CS47300: Web Information Search and Management

Prof. Chris Clifton
30 August 2017
Material adapted from course created by Dr. Luo Si, now leading Alibaba research group

Retrieval Models
Overview of Retrieval Models

Retrieval Models
- Boolean
- Vector space
  - Basic vector space
  - Extended Boolean
- Probabilistic models
  - Statistical language models
  - Two Possion model
  - Bayesian inference networks
- Citation/Link analysis models
  - Page rank
  - Hub & authorities

SMART, LUCENE
Lemur Project (Indri, Galago)
Okapi
Inquery
Google
Clever

Retrieval Models: Outline

Retrieval Models
- Exact-match retrieval method
  - Unranked Boolean retrieval method
  - Ranked Boolean retrieval method
- Best-match retrieval method
  - Vector space retrieval method
  - Latent semantic indexing
Retrieval Models: Unranked Boolean

Unranked Boolean: Exact match method

• Selection Model
  – Retrieve a document iff it matches the precise query
  – Often return unranked documents (or with chronological order)

• Operators
  – Logical Operators: AND OR, NOT
  – Proximity operators:
    • #1(white house) (i.e., within one word distance, phrase)
    • #sen(Iraq weapon) (i.e., within a sentence)
  – String matching operators: Wildcard (e.g., ind* for india and indonesia)
  – Field operators: title(information and retrieval)...

A query example
(#2(distributed information retrieval) OR (#1 (federated search))) AND author(#1(Jamie Callan) AND NOT (Steve))
Retrieval Models: Unranked Boolean

WestLaw system: Commercial Legal/Health/Finance Information Retrieval System
- Logical operators
- Proximity operators: Phrase, word proximity, same sentence/paragraph
- String matching operator: wildcard (e.g., ind*)
- Field operator: title(#1("legal retrieval")) date(2000)
- Citations: Cite (Salton)

Advantages:
- Work well if user knows exactly what to retrieve
- Predictable; easy to explain
- Very efficient

Disadvantages:
- Difficult to design a good query
  - Users may be too optimistic
- Results are unordered
Retrieval Models: Unranked Boolean

Disadvantages:

• It is difficult to design the query
  – “Loose” query (information OR retrieval): Low precision
  – “Strict” query (information AND retrieval): Low recall
    • Users may assume most/all relevant documents found

• Results are unordered
  – Low precision queries not very useful

Retrieval Models: Ranked Boolean

Ranked Boolean: Exact match

- Similar to unranked Boolean but documents are ordered by some criterion
- Reflect importance of document by its words

Retrieve docs from Wall Street Journal Collection

Query: (Thailand AND stock AND market)

Which word is more important?
Many “stock” and “market”, but fewer “Thailand”. Fewer may be more indicative

Term Frequency (TF): Number of occurrence in query/doc; larger number means more important

Inversed Document Frequency (IDF):
Larger means more important

There are many variants of TF, IDF: e.g., consider document length
Retrieval Models: Ranked Boolean

- Ranked Boolean: Calculate doc score
- Term evidence: Evidence from term i occurred in doc j: \( (tf(i,j)) \) and \( (tf(i,j) \times idf(i)) \)
- AND weight: minimum of argument weights
- OR weight: maximum of argument weights

Query: (Thailand AND stock AND market)

Retrieval Models: Ranked Boolean

Advantages:
- All advantages from unranked Boolean algorithm
  - Works well when query is precise; predictive; efficient
- Results in a ranked list (not a full list); easier to browse and find the most relevant ones than Boolean
- Rank criterion is flexible: e.g., different variants of term evidence

Disadvantages:
- Still an exact match (document selection) model: inverse correlation for recall and precision of strict and loose queries
- Predictability makes user overestimate retrieval quality
Retrieval Models:
Vector Space Model

• Any text object can be represented by a term vector
  – Documents, queries, passages, sentences
  – A query can be seen as a short document
• Similarity is determined by distance in the vector space
  – Example: cosine of the angle between two vectors

(Research) Famous Examples
• The SMART system
  – Developed at Cornell University: 1960-1999
  – Still quite popular
• The Lucene system
  – Open source information retrieval library; (Based on Java)
  – Works with Hadoop (Map/Reduce) in large scale app (e.g., Amazon Book)

Retrieval Models:
Vector Space Model

Vector space model vs. Boolean model
• Boolean models
  – Query: a Boolean expression that a document must satisfy
  – Retrieval: Deductive inference
• Vector space model
  – Query: viewed as a short document in a vector space
  – Retrieval: Find similar vectors/objects
Retrieval Models:
Vector Space Model

• Vector representation

<table>
<thead>
<tr>
<th></th>
<th>Java</th>
<th>Sun</th>
<th>Starbucks</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Query</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
</tr>
</tbody>
</table>
Retrieval Models: Vector Space Model

Give two vectors of query and document

- query \( \vec{q} = (q_1, q_2, \ldots, q_n) \)
- document \( \vec{d}_j = (d_{j1}, d_{j2}, \ldots, d_{jn}) \)

- calculate the similarity

Cosine similarity: Angle between vectors

\[
sim(\vec{q}, \vec{d}_j) = \cos(\theta(\vec{q}, \vec{d}_j))
\]

\[
\cos(\theta(\vec{q}, \vec{d}_j)) = \frac{\vec{q} \cdot \vec{d}_j}{\|\vec{q}\| \|\vec{d}_j\|} = \frac{q_1d_{j1} + q_2d_{j2} + \ldots + q_n d_{jn}}{\sqrt{q_1^2 + \ldots + q_n^2} \sqrt{d_{j1}^2 + \ldots + d_{jn}^2}}
\]

---

Vector representation

<table>
<thead>
<tr>
<th></th>
<th>Java</th>
<th>Sun</th>
<th>Starbucks</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>D2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Query</td>
<td>1</td>
<td>0.2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Similarity Score</th>
<th>D1</th>
<th>D2</th>
<th>D3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>0.59</td>
<td>0.99</td>
<td>0.70</td>
</tr>
</tbody>
</table>
Retrieval Models: Vector Space Model

• Vector Coefficients
• The coefficients (vector elements) represent term evidence/term importance
• Derived from several elements
  – Document term weight: Evidence of the term in the document/query
  – Collection term weight: Importance of term from observation of collection
  – Length normalization: Reduce document length bias
• Naming convention for coefficients:

\[ q_k \cdot d_{j,k} = DCL.DCL \]

First triple represents query term; second for document term

Retrieval Models: Vector Space Model

• Common vector weight components:
• Inc.ltc: widely used term weight
  – “l”: log(tf)+1
  – “n”: no weight/normalization
  – “t”: log(N/df)
  – “c”: cosine normalization

\[
\frac{q_1 d_{j,1} + q_2 d_{j,2} + \cdots + q_m d_{j,m}}{\sqrt{\sum_k [\log(tf_k) + 1][\log(tf_k) + 1][\log(N/df_k)]}} = \sqrt{\sum_k [\log(tf_k) + 1][\log(tf_k) + 1][\log(N/df_k)]} \]
Retrieval Models: Vector Space Model

• Common vector weight components:
  • dnn.dtb: handle varied document lengths
    – “d”: 1+ln(1+ln(tf))
    – “t”: log((N/df)
    – “b”: 1/(0.8+0.2*docleng/avg_doclen)

• Standard vector space
  – Represent query/documents in a vector space
  – Each dimension corresponds to a term in the vocabulary
  – Use a combination of components to represent the term evidence in both query and document
  – Use similarity function to estimate the relationship between query/documents (e.g., cosine similarity)
Retrieval Models: Vector Space Model

Advantages:
- Best match method; it does not need a precise query
- Generated ranked lists; easy to explore the results
- Simplicity: easy to implement
- Effectiveness: often works well
- Flexibility: can utilize different types of term weighting methods
- Used in a wide range of IR tasks: retrieval, classification, summarization, content-based filtering…

Disadvantages:
- Hard to choose the dimension of the vector (“basic concept”); terms may not be the best choice
- Assume independent relationship among terms
- Heuristic for choosing vector operations
  - Choose of term weights
  - Choose of similarity function
- Assume a query and a document can be treated in the same way
Retrieval Models: Vector Space Model

Disadvantages:

- Hard to choose the dimension of the vector (“basic concept”);
  - terms may not be the best choice
- Assumes independent relationship among terms
- Heuristic for choosing vector operations
  - Choose of term weights
  - Choose of similarity function
- Assume a query and a document can be treated in the same way

What is a good vector representation?

- Orthogonal: the dimensions are linearly independent (“no overlapping”)
- No ambiguity (e.g., Java)
- Wide coverage and good granularity
- Good interpretation (e.g., representation of semantic meaning)
- Many possibilities: words, stemmed words, “latent concepts”...