: If the first page displayed doesn't include any interesting images, click browse to see the next page.
- **Search**: Once you find some initial images you are interested, click on them to select and press search.
- **Iterate**: After the search results are displayed, select/unselect more relevant images and click search.

The system is based on relevance feedback and it learns while you select more images and iterate.
About Relevance Feedback

Relevance feedback example 1

Instructions:
Browse: If the first page displayed doesn't include any interesting images, click browse to see the next page.
Search: Once you find some initial images you are interested, click on them to select and press search.
Iterate: After the search results are displayed, select/unselect more relevant images and click search.
The system is based on relevance feedback and it learns while you select more images and iterate.

Initialize a new search: bike

Displayed: 1 to 24 of 24  Index: 24

Browse  Search  Prev  Next  Random
Example 2

*國立歷史博物館/師大/新視提供
The Rocchio algorithm

- Standard algorithm for relevance feedback (SMART, 70s)
- Integrates a measure of relevance feedback into the Vector Space Model
- Idea: we want to find a query vector $\vec{q}_{opt}$
  - maximizing the similarity with relevant documents while
  - minimizing the similarity with non-relevant documents

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

With the cosine similarity, this gives:

$$
\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\vec{d}_j \in C_r} \vec{d}_j - \frac{1}{|C_{nr}|} \sum_{\vec{d}_j \in C_{nr}} \vec{d}_j
$$
The Rocchio algorithm (continued)

- Problem with the above metrics: the set of relevant documents is unknown

- Instead, we produce the modified query $m$:

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

where:

- $q_0$ is the original query vector
- $D_r$ is the set of known relevant documents
- $D_{nr}$ is the set of known non-relevant documents
- $\alpha, \beta, \gamma$ are balancing weights (judge vs system)
Relevance feedback on initial query

- **x** known non-relevant documents
- **o** known relevant documents

Initial query

Revised query
The Rocchio algorithm (continued)

Remarks:

- Negative weights are usually ignored
- Rocchio-based relevance feedback improves both recall and precision
- For reaching high recall, many iterations are needed
- Empirically determined values for the balancing weights:
  \[ \alpha = 1 \quad \beta = 0.75 \quad \gamma = 0.15 \]
- Positive feedback is usually more valuable than negative feedback: \[ \beta > \gamma \]
Rocchio algorithm: exercise

Consider the following collection (one doc per line):

good movie trailer shown
trailer with good actor
unseen movie

a dictionary made of the words movie, trailer and good, and
an IR system using the standard \( tf - idf \) weighting (without normalisation).

Assuming a user judges the first 2 documents relevant for the query movie trailer. What would be the Rocchio-revised query?
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 (wrong) / Britney Spears (correct)
  - Cross-language information retrieval.
  - Mismatch of searcher’s vocabulary vs. collection vocabulary
    - hotel / inn / tavern
Violation of A2

- There are several relevance prototypes.
- Examples:
  - Burma/Myanmar
  - Pop stars that worked at Burger King
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
Evaluation of Relevance Feedback strategies

- Note that improvements brought by the relevance feedback decrease with the number of iterations, usually one round gives good results.

- Several evaluation strategies:
  
  (a) **comparative evaluation**
  
  query \( q_0 \rightarrow \text{prec/recall graph} \)
  
  query \( q_m \rightarrow \text{prec/recall graph} \)

  \( \rightarrow \) usually +50% of mean average precision
  
  (partly comes from the fact that known relevant documents are higher ranked)
Evaluation of Relevance Feedback strategies (continued)

- Evaluation strategies:
  
  (b) residual collection

  → same technique as above but by looking at the set of retrieved documents - the set of assessed relevant documents

  → the performance measure drops

  (c) using two similar collections

  collection #1 is used for querying and giving relevance feedback
  collection #2 is used for comparative evaluation

  → $q_0$ and $q_m$ are compared on collection #2
Evaluation of Relevance Feedback strategies (continued)

- Evaluation strategies:
  - (d) user studies
    - e.g. time-based comparison of retrieval, user satisfaction, etc.
    - user utility is a fair evaluation as it corresponds to real system usage
Other local methods for query expansion

Pseudo Relevance Feedback

- Aka *blind relevance feedback*

- No need of an extended interaction between the user and the system

- Method:
  - normal retrieval to find an initial set of most relevant documents
  - *assumption* that the top $k$ documents are relevant
  - relevance feedback defined accordingly

- Works with the TREC Ad Hoc task
  - Inc.ltc (precision at $k = 50$): no-RF 62.5 %, RF 72.7 %

- Problem: distribution of the documents may influence the results
Indirect Relevance Feedback

- Uses evidences rather than explicit feedback
- Example: number of clicks on a given retrieved document
- Not user-specific
- More suitable for web IR, since it does not need an extra action from the user
Vocabulary tools for query reformulation

Tools displaying:

- a list of close terms belonging to the dictionary
- information about the query words that were omitted (cf stop-list)
- the results of stemming

→ ≈ debugging environnement
Query logs and thesaurus

- Users select among query suggestions that are built either from query logs or thesaurus

- Replacement words are extracted from thesaurus according to their proximity to the initial query word

- Thesaurus can be developed:
  - manually (e.g. biomedicine)
  - automatically (cf below)

- NB: query expansion
  - (i) increases recall
  - (ii) may need users’ relevance on query terms (≠ documents)
Automatic thesaurus generation

Analyze of the collection for building the thesaurus automatically:

1. Using word co-occurrences (co-occurring words are more likely to belong to the same query field) → may contain false positives (example: apple)

2. Using a shallow grammatical analyzes to find out relations between words example: cooked, eaten, digested → food

Note that co-occurrence-based thesaurus are more robust, but grammatical-analyzes thesaurus are more accurate
Co-occurrence Thesaurus

- Simplest way to compute one is based on term-term similarities in \( C = AA^T \) where \( A \) is term-document matrix.

- \( w_{i,j} = \) (normalized) weight for \((t_i, d_j)\)

- For each \( t_i \), pick terms with high values in \( C \)
### Automatic Thesaurus Generation Example

<table>
<thead>
<tr>
<th>word</th>
<th>ten nearest neighbors</th>
</tr>
</thead>
<tbody>
<tr>
<td>absolutely</td>
<td>absurd whatsoever totally exactly nothing</td>
</tr>
<tr>
<td>bottomed</td>
<td>dip copper drops topped slide trimmed slight</td>
</tr>
<tr>
<td>captivating</td>
<td>shimmer stunningly superbly plucky witty</td>
</tr>
<tr>
<td>doghouse</td>
<td>dog porch crawling beside downstairs gazed</td>
</tr>
<tr>
<td>Makeup</td>
<td>repellent lotion glossy sunscreen Skin gel pores</td>
</tr>
<tr>
<td>mediating</td>
<td>reconciliation negotiate cease conciliation p</td>
</tr>
<tr>
<td>keeping</td>
<td>hoping bring wiping could some would other</td>
</tr>
<tr>
<td>lithographs</td>
<td>drawings Picasso Dali sculptures Gauguin</td>
</tr>
<tr>
<td>pathogens</td>
<td>toxins bacteria organisms bacterial parasite</td>
</tr>
<tr>
<td>senses</td>
<td>grasp psyche truly clumsy naive innate awake</td>
</tr>
</tbody>
</table>
Conclusion

➤ Query expansion using either local methods:
   • Rocchio algorithm for Relevance Feedback
   • Pseudo Relevance Feedback
   • Indirect Relevance Feedback

➤ or global ones:
   • Query logs
   • Thesaurus

➤ Thesaurus-based query expansion increases recall but may decrease precision (cf ambiguous terms)

➤ High cost of thesaurus development and maintenance

➤ Thesaurus-based query expansion is less efficient than Rocchio Relevance Feedback but may be as good as Pseudo Relevance Feedback
References

- C. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval

- Chris Buckley and Gerard Salton and James Allan
  The effect of adding relevance information in a relevance feedback environment (1994)
  http://citeseer.ist.psu.edu/buckley94effect.html

- Ian Ruthven and Mounia Lalmas
  A survey on the use of relevance feedback for information access systems (2003)