Information Storage and Retrieval
IIR 9: Relevance Feedback & Query Expansion

Paul Gray

Department of Computer Science

2017-04-04
Overview

1. Tidbit from Chapter 8
2. Motivation
3. Relevance feedback: Basics
4. Relevance feedback: Details
5. Query expansion
Outline

1. Tidbit from Chapter 8
2. Motivation
3. Relevance feedback: Basics
4. Relevance feedback: Details
5. Query expansion
Precision and recall

- **Precision** ($P$) is the fraction of retrieved documents that are relevant.

\[
\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant} \mid \text{retrieved})
\]

- **Recall** ($R$) is the fraction of relevant documents that are retrieved.

\[
\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved} \mid \text{relevant})
\]
A combined measure: $F$

- $F$ allows us to trade off precision against recall.
- Balanced $F$:
  \[ F_1 = \frac{2PR}{P + R} \]
- This is a kind of soft minimum of precision and recall.
Interactive relevance feedback: improve initial retrieval results by telling the IR system which docs are relevant / nonrelevant

Best known relevance feedback method: Rocchio feedback

Query expansion: improve retrieval results by adding synonyms / related terms to the query

Sources for related terms: Manual thesauri, automatic thesauri, query logs
Overview

1. Tidbit from Chapter 8
2. Motivation
3. Relevance feedback: Basics
4. Relevance feedback: Details
5. Query expansion
Outline

1. Tidbit from Chapter 8
2. Motivation
3. Relevance feedback: Basics
4. Relevance feedback: Details
5. Query expansion
How can we improve recall in search?

- Main topic today: two ways of improving recall: relevance feedback and query expansion
- As an example consider query $q$: [aircraft] . . .
- . . . and document $d$ containing “plane”, but not containing “aircraft”
- A simple IR system will not return $d$ for $q$.
- Even if $d$ is the most relevant document for $q$!
- We want to change this:
  - Return relevant documents even if there is no term match with the (original) query
Recall

- Loose definition of recall in this lecture: “increasing the number of relevant documents returned to user”
- This may actually decrease recall on some measures, e.g., when expanding “jaguar” with “panthera”
  - ...which eliminates some relevant documents, but increases relevant documents returned on top pages
- In case you are not motivated to look it up, *Panthera* is a genus of large, wild cats in the mammalian family, Felidae, and includes the four, well-known living species of the lion.
  - (Yes, I looked that up. The above is from http://www.newworldencyclopedia.org/entry/Panthera)
- So replacing jaguar (a specific word) with panthera (a more general term that includes jaguar but also other related mammal families) would bring in more documents that relate to wild cats, but leave out those that might specifically address jaguars.
CanIHasCheezburger?

...likely bringing into the returned documents all of those cheezeburger/tag/panther memes.
Options for improving recall

- **Local**: Do a “local”, on-demand analysis for a user query
  - Main local method: relevance feedback
  - Part 1

- **Global**: Do a global analysis once (e.g., of collection) to produce thesaurus
  - Use thesaurus for query expansion
  - Part 2
Relevance feedback: Basic idea

- The user issues a (short, simple) query.
- The search engine returns a set of documents.
- User marks some docs as relevant, some as nonrelevant.
- Search engine computes a new representation of the information need. Hope: better than the initial query.
- Search engine runs new query and returns new results.
- New results have (hopefully) better recall.
- We can iterate this: several rounds of relevance feedback.
- We will use the term ad hoc retrieval to refer to regular retrieval without relevance feedback.
Relevance feedback: Examples

We will now look at three different examples of relevance feedback that highlight different aspects of the process.
Relevance Feedback: Example 1

Shopping related 607,000 images are indexed and classified in the database
Only One keyword is allowed!!!

bike

Search

Designed by Baris Sumengen and Shawn Newsam

Powered by JLAMP2000 (Java, Linux, Apache, Mysql, Perl, Windows2000)
Results for initial query
User feedback: Select what is relevant
Results after relevance feedback
Vector space example: query "canine" (1)
Similarity of docs to query “canine”

source: Fernando Díaz
User feedback: Select relevant documents

source: Fernando Díaz
Results after relevance feedback

source: Fernando Díaz
Example 3: A real (non-image) example

Initial query: [new space satellite applications]

Results for initial query: ($r = \text{rank}$)

- $r + 1\ 0.539\ \text{NASA Hasn't Scrapped Imaging Spectrometer}$
- $r + 2\ 0.533\ \text{NASA Scratches Environment Gear From Satellite Plan}$
- $3\ 0.528\ \text{Science Panel Backs NASA Satellite Plan, But Urges Launches of Smaller Probes}$
- $4\ 0.526\ \text{A NASA Satellite Project Accomplishes Incredible Feat: Staying Within Budget}$
- $5\ 0.525\ \text{Scientist Who Exposed Global Warming Proposes Satellites for Climate Research}$
- $6\ 0.524\ \text{Report Provides Support for the Critics Of Using Big Satellites to Study Climate}$
- $7\ 0.516\ \text{Arianespace Receives Satellite Launch Pact From Telesat Canada}$
- $+\ 8\ 0.509\ \text{Telecommunications Tale of Two Companies}$

User then marks relevant documents with “+”.
### Expanded query after relevance feedback

<table>
<thead>
<tr>
<th>Term</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>new</td>
<td>2.074</td>
</tr>
<tr>
<td>satellite</td>
<td>30.816</td>
</tr>
<tr>
<td>nasa</td>
<td>5.991</td>
</tr>
<tr>
<td>launch</td>
<td>4.196</td>
</tr>
<tr>
<td>instrument</td>
<td>3.516</td>
</tr>
<tr>
<td>bundespost</td>
<td>3.004</td>
</tr>
<tr>
<td>rocket</td>
<td>2.790</td>
</tr>
<tr>
<td>broadcast</td>
<td>2.003</td>
</tr>
<tr>
<td>oil</td>
<td>0.836</td>
</tr>
<tr>
<td>space</td>
<td>15.106</td>
</tr>
<tr>
<td>application</td>
<td>5.660</td>
</tr>
<tr>
<td>eos</td>
<td>5.196</td>
</tr>
<tr>
<td>aster</td>
<td>3.972</td>
</tr>
<tr>
<td>arianespace</td>
<td>3.446</td>
</tr>
<tr>
<td>ss</td>
<td>2.806</td>
</tr>
<tr>
<td>scientist</td>
<td>2.053</td>
</tr>
<tr>
<td>earth</td>
<td>1.172</td>
</tr>
<tr>
<td>measure</td>
<td>0.646</td>
</tr>
</tbody>
</table>

Compare to original query: [new space satellite applications]
<table>
<thead>
<tr>
<th>Rank</th>
<th>Old Rank</th>
<th>Relevance Score</th>
<th>Article Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>0.513</td>
<td>NASA Scratches Environment Gear From Satellite Plan</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.500</td>
<td>NASA Hasn’t Scrapped Imaging Spectrometer</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.493</td>
<td>When the Pentagon Launches a Secret Satellite, Space Sleuths Do Some Spy Work of Their Own</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.493</td>
<td>NASA Uses ‘Warm’ Superconductors For Fast Circuit</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>0.492</td>
<td>Telecommunications Tale of Two Companies</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>0.491</td>
<td>Soviets May Adapt Parts of SS-20 Missile For Commercial Use</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>0.490</td>
<td>Gaping Gap: Pentagon Lags in Race To Match the Soviets In Rocket Launchers</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>0.490</td>
<td>Rescue of Satellite By Space Agency To Cost $90 Million</td>
</tr>
</tbody>
</table>
Outline

1. Tidbit from Chapter 8
2. Motivation
3. Relevance feedback: Basics
4. Relevance feedback: Details
5. Query expansion
The centroid is the center of mass of a set of points.

Recall that we represent documents as points in a high-dimensional space.

Thus: we can compute centroids of documents.

Definition:

$$\mu(D) = \frac{1}{|D|} \sum_{d \in D} \vec{v}(d)$$

where $D$ is a set of documents and $\vec{v}(d) = \vec{d}$ is the vector we use to represent document $d$. 
Centroid: Examples
Rocchio algorithm

- The Rocchio algorithm implements relevance feedback in the vector space model.
- Rocchio chooses the query $\vec{q}_{opt}$ that maximizes

$$\vec{q}_{opt} = \arg \max_{\vec{q}} [\text{sim}(\vec{q}, \mu(D_r)) - \text{sim}(\vec{q}, \mu(D_{nr}))]$$

$D_r$: set of relevant docs; $D_{nr}$: set of nonrelevant docs

- Intent: $\vec{q}_{opt}$ is the vector that separates relevant and nonrelevant docs maximally.
- Making some additional assumptions, we can rewrite $\vec{q}_{opt}$ as:

$$\vec{q}_{opt} = \mu(D_r) + [\mu(D_r) - \mu(D_{nr})]$$
Rocchio algorithm

- The optimal query vector is:

\[ \tilde{q}_{opt} = \mu(D_r) + [\mu(D_r) - \mu(D_{nr})] \]

\[ = \frac{1}{|D_r|} \sum_{\tilde{d}_j \in D_r} \tilde{d}_j + \left[ \frac{1}{|D_r|} \sum_{\tilde{d}_j \in D_r} \tilde{d}_j - \frac{1}{|D_{nr}|} \sum_{\tilde{d}_j \in D_{nr}} \tilde{d}_j \right] \]

- We move the centroid of the relevant documents by the difference between the two centroids.
Exercise: Compute Rocchio vector

circles: relevant documents, Xs: nonrelevant documents
compute: $\tilde{q}_{opt} = \mu(D_r) + [\mu(D_r) - \mu(D_{nr})]$
Rocchio illustrated

circles: relevant documents, Xs: nonrelevant documents
Rocchio illustrated

$\vec{q}_{opt}$: centroid of relevant documents

$\vec{\mu}_R$: centroid of relevant documents

$\vec{\mu}_{NR}$: centroid of non-relevant documents
Rocchio illustrated

$\vec{\mu}_R$ does not separate relevant/nonrelevant.
Rocchio illustrated

\( \vec{\mu}_{NR} \): centroid of nonrelevant documents
Rocchio illustrated

\[ \vec{\mu}_{R} \]
\[ \vec{\mu}_{NR} \]
\[ \vec{\mu}_{opt} \]

\[ \vec{q}_{opt} \]
Rocchio illustrated

\[ \vec{q}_{opt} \]

\[ \vec{\mu}_R - \vec{\mu}_{NR} : \text{difference vector} \]
Rocchio illustrated

Add difference vector to $\bar{\mu}_R$ ...
Rocchio illustrated

...to get $\tilde{q}_{opt}$
Rocchio illustrated

\[ \vec{q}_{opt} \] separates relevant/nonrelevant perfectly.
Rocchio illustrated

\[ \tilde{q}_{opt} \] separates relevant/nonrelevant perfectly.
So far, we have used the name Rocchio for the theoretically better motivated original version of Rocchio.

The implementation that is actually used in most cases is the SMART implementation – this SMART version of Rocchio is what we will refer to from now on.
Rocchio 1971 algorithm (SMART)

- Used in practice:

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

- \(q_m\): modified query vector; \(q_0\): original query vector; \(D_r\) and \(D_{nr}\): sets of known relevant and nonrelevant documents respectively; \(\alpha\), \(\beta\), and \(\gamma\): weights

- New query moves towards relevant documents and away from nonrelevant documents.

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

- Set negative term weights to 0.

- “Negative weight” for a term doesn’t make sense in the vector space model.
Positive vs. negative relevance feedback

- Positive feedback is more valuable than negative feedback.
- For example, set $\beta = 0.75$, $\gamma = 0.25$ to give higher weight to positive feedback.
- Many systems only allow positive feedback.
When can relevance feedback enhance recall?

Assumption A1: The user knows the terms in the collection well enough for an initial query.

Assumption A2: Relevant documents contain similar terms (so I can “hop” from one relevant document to a different one when giving relevance feedback).
Violation of A1

- Assumption A1: The user knows the terms in the collection well enough for an initial query.
- Violation: Mismatch of searcher’s vocabulary and collection vocabulary
- Example: cosmonaut / astronaut
Violation of A2

- Assumption A2: Relevant documents are similar.
- Example for violation: [contradictory government policies]
- Several unrelated “prototypes”
  - Subsidies for tobacco farmers vs. anti-smoking campaigns
  - Aid for developing countries vs. high tariffs on imports from developing countries
- Relevance feedback on tobacco docs will not help with finding docs on developing countries.
Relevance feedback: Assumptions

- When can relevance feedback enhance recall?

- Assumption A1: The user knows the terms in the collection well enough for an initial query.

- Assumption A2: Relevant documents contain similar terms (so I can “hop” from one relevant document to a different one when giving relevance feedback).
Relevance feedback: Evaluation

- Pick an evaluation measure, e.g., precision in top 10: $P@10$
- Compute $P@10$ for original query $q_0$
- Compute $P@10$ for modified relevance feedback query $q_1$
- In most cases: $q_1$ is spectacularly better than $q_0$!
- Is this a fair evaluation?
Relevance feedback: Evaluation

- Fair evaluation must be on “residual” collection: docs not yet judged by user.
- Studies have shown that relevance feedback is successful when evaluated this way.
- Empirically, one round of relevance feedback is often very useful. Two rounds are marginally useful.
Evaluation: Caveat

- True evaluation of usefulness must compare to other methods taking the same amount of time.
- Alternative to relevance feedback: User revises and resubmits query.
- Users may prefer revision/resubmission to having to judge relevance of documents.
- There is no clear evidence that relevance feedback is the “best use” of the user’s time.
Exercise

- Do search engines use relevance feedback?
- Why?
Relevance feedback: Problems

- Relevance feedback is expensive.
  - Relevance feedback creates long modified queries.
  - Long queries are expensive to process.
- Users are reluctant to provide explicit feedback.
- It’s often hard to understand why a particular document was retrieved after applying relevance feedback.
- The search engine Excite had full relevance feedback at one point, but abandoned it later.
Pseudo-relevance feedback

- Pseudo-relevance feedback automates the “manual” part of true relevance feedback.

- Pseudo-relevance feedback algorithm:
  - Retrieve a ranked list of hits for the user’s query
  - **Assume that the top \( k \) documents are relevant.**
  - Do relevance feedback (e.g., Rocchio)

- Works very well on average

- But can go horribly wrong for some queries.
  - Because of **query drift**
  - If you do several iterations of pseudo-relevance feedback, then you will get query drift for a large proportion of queries.
Pseudo-relevance feedback at TREC4

- Cornell SMART system
- Results show number of relevant documents out of top 100 for 50 queries (so total number of documents is 5000):

<table>
<thead>
<tr>
<th>method</th>
<th>number of relevant documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>lnc.ltc</td>
<td>3210</td>
</tr>
<tr>
<td>lnc.ltc-PsRF</td>
<td>3634</td>
</tr>
<tr>
<td>Lnu.ltu</td>
<td>3709</td>
</tr>
<tr>
<td>Lnu.ltu-PsRF</td>
<td>4350</td>
</tr>
</tbody>
</table>

Results contrast two length normalization schemes (L vs. l) and pseudo-relevance feedback (PsRF).
- The pseudo-relevance feedback method used added only 20 terms to the query. (Rocchio will add many more.)
- This demonstrates that pseudo-relevance feedback is effective on average.
Outline

1. Tidbit from Chapter 8
2. Motivation
3. Relevance feedback: Basics
4. Relevance feedback: Details
5. Query expansion
Query expansion: Example

Yahoo! Search

Web | Images | Video | Audio | Directory | Local | News | Shopping | More »

Search for "palm"

Search Results

Also try: palm springs, palm pilot, palm trees, palm reading

- **Official Palm Store**
  - Store.palm.com
  - Free shipping on all handhelds and more at the official Palm store.

- **Palms Hotel - Best Rate Guarantee**
  - www.vegas.com
  - Book the Palms Hotel Casino with our best rate guarantee at VEGAS.com, the official Vegas travel site.

- **Palm Pilots - Palm Downloads**
  - Yahoo! Shortcut - About

1. **Palm, Inc.**
   - Maker of handheld PDA devices that allow mobile users to manage schedules, contacts, and other personal and business information.
   - Category: B2B > Personal Digital Assistants (PDAs)
   - www.palm.com - 20k - Cached - More from this site - Save

SPONSOR RESULTS

- **Palm Memory**
  - Memory Giant is fast and easy.
  - Guaranteed compatible memory.
  - Great...
  - www.memorygiant.com

- **The Palms, Turks and Caicos Islands**
  - Resort/Condo photos, rates, availability and reservations....
  - www.worldwidereservationsystems.com

- **The Palms Casino Resort, Las Vegas**
  - Low price guarantee at the Palms Casino resort in Las Vegas. Book...
  - lasvegas.hotelscorp.com
Types of user feedback

- User gives feedback on **documents**.
  - More common in relevance feedback
- User gives feedback on **words or phrases**.
  - More common in query expansion
Query expansion

- Query expansion is another method for **increasing recall**.
- We use “global query expansion” to refer to “global methods for query reformulation”.
- In global query expansion, the query is modified based on some global resource, i.e. a resource that is not query-dependent.
- Main information we use: (near-)synonymy
“Global” resources used for query expansion

- A publication or database that collects (near-)synonyms is called a **thesaurus**.
- Manual thesaurus (maintained by editors, e.g., PubMed)
- Automatically derived thesaurus (e.g., based on co-occurrence statistics)
- Query-equivalence based on query log mining (common on the web as in the “palm” example)
Thesaurus-based query expansion

- For each term $t$ in the query, expand the query with words the thesaurus lists as semantically related with $t$.
- Example from earlier: HOSPITAL $\rightarrow$ MEDICAL
- Generally increases recall
- May significantly decrease precision, particularly with ambiguous terms
  - INTEREST RATE $\rightarrow$ INTEREST RATE FASCINATE
- Widely used in specialized search engines for science and engineering
- It’s very expensive to create a manual thesaurus and to maintain it over time.
Example for manual thesaurus: PubMed

PubMed Query:

"neoplasms"[MeSH Terms] OR cancer[Text Word]
Automatic thesaurus generation

- Attempt to generate a thesaurus automatically by analyzing the distribution of words in documents
- Fundamental notion: similarity between two words
- Definition 1: Two words are similar if they co-occur with similar words.
  - “car” \(\approx\) “motorcycle” because both occur with “road”, “gas” and “license”, so they must be similar.
- Definition 2: Two words are similar if they occur in a given grammatical relation with the same words.
  - You can harvest, peel, eat, prepare, etc. apples and pears, so apples and pears must be similar.
- Co-occurrence is more robust, grammatical relations are more accurate.
## Co-occurrence-based thesaurus: Examples

<table>
<thead>
<tr>
<th>Word</th>
<th>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</td>
</tr>
<tr>
<td>captivating</td>
<td>shimmer stunningly superbly plucky witty</td>
</tr>
<tr>
<td>doghouse</td>
<td>dog porch crawling beside downstairs</td>
</tr>
<tr>
<td>makeup</td>
<td>repellent lotion glossy sunscreen skin gel</td>
</tr>
<tr>
<td>mediating</td>
<td>reconciliation negotiate case conciliation</td>
</tr>
<tr>
<td>keeping</td>
<td>hoping bring wiping could some would</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</td>
</tr>
</tbody>
</table>

WordSpace demo on web
Query expansion at search engines

- Main source of query expansion at search engines: query logs
- Example 1: After issuing the query [herbs], users frequently search for [herbal remedies].
  - → “herbal remedies” is potential expansion of “herb”.
- Example 2: Users searching for [flower pix] frequently click on the URL photobucket.com/flower. Users searching for [flower clipart] frequently click on the same URL.
  - → “flower clipart” and “flower pix” are potential expansions of each other.