Lecture Objectives

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

- **Implement**:
  - P, R
  - MAP
  - nDCG
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?

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)
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:

<table>
<thead>
<tr>
<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>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>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>
</tr>
<tr>
<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>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>
</tr>
<tr>
<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>R</td>
<td>R</td>
<td>R</td>
</tr>
</tbody>
</table>
Precision and Recall

- **Precision**: What fraction of these retrieved docs are relevant?
  \[ P = \frac{rel \cap ret}{retrieved} = \frac{TP}{TP + FP} \]

- **Recall**: What fraction of the relevant docs were retrieved?
  \[ R = \frac{rel \cap ret}{relevant} = \frac{TP}{TP + FN} \]
**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**

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

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

---

*Walid Magdy, TTDS 2018/2019*
What about Accuracy?

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

*irrelevant >>>>> relevant*  
*(needle in a haystack)*

One Measure? F-measure

- 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 \(\rightarrow F1\)
  - \(\beta = 5\), 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>Rank</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>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>8</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>9</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>10</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>11</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>12</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></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?

---

**P@5**

- 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>R</td>
<td></td>
<td>R</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
<td>R</td>
</tr>
</tbody>
</table>

---
**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>R</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>R</td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>8</td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>R</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R</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, users 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 → 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)

**Q₁** (has 4 rel. docs)

- 1 → R 1/1 = 1.00
- 2 → R 2/2 = 1.00
- 3 → 
- 4 → 
- 5 → R 3/5 = 0.60
- 6 → 
- 7 → 
- 8 → 
- 9 → R 4/9 = 0.44
- 10 → 

**Q₂** (has 3 rel. docs)

- 1 → R 1/3 = 0.33
- 2 → 
- 3 → 
- 4 → 
- 5 → 
- 6 → 
- 7 → R 2/7 = 0.29
- 8 → 
- 9 → 
- 10 → 

**Q₃** (has 7 rel. docs)

- 1 → R 1/2 = 0.50
- 2 → 
- 3 → 
- 4 → 
- 5 → R 2/5 = 0.40
- 6 → 
- 7 → R 3/8 = 0.375
- 8 → 
- 9 → 
- 10 → 

\[AP = \frac{3.04}{4} = 0.76\]
\[AP = \frac{0.62}{3} = 0.207\]
\[AP = \frac{1.275}{7} = 0.182\]

### Mean Average Precision (MAP)

**Q₁** (has 4 rel. docs)

- 1 → R 1/1 = 1.00
- 2 → R 2/2 = 1.00
- 3 → 
- 4 → 
- 5 → R 3/5 = 0.60
- 6 → 
- 7 → 
- 8 → 
- 9 → R 4/9 = 0.44
- 10 → 

\[AP = 0.76\]

**Q₂** (has 3 rel. docs)

- 1 → R 1/3 = 0.33
- 2 → 
- 3 → 
- 4 → 
- 5 → 
- 6 → 
- 7 → R 2/7 = 0.29
- 8 → 
- 9 → 
- 10 → 

\[AP = 0.207\]

**Q₃** (has 7 rel. docs)

- 1 → R 1/2 = 0.50
- 2 → 
- 3 → 
- 4 → 
- 5 → R 2/5 = 0.40
- 6 → 
- 7 → R 3/8 = 0.375
- 8 → 
- 9 → 
- 10 → 

\[AP = 0.182\]

\[MAP = (0.76 + 0.207 + 0.182)/3 = 0.383\]
**AP & MAP**

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

where, \( r \): number of relevant docs for a given query
\( n \): number of documents retrieved
\( P(k) \): precision @ \( k \)
\( rel(k) \): 1 if retrieved doc @ \( k \) is relevant, 0 otherwise.

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

where, \( Q \): number of queries in the test collection

**AP/MAP**

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

- 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: rel = 0/1
**MAP**

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

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.500</td>
<td>0.625</td>
<td>0.556</td>
<td>0.424</td>
</tr>
<tr>
<td>B</td>
<td>0.500</td>
<td>0.750</td>
<td>0.600</td>
<td>0.290</td>
</tr>
<tr>
<td>C</td>
<td>0.417</td>
<td>0.625</td>
<td>0.500</td>
<td>0.433</td>
</tr>
<tr>
<td>D</td>
<td>0.333</td>
<td>0.500</td>
<td>0.400</td>
<td>0.475</td>
</tr>
<tr>
<td>E</td>
<td>0.375</td>
<td>0.375</td>
<td>0.375</td>
<td>0.262</td>
</tr>
<tr>
<td>F</td>
<td>0.500</td>
<td>0.750</td>
<td>0.600</td>
<td>0.420</td>
</tr>
<tr>
<td>G</td>
<td>0.800</td>
<td>0.500</td>
<td>0.615</td>
<td>0.340</td>
</tr>
</tbody>
</table>

**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 \(\frac{1}{\log_2(\text{rank})}\)
  - 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@$k = \frac{DCG@k}{iDCG@k}$ (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

<table>
<thead>
<tr>
<th>k</th>
<th>G</th>
<th>DG</th>
<th>DCG@$k$</th>
<th>iG</th>
<th>iDG</th>
<th>iDCG@$k$</th>
<th>nDCG@$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3.00</td>
<td>3</td>
<td>3.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2.5</td>
<td>3.00</td>
<td>6</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1.89</td>
<td>1.89</td>
<td>7.89</td>
<td>0.87</td>
<td>0.87</td>
<td>0.87</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>6.89</td>
<td>2.00</td>
<td>8.89</td>
<td>0.78</td>
<td>0.78</td>
<td>0.78</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>6.89</td>
<td>2.00</td>
<td>9.75</td>
<td>0.71</td>
<td>0.71</td>
<td>0.71</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0.39</td>
<td>0.77</td>
<td>10.52</td>
<td>0.69</td>
<td>0.69</td>
<td>0.69</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0.71</td>
<td>0.36</td>
<td>10.88</td>
<td>0.73</td>
<td>0.73</td>
<td>0.73</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>0.67</td>
<td>0.00</td>
<td>10.88</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>0.95</td>
<td>0.00</td>
<td>10.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.96</td>
<td>0.00</td>
<td>10.88</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
</tr>
</tbody>
</table>
Summary:

- IR test collection:
  - Document collection
  - Query set
  - Relevant judgements
  - IR measures
- IR measures:
  - R, P, F → not commonly used
  - P@k, R-precision → used sometimes
  - MAP → the most used IR measure
  - nDGC → the most used measure for web search

Resources

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