Text Technologies for Data Science
INFR11145

IR Evaluation

Instructor:
Walid Magdy

19-Oct-2022

Lecture Objectives

• Learn about how to evaluate IR
  • Evaluation measures
  • P, R, F
  • MAP
  • nDCG

• Implement: (as part of CW2)
  • P, R
  • MAP
  • nDCG
Search Process

- Help user formulate the query by suggesting what he could search for
- Log user's actions: clicks, hovering, giving up
- Fetch a set of results, present to the user
- Iterate!

IR as an Experimental Science!

- Formulate a research question: the hypothesis
- Design an experiment to answer the question
- Perform the experiment
  - Compare with a baseline "control"
- Does the experiment answer the question?
  - Are the results significant? Or is it just luck?
- Report the results!
- Iterate …
- e.g. stemming improves results? (university → univers)
**Lab 3 output**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 65, 4.8040</td>
<td>2, 3549, 7.0396</td>
<td>3, 3354, 4.6113</td>
</tr>
<tr>
<td>1, 3533, 4.7264</td>
<td>2, 305, 6.8394</td>
<td>3, 335, 4.5087</td>
</tr>
<tr>
<td>1, 3562, 3.5454</td>
<td>2, 288, 6.6742</td>
<td>3, 268, 3.6606</td>
</tr>
<tr>
<td>1, 3608, 3.4910</td>
<td>2, 223, 6.1252</td>
<td>3, 328, 3.4825</td>
</tr>
<tr>
<td>1, 141, 3.3262</td>
<td>2, 219, 4.8626</td>
<td>3, 21, 3.3984</td>
</tr>
<tr>
<td>1, 361, 3.3262</td>
<td>2, 3762, 4.8626</td>
<td>3, 304, 3.3722</td>
</tr>
<tr>
<td>1, 92, 3.2311</td>
<td>2, 3663, 4.5415</td>
<td>3, 313, 3.3436</td>
</tr>
<tr>
<td>1, 3829, 3.1818</td>
<td>2, 3766, 3.9924</td>
<td>3, 3790, 3.1796</td>
</tr>
<tr>
<td>1, 3420, 3.1273</td>
<td>2, 188, 3.8844</td>
<td>3, 55, 3.0462</td>
</tr>
<tr>
<td>1, 3734, 3.0561</td>
<td>2, 3360, 3.0988</td>
<td>3, 217, 2.8492</td>
</tr>
<tr>
<td>1, 3387, 2.9626</td>
<td>2, 3408, 3.0315</td>
<td>3, 361, 2.8348</td>
</tr>
<tr>
<td>1, 3599, 2.9626</td>
<td>2, 3390, 2.8498</td>
<td>3, 3789, 2.7158</td>
</tr>
</tbody>
</table>

Is that a good performance?

---

**Configure your system**

- **About the system:**
  - Stopping? Tokenise? Stemming? n-gram char?
  - Use synonyms improve retrieval performance?
- **Corresponding experiment?**
  - Run your search for a set of queries with each setup and find which one will achieve the best performance
- **About the user:**
  - Is letting users weight search terms a good idea?
- **Corresponding experiment?**
  - Build two different interfaces, one with term weighting functionality, and one without; run a user study
Types of Evaluation Strategies

- **System-centered studies:**
  - Given documents, queries, and relevance judgments
  - Try several variations of the system
  - Measure which system returns the “best” hit list
  - Laboratory experiment

- **User-centered studies**
  - Given several users, and at least two retrieval systems
  - Have each user try the same task on both systems
  - Measure which system works the “best”

Importance of Evaluation

- The ability to measure differences underlies experimental science
  - How well do our systems work?
  - Is A better than B?
  - Is it really?
  - Under what conditions?

- Evaluation drives what to research
  - Identify techniques that work and don’t work
The 3-dimensions of Evaluation

- **Effectiveness**
  - How “good” are the documents that are returned?
  - System only, human + system

- **Efficiency**
  - Retrieval time, indexing time, index size

- **Usability**
  - Learnability, flexibility
  - Novice vs. expert users

---

**Cranfield Paradigm (Lab setting)**

1. **Query**
2. **IR System**
3. **Search Results**
4. **Evaluation Module**
5. **Relevance Judgments**
6. **Measure of Effectiveness**

---

**Cranfield Paradigm (Lab setting)**

1. **Query**
2. **Document Collection**
3. **IR System**
4. **Search Results**
5. **Evaluation Module**
6. **Relevance Judgments**
7. **Measure of Effectiveness**
Reusable IR Test Collection

- **Collection of Documents**
  - Should be “representative” to a given IR task
  - Things to consider: size, sources, genre, topics, …

- **Sample of information need**
  - Should be “randomized” and “representative”
  - Usually formalized **topic** statements (query + description)

- **Known relevance judgments**
  - Assessed by humans, for each topic-document pair
  - Binary/Graded

- **Evaluation measure**

Good Effectiveness Measures

- Should capture some aspect of what the user wants
  - IR → Do the results satisfy user’s information need?

- Should be easily replicated by other researchers

- Should be easily comparable
  - Optimally, expressed as a single number
    - Curves and multiple numbers are still accepted, but single numbers are much easier for comparison

- Should have predictive value for other situations
  - What happens with different queries on a different document collection?
Set Based Measures

- Assuming IR system returns sets of retrieved results without ranking
- Suitable with Boolean Search
- No certain number of results per query

Which looks the best IR system?

- For query Q, collection has 8 relevant documents:
Precision and Recall

- **Precision:**
  
  What fraction of these retrieved docs are relevant?

  $$P = \frac{\text{rel} \cap \text{ret}}{\text{retrieved}} = \frac{TP}{TP + FP}$$

- **Recall:**
  
  What fraction of the relevant docs were retrieved?

  $$R = \frac{\text{rel} \cap \text{ret}}{\text{relevant}} = \frac{TP}{TP + FN}$$
Which looks the best IR system?

- For query Q, collection has 8 relevant documents:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td></td>
<td>R</td>
<td>R</td>
<td></td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td></td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>P=5/10</td>
<td>P=6/12</td>
<td>P=5/12</td>
<td>P=5/12</td>
<td>P=6/12</td>
<td>R=5/8</td>
<td>R=6/8</td>
<td></td>
</tr>
</tbody>
</table>

Trade-off between P & R

- Precision: The ability to retrieve top-ranked docs that are mostly relevant.
- Recall: The ability of the search to find all of the relevant items in the corpus.
- Retrieve more docs:
  - Higher chance to find all relevant docs $\rightarrow$ $R \uparrow \uparrow$
  - Higher chance to find more irrelevant docs $\rightarrow$ $P \downarrow \downarrow$
**Trade-off between P & R**

- Returns relevant documents but misses many useful ones too
- The ideal
- Returns most relevant documents but includes lots of junk

![Graph showing the trade-off between Precision and Recall]

**What about Accuracy?**

- **Accuracy:**
  What fraction of docs was classified correctly?
  \[ A = \frac{TP + TN}{TP + FP + TN + FN} \]

![Venn diagram illustrating accuracy metrics]

10/15/22
One Measure? F-measure

\[
F_1 = \frac{2 \cdot P \cdot R}{P + R}
\]

\[
F_\beta = \frac{(\beta^2 + 1)P \cdot R}{\beta^2 P + R}
\]

- Harmonic mean of recall and precision
  - Emphasizes the importance of small values, whereas the arithmetic mean is affected more by outliers that are unusually large
- Beta (\(\beta\)) controls relative importance of P and R
  - \(\beta = 1\), precision and recall equally important \(\rightarrow F_1\)
  - \(\beta = 5\), recall five times more important than precision

Rank-based IR measures

- Consider systems A & B
  - Both retrieved 10 docs, only 5 are relevant
  - P, R & F are the same for both systems
  - Should their performances considered equal?
- Ranked IR requires taking “ranks” into consideration!
- How to do that?
Which is the best ranked list?

- For query Q, collection has 8 relevant documents:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td>1</td>
<td>R</td>
<td>1</td>
<td>R</td>
<td>1</td>
<td>R</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>R</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>R</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>R</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>R</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>R</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>R</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>R</td>
</tr>
<tr>
<td>9</td>
<td>R</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>R</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>R</td>
</tr>
<tr>
<td>11</td>
<td>R</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>R</td>
</tr>
<tr>
<td>12</td>
<td>R</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>R</td>
</tr>
</tbody>
</table>

Precision @ K

- k (a fixed number of documents)
- Have a cut-off on the ranked list at rank k, then calculate precision!
- Perhaps appropriate for most of web search: most people only check the top k results
- But: averages badly, Why?
• For query $Q$, collection has 8 relevant documents:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td>1</td>
<td>R</td>
<td>1</td>
<td>R</td>
<td>1</td>
<td>R</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

- $P@5$

$\text{R-Precision}$

• For a query with known $r$ relevant documents $\Rightarrow$ R-precision is the precision at rank $r$ ($P@r$)

• $r$ is different from one query to another

• Concept:
  It examines the ideal case: getting all relevant documents in the top ranks

• Is it realistic?
### R-Precision

- For query $Q$, collection has 8 relevant documents:

```
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td></td>
<td>1</td>
<td>R</td>
<td>1</td>
<td>R</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>2</td>
<td>R</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td>3</td>
<td>3</td>
<td>R</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>4</td>
<td>R</td>
<td>4</td>
<td>4</td>
<td>R</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>5</td>
<td>5</td>
<td>R</td>
<td>5</td>
<td>5</td>
<td>R</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>6</td>
<td>6</td>
<td></td>
<td>6</td>
<td>R</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>7</td>
<td>7</td>
<td>R</td>
<td>7</td>
<td>R</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>R</td>
<td>8</td>
<td>8</td>
<td>R</td>
<td>8</td>
<td>R</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>R</td>
<td>9</td>
<td>R</td>
<td>9</td>
<td></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td>10</td>
<td>10</td>
<td></td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>R</td>
<td>11</td>
<td>11</td>
<td>R</td>
<td>11</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>12</td>
<td>R</td>
<td>12</td>
<td>R</td>
<td>12</td>
<td></td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>
```

### User Satisfaction??

- It is assumed that users need to find relevant docs at the highest possible ranks
  - Precision is a good measure
- But, user would cut-off (stop inspecting results) at some point, say rank $x$
  - $P@x$
- What is the optimal $x$?
  - When you think a user can stop?
When a user can stop?

- IR objective: “satisfy user information need”
- Assumption: a user will stop once his/her information need is satisfied
- How? user will keep looking for relevant docs in the ranked list, read them, then stop once he/she feels satisfied
- $P@x \to x$ can be any rank where a relevant document appeared (assume uniform distribution)
- What about calculating the averages over all $x$'s?
  - every time you find relevant doc, calculate $P@x$, then take the average at the end

### Average Precision (AP)

<table>
<thead>
<tr>
<th>$Q_1$ (has 4 rel. docs)</th>
<th>$Q_2$ (has 3 rel. docs)</th>
<th>$Q_3$ (has 7 rel. docs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 R 1/1=1.00</td>
<td>1 R 1/3=0.33</td>
<td>1 R 1/2=0.50</td>
</tr>
<tr>
<td>2 R 2/2=1.00</td>
<td>2 R 2/7=0.29</td>
<td>2 R 2/5=0.40</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5 R 3/5=0.60</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7 R 2/7=0.29</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8 R 3/8=0.375</td>
</tr>
<tr>
<td>9 R 4/9=0.44</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10</td>
</tr>
</tbody>
</table>

- $\text{AP} = 3.04 / 4 = 0.76$
- $\text{AP} = 0.76$
- $\text{AP} = 1.275 / 7 = 0.182$
### Mean Average Precision (MAP)

<table>
<thead>
<tr>
<th>Query</th>
<th>Relevant Documents</th>
<th>Precision</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q₁</td>
<td>R 1/1 = 1.00</td>
<td>1</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>R 2/2 = 1.00</td>
<td>2</td>
<td>0.207</td>
</tr>
<tr>
<td></td>
<td>R 3/5 = 0.60</td>
<td>3</td>
<td>0.182</td>
</tr>
<tr>
<td>Q₂</td>
<td>R 1/3 = 0.33</td>
<td>1</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>R 2/7 = 0.29</td>
<td>2</td>
<td>0.207</td>
</tr>
<tr>
<td></td>
<td>R 3/8 = 0.375</td>
<td>3</td>
<td>0.182</td>
</tr>
<tr>
<td>Q₃</td>
<td>R 1/2 = 0.50</td>
<td>1</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>R 2/5 = 0.40</td>
<td>2</td>
<td>0.207</td>
</tr>
<tr>
<td></td>
<td>R 3/8 = 0.375</td>
<td>3</td>
<td>0.182</td>
</tr>
</tbody>
</table>

\[
\text{MAP} = \frac{(0.76+0.207+0.182)}{3} = 0.383
\]

### AP & MAP

\[
AP = \frac{1}{r} \sum_{k=1}^{n} P(k) \times \text{rel}(k)
\]

where, \( r \): number of relevant docs for a given query

\( n \): number of documents retrieved

\( P(k) \): precision @ \( k \)

\( \text{rel}(k) \): 1 if retrieved doc @ \( k \) is relevant, 0 otherwise.

\[
\text{MAP} = \frac{1}{Q} \sum_{q=1}^{Q} AP(q)
\]

where, \( Q \): number of queries in the test collection
**AP/MAP**

\[ AP = \frac{1}{r} \sum_{k=1}^{n} P(k) \times \text{rel}(k) \]

- A mix between precision and recall
- Highly focus on finding relevant document as early as possible
- When \( r = 1 \) \( \rightarrow \) MAP = MRR (mean reciprocal rank \( \frac{1}{k} \))
- MAP is the most commonly used evaluation metric for most IR search tasks
- Uses binary relevance: \( \text{rel} = 0/1 \)

---

**Binary vs. Graded Relevance**

- Some docs are more relevant to a query than other relevant ones!
  - We need non-binary relevance
- Binary Relevance:
  - Relevant \( 1 \)
  - Irrelevant \( 0 \)
- Graded Relevance:
  - Perfect \( 4 \)
  - Excellent \( 3 \)
  - Good \( 2 \)
  - Fair \( 1 \)
  - Bad \( 0 \)
**Binary vs. Graded Relevance**

- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant
  - The lower the ranked position of a relevant document, the less useful it is for the user, since it is less likely to be examined

- Discounted Cumulative Gain (DCG)
  - Uses graded relevance as a measure of the usefulness
  - The most popular for evaluating web search

**Discounted Cumulative Gain (DCG)**

- Gain is accumulated starting at the top of the ranking and may be reduced (discounted) at lower ranks

- Users care more about high-ranked documents, so we discount results by $1/\log_2(\text{rank})$
  - the discount at rank 4 is $1/2$, and at rank 8 is $1/3$

- $\text{DCG}_k$ is the total gain accumulated at a particular rank $k$ (sum of DG up to rank $k$):

\[
\text{DCG}_k = \text{rel}_1 + \sum_{i=2}^{k} \frac{\text{rel}_i}{\log_2(i)}
\]
Normalized DCG (nDCG)

• DCG numbers are averaged across a set of queries at specific rank values (DCG@k)
  • e.g., DCG at rank 5 is 6.89 and at rank 10 is 9.61
  • Can be any positive real number!

• DCG values are often normalized by comparing the DCG at each rank with the DCG value for the perfect ranking
  • makes averaging easier for queries with different numbers of relevant documents

• nDCG@k = DCG@k / iDCG@k (divide actual by ideal)

• nDCG ≤ 1 at any rank position

• To compare DCGs, normalize values so that an ideal ranking would have a normalized DCG of 1.0
Summary:

- **IR test collection:**
  - Document collection
  - Query set
  - Relevant judgements
  - IR measures

- **IR measures:**
  - R, P, F \( \rightarrow \) not commonly used
  - P@k, R-precision \( \rightarrow \) used sometimes
  - MAP \( \rightarrow \) the most used IR measure
  - nDGC \( \rightarrow \) the most used measure for web search
Resources

• Text book 1: Intro to IR, Chapter 8
• Text book 2: IR in Practice, Chapter 8