Fall 2016
CS646: Information Retrieval

Lecture 2 - Introduction to Search Result Ranking

Jiepu Jiang
University of Massachusetts Amherst
2016/09/12
More course information

Programming Prerequisites
• Proficiency in Java is highly recommended

The Difference between 646 and 597R(546)
• 646 satisfies an AI core requirement
• 646 focuses on research-oriented topics

Subsequent Important Dates
• 9/14 (Wednesday, this week): HW1 out
• 9/19 (Monday, next week): add/drop with no record deadline
About the IR Community

Related Community
• ACM special interest group in information retrieval (ACM SIGIR)
• Mailing list: SIG-IRList (just Google it)

Related Conferences & Journals
• Annual SIGIR conference, and sub-conferences (ICTIR & CHIIR)
• Related, but not IR-focused: CIKM, WSDM, WWW, ACL, EMNLP, …
• Regional: ECIR, AIRS, DIR, ADCS, …
• Evaluation workshops: TREC, CLEF, NTCIR, FIRE, …
• Journals: IR journal, ACM TOIS, JASIST, IP&M, J Doc, TKDE, …

Achievement Award
• Gerard Salton Award, Karen Spärck Jones award, Tony Kent Strix award
Previously on CS646 …

Information retrieval (IR) system
• Input: a user query
• Output: a ranked list of documents

A few notions and techniques
• Types of information needs
• Bag of words representation
  • Inverted Index
• Ranking
• Evaluation
A simple flow of the retrieval process

From James Allan’s slides on CS646
The user’s side

Information Need

Representation

Query

User’s Mind

I plan to take CS646 information retrieval. What’s that?

Note that a user may issue many different queries for the same information need.
The system’s side

Text Objects

Representation

Indexed Objects

Bag-of-words representation of documents

Doc#1
- information x 1
- computer x 2
- data x 1
- science x 1

Doc#2
- retrieval x 1
- query x 2
- index x 1
- data x 1

Doc#3
- retrieval x 1
- information x 1
- data x 2
- search x 1

Inverted Index (Logical Structure)

<table>
<thead>
<tr>
<th>Terms</th>
<th>Term Occurrences in Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>(Doc#1, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#2, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#3, 2)</td>
</tr>
<tr>
<td>information</td>
<td>(Doc#1, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#3, 1)</td>
</tr>
<tr>
<td>query</td>
<td>(Doc#2, 2)</td>
</tr>
<tr>
<td>retrieval</td>
<td>(Doc#2, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#3, 1)</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>
Matching query terms & indexed objects

Query: Information Retrieval

Indexed Objects

Retrieved Objects

**Terms** | **Term Occurrences in Documents**
--- | ---
data | (Doc#1, 1) | (Doc#2, 1) | (Doc#3, 1)
information | (Doc#1, 1) | (Doc#3, 1)
query | (Doc#2, 2)
retrieval | (Doc#2, 1) | (Doc#3, 1)
....
Boolean Search vs. Best Match Search

Boolean Search

- e.g., information **AND** retrieval, information **OR** retrieval
- Returns *a set* of results
- Results are similar to those by the following SQL:

```
SELECT title, author, content, year
FROM articles
WHERE
    content LIKE ":information:%"
OR
    content LIKE ":retrieval:%"
```

Best Match Search (the most popular form of search)

- Returns *a ranked list* of search
- Sorted by some scores (indicating the inferred relevance of results)
Ranking Search Results

Query: *Information Retrieval*

Information Retrieval

<table>
<thead>
<tr>
<th>Information</th>
<th>retrieval</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Doc#1, 1)</td>
<td>(Doc#2, 1)</td>
</tr>
<tr>
<td>(Doc#3, 1)</td>
<td>(Doc#3, 1)</td>
</tr>
</tbody>
</table>

Merge (Boolean OR)

Doc#1 | Doc#2 | Doc#3

Scoring & Ranking (*best match*)

Doc#3
Score: 4.5

Doc#2
Score: 3

Doc#1
Score: 1.5
Although Google does not report any scores to you, it still computes certain scores for the search results and rank them by the scores.
In fact, it’s an old idea used by publishers and libraries.
The old Tillamook County Library card catalog was recently put to new use as a holder of seed packets. (Photo by LeeAnn Neal)
How to Compute Relevance score?

Many factors may contribute to score\( (q, d) \)

- Occurrence or non-occurrence
- Term frequency
- Document length
- Term importance
- Many others ….

Many models break down score\( (q, d) \) into a linear combination of score\( (w, d) \) – the score of each term \( w \) in the document \( d \).

\[
\text{score}(q, d) = \sum_{w \in q} \text{score}(w, d)
\]
Occurrence or Non-occurrence

\[
\text{score}(w, d) = \begin{cases} 
1 & \text{if } w \text{ appears in } d \\
0 & \text{if } w \text{ does not appear}
\end{cases}
\]

Query: Information Retrieval

<table>
<thead>
<tr>
<th>Doc#1</th>
<th>Doc#2</th>
<th>Doc#3</th>
</tr>
</thead>
<tbody>
<tr>
<td>information x 1</td>
<td>retrieval x 1</td>
<td>retrieval x 1</td>
</tr>
<tr>
<td>computer x 2</td>
<td>query x 2</td>
<td>information x 1</td>
</tr>
<tr>
<td>data x 1</td>
<td>query x 2</td>
<td>data x 2</td>
</tr>
<tr>
<td>science x 1</td>
<td>index x 1</td>
<td>search x 1</td>
</tr>
</tbody>
</table>
(Within-document) Term Frequency (TF)

Matching a frequent term in a document is more important than matching a less frequent one.

“… the more frequently a notion and combination of notions occur, the more importance the author attaches to them as reflecting the essence of his overall idea.”

Term Frequency (TF)

\[ \text{score}(w, d) = \text{count}(w, d) \]

**Query: Information Retrieval**

- Doc#1:
  - information x 2
  - computer x 2
  - data x 1
  - science x 1

- Doc#2:
  - retrieval x 3
  - query x 2
  - index x 1
  - data x 1

- Doc#3:
  - retrieval x 1
  - information x 2
  - data x 2
  - search x 1
Term Frequency (TF): Variants

\[
\text{score}(w, d) = \begin{cases} 
1 + \log \text{count}(w, d) & \text{count}(w, d) > 1 \\
0 & \text{count}(w, d) = 0
\end{cases}
\]

- Avoid assigning a too strong impact on a very frequent term.
- Make sure the difference between 0 (no occurrence) and 1 (appearing once) is greater than that between 1 and 2 and so on

\[
\frac{dy_1}{dx} = 1 \quad \frac{dy_2}{dx} = \frac{1}{x} \leq 1
\]

\[
\lim_{x \to \infty} \frac{dy_2}{dx} = 0
\]

\[y_1 = x\]

\[y_2 = 1 + \log x\]
Document length matters – It’s easy to match a query term in a long document

Many retrieval models normalize TF by document length.

$$score(w,d) = \log \frac{\text{count}(w,d) + 1}{\text{length}(d) + 1}$$

Query: *information retrieval*

Which of the following seem more relevant?

Doc#1: a 5000-word paper

*Information x 10*

*retrieval x 10*

....

OR

Doc#2: a 300-page book

*Information x 10*

*retrieval x 10*

....
Not All Terms are Equally Important

- A general belief is that more specific terms are more important, but we need a measure for specificity.

Query: *information retrieval*

Which of the following two docs seem more relevant?

Doc#1

- information x 1
- computer x 2
- data x 1
- science x 1

Doc#2

- retrieval x 1
- query x 2
- index x 1
- search x 1
Inverse Document Frequency (IDF)

\[ \text{score}(w, d) = IDF = \log \frac{N}{n} \]

\(N\): the total number of documents in the corpus
\(n\): the number of documents containing the term
The log base is not important.

Rare terms are more discriminative and important.

Karen Spärck Jones (1935-2007)
(Gerard Salton Award, 1988)

Inverse Document Frequency (IDF)

\[
score(w, d) = IDF = \log \frac{N}{n}
\]

\(N\): the total number of documents in the corpus
\(n\): the number of documents containing the term

Rare terms are more discriminative and important.

<table>
<thead>
<tr>
<th>Word</th>
<th>N</th>
<th>n</th>
<th>IDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>information</td>
<td>278123</td>
<td>60938</td>
<td>1.52</td>
</tr>
<tr>
<td>retrieval</td>
<td>278123</td>
<td>2034</td>
<td>4.92</td>
</tr>
</tbody>
</table>
Why IDF works?

\[ \hat{P}(w) = \frac{n}{N} \quad \text{IDF} = \log \frac{N}{n} \]

- The ratio \( n/N \) is an estimate on how likely the term is to appear in a document by chance.
- A term with a low IDF is one that is likely to appear in documents just by chance.
- So the occurrence of a term with a low IDF is less important/informative compared with one with a high IDF.

TF-IDF Weighting

\[ \text{score}(q,d) = \sum_{w \in q} \text{tf}(w,d) \cdot \text{idf}(w) \]

- Since both TF and IDF are important factors, we just combine them.

Gerard Salton
(1927-1995)
(Gerard Salton Award, 1983)

TF-IDF Weighting: Variants of TF

\[ \text{score}(q, d) = \sum_{w \in q} \text{tf}(w, d) \cdot \text{idf}(w) \]

<table>
<thead>
<tr>
<th></th>
<th>( \text{tf}(w, d) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>0 (not occur) or 1 (any occurrence).</td>
</tr>
<tr>
<td>Raw TF</td>
<td>Just the raw frequency.</td>
</tr>
<tr>
<td>Log TF</td>
<td>1 + log freq</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>
TF-IDF Weighting: Variants of IDF

\[ \text{score}(q, d) = \sum_{w \in q} tf(w, d) \cdot idf(w) \]

<table>
<thead>
<tr>
<th>idf(w)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not weight</td>
</tr>
<tr>
<td>KSJ</td>
</tr>
<tr>
<td>BM25</td>
</tr>
<tr>
<td>..</td>
</tr>
</tbody>
</table>
Similarity implies relevance

“The more two representations agreed in given elements and their distribution, the higher would be the probability of their representing similar information.”

Ranking results: query-document similarity

Vector Space Model (SMART system)

- Represent both query and document as vectors of terms (words).
- Measure similarity using cosine.


The VSM figure comes from the WBC textbook (page 240).

Vector Space Model

• A document similar to the query has a small angle in the N-dimensional space, and thus a greater $\cos(q, d)$.

• We’ll talk more details in Lecture 6.
These 1970s ideas are still being used today, but are implemented in a more complex way.
Remember they are just ideas, assumptions, or heuristics. They may not be correct and they may not work well for all occasions.
Later in this course, we will introduce

- More on Boolean search & Vector Space Model (Lecture 6)
  - VSM is still quite popular today!
  - The default model of Apache Lucene is a variant of VSM.

- Binary independence model and BM25 (Lecture 7)
  - An implementation of TF-IDF
  - Becomes popular since 1995
  - Still one of the best-performing ad hoc search model.

- Language modeling approach for IR (Lecture 8 and Lecture 9)
  - Comes from the statistical language models in NLP
  - Becomes popular since 1998
  - Representation + retrieval model
  - Still one of the best-performing ad hoc search model.
Other ideas: query term proximity

- A document where the query terms appeared adjacent to each other is more likely relevant than another where the terms are far away from each other.
- Example: phrase match
- Sequential dependence model (Lecture 9)

Query: *information retrieval*

Which of the following two docs seem more relevant?

Doc#1

... *information retrieval* ...

OR

Doc#2

... *information* ... *(1000 words)*... *retrieval* ...
Vocabulary Mismatch

- Searchers and authors of documents use different words to refer to the same notion, causing difficulties in matching.

Query: *search engine*

Document: *… information retrieval …*

Comparison: 

$\text{Score}(q, d) = 0$
Query representation

Improving query representation

- (Automatic) query expansion (Lecture 10)

<table>
<thead>
<tr>
<th>“Monica Lewinsky Case”</th>
<th>“Rats in Space”</th>
</tr>
</thead>
<tbody>
<tr>
<td>(P(w</td>
<td>Q))</td>
</tr>
<tr>
<td>0.041</td>
<td>lewinsky</td>
</tr>
<tr>
<td>0.038</td>
<td>monica</td>
</tr>
<tr>
<td>0.027</td>
<td>jury</td>
</tr>
<tr>
<td>0.026</td>
<td>grand</td>
</tr>
<tr>
<td>0.019</td>
<td>confidant</td>
</tr>
<tr>
<td>0.016</td>
<td>talk</td>
</tr>
<tr>
<td>0.015</td>
<td>case</td>
</tr>
<tr>
<td>0.014</td>
<td>president</td>
</tr>
<tr>
<td>0.013</td>
<td>clinton</td>
</tr>
<tr>
<td>0.010</td>
<td>starr</td>
</tr>
</tbody>
</table>

Document representation

Improving document representation (Lecture 19)
- Latent semantic indexing (LSI) and Probabilistic LSI (PLSI)
- Cluster-based retrieval and topic model (e.g., LDA)

A topic model for "information retrieval"

<table>
<thead>
<tr>
<th>Word</th>
<th>$P(w \mid D)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>0.087</td>
</tr>
<tr>
<td>Retrieval</td>
<td>0.068</td>
</tr>
<tr>
<td>Search</td>
<td>0.043</td>
</tr>
<tr>
<td>Engine</td>
<td>0.032</td>
</tr>
<tr>
<td>index</td>
<td>0.028</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>
Distributed representations of words

Word embeddings (Lecture 20 & 21)

• Representing words as vectors (of a low-dimensional space comparing to the vocabulary size).
• A guest lecture on deep learning for IR (from CIIR grad students)
Not all documents are equally important (Lecture 16)

Link analysis
- PageRank and variants
- HITS and variants

Spam detection
- Determine whether a web page is a spam web page or not
Contextual and Personalized Search (Lecture 17)

Taking into account context

• Who you are
• Where you are
• Short-term interest
• Long-term interest
• Your previous search activities

Example: Google always ranks “UMASS moodle” at the top when you search the query “moodle” in Amherst.
Learning-to-rank (Lecture 11)

Since 2005, a popular method is to combine many different factors into search result ranking and learn the weights of factors automatically by machine learning techniques.

An example:

\[ \text{Score}(q, d) = \\
3.2 \times \text{score}(\text{information}, d) + 2.8 \times \text{score}(\text{retrieval}, d) \\
+ 5.9 \times ("\text{information retrieval}", d) \\
+ 0.8 \times \text{PageRank}(d) + 1.2 \times \text{HITS}(d) - 0.8 \times \text{Spam}(d) \\
+ 3.5 \times \text{similarity}(\text{previous queries}, d) \\
+ 6.5 \times \text{similarity}(\text{previous clicks}, d) \]
Query suggestion (Lecture 18)
Next Lecture

• Some basic ideas of IR evaluation