What we’ll learn today

▶ How (in principle) to build a reusable test collection for evaluating IR systems
▶ How to evaluate and compare IR systems against such a test collection, using effectiveness metrics
Meeting human information needs

[ SELECT * FROM customers WHERE city=’Sydney’ AND age > 45 ]
[ ( jaguar OR irvine OR webber ) AND ( race OR competition OR ”grand prix” ) w/10 ( statistics OR results OR scores ) ]
[ jaguar race statistics ]

▶ Free text queries are not formal representations of information sought (unlike SQL or Boolean queries)
▶ Rather, they are informal, suggestive approximations of what user wants (which user themselves may not exactly know)
▶ “Correct” answers are not formally definable
▶ “Models” only guides, don’t determine theoretical correctness

All models are wrong, but some are useful – George E. P. Box, 1987

▶ System must do “best it can” to:
  ▶ Infer user’s intent
  ▶ Predict result responsiveness to this intent
Correct answers not formally definable

- How well the system’s results (for a query, for all queries) meet a user’s need is referred to as the system’s *effectiveness*.
- And the process of determining this effectiveness (for a given query, a given set of queries, or in general) is known as *effectiveness evaluation*.
- Cannot use evaluation regimes such as a *correct system is one in which all documents returned contain all query keywords*.
- Ultimately, effectiveness defined by user’s satisfaction with or utility from results.
Direct human evaluation

- Obvious evaluation method: direct evaluation with human users, effectiveness measure from:
  - reported satisfaction
  - completion of tasks

- But method too expensive, slow for comparing, tuning many different formulae or parameters:
  - $\text{TF} = f_{d,t} \ OR \ \log(f_{d,t} + 1) \ OR \ \ldots$
  - Pivoted DLN slope $s = 1.0 \ OR \ 0.9 \ OR \ \ldots$
  - PRF with 1 or 3 or 5 or \ldots top documents
  - Rocchio parameter $\alpha = 0.4 \ OR \ 0.5 \ OR \ 0.6 \ OR \ \ldots$
  - Across 200+ different queries

- Complexities of experimental setup (user to evaluate 20 results for one query, without learning or fatiguing)
Automated testing

- We want evaluation setup that can be run automatically
- While still being based upon human perceptions of effectiveness
- To achieve this, we will have to make some simplifications!
- Begin with “maximal” set of simplifications applied
- ...to create (traditional, TREC, Cranfield) test collection model
Framework

- User has information need
- Express this need as a query
- System runs query against corpus
- Returns ranked list of documents
- Effectiveness is how well this ranked list satisfies information need
Simplifying assumption 1: Ad-hoc

Retrieval is Ad-Hoc

- Query is made once
  - No opportunity for refinement, feedback
- We have no prior knowledge of the user (their interests, preferences)
- We have no prior knowledge of behaviour of other users for this query
Simplifying assumption 2: Relevance

Effectiveness based upon *relevance*

- Each document is either relevant or irrelevant to *information need*
  - Note: more exact to speak of “relevance to information need” than “relevance to query”
- Relevance is binary (document is either wholly relevant or wholly irrelevant)
- Relevance of one document in result independent of relevance of other documents in result (no redundancy, diversity)
- Effectiveness of result is function of relevance of documents in result
Example query from TREC

Vikings in Scotland
Description: What hard evidence proves that the Vikings visited or lived in Scotland?
Narrative: A document that merely states that the Vikings visited or lived in Scotland is not relevant. A relevant document must mention the source of the information, such as relics, sagas, runes or other records from those times.

Source http://trec.nist.gov/data/topics_eng/topics.501-550.txt
Test collection

With these assumptions, automated effectiveness evaluation performable with a reusable *test collection*, consisting of three (main) components:

- **Corpus** set of documents
- **Queries** set of queries to run against corpus
  - Sometimes supplemented by fuller descriptions of underlying information need
  - In which case we speak of “topics”
- **Qrels** for each document and query, a (human) judgment of whether that document is relevant to (the information need underlying) that query
Converting document ranking into relevance vector

<table>
<thead>
<tr>
<th>Retrieval run</th>
<th>Qrels</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Docid</strong></td>
<td><strong>Score</strong></td>
</tr>
<tr>
<td>CR93H-9548</td>
<td>0.5436</td>
</tr>
<tr>
<td>CR93H-12789</td>
<td>0.4958</td>
</tr>
<tr>
<td>CR93H-10580</td>
<td>0.4633</td>
</tr>
<tr>
<td>CR93H-14389</td>
<td>0.4616</td>
</tr>
<tr>
<td>AP880828-0030</td>
<td>0.4523</td>
</tr>
<tr>
<td>CR93H-10986</td>
<td>0.4383</td>
</tr>
<tr>
<td>CR93H-10986</td>
<td>0.4383</td>
</tr>
<tr>
<td>CR93H-12789</td>
<td>0.4383</td>
</tr>
</tbody>
</table>

Relevance vector

$$\langle 1, 0, 0, 0, 0, 1, \ldots \rangle$$ (1)

- Take retrieval run as a ranking of document ids (already a very abstracted representation!)
- Look up relevance of document ids in qrels dictionary
- Convert run into relevance vector
**Effectiveness of result is function of relevance of documents in result**

- Need function to express effectiveness of relevance vector as a single number

\[ m(\langle <1,0,1,0,0,1,1,\ldots> \rangle) \to 0.8 \quad (2) \]

- This function an *effectiveness metric*
- And the number it reports an *effectiveness score*
Recall and precision

Two fundamental (set-based) measures:

**Recall**  Proportion of relevant documents retrieved, \(\frac{|R \cap F|}{|R|}\)

**Precision**  Proportion of retrieved documents relevant, \(\frac{|R \cap F|}{|F|}\)

Why not **accuracy**, \(\frac{|R \cap F| + |\neg R \cap \neg F|}{|D|}\)?
Recall and precision for boolean queries

Recall and precision are set-based measures

- Can evaluate against the set of returned documents, $\mathcal{F}$ for boolean queries
- Not (directly) applicable to ranking models

Consider the contingency table

<table>
<thead>
<tr>
<th></th>
<th>$\mathcal{R}$</th>
<th>$\neg\mathcal{R}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{F}$</td>
<td>TP</td>
<td>FP</td>
</tr>
<tr>
<td>$\neg\mathcal{F}$</td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>

reporting True and False Positives; and False and True Negatives.

Can compute precision and recall directly

$$p = \frac{TP}{TP + FP}$$

$$r = \frac{TP}{TP + FN}$$
Balancing Precision and Recall

Often the two are mutually exclusive

- High recall only possible from returning many results
- But precision tends to suffer
- Some users care about recall (e.g., legal queries)
- Other users just want a one or two good results (precision)

How to balance the two components? F-score is one answer

\[
F_\beta = \frac{(1 + \beta^2)p \times r}{\beta^2 p + r}
\]

where \( \beta \in [0, \infty) \) encodes the balance. Typically see

\[
F_1 = \frac{2p \times r}{p + r}
\]
Issues with Recall

Recall is fundamentally harder than precision to capture; why?

- Our corpus $\mathcal{D}$ is very large
- How to find all relevant documents, $\mathcal{R}_q$?
- Could laboriously look at every document . . . too slow!

Pooling as a solution

- Get top $k$ results for several different IR systems
- Judge only entries in the union of these results
- Assumes that all relevant documents will be found by at least one system
- But this is often not true, underestimates $\mathcal{R}_q$
- Conveys bias to recall based metrics
Precision @ k

Simple measure, \( prec@k \):

- Truncate ranking to depth \( k \)
- Calculate precision of prefix

\[
p@k(\vec{f}) = \frac{1}{k} \sum_{i=1}^{k} f_i \tag{3}
\]

\[
p@5(\langle 1, 1, 0, 1, 0, 1, 0, 1, 0 \ldots \rangle) = p@5(\langle 1, 1, 0, 1, 0 \rangle) = \frac{1}{5} \cdot 3 = 0.6 \tag{4}
\]

\[
rec@k(\vec{f}) = c_q \cdot p@k(\vec{f}), \text{ where } c_q \text{ is a query-dependent constant:}
\]

- Why?
- What is \( c_q \)?
**Precision @ k**

Two objections to Precision @ k:

**Not rank-sensitive**

- Doesn’t reward better rankings up to $k$:
  
  $$p@5(⟨1, 0, 0, 0, 0⟩) = p@5(⟨0, 0, 0, 0, 1⟩)$$  \hspace{1cm} (5)

- More exactly: rank-sensitivity is very coarse; ranks up to $k$ get same weight of $1/k$; ranks beyond $k$ get weight of 0

**Not recall-sensitive**

- Ignores number of relevant documents for query, $R_q = |\mathcal{R}_q|$
- Maximum $p@100$ when $R_q = 1$ is 0.01.
- $p@5 = 1.0$ easier where $R_q = 1000$ than $R_q = 5$

This important where aggregating scores over multiple queries
Incorporating Recall: the P-R curve

Rather than use a static $k$, consider precision at all levels of $k$ and record the precision and recall:

(Figure from MRS, IIR chapter 8)

- Recall of 1 results from returning all documents
- Good systems have high precision for the top ranked documents (i.e., for $r \approx 0$, usually $p \approx 1$)
Mean average precision (MAP)

Mean average precision:

The average precision at each point in the ranking a relevant document occurs:

In practice

- relevant documents not in ranking given precision 0

\[
AP(\vec{f}; q) = \frac{1}{R_q} \sum_{i=1}^{R_q} p@r(i)(\vec{f})
\]  

(6)

where \( r(i) \) is the depth in the ranking needed to recover the \( i^{th} \) relevant document. More simply,

\[
AP(\vec{f}; k, q) = \frac{1}{R_q} \sum_{i=1}^{k} f_i \cdot p@i(\vec{f})
\]  

(7)

- ranking generally truncated at some depth \( k \) (e.g. \( k = 1000 \))
- approximates the area under the P-R curve
Since early 1990s, academic IR evaluation focused around collaborative evaluation “competitions”, that:
- share effort of creating collection (particularly, evaluating documents for relevance to queries)
- provide common benchmark for performance

First and most famous of these is TREC (Text REtrieval Conference), run annually, based at NIST in US.

Typical ad-hoc TREC collection contains:
- 50 topics (queries, with more extended relevance statements), authored by experience independent searchers
- Qrels for top 100 results returned by each participant to each query (pooling) (remaining documents assumed irrelevant), judged by topic authors
- (Externally) results submitted by participants
Example TREC datasets

TREC 5, 1996

Topic

<table>
<thead>
<tr>
<th>num</th>
<th>Number: 252</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>Topic: Combating Alien Smuggling</td>
</tr>
<tr>
<td>desc</td>
<td>Description: What steps are being taken by governmental or even private entities world-wide to stop the smuggling of aliens.</td>
</tr>
<tr>
<td>narr</td>
<td>Narrative: To be relevant, a document must describe an effort being made (other than routine border patrols) in any country of the world to prevent the illegal penetration of aliens across borders.</td>
</tr>
</tbody>
</table>

Qrels

<table>
<thead>
<tr>
<th>Topic</th>
<th>Docid</th>
<th>Rel</th>
</tr>
</thead>
<tbody>
<tr>
<td>252</td>
<td>AP881226-0140</td>
<td>1</td>
</tr>
<tr>
<td>252</td>
<td>AP881227-0083</td>
<td>0</td>
</tr>
<tr>
<td>252</td>
<td>CR93E-10038</td>
<td>0</td>
</tr>
<tr>
<td>252</td>
<td>CR93E-1004</td>
<td>0</td>
</tr>
<tr>
<td>252</td>
<td>CR93E-10211</td>
<td>0</td>
</tr>
<tr>
<td>252</td>
<td>CR93E-10529</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Runfile

<table>
<thead>
<tr>
<th>Topic</th>
<th>Docid</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>252</td>
<td>CR93H-9548</td>
<td>0.5436</td>
</tr>
<tr>
<td>252</td>
<td>CR93H-12789</td>
<td>0.4958</td>
</tr>
<tr>
<td>252</td>
<td>CR93H-10580</td>
<td>0.4633</td>
</tr>
<tr>
<td>252</td>
<td>CR93H-14389</td>
<td>0.4616</td>
</tr>
<tr>
<td>252</td>
<td>AP880828-0030</td>
<td>0.4523</td>
</tr>
<tr>
<td>252</td>
<td>CR93H-10986</td>
<td>0.4383</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparing systems on a test collection

Compare two systems on test collection:

- Run each system against each topic
- Calculate per-topic effectiveness score under selected metric (e.g. AP)
- Calculate systems score on collection as mean of topic scores
- Compare systems by mean score
- Test mean score differences for statistical significance
Extending test collection model: diversity

- Similar documents make each other redundant in results list
- Query may have many intents or aspects
  - **Intents**: different topics underlying same query
  - **Aspects**: different parts of information about the one topic
- Want to avoid redundancy, reward diversity in results list

**Pros**
- Very important aspect of practical retrieval satisfaction, utility

**Cons**
- Places a much heavier load on assessor / organizers
Extending test collection model: multi-session

- In practice, a user can refine their query, search interactively
- System should respond to a query differently if it is a refinement
- Recent attempts to do this in a test collection
- ... but very difficult!
- May have to be approached through interaction studies (see next)
Automatic user feedback methods available on working, heavily-used system (e.g. web search engine):

- Click-through statistics (if a user clicks on a result, treat that result as “correct”)
- Try different result lists on users, and observe click and other behaviour:
  - A/B testing (show different result lists to different users)
  - Result interleaving (interleave results from two algorithms in the one list)
  - Randomised treatments over more than two options, multivariate regression analysis
Looking back and forward

Retrieval effectiveness must be measured against human perception

Human-in-loop too expensive for regular experiments

Test collection “cans” human as qrels

Metric calculates score from relevance vector

Compare two systems by scores on set of topics from one collection
Looking back and forward

Forward

▶ Next lecture looks at exploiting relevance judgements for query refinement, aka “relevance-feedback”
▶ Also other methods of query expansion using knowledge sources such as thesauri
Further reading

- Chapter 8, “Evaluation” of Manning, Raghavan, and Schutze, *Introduction to Information Retrieval*
- Overview of one of the TREC conferences, for instance TREC 5 http://trec.nist.gov/pubs/trec5/papers/overview.ps.gz (note: gzipped postscript)