Query Operation

Reformulate queries automatically to improve queries.
- a query only approximates an information need
- users often start with short queries
- users can often improve queries after seeing relevant and irrelevant documents
  - by adding and removing terms
  - by reweighting terms
  - by adding structures (or, and, not …)
Query Expansion

- Relevance Feedback
  reweight query terms based on the feedback information from the user

- Term Correlation
  - Local analysis
    term correlation is created based on a subset of relevant and irrelevant documents
  - Global analysis
    term correlation is created based on the entire collection of documents
Relevance Feedback

- User is presented with a small list of retrieved documents and marks relevant documents

- Reward the terms frequently occurred in relevant documents and punish the terms frequently occurred in non-relevant documents

- Move the query from non-relevant documents to relevant documents
Relevance Feedback in Vector Model

Before

After
Optimized Query

\[ \tilde{q}_{opt} = \frac{1}{|R|} \sum_{d_j \in R} \tilde{d}_j - \frac{1}{N-|R|} \sum_{d_j \in \bar{R}} \tilde{d}_j \]

R is the set of relevant documents
N is the number of documents in the collection
Query Expansion

Original Rocchio:

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

$D_r$ is the set of retrieved relevant documents

$D_n$ is the set of retrieved non-relevant documents

$\alpha$ and $\beta$ are constants representing the relative importance of the positive feedback and negative feedback.
Relevance Feedback Example

Original Query: $(5\ 0\ 3\ 0\ 1)$
Relevant document $d_1 = (2\ 1\ 2\ 0\ 0)$
Non-relevant document $d_2 = (1\ 0\ 0\ 0\ 2)$
$\alpha = 0.5$, $\beta = 0.25$

$q_{lm} = (5\ 0\ 3\ 0\ 1) + 0.5\ (2\ 1\ 2\ 0\ 0) - 0.25\ (1\ 0\ 0\ 0\ 2)$
$= (5.75\ 0.5\ 4.0\ 0.0\ 0.5)$
Relevance Feedback Variations

- Standard Rocchio
  \[ \tilde{q}_m = \alpha \tilde{q} + \frac{\beta}{|D_r|} \sum_{d_j \in R} \tilde{d}_j - \frac{\lambda}{|D_n|} \sum_{d_j \in \tilde{R}} \tilde{d}_j \]

- Ide Regular (Don’t normalize):
  \[ \tilde{q}_m = \alpha \tilde{q} + \beta \sum_{d_j \in R} \tilde{d}_j - \gamma \sum_{d_j \in \tilde{R}} \tilde{d}_j \]

- Ide_Dec_Hi (highest ranked non-relevant document):
  \[ \tilde{q}_m = \alpha \tilde{q} + \beta \sum_{d_j \in R} \tilde{d}_j - \gamma \text{Highest Rank}(D_n) \]
Term Selection

- Original query terms only
- All terms
- Most common terms
- Most highly weighted terms
Relevance Feedback Variation

- Ide_Dec_Hi is effective for a few judged documents
- Rocchio ($\alpha = 0.75, \beta = 0.25$) is effective for many judged documents
- Using all terms is the best
- Using most common terms works well too.
- Using highly weighted terms is inferior (questionable)
Machine Learning Approach

Learning classification rules:
For two or more classes of objects:
  Given: a set of training examples from each class
  Result: rules that correctly classify all objects

IR is a classification problem:
  two classes: relevant and non-relevant
  learning rules to classify relevant and non-relevant
Learn to identify relevant documents

- Represent a query as:
  \[ w_1 t_2 + w_2 t_2 + \ldots + w_n t_n \]
- The goal is to learn all weights \( w_i \) (\( i = 1, \ldots, n \)) from given relevant and non-relevant documents to best separate relevant documents from non-relevant documents.
- \( w_1 t_2 + w_2 t_2 + \ldots + w_n t_n \) is a linear discriminator.
- Queries may be represented in other formats such as decision trees and neural networks.
Learning Algorithms

- Least Squared Error
- Perceptron (Rocchio)
- Neural Network
- Decision Tree
Relevance Feedback in Probabilistic Model

Relevance feedback is used to compute the conditional probabilities: $P(k_i|R)$ and $P(k_i|NR)$.

$$P(k_i|R) = \frac{|D_{r,i}| + \frac{n_i}{N}}{|D_r| + 1}$$

$$P(k_i|NR) = \frac{n_i - |D_{r,i}| + \frac{n_i}{N}}{N - |D_r| + 1}$$
Clustering

• Clustering is to divide objects into groups.

• Clustering is based on the similarity between two objects.

• Objects in the same cluster should be similar to each other and objects in different clusters should be dissimilar.

• Clustering in IR: document clustering and term clustering
Term Clustering

• Terms with similar meanings are grouped together to form thesaurus classes

• Many term clustering algorithms are based term co-occurrence in documents

• Term clusters are used for query expansion

• Term clusters are used for indexing documents to reducing dimensionality.
Local Feedback Analysis

• Analysis is conducted on all documents retrieved for the initial query. No user involvement is need.

• Query is expanded with terms correlated with query terms.

• Correlated terms of a query term are generated as a term cluster.
Association Clusters

- F is a t (number of terms) by N (number of documents) association (term frequency) matrix, $f_{ij}$ is the term frequency of term $t_i$ in document $d_j$.
- The term correlation matrix $C = F F^t$
  \[ c_{ij} = f_{i1} * f_{j1} + f_{i2} * f_{j2} + \ldots + f_{iN} * f_{jN} \]
  $c_{ij}$ is the correlation between $t_i$ and $t_j$.
- Normalized term correlation
  \[ s_{ij} = c_{ij} / (c_{ii} + c_{jj} - c_{ij}) \]
Association Clusters

- Local association cluster of term $t_i$:
  the $n$ terms with largest $c_{ij}$

- For each query term $q_i$, find its local association cluster.

- Expand the query with the terms in the association cluster.
Other Methods to Derive Term Correlation

• Metric Cluster
term correlation is derived based on the distance between the two terms in a document (the number of terms between the two terms). The closer the two terms, the more correlated they are.

• Scalar Cluster
Two terms with similar correlated terms are correlated.
Local Context Analysis

• Retrieved top $n$ ranked passages
  – 300 fix word length passages
  – avoid the problem with long documents on multiple topics
  – more efficient

• Rank concepts (noun groups) instead of individual terms based on their similarity to the query

• Measure the similarity between the concept and the entire query (not individual terms)

• Top $m$ ranked concepts are added to the original query.
Similarity between Concept and Query

- the more correlated the query terms are with concept terms, the more similar the query is to the concept

- the fewer passages contain the concept terms, the more similar the query is to the concept

- the fewer passages contain the query terms, the more similar the query is to the concept
Global Analysis

- Use the entire collection of documents
- Use concepts (noun groups) created during the document indexing phase
- Rank concepts based on their similarity to the query
- Select the top ranked concepts as expansion concepts
Similarity Measure

- Term (concept) correlation matrix

- Query terms are weighted

- Terms highly correlated with important query terms have high similarity to the query

- Terms are selected based their similarity to the all terms of the query (not individual query terms).