Information Retrieval (Part IV)

[DAT640] Information Retrieval and Text Mining

Krisztian Balog
University of Stavanger

September 24, 2019

Outline

- ~~Search engine architecture, indexing~~
- ~~Evaluation~~
- **Retrieval models** $\Leftarrow$ today
- Query modeling
- Learning-to-rank, Neural IR
- Semantic search

Recap

- Common form of a retrieval function

$$score(d, q) = \sum_{t \in q} w_{t,d} \times w_{t,q}$$

BM25

- Retrieval model is based on the idea of query-document similarity. Three main components:
  - Term frequency
  - Inverse document frequency
  - Document length normalization
- Retrieval function

$$score(d,q) = \sum_{t \in q} \frac{f_{t,d} \times (1+k_1)}{f_{t,d} + k_1(1 - b + b\frac{|d|}{avgdl})} \times idf_t$$

  - Parameters
    - k_1: calibrating term frequency scaling ($k_1 \in [1.2..2]$)
    - b: document length normalization ($b \in [0,1]$)

Language models

- Retrieval model is based on the probability of observing the query given that document
- Log query likelihood scoring

$$score(d, q) = \log P(q|d) = \sum_{t \in q} \log P(t|\theta_d) \times f_{t,q}$$

- Jelinek-Mercer smoothing

$$score(d, q) = \sum_{t \in q} \log \left((1 - \lambda) \frac{f_{t,d}}{|d|} + \lambda P(t|C) \right) \times f_{t,q}$$

- Dirichlet smoothing

$$score(d, q) = \sum_{t \in q} \log \frac{f_{t,d} + \mu P(t|C)}{|d| + \mu} \times f_{t,q}$$

Discussion

Question

How to compute these retrieval functions for all document in the collection?

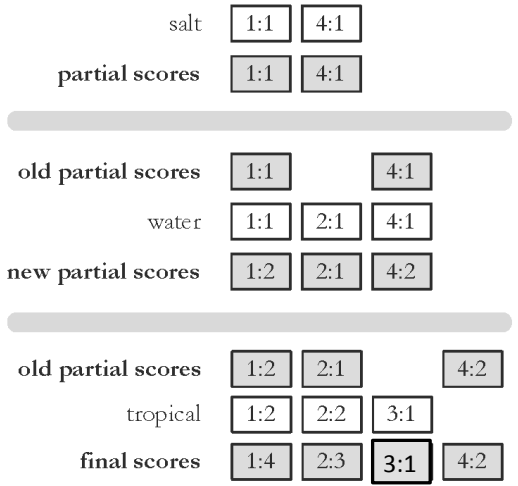
Query processing

- Strategies for processing the data in the index for producing query results
  - We benefit from the inverted index by scoring only documents that contain at least one query term
- Term-at-a-time
  - Accumulates scores for documents by processing term lists one at a time
- Document-at-a-time
  - Calculates complete scores for documents by processing all term lists, one document at a time
- Both approaches have optimization techniques that significantly reduce time required to generate scores

Term-at-a-time query processing

$scores = \{\}$ // score accumulator maps doc IDs to scores
for $w \in q$ **do**
  for $d, count \in Idx.fetch\_docs(w)$ **do**
    $scores[d] = scores[d] + score\_term(count)$
  end for
end for
return top k documents from $scores$

Term-at-a-time query processing



Exercise #1

- Implement term-at-a-time scoring
- Code skeleton on GitHub: `exercises/lecture_10/exercise_1.ipynb` (make a local copy)

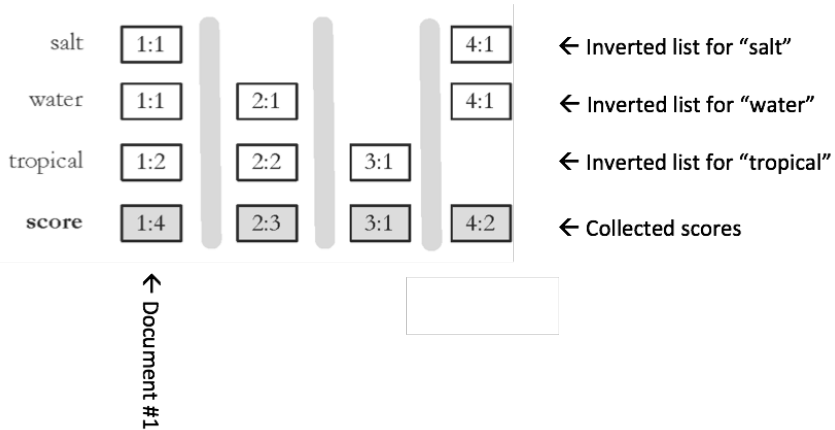
From term-at-a-time to document-at-a-time query processing

- Term-at-a-time query processing
  - Advantage: simple, easy to implement
  - Disadvantage: the score accumulator will be the size of document matching at least one query term
- Document-at-a-time query processing
  - Make the score accumulator data structure smaller by scoring entire documents at once. We are typically interested only in top-k results
  - Idea #1: hold the top-k best completely scored documents in a priority queue
  - Idea #2: Documents are sorted by document ID in the posting list. If documents are scored ordered by their IDs, then it is enough to iterate through each query term's posting list only once
    - Keep a pointer for each query term. If the posting equals the document currently being scored, then get the term count and move the pointer; otherwise the current document does not contain the query term

Document-at-a-time query processing

```
context = {}        // maps a document to a list of matching terms
for w ∈ q do
    for d, count ∈ Idx.fetch_docs(w) do
        context[d].append(count)
    end for
end for
priority_queue = {}        // low score is treated as high priority
for d, term_counts ∈ context do
    score = 0
    for count ∈ term_counts do
        score = score + score_term(count)
    end for
    priority_queue.push(d, score)
    if priority_queue.size() > k then
        priority_queue.pop()        // removes lowest score so far
    end if
end for
Return sorted documents from priority_queue
```

Document-at-a-time query processing



Exercise #2

- Implement document-at-a-time scoring
- Code skeleton on GitHub: `exercises/lecture_10/exercise_2.ipynb` (make a local copy)

Discussion

Question

What other statistics are needed to compute these retrieval functions (in addition to term frequencies)?

BM25

- Total number of documents in the collection (for IDF computation) (int)
- Document length for each document (dictionary)
- Average document length in the collection (int)
- (optionally pre-computed) IDF score for each term (dictionary)

Language models

- Document length for each document (dictionary)
- Sum TF for each term (dictionary)
- Sum of all document lengths in the collection (int)
- (optionally pre-computed) Collection term probability $P(t|C)$ for each term (dictionary)

Fielded (variants of) Retrieval Models

Motivation

- Documents are composed of multiple fields
  - E.g., title, body, anchors, etc.
- Modeling internal document structure may be beneficial for retrieval

Example


promise
PROMISE
Participative Research labOratory for Multimedia and Multilingual Information Systems Evaluation
Log-in
Overview
Achievements
Use cases
Publications
Events
CLEF
Media Center
Contacts
Search
Events
Winter School 2013
PROMISE Winter School 2013
Bressanone, Italy
Winter School 2013
Programme
Lecturers
Venue
Registration and Accomodation
Sponsor and Patronage
Flyer
Important Dates
Registration Deadline (extended): 28th
PROMISE Winter School 2013
Bridging between Information Retrieval and Databases
Bressanone, Italy 4 - 8 February 2013
The aim of the PROMISE Winter School 2013 on "Bridging between Information Retrieval and Databases" is to give participants a grounding in the core topics that constitute the multidisciplinary area of information access and retrieval to unstructured, semistructured, and structured information. The school is a week-long event consisting of guest lectures from invited speakers who are recognized experts in the field. The school is intended for PhD students, Masters students or senior researchers such as post-doctoral researchers form the fields of databases, information retrieval, and related fields.

Unstructured representation

PROMISE Winter School 2013
Bridging between Information Retrieval and Databases
Bressanone, Italy 4 - 8 February 2013
The aim of the PROMISE Winter School 2013 on "Bridging between Information Retrieval and Databases" is to give participants a grounding in the core topics that constitute the multidisciplinary area of information access and retrieval to unstructured, semistructured, and structured information. The school is a week-long event consisting of guest lectures from invited speakers who are recognized experts in the field. The school is intended for PhD students, Masters students or senior researchers such as post-doctoral researchers form the fields of databases, information retrieval, and related fields. [...]

Example

```
<html>
<head>
   <title>Winter School 2013</title>
   <meta name="keywords" content="PROMISE, school, PhD, IR, DB, [...]" />
   <meta name="description" content="PROMISE Winter School 2013, [...]" />
</head>
<body>
   <h1>PROMISE Winter School 2013</h1>
   <h2>Bridging between Information Retrieval and Databases</h2>
   <h3>Bressanone, Italy 4 - 8 February 2013</h3>
   <p>The aim of the PROMISE Winter School 2013 on "Bridging between
   Information Retrieval and Databases" is to give participants a grounding
   in the core topics that constitute the multidisciplinary area of
   information access and retrieval to unstructured, semistructured, and
   structured information. The school is a week-long event consisting of
   guest lectures from invited speakers who are recognized experts in the
   field. The school is intended for  PhD students, Masters students or
   senior researchers such as post-doctoral researchers form the fields of
   databases, information retrieval, and related fields. </p>
   [...]
</body>
</html>
```

Fielded representation (based on HTML markup)

title	Winter School 2013
meta	PROMISE, school, PhD, IR, DB, [...] PROMISE Winter School 2013, [...]
headings	PROMISE Winter School 2013 Bridging between Information Retrieval and Databases Bressanone, Italy 4-8 February 2013
body	The aim of the PROMISE Winter School 2013 on "Bridging between Information Retrieval and Databases" is to give participants a grounding in the core topics that constitute the multidisciplinary area of information access and retrieval to unstructured,semistructured, and structured information. The school is a week-long event consisting of guest lectures from invited speakers who are recognized experts in the field. The school is intended for PhD students, Masters students or senior researchers such as postdoctoral researchers form the fields of databases, information retrieval, and related fields.

Fielded extension of retrieval models

- BM25 $\Rightarrow$ BM25F
- Language Models (LM) $\Rightarrow$ Mixture of Language Models (MLM)

BM25F

- Extension of BM25 incorporating multiple fields
- The soft normalization and term frequencies need to be adjusted
- Original BM25 retrieval function:

$$score(d, q) = \sum_{t \in q} \frac{f_{t,d} \times (1 + k_1)}{f_{t,d} + k_1 \times B} \times idf_t$$

- where B is is the soft normalization:

$$B = (1 - b + b\frac{|d|}{avgdl})$$

BM25F

- Replace term frequencies $f_{t,d}$ with *pseudo term frequencies* $\tilde{f}_{t,d}$
- BM25F retrieval function:

$$score(d,q) = \sum_{t \in q} \frac{\tilde{f}_{t,d}}{k_1 + \tilde{f}_{t,d}} \times idf_t$$

- Pseudo term frequency calculation

$$\tilde{f}_{t,d} = \sum_i w_i \times \frac{f_{t,d_i}}{B_i}$$

- where
  - i corresponds to the field index
  - w_i is the field weight (such that $\sum_i w_i = 1$)
  - B_i is soft normalization for field i, where b_i becomes a field-specific parameter

$$B_i = (1 - b_i + b_i \frac{|d_i|}{avgdl_i})$$

Mixture of Language Models (MLM)

- Idea: Build a separate language model for each field, then take a linear combination of them

$$P(t|\theta_d) = \sum_i w_i P(t|\theta_{d_i})$$

- where
  - i corresponds to the field index
  - w_i is the field weight (such that $\sum_i w_i = 1$)
  - $P(t|\theta_{d_i})$ is the field language model

Field language model

- Smoothing goes analogously to document language models, but term statistics are restricted to the given field i
- Using Jelinek-Mercer smoothing:

$$P(t|\theta_{d_i}) = (1 - \lambda_i)P(t|d_i) + \lambda_i P(t|C_i)$$

- where both the empirical field model ($P(t|d_i)$) and the collection field model ($P(t|C_i)$) are maximum likelihood estimates:

$$P(t|d_i) = \frac{f_{t,d_i}}{|d_i|} \qquad P(t|d_i) = \frac{\sum_{d'} f_{t,d'_i}}{\sum_{d'} |d'_i|}$$

Exercise #3

- Document retrieval using fielded language models (paper-based)

Setting parameter values

- Retrieval models often contain parameters that must be tuned to get the best performance for specific types of data and queries
- For experiments
  - Use training and test data sets
  - If less data available, use cross-validation by partitioning the data into k subsets
- Many techniques exist to find optimal parameter values given training data
  - Standard problem in machine learning
- For standard retrieval models, involving few parameters, *grid search* is feasible
  - Perform a sweep over the possible values of each parameter, e.g., from 0 to 1 in steps of 0.1

Reading

- Text Data Management and Analysis (Zhai&Massung)
  - Chapter 6
  - Section 8.3