Automatic Global Analysis

Cap. 5: Query Operations
Section 5.4: Automatic Global Analysis

27 de Outubro de 1999
Motivation

- Methods of local analysis extract information from local set of documents retrieved to expand the query.
- An alternative is to expand the query using information from the whole set of documents.
- Until the beginning of the 1990s, this technique failed to yield consistent improvements in retrieval performance.
- Now, with moderns variants, sometimes based in thesaurus, this perception has changed.
Automatic Local Analysis

- There are two modern variants based on a thesaurus-like structure built using all documents in collection
  - Query Expansion based on a Similarity Thesaurus
  - Query Expansion based on a Statistical Thesaurus
The similarity thesaurus is based on term to term relationships rather than on a matrix of co-occurrence.

This relationship are not derived directly from co-occurrence of terms inside documents.

They are obtained by considering that the terms are concepts in a concept space.

In this concept space, each term is indexed by the documents in which it appears.

Terms assume the original role of documents while documents are interpreted as indexing elements.
The following definitions establish the proper framework:

- $t$: number of terms in the collection
- $N$: number of documents in the collection
- $f_{i,j}$: frequency of occurrence of the term $k_i$ in the document $d_j$
- $t_j$: vocabulary of document $d_j$
- $itf_j$: inverse term frequency for document $d_j$
Similarity Thesaurus

- Inverse term frequency for document $d_j$
  $$itf_j = \log \frac{t}{t_j}$$

- To $k_i$ is associated a vector
  $$k_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,N})$$
Similarity Thesaurus

where \( w_{i,j} \) is a weight associated to index-document pair \([k_i,d_j]\). These weights are computed as follows

\[
\begin{align*}
\text{Similarity} &= \sum_{l=1}^{N} (0.5 + 0.5 \times \frac{f_{i,j}}{\max_j(f_{i,j})}) \times \text{itf}_j \\
&= \frac{(0.5 + 0.5 \times \frac{f_{i,j}}{\max_j(f_{i,j})}) \times \text{itf}_j}{\sqrt{\sum_{l=1}^{N} (0.5 + 0.5 \times \frac{f_{i,l}}{\max_l(f_{i,l})})^2 \text{itf}_{ji}^2}}
\end{align*}
\]
Similarity Thesaurus

- The relationship between two terms $k_u$ and $k_v$ is computed as a correlation factor $c_{u,v}$ given by

$$c_{u,v} = \textbf{k}_u \cdot \textbf{k}_v = \sum_{\forall d_j} w_{u,j} \times w_{v,j}$$

- The global similarity thesaurus is built through the computation of correlation factor $C_{u,v}$ for each pair of indexing terms $[k_u,k_v]$ in the collection.
Similarity Thesaurus

- This computation is expensive
- Global similarity thesaurus has to be computed only once and can be updated incrementally
Query Expansion based on a Similarity Thesaurus

Query expansion is done in three steps as follows:

1. Represent the query in the concept space used for representation of the index terms.
2. Based on the global similarity thesaurus, compute a similarity \( \text{sim}(q, kv) \) between each term \( kv \) correlated to the query terms and the whole query \( q \).
3. Expand the query with the top \( r \) ranked terms according to \( \text{sim}(q, kv) \).
Query Expansion - step one

- To the query $q$ is associated a vector $\vec{q}$ in the term-concept space given by

$$\vec{q} = \sum_{k_i \in q} w_{i,q} \vec{k}_i$$

- where $w_{i,q}$ is a weight associated to the index-query pair $[k_i, q]$
Query Expansion - step two

- Compute a similarity $\text{sim}(q, kv)$ between each term $kv$ and the user query $q$

  $$\text{sim}(q, k_v) = \vec{q} \cdot \vec{k_v} = \sum_{k_u \in q} w_{u,q} \times c_{u,v}$$

- where $c_{u,v}$ is the correlation factor
Query Expansion - step three

- Add the top $r$ ranked terms according to $\text{sim}(q, kv)$ to the original query $q$ to form the expanded query $q'$
- To each expansion term $kv$ in the query $q'$ is assigned a weight $w_{v,q'}$ given by

$$w_{v,q'} = \frac{\text{sim}(q, k_v)}{\sum_{k_u \in q} w_{u,q}}$$

- The expanded query $q'$ is then used to retrieve new documents to the user
Query Expansion Sample

- Doc1 = D, D, A, B, C, A, B, C
- Doc2 = E, C, E, A, A, D
- Doc3 = D, C, B, B, D, A, B, C, A
- Doc4 = A

- $c(A,A) = 10.991$
- $c(A,C) = 10.781$
- $c(A,D) = 10.781$
- ...
- $c(D,E) = 10.398$
- $c(B,E) = 10.396$
- $c(E,E) = 10.224$
Query Expansion Sample

- Query: \( q = A \ E \ E \)
  - \( \text{sim}(q,A) = 24.298 \)
  - \( \text{sim}(q,C) = 23.833 \)
  - \( \text{sim}(q,D) = 23.833 \)
  - \( \text{sim}(q,B) = 23.830 \)
  - \( \text{sim}(q,E) = 23.435 \)

- New query: \( q' = A \ C \ D \ E \ E \)
  - \( \text{w}(A,q')= 6.88 \)
  - \( \text{w}(C,q')= 6.75 \)
  - \( \text{w}(D,q')= 6.75 \)
  - \( \text{w}(E,q')= 6.64 \)
Global thesaurus is composed of classes which group correlated terms in the context of the whole collection.

Such correlated terms can then be used to expand the original user query.

This terms must be low frequency terms.

However, it is difficult to cluster low frequency terms.

To circumvent this problem, we cluster documents into classes instead and use the low frequency terms in these documents to define our thesaurus classes.

This algorithm must produce small and tight clusters.
Complete link algorithm

- This is document clustering algorithm with produces small and tight clusters
  - Place each document in a distinct cluster.
  - Compute the similarity between all pairs of clusters.
  - Determine the pair of clusters \([Cu,Cv]\) with the highest inter-cluster similarity.
  - Merge the clusters \(Cu\) and \(Cv\)
  - Verify a stop criterion. If this criterion is not met then go back to step 2.
  - Return a hierarchy of clusters.

- Similarity between two clusters is defined as the minimum of similarities between all pair of inter-cluster documents
Selecting the terms that compose each class

- Given the document cluster hierarchy for the whole collection, the terms that compose each class of the global thesaurus are selected as follows:
  - Obtain from the user three parameters:
    - TC: Threshold class
    - NDC: Number of documents in class
    - MIDF: Minimum inverse document frequency
Selecting the terms that compose each class

- Use the parameter TC as threshold value for determining the document clusters that will be used to generate thesaurus classes.

- This threshold has to be surpassed by $\text{sim}(C_u, C_v)$ if the documents in the clusters $C_u$ and $C_v$ are to be selected as sources of terms for a thesaurus class.
Selecting the terms that compose each class

- Use the parameter NDC as a limit on the size of clusters (number of documents) to be considered.

- A low value of NDC might restrict the selection to the smaller cluster Cu+v
Selecting the terms that compose each class

- Consider the set of document in each document cluster pre-selected above.
- Only the lower frequency documents are used as sources of terms for the thesaurus classes.
- The parameter MIDF defines the minimum value of inverse document frequency for any term which is selected to participate in a thesaurus class.
Query Expansion based on a Statistical Thesaurus

- Use the thesaurus class to query expansion.
- Compute an average term weight $w_{tc}$ for each thesaurus class $C$

$$w_{tc} = \frac{\sum_{i=1}^{\left|C\right|} w_{i,C}}{\left|C\right|}$$
Query Expansion based on a Statistical Thesaurus

- $w_{tc}$ can be used to compute a thesaurus class weight $w_{c}$ as

$$W_{c} = \frac{w_{tc}}{|C|} \times 0.5$$
Query Expansion Sample

Doc1 = D, D, A, B, C, A, B, C
Doc2 = E, C, E, A, A, D
Doc3 = D, C, B, B, D, A, B, C, A
Doc4 = A

sim(1,3) = 0.99
sim(1,2) = 0.40
sim(2,3) = 0.29
sim(4,1) = 0.00
sim(4,2) = 0.00
sim(4,3) = 0.00

didf A = 0.0
didf B = 0.3
didf C = 0.12
didf D = 0.12
didf E = 0.60

TC = 0.90 NDC = 2.00 MIDF = 0.2

q' = A B E

q = A E E
Query Expansion based on a Statistical Thesaurus

- Problems with this approach
  - initialization of parameters TC, NDC and MIDF
  - TC depends on the collection
  - Inspection of the cluster hierarchy is almost always necessary for assisting with the setting of TC.
  - A high value of TC might yield classes with too few terms
Conclusion

- Thesaurus is an efficient method to expand queries.
- The computation is expensive but it is executed only once.
- Query expansion based on similarity thesaurus may use high term frequency to expand the query.
- Query expansion based on statistical thesaurus need well defined parameters.