Relevance Feedback and Query Reformulation

Lecture 10
CS 510
Information Retrieval on the Internet
Thanks to Susan Price

Outline

• Query reformulation
• Sources of relevance for feedback
• Using relevance feedback
  – Vector model
• Clustering approaches
  – Local analysis
  – Global analysis
• Examples of relevance feedback and query expansion
• Inputs to document ranking other than similarity
The basic problem

Queries are not always successful

- Users not aware of document content
- Don’t know what terms to use

Who made this vase?

My strategy:
- Try to find images like it
- See if there pages mention the maker

My problem:
- What’s it called?
Possible solutions

- Retrieve new documents (boost recall)
  - Add new disjuncts to the query
- Avoid irrelevant documents (boost prec.)
  - Add new conjuncts to the query

Possible solutions

- Re-order the documents
  - Re-weight terms
Approaches

Query reformulation
– Query expansion
– Query refinement

Relevance feedback
Combine query with other sources of information

How do we get additional info?

From known (or presumed) relevant (and irrelevant) documents

Assumption:
– Relevant documents are more similar to each other than they are to non-relevant documents

From thesaurus or term-relatedness knowledge
– Can be hand built
– Can be derived from documents
How to get relevant documents?

1. Ask the user
   - Systems returns documents for initial query
   - User marks which ones are relevant
   - System uses as input to new query
   - Assumption: user will recognize documents that are useful, or close to what he wants (and is willing to mark them)

2. Assume top-ranked documents are relevant; automatically reformulate query

Relevance feedback: user
Blind feedback

(1) Query

(2) Term re-weighting based on top-ranked documents

(3) Results

Using relevance feedback

• Vector model: use terms and term weights in relevant documents to reformulate the query
  – Add new terms (query expansion)
  – Re-weight existing terms

• Basic approach:
  \[ \text{Vec}_{\text{newQ}} = \text{Vec}_{\text{oldQ}} + \text{Vec}_{\text{posExamples}} - \text{Vec}_{\text{negExamples}} \]
Relevance feedback: Vector model

Three classic query reformulations: \((\alpha, \beta, \gamma\) originally set to 1)

\(D_r\) is the set of relevant docs; \(D_n\) is the set of non-relevant docs

1. **Rocchio**

\[ \vec{q}_m = \alpha \vec{q} + \frac{\beta}{|D_r|} \sum_{\forall d_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \sum_{\forall d_j \in D_n} \vec{d}_j \]

Divide positive & negative vectors by size of example sets

2. **Ide**

\[ \vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall d_j \in D_r} \vec{d}_j - \gamma \sum_{\forall d_j \in D_n} \vec{d}_j \]

3. **Ide Dec-Hi**

\[ \vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall d_j \in D_r} \vec{d}_j - \gamma \max_{\text{non-relevant}}(\vec{d}_j) \]

Use only top-ranked non-relevant document

\[ \text{Vec}_{\text{newQ}} = \text{Vec}_{\text{oldQ}} + \text{Vec}_{\text{posExamples}} - \text{Vec}_{\text{negExamples}} \]

Using relevance feedback

- Probabilistic model: use relevant documents to re-calculate probabilities of relevance
  - Re-weight terms
  - No query expansion
Clustering approaches

• Cluster hypothesis: “closely associated documents tend to be relevant to the same requests”\(^1\)

• Instead of using weights of individual terms in relevant documents, use relevant documents to describe a cluster of desirable documents


Clustering for query reformulation

• Local analysis
  – Operate only on documents returned for the current query – the local set
  – Use term co-occurrences to select terms for query expansion

• Global analysis
  – Use information from all the documents in the collection
Local Clustering

- Use term co-occurrence data to calculate correlation between terms $u, v$: $C_{u,v}$
- Add the $n$ query terms with the highest $C_{u,v}$ where $u$ is a term in the original query
- Reflects term clustering

Global analysis

- Similarity thesaurus
- Statistical thesaurus
Creating a similarity thesaurus

• Index a term based on documents it appears in
  – each term associated with a vector
    \[ \overrightarrow{k_i} = (w_{i,1}, w_{i,2}, \ldots, w_{i,N}) \]
  – where \( w_{i,j} \) is the weight of document \( j \) associated with term \( i \)
  – \( w_{i,j} \) is calculated similarly to \( tf-idf \) except it uses inverse term frequency (not inverse document freq.)
  – length of vector is number of documents in collection
• Calculate a matrix of term correlations
  \[ c_{u,v} = \overrightarrow{k_u} \cdot \overrightarrow{k_v} = \sum_{d \in D} w_{u,d} \times w_{v,d} \]

Using a similarity thesaurus

• Compare candidate terms to the query “concept” or centroid
  – Calculate virtual query vector that combines vectors for each query term, weighted by \( w_{i,q} \)
• Calculate similarity of each term in collection to the query concept
• Add top \( r \) terms to original query
  – Weighted by similarity to original query
Relevance feedback on the Web

Query expansion on the Web

GoogleSuggest FAQs:

“That's pretty cool. How does it do that? Our algorithms use a wide range of information to predict the queries users are most likely to want to see. For example, Google Suggest uses data about the overall popularity of various searches to help rank the refinements it offers.”
What if we search using the medical term, *varicella*?

Textpresso

- Text mining application for scientific literature
- Can search terms or categories of terms
  - Biological concepts (gene, cell, nucleic acid)
  - Relationships (association, regulation)
  - Biological processes
- Can search for “gene” or do query expansion to search for any gene name
Specify a biologic concept
Expand from category to instances

View matches
Document ranking

- Relevance feedback or query expansion can be used for ranking (if it gives a similarity score) instead of re-searching
- Using considerations besides similarity
  - Filters
  - Ranking adjustments
Possible filters

- Language (English, Spanish, etc)
- Source
- Other metadata
- Novelty
  - Suppress documents already seen, duplicates, near-duplicates from same site

Possible ranking inputs

- Inputs other than similarity
  - Importance (page rank)
  - Popularity
  - Reading level
  - Currency
  - Quality
    - What is quality? What are the criteria? Who applies the criteria?