RSLIS at INEX 2011
Social Book Search track

Toine Bogers
Kirstine Wilfred Christensen
Birger Larsen

Royal School of Library & Information Science / DBC
Copenhagen, Denmark
Outline

• Methodology
  - Pre-processing
  - Indexing & topics
• Content-based retrieval
• Social re-ranking
• Submitted runs
• Discussion
Methodology
Pre-processing

- Removed 22 XML fields not likely to contribute to retrieval
  - Example: `<image>`, `<listprice>`, `<binding>`

- Retained 19 content-bearing XML fields
  - `<isbn>`, `<title>`, `<publisher>`, `<editorial>`, `<creator>`, `<series>`, `<award>`, `<character>`, `<place>`, `<blurber>`, `<epigraph>`, `<firstwords>`, `<lastwords>`, `<quotation>`, `<dewey>`, `<subject>`, `<browseNode>`, `<review>`, and `<tag>`
Indexing

- Created six different indexes
  - All fields (all-doc-fields)
    - All 19 content-bearing XML fields
  - Metadata (metadata)
    - Immutably tied to the book, provided by publisher
      - <title>, <publisher>, <editorial>, <creator>, <series>, <award>, <character>, and <place>
Indexing

- Content (content)
  - Fields that contain some part of the book text
  - <blurber>, <epigraph>, <firstwords>, <lastwords>, and <quotation>

- Controlled metadata (controlled-metadata)
  - Subject descriptions curated by library professionals
  - <browseNode>, <dewey>, and <subject>
Indexing

- Tags (tags)
  - User-generated subject descriptions
  - <tag>

- User reviews
  - Book-centric index reviews (all reviews belonging to the same book aggregated into a single representation)
  - Review-centric index reviews-split (each review indexed separately)
Topics

- Four different topic representations
  - Title (title)
  - Group (group)
  - Narrative (narrative)
  - All three topic fields combined (all-topic-fields)
Content-based retrieval
Approach

• Pairwise combinations of all indexes and topic representations
  - 6 indexes × 4 representations = 24 different runs

• Algorithm
  - Language modeling using JM smoothing
  - \( \lambda \) optimized in steps of 0.1 in \([0, 1]\) range
  - Stopword filtering & Krovetz stemming
## Results

<table>
<thead>
<tr>
<th>Document fields</th>
<th>Topic fields</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>title</td>
</tr>
<tr>
<td>metadata</td>
<td>0.2756</td>
</tr>
<tr>
<td>content</td>
<td>0.0083</td>
</tr>
<tr>
<td>controlled-metadata</td>
<td>0.0663</td>
</tr>
<tr>
<td>tags</td>
<td>0.2848</td>
</tr>
<tr>
<td>reviews</td>
<td>0.3020</td>
</tr>
<tr>
<td>all-doc-fields</td>
<td>0.2644</td>
</tr>
</tbody>
</table>

The inclusion of user-generated metadata in the Amazon/LibraryThing collection gives the track participants the opportunity to examine the effectiveness of using social features to re-rank or improve the initial content-based search results. One such source of social data are the tags assigned by LibraryThing users to the books in the collection. The results in the previous section showed that even when treating these as a simple content-based representation of the collection using our tags index, we can achieve relatively good performance. In this section, we turn our attention to the book reviews entered by Amazon's large user base. We mentioned in Section 5 that we indexed the user reviews from the `<review>` fields in two different ways: all user reviews belonging to a single book were combined in a single document representation for that book and each book review was indexed and retrieved separately. The results of the content-based runs in the previous section showed that a book-centric approach to indexing reviews provided good performance. Review-centric retrieval, however, not all user reviews are equal. Some reviewers provide more accurate/more in-depth reviews than others, and in some cases, reviews may even be misleading or deceptive. This problem of spam reviews on online shopping websites such as Amazon.com is well-documented. This suggests that indexing and retrieving reviews individually and then aggregating the individually retrieved reviews could be beneficial by matching the best, most topical reviews against our topics. Our review-centric retrieval approach works as follows: First, we index all reviews separately in our `reviews-split` index. We then retrieve the top r reviews for each topic. This can result in several reviews covering the same book occurring in our result list, which then need to be aggregated into a single relevance score for each separate book. This problem is similar to the problem of results fusion in IR, where the results of different retrieval algorithms on the same collection are combined. This suggests the applicability of standard methods for results fusion as...
Social re-ranking
Approach

• Tags
  - Tag index *tags* performed well

• Reviews
  - Book-centric index *reviews* performed well
  - What about the review-centric index *reviews*-split?
Approach

• **Review-centric** retrieval

  1. Retrieve *individual* reviews

  2. **Aggregate scores** for individual reviews into a single relevance score for each occurring book

     ‣ Similar to results fusion in IR!

     ‣ Can use methods like CombMAX, CombSUM, etc.
Approach

- **Unweighted** review fusion
  - CombMAX, CombSUM, and CombMNZ

- **Weighted** review fusion
  - Weighting based on **review helpfulness**
    \[
    score_{\text{weighted}}(i) = score_{\text{org}}(i) \times \frac{\text{helpful vote count}}{\text{total vote count}}
    \]
  - Weighting based on normalized **book ratings**
    \[
    score_{\text{weighted}}(i) = score_{\text{org}}(i) \times \frac{r}{5}
    \]
## Results

<table>
<thead>
<tr>
<th>Runs</th>
<th>Topic fields</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>title</td>
</tr>
<tr>
<td>CombMAX</td>
<td>0.3117</td>
</tr>
<tr>
<td>CombSUM</td>
<td>0.3377</td>
</tr>
<tr>
<td>CombMNZ</td>
<td>0.3350</td>
</tr>
<tr>
<td>CombMAX - Helpfulness</td>
<td>0.2603</td>
</tr>
<tr>
<td>CombSUM - Helpfulness</td>
<td>0.2993</td>
</tr>
<tr>
<td>CombMNZ - Helpfulness</td>
<td>0.3083</td>
</tr>
<tr>
<td>CombMAX - Ratings</td>
<td>0.2882</td>
</tr>
<tr>
<td>CombSUM - Ratings</td>
<td>0.3199</td>
</tr>
<tr>
<td>CombMNZ - Ratings</td>
<td>0.3230</td>
</tr>
<tr>
<td>reviews</td>
<td>0.3020</td>
</tr>
</tbody>
</table>
Submitted runs
Submitted runs

- Four submitted runs
  - Run 1: title.all-doc-fields
  - Run 2: all-topic-fields.all-doc-fields
  - Run 3: title.reviews-split.CombSUM
  - Run 4: all-topic-fields.reviews-split.CombSUM
Results

- Best-performing runs
  - Run 2: all-topic-fields.all-doc-fields
  - Run 4: all-topic-fields.reviews-split.CombSUM

- Means there is hope for the social re-ranking approach...
Discussion
What did we learn?

• Best performance when combining all available information
  – Support for principle of polyrepresentation
    ‣ Ingwersen (1996) and Belkin (1993)

• User-generated metadata » curated metadata

• Book-centric vs. review-centric undecided
  – Helpfulness and ratings do not contribute enough in the current approach
Questions?