NLU & IR: CLASSICAL IR

Omar Khattab

CS224U: Natural Language Understanding
Spring 2021
Ranked Retrieval

- **Scope:** A large corpus of text documents (e.g., Wikipedia)
- **Input:** A textual query (e.g., a natural-language question)
- **Output:** Top-K Ranking of relevant documents (e.g., top-100)
How do we conduct ranked retrieval?

■ We’ve touched on one way before: the **Term–Document Matrix**

<table>
<thead>
<tr>
<th></th>
<th>d1</th>
<th>d2</th>
<th>d3</th>
<th>d4</th>
<th>d5</th>
<th>d6</th>
<th>d7</th>
<th>d8</th>
<th>d9</th>
<th>d10</th>
</tr>
</thead>
<tbody>
<tr>
<td>against</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>age</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>agent</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ages</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ago</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>agree</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ahead</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ain't</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>air</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>aka</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

■ With good weights, this allows us to answer **single-term** queries!
How do we conduct ranked retrieval?

■ For multi-term queries, classical IR models would tokenize and then treat the tokens independently.

\[
RelevanceScore(query, doc) = \sum_{term \in query} Weight_{doc,term}
\]

■ This reduces a large fraction of classical IR to:
  - How do we best tokenize (and stem) queries and documents
  - **How do we best weight each term-document pair**
Term–Document Weighting: Intuitions

- **Frequency** of occurrence will remain a primary factor
  - If a term $t$ occurs frequently in document $d$, the document is more likely to be relevant for queries including $t$

- **Normalization** will remain a primary component too
  - If that term $t$ is rather rare, then document $d$ is even more likely to be relevant for queries including $t$
  - If that document $d$ is rather short, this also improves its odd

- Amplify the important, the trustworthy, the unusual; deemphasize the mundane and the quirky.
Term–Document Weighting: TF-IDF

- Let $N = |\text{Collection}|$ and $df(\text{term}) = |\{\text{doc} \in \text{Collection} : \text{term} \in \text{doc}\}|$

\[
TF(\text{term}, \text{doc}) = \log(1 + Freq(\text{term}, \text{doc}))
\]

\[
IDF(\text{term}) = \log \frac{N}{df(\text{term})}
\]

\[
TF.IDF(\text{term}, \text{doc}) = TF(\text{term}, \text{doc}) \times IDF(\text{term})
\]

\[
TF.IDF(\text{query}, \text{doc}) = \sum_{\text{term} \in \text{query}} TF.IDF(\text{term}, \text{doc})
\]

TF and IDF both grow sub-linearly with frequency and $1/df$ (in particular, logarithmically).
Term–Document Weighting: BM25

Or “Finding the best match, seriously this time! Attempt #25” :-(

\[
IDF(term) = \log(1 + \frac{N - df(term) + 0.5}{df(term) + 0.5})
\]

\[
TF(term, doc) = \frac{Freq(term, doc) \times (k + 1)}{Freq(term, doc) + k \times (1 - b + b \times \frac{|doc|}{avgdoclen})}
\]

\[
BM25(term) = BM25:TF(term, doc) \times BM25:IDF(term)
\]

\[
BM25(query, doc) = \sum_{term \in query} BM25(term, doc)
\]

k, b are parameters.

Unlike TF-IDF, term frequency in BM25 saturates and penalizes longer documents!

Efficient Retrieval: Inverted Indexing

- Raw Collection: Document $\rightarrow$ Terms

- Term–document matrix: Term $\rightarrow$ Documents
  - But it’s extremely sparse and thus wastes space!

- The inverted index is just a sparse encoding of this matrix
  - Mapping each unique term $t$ in the collection to a posting list
  - The posting list enumerates non-zero $<\text{Freq}, \text{DocID}>$ for $t$
Beyond term matching in classical IR...

- Query and Document expansion

- Term dependence and phrase search

- Learning to Rank with various features:
  - Different document fields (e.g., title, body, anchor text)
  - Link Analysis (e.g., PageRank)

Lots of IR exploration into these! However, BM25 was a very strong baseline on the best you can do “ad-hoc”—until 2019 with BERT-based ranking!
IR Evaluation

- A search system must be **efficient** and **effective**
  - If we had infinite resources, we’d just hire experts to look through all the documents one by one!

- **Efficiency**
  - **Latency** (milliseconds; for one query)
  - Throughput (queries/sec)
  - Space (GBs for the index? TBs?)
  - Hardware required (one CPU core? Many cores? GPUs?)
  - Scaling to various collection sizes, under different loads
IR Effectiveness

- Do our top-k rankings fulfill users’ information needs?
  - Often harder to evaluate than classification/regression!

- If you have lots of users, you can run online experiments…

- But we’re typically interested in reusable test collections
Test Collections

- Document Collection (or “Corpus”)
- Test Queries (or “Topics”)
  - Could also include a train/dev split, if resources allow!
  - Or, in some cases, cross-validation could be used.
- Query–Document Relevance Assessments
  - Is document \( j \) relevant to query \( i \)?
    - Binary judgments: relevant (0) vs. non-relevant (1)
    - Graded judgments: \{-1, 0, 1, 2\} (e.g., junk, irrelevant, relevant, key)

We typically have to make the (significant!) assumption that unjudged documents are irrelevant. Some test collections would only label a few positives per query.
Test Collections: TREC

■ Text REtrieval Conference (TREC) includes numerous annual tracks for comparing IR systems.

■ The 2021 iteration has tracks for Conversational Assistance, Health Misinformation, Fair Ranking, “Deep Learning”.

■ TREC tends to emphasize careful evaluation with a very small set of queries (e.g., 50 queries, each with >100 annotated documents)
  - Having only few test queries does not imply few documents!
Test Collections: MS MARCO Ranking Tasks

- MS MARCO Ranking is the largest public IR benchmark
  - adapted from a Question Answering dataset
  - consists of more than 500k **Bing search queries**
    - Sparse labels: approx. one relevance label per query!
    - Fantastic for training IR models!

- MS MARCO Passage Ranking (9M short passages; sparse labels)
- MS MARCO Document Ranking (3M long documents; sparse labels)
- TREC DL’19 and DL’20 (short&long; dense labels for few queries)
Test Collections: Other Benchmarks

- Lots of small or domain-specific benchmarks!
- BEIR is a recent effort to use those for testing models in “zero-shot” scenarios

We will also see later that OpenQA benchmarks can serve as large IR benchmarks too!

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Passage-Retrieval</td>
<td>Misc.</td>
<td>MSMARCO</td>
<td></td>
<td>Binary</td>
<td>532,761</td>
<td>6,980</td>
<td>8,841,823</td>
<td>1.1</td>
<td>5.96</td>
<td>55.98</td>
</tr>
<tr>
<td></td>
<td>Bio-Medical</td>
<td>Bio-Medical</td>
<td>(1) TREC-COVID</td>
<td>3-level</td>
<td>50</td>
<td>171,332</td>
<td>493.5</td>
<td>10.60</td>
<td>160.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Information</td>
<td>Bio-Medical</td>
<td>(2) NCForum</td>
<td>3-level</td>
<td>323</td>
<td>3,633</td>
<td>38.2</td>
<td>3.30</td>
<td>232.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Retrieval (IR)</td>
<td>Bio-Medical</td>
<td>(3) BioASQ</td>
<td>3-level</td>
<td>500</td>
<td>14,914,602</td>
<td>4.7</td>
<td>8.05</td>
<td>202.61</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Question</td>
<td>Wikipedia</td>
<td>(4) NQ</td>
<td></td>
<td>Binary</td>
<td>132,803</td>
<td>3,452</td>
<td>2,681,468</td>
<td>1.2</td>
<td>9.16</td>
<td>78.88</td>
</tr>
<tr>
<td></td>
<td>Answering</td>
<td>Wikipedia</td>
<td>(5) HotpotQA</td>
<td>3-level</td>
<td>5,447</td>
<td>7,405</td>
<td>5,233,329</td>
<td>2.0</td>
<td>17.61</td>
<td>46.30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(QA)</td>
<td>Finance</td>
<td>(6) FiQA-2018</td>
<td>3-level</td>
<td>648</td>
<td>576,682</td>
<td>2.6</td>
<td>10.77</td>
<td>132.32</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tweet-Retrieval</td>
<td>Twitter</td>
<td>(7) Signal-1M (RT)</td>
<td>3-level</td>
<td>97</td>
<td>2,866,316</td>
<td>19.6</td>
<td>9.30</td>
<td>13.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>News-Retrieval</td>
<td>News</td>
<td>(8) TREC-NEWS</td>
<td>3-level</td>
<td>57</td>
<td>594,977</td>
<td>19.6</td>
<td>11.14</td>
<td>634.79</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Argument</td>
<td>Misc.</td>
<td>(9) ArguAna</td>
<td>3-level</td>
<td>49</td>
<td>382,545</td>
<td>49.2</td>
<td>6.55</td>
<td>292.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Retrieval</td>
<td>Misc.</td>
<td>(10) Touche-2020</td>
<td>3-level</td>
<td>1,406</td>
<td>8,674</td>
<td>1.0</td>
<td>192.98</td>
<td>166.80</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Duplicate-Question</td>
<td>StackEx</td>
<td>(11) CQADupStack</td>
<td>3-level</td>
<td>49</td>
<td>382,545</td>
<td>49.2</td>
<td>6.55</td>
<td>292.37</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Retrieval</td>
<td>Quora</td>
<td>(12) Quora</td>
<td>3-level</td>
<td>13,145</td>
<td>457,199</td>
<td>1.4</td>
<td>8.59</td>
<td>129.09</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Entity-Retrieval</td>
<td>Wikipedia</td>
<td>(13) DBPedia</td>
<td>3-level</td>
<td>67</td>
<td>4,635,922</td>
<td>38.2</td>
<td>5.39</td>
<td>49.68</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Citation-Prediction</td>
<td>Scientific</td>
<td>(14) SCIDOCS</td>
<td>3-level</td>
<td>1,000</td>
<td>25,657</td>
<td>4.9</td>
<td>9.38</td>
<td>176.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Fact Checking</td>
<td>Wikipedia</td>
<td>(15) FEVER</td>
<td>3-level</td>
<td>6,666</td>
<td>6,666</td>
<td>5,416,568</td>
<td>1.2</td>
<td>8.13</td>
<td>84.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wikipedia</td>
<td>(16) Climate-FEVER</td>
<td>3-level</td>
<td>1,535</td>
<td>5,416,593</td>
<td>3.0</td>
<td>20.13</td>
<td>84.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scientific</td>
<td>(17) SciFact</td>
<td>3-level</td>
<td>300</td>
<td>5,183</td>
<td>1.1</td>
<td>12.37</td>
<td>213.63</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Statistics of all the tasks, domains and datasets included in BEIR. Few datasets contain documents without titles. Relevancy column indicates the relation between the query and document: binary (relevant, irrelevant) or further graded into sub-levels. Avg. Docs/Query column indicates the average relevant documents per question.

IR Effectiveness Metrics

- We’ll use “metric”@K, often with K in {5, 10, 100, 1000}.
  - Selection of the metric (and the cutoff K) depends on the task.

- For all metrics here, we’ll [macro-]average across all queries.
  - All queries will be assigned equal weight, for our purposes.
IR Effectiveness Metrics: Success & MRR

- Let $rank \in \{1, 2, 3, \ldots \}$ be the position of the first relevant document

- $Success@K = \begin{cases} 1 & \text{if } rank \leq K \\ 0 & \text{otherwise} \end{cases}$

- $ReciprocalRank@K = \begin{cases} 1/rank & \text{if } rank \leq K \\ 0 & \text{otherwise} \end{cases}$

  - This is MRR (M for “mean”), but dropped the M as we’re looking at only one query
IR Effectiveness Metrics: Precision & Recall

- Let $Ret(K)$ be the top-K retrieved documents
- Let $Rel$ be the set of all documents judged as relevant

- Precision@K = $\frac{|Ret(K) \cap Rel|}{K}$

- Recall@K = $\frac{|Ret(K) \cap Rel|}{|Rel|}$
IR Effectiveness Metrics: MAP

- (M)AP = (Mean) Average Precision

- Let $rank_1, rank_2, ..., rank_{|Rel|}$ be the positions of all relevant documents
  - Compute precision@i at each of those positions—and average!

- Equivalently, AveragePrecision@K =

$$\sum_{i=1}^{K} \begin{cases} 
\text{Precision@i} & \text{if relevant? (i}^{\text{th}} \text{ document)} \\
0 & \text{otherwise}
\end{cases} \frac{}{|Rel|}$$
IR Effectiveness Metrics: DCG

■ Discounted Cumulative Gain
  - Not inherently normalized, so we also consider Normalized DCG

\[
DCG@K = \sum_{i=1}^{K} \frac{\text{graded_relevance}(i^{th \ document})}{\log_2(i + 1)}
\]

\[
NDCG@K = \frac{DCG@K}{\text{ideal \ DCG@K}}
\]
Next...

- Neural IR.
References


