Query Processing
Relevance feedback; query expansion;
Web Search
Overview

Indexing → Indexes → Ranking → Results → Application

Information analysis → Query processing

Crawler → Multimedia documents

Query → Documents → User
Query assist

trump
trump news
trump cabinet
trump tower
trump executive orders
trump memes
trump impeachment
trump russia
trump trumpet
trump latest
Query assist

How can we revise the user query to improve search results?
How do we augment the user query?

- Local analysis (relevance feedback)
  - Based on the query-related documents (initial search results)

- Global analysis (statistical query expansion)
  - Automatically derived thesaurus from the full collection
  - Refinements based on query log mining

- Manual expansion (thesaurus query expansion)
  - Linguistic thesaurus: e.g. MedLine: physician, syn: doc, doctor, MD, medico
  - Can be query rather than just synonyms
Relevance feedback

• Given the initial search results, the user marks some documents as important or non-important.
  • This information is used for a second search iteration where these examples are used to refine the results

• The characteristics of the positive examples are used to boost documents with similar characteristics

• The characteristics of the negative examples are used to penalize documents with similar characteristics
Example: UX perspective

Results for initial query

User feedback

Results after Relevance Feedback
Example: geometric perspective

Results for Initial Query

User feedback

Results after Relevance Feedback
Key concept: Centroid

• The **centroid** is the center of mass of a set of points
  - Recall that we represent documents as points in a high-dimensional space

• The centroid of a set of documents $C$ is defined as:

$$
\bar{\mu}(C) = \frac{1}{|C|} \sum_{d \in C} \vec{d}
$$
Rocchio algorithm

• The Rocchio algorithm uses the vector space model to pick a relevance fed-back query
  • Rocchio seeks the query \( q_{opt} \) that maximizes

\[
\tilde{q}_{opt} = \arg \max_{\tilde{q}} \left[ \cos(\tilde{q}, \tilde{\mu}(C_r)) - \cos(\tilde{q}, \tilde{\mu}(C_{nr})) \right]
\]

• Tries to separate documents marked as relevant and non-relevant

\[
\tilde{q}_{opt} = \frac{1}{|C_r|} \sum_{d_j \in C_r} \tilde{d}_j - \frac{1}{|C_{nr}|} \sum_{d_j \not\in C_r} \tilde{d}_j
\]

• Problem: we don’t know the truly relevant docs
The theoretically best query

Optimal query

x non-relevant documents
○ relevant documents

Sec 9.1.1
Relevance feedback on initial query

Initial query

Revised query

x known non-relevant documents
o known relevant documents
Rocchio 1971 Algorithm (SMART)

• Used in practice:

\[
\tilde{q}_m = \alpha \tilde{q}_0 + \beta \frac{1}{|D_r|} \sum_{d_j \in D_r} \tilde{d}_j - \gamma \frac{1}{|D_{nr}|} \sum_{d_j \in D_{nr}} \tilde{d}_j
\]

• \(D_r\) = set of known relevant doc vectors
• \(D_{nr}\) = set of known irrelevant doc vectors
  • Different from \(C_r\) and \(C_{nr}\)
• \(q_m\) = modified query vector; \(q_0\) = original query vector; \(\alpha, \beta, \gamma\): weights (hand-chosen or set empirically)

• The new query moves toward relevant documents and away from irrelevant documents
Subtleties to note

- Tradeoff $\alpha$ vs. $\beta/\gamma$: If we have a lot of judged documents, we want a higher $\beta/\gamma$.

- Some weights in query vector can go negative
  - Negative term weights are ignored (set to 0)
Google A/B testing of relevance feedback
Relevance feedback: Why is it not used?

• Users are often reluctant to provide explicit feedback

• Implicit feedback and user session monitoring is a better solution

• RF works best when relevant documents form a cluster

• In general negative feedback doesn’t hold a significant improvement
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).
  • Cross-language information retrieval (hígado).
  • Mismatch of searcher’s vocabulary vs. collection vocabulary
    • Cosmonaut/astronaut
Violation of A2

• There are several relevance prototypes.

• Examples:
  • Burma/Myanmar
  • Contradictory government policies
  • Pop stars that worked at Burger King

• Often: instances of a general concept

• Good editorial content can address problem
  • Report on contradictory government policies
Evaluation: Caveat

• True evaluation of usefulness must compare to other methods taking the same amount of time.

• There is no clear evidence that relevance feedback is the “best use” of the user’s time

Users may prefer revision/resubmission to having to judge relevance of documents.
Pseudo-relevance feedback

• Given the initial query search results...
  • a few examples are taken from the top of this rank and a new query is formulated with these positive examples.

• It is important to choose the right number of documents and the terms to expand the query
Pseudo-relevant feedback

- The most frequent terms of all top documents are considered the pseudo-relevant terms:

$$topDocTerms = \sum_{i=1}^{#topDocs} d_{retDocId(q_0,i)}$$

$$prfterms_i = \begin{cases} 
topDocTerms_i & \text{topDocTerms}_i < th \\
0 & \text{topDocTerms}_i < th 
\end{cases}$$

, s.t. $\|prfterms\|_0 = #topterms$

- The expanded queries then become: $q = \gamma \cdot q_0 + (1 - \gamma) \cdot prfterms$

- Other strategies can be thought to automatically select “possibly” relevant documents
## Experimental comparison

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P@10</td>
<td>MAP</td>
<td>P@10</td>
<td>MAP</td>
<td>P@10</td>
</tr>
<tr>
<td>Cosine TF-IDF</td>
<td>0.264</td>
<td>0.126</td>
<td>0.252</td>
<td>0.135</td>
<td>0.120</td>
</tr>
<tr>
<td>Proximity</td>
<td>0.396</td>
<td>0.124</td>
<td>0.370</td>
<td>0.146</td>
<td>0.425</td>
</tr>
<tr>
<td>No length norm. (rawTF)</td>
<td>0.266</td>
<td>0.106</td>
<td>0.240</td>
<td>0.120</td>
<td>0.298</td>
</tr>
<tr>
<td>D: rawTF+ no IDF Q: IDF</td>
<td>0.342</td>
<td>0.132</td>
<td>0.328</td>
<td>0.154</td>
<td>0.400</td>
</tr>
<tr>
<td>Binary</td>
<td>0.256</td>
<td>0.141</td>
<td>0.224</td>
<td>0.148</td>
<td>0.069</td>
</tr>
<tr>
<td>2-Poisson</td>
<td>0.402</td>
<td>0.177</td>
<td>0.406</td>
<td>0.207</td>
<td>0.418</td>
</tr>
<tr>
<td>BM25</td>
<td>0.424</td>
<td>0.178</td>
<td>0.440</td>
<td>0.205</td>
<td>0.471</td>
</tr>
<tr>
<td>LMD</td>
<td>0.450</td>
<td>0.193</td>
<td>0.428</td>
<td>0.226</td>
<td>0.484</td>
</tr>
<tr>
<td>BM25F</td>
<td></td>
<td></td>
<td>0.482</td>
<td>0.242</td>
<td>0.544</td>
</tr>
<tr>
<td>BM25+PRF</td>
<td>0.452</td>
<td>0.239</td>
<td>0.454</td>
<td>0.249</td>
<td>0.567</td>
</tr>
<tr>
<td>RRF</td>
<td>0.462</td>
<td>0.215</td>
<td>0.464</td>
<td>0.252</td>
<td>0.543</td>
</tr>
<tr>
<td>LR</td>
<td>0.446</td>
<td>0.266</td>
<td></td>
<td></td>
<td>0.588</td>
</tr>
<tr>
<td>RankSVM</td>
<td>0.420</td>
<td>0.234</td>
<td></td>
<td></td>
<td>0.556</td>
</tr>
</tbody>
</table>
How do we augment the user query?

• Local analysis (relevance feedback)
  • Based on the query-related documents (initial search results)

• Global analysis (statistical query expansion)
  • Automatically derived thesaurus from the full collection
  • Refinements based on query log mining

• Manual expansion (thesaurus query expansion)
  • Linguistic thesaurus: e.g. MedLine: physician, syn: doc, doctor, MD, medico
  • Can be query rather than just synonyms
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 $C$

What does $C$ contain if $A$ is a term-doc incidence (0/1) matrix?
Automatic thesaurus generation

• Attempt to generate a thesaurus automatically by analyzing the collection of documents

• Fundamental notion: similarity between two words
  • Definition 1: Two words are similar if they co-occur with similar words.
  • Definition 2: Two words are similar if they occur in a given grammatical relation with the same words.

• Co-occurrence based is more robust, grammatical relations are more accurate.
Example: Automatic thesaurus generation

<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 sligh</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</td>
</tr>
<tr>
<td>mediating</td>
<td>reconciling negotiate cease conciliation</td>
</tr>
<tr>
<td>keeping</td>
<td>hoping bringing 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 awk</td>
</tr>
</tbody>
</table>

If the initial query has 3 terms, the query that “hits” the index may end-up having 30 terms!!!

Retrieval precision improves, but, how is retrieval efficiency affected by this?
How do we augment the user query?

• Local analysis (relevance feedback)
  • Based on the query-related documents (initial search results)

• Global analysis (statistical query expansion)
  • Automatically derived thesaurus from the full collection
  • Refinements based on query log mining

• Manual expansion (thesaurus query expansion)
  • Linguistic thesaurus: e.g. MedLine: physician, syn: doc, doctor, MD, medico
  • Can be query rather than just synonyms
Linguistic thesaurus-based query expansion

- Find synonyms and other morphological forms
  - WordNet provides natural language based expansions
    - [http://wordnet.princeton.edu/](http://wordnet.princeton.edu/)

```java
public void getSynonyms(IDictionary dict) {
    // look up first sense of the word "dog"
    IIndexWord idxWord = dict.getIndexWord("dog", POS.NOUN);
    IWordID wordID = idxWord.getIndexWord().get(0); // 1st meaning
    IWord word = dict.getWord(wordID);
    ISynset synset = word.getSynset();

    // iterate over words associated with the synset
    for (IWord w : synset.getWords()) {
        System.out.println(w.getLemma());
    }
}
```

Manual thesaurus-based query expansion

• For each term, \( t \), in a query, expand the query with synonyms and related words of \( t \) from the thesaurus
  • feline \( \rightarrow \) feline cat
  • May weight added terms less than original query terms.

• Generally **increases recall**
  • Widely used in many science/engineering fields

• May significantly **decrease precision**, particularly with ambiguous terms.
  • “interest rate” \( \rightarrow \) “interest rate fascinate evaluate”
  • There is a high cost of manually producing a thesaurus
Summary

• PRF improves top precision and QE improves recall but...

• It’s often harder to understand why a particular document was retrieved after applying RF or QE

• 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