Evaluation of document ranking

CS6101

How good is a retrieval system?

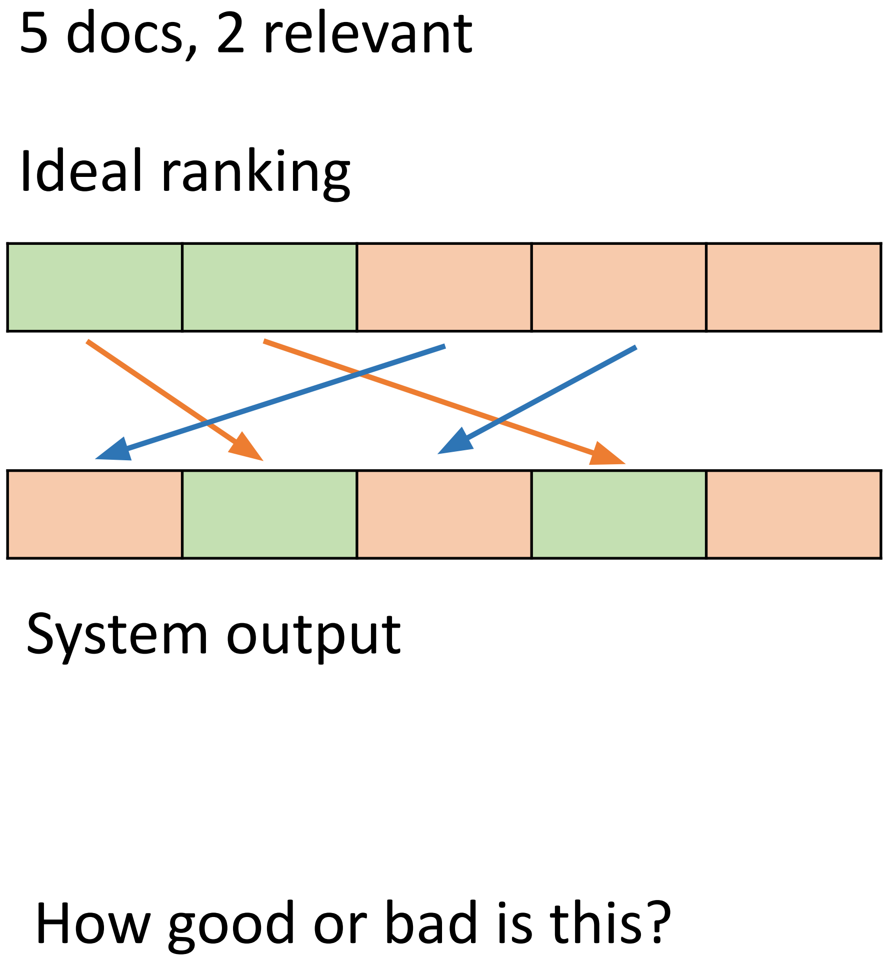
- Depends on operating objectives
  - Payoff to one/many users, fairness, diversity, ad revenue
  - Latency, throughput, RAM, cache, core footprint, etc.
- Depends on expected form of output, e.g., unordered set vs. ordered list
- Depends on form of gold relevance judgments available
  - Incomplete set of relevant docs (most common) ②
  - Incomplete set of pairwise preferences (also common) ③
  - Complete set of relevant docs ①
  - Total order over relevant docs
  - Total order over all corpus docs (impossible)
- Meanwhile, our system capability is to assign a score to each corpus doc
  - Can threshold to convert to set

① Complete relevant set known

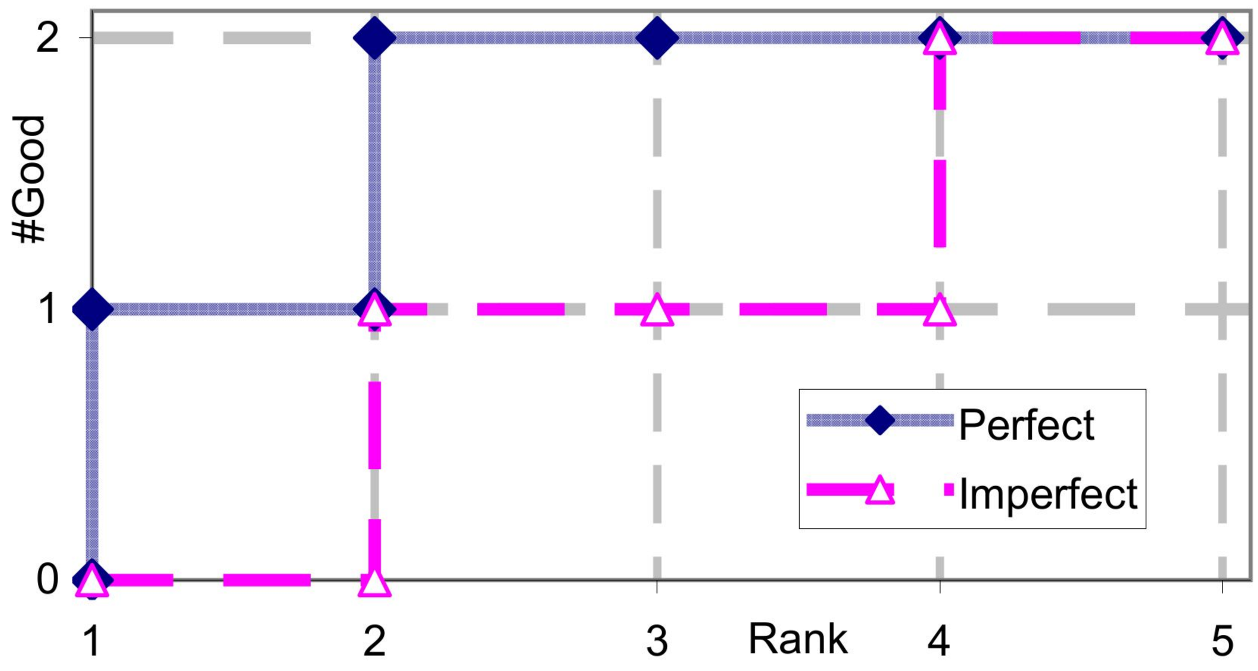
- Say $\mathcal{D}$ or D is the whole corpus
- Query q, gold relevant doc set $D_q^{\oplus}$ or $D_{q\oplus}$
  - Irrelevant doc set $D_{q\ominus} = \mathcal{D} \setminus D_{q\oplus}$
  - Usually $|D_{q\oplus}| \ll |D_{q\ominus}| \approx |\mathcal{D}|$ ("mostly irrelevant")
- If $D_{q\ominus}$ too large, ML systems draw negative samples
  - These samples are very unlikely to contain relevant documents
  - ML systems frequently compare scores of known relevant and sample assumed irrelevant doc pairs

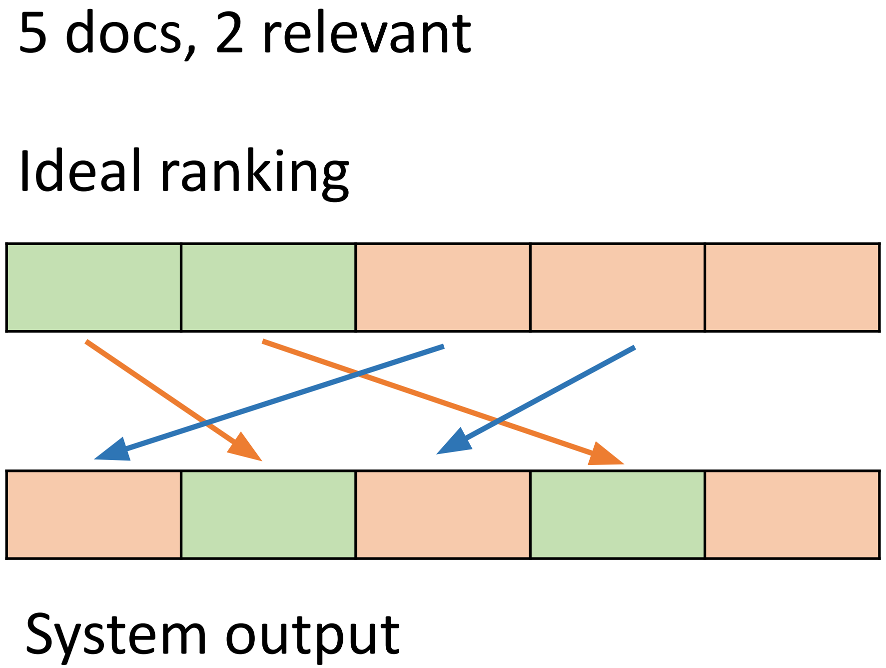
Case: System outputs *ranking*

- Gold is known as unordered set
  - Aka *binary* relevance
- System outputs a total order
- Ideally, all known relevant docs should be at the top of the list
- If that does not happen, measure how far we are from the ideal ranking



Area under the curve





- Two related ways to measure
  - Area between ideal and system curves
  - Number of cross-overs (“discordant pairs”) — will return to this soon
- All relevants are equivalent and so are irrelevants (don’t shuffle among these)

Mean reciprocal rank (MRR)

- Consider navigational query q with one relevant doc
  - Say system places it at rank $p_{q,1}$
  - If more than one, consider only top-ranking relevant doc
- Large $\sum_q p_{q,1}$ means ranking is ineffective
  - Sum can be dominated by few unfixable queries (outliers)
- Instead, consider $1/p_{q,1}$ as a reward
  - 1 for nailing first place, $^1/_2$ for second place, …
  - Dropping from 1 to 2 as bad as dropping from 2 to ∞
  - “Mean” reciprocal
- Average over query set Q to get $\frac{1}{|Q|}\sum_q \frac{1}{p_{q,1}}$
- Sometime truncated at “patience limit” rank K to get $\frac{1}{|Q|}\sum_q \frac{1}{p_{q,1}}[\![p_{q,1} \le K]\!]$

- Untrained model
- Ranks 2, 2, 2, 2, 2, 100
- Train model A
- Ranks 1, 1, 1, 1, 1, 100
- Train model B
- Ranks 2, 2, 2, 2, 2, 50
- Which is better, A or B?

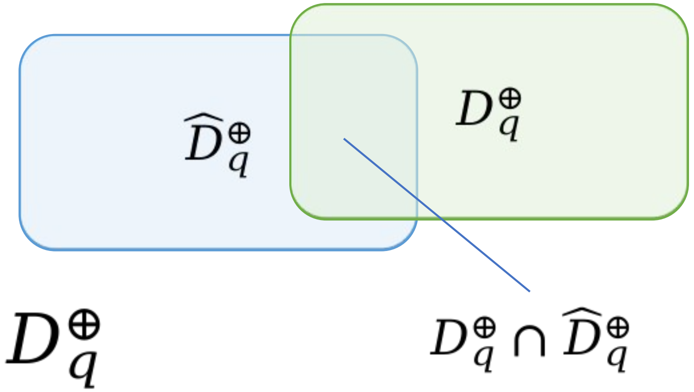
Mean average precision (MAP)

- Suppose ("informational") query q has R_q relevant docs
- Ideal ranks would be $1, 2, \cdots, R_q$
- System places them at ranks $1 \le p_{q,1} < p_{q,2} < \cdots < p_{q,R_q}$
  - 1 of first $p_{q,1}$ is relevant: precision $= 1/p_{q,1}$
  - 2 of first $p_{q,2}$ are relevant: precision $= 2/p_{q,2}$
  - Average over these: $\frac{1}{R_q}\sum_{r=1}^{R_q} \frac{r}{p_{q,r}}$

    Payoff (relevant docs)

    Effort (inspected docs)
  - Average precision in $[0,1]$
- And then find mean over queries
- Robust performance metric, rewards incremental progress of learning-to-rank algorithms

Case: System outputs *set*



- System outputs claimed relevant set $\widehat{D}$ or $\widehat{D}_q^{\oplus}$
  - E.g. using a cut-off at rank K
- Various ways to measure the goodness of $\widehat{D}_q^{\oplus}$ wrt gold $D_q^{\oplus}$
  - What fraction of gold are we recalling? $r = \frac{|D_q^{\oplus} \cap \widehat{D}_q^{\oplus}|}{|D_q^{\oplus}|}$
  - What fraction of the system response is relevant aka precise? $p = \frac{|D_q^{\oplus} \cap \widehat{D}_q^{\oplus}|}{|\widehat{D}_q^{\oplus}|}$
  - Can trade off one for the other
  - Harmonic mean $F_1 = \frac{1}{\frac{1}{2}\left(\frac{1}{p} + \frac{1}{r}\right)} = \frac{2pr}{p+r}$ discourages imbalanced trade-off
- Recall at K, precision at K — usually a tradeoff between them

Set-wise losses

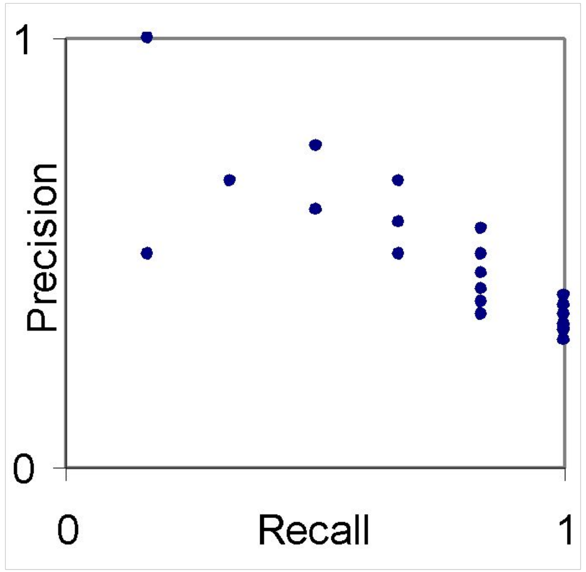
- Gold sets $D_{q\oplus} \cup D_{q\ominus} = D$
- System retrieves $\widehat{D}$ (perhaps rank and cut-off)
- $\text{recall} = \frac{|\widehat{D} \cap D_{q\oplus}|}{|D_{q\oplus}|}$ and $\text{precision} = \frac{|\widehat{D} \cap D_{q\oplus}|}{|\widehat{D}|}$
- Recall measures what fraction of relevant docs are retrieved
- Precision measures what fraction of retrieved docs are relevant
- Harmonic mean $F_1 = \frac{1}{\frac{1}{2}\left[\frac{1}{R}+\frac{1}{P}\right]} = \frac{2PR}{P+R}$
  - Penalizes extreme lop-sided trade-offs
- Eval involves discrete counts, needs work to find surrogate smooth losses

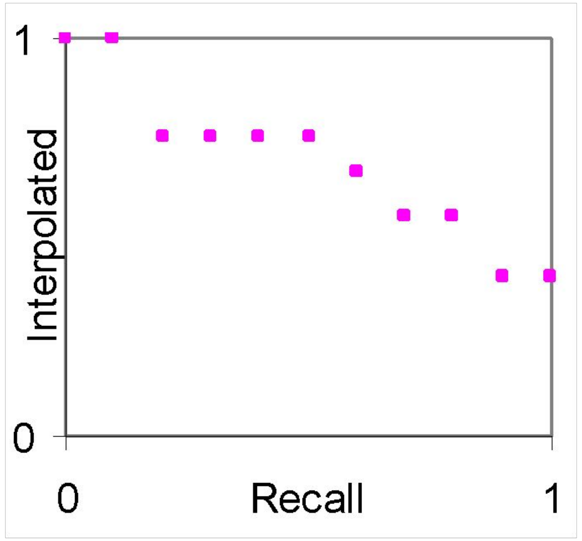
Sets via rank-and-cutoff

- System provides ranking
- We inspect up to rank k, corresponding to doc subset $D_k \subset D$
- These define recall@k and precision@k
- If $k = 0$, precision is 1 by convention
  - "Silent and correct" vs "verbose and often wrong"
- If $k = |D|$, recall is 1, precision is $\frac{|D_{q\oplus}|}{|D|} \to 0$
- P@k, R@k usually, but not always, negatively related
- Interpolated precision; average over queries

Precision-recall trade-off

k	r_k
1	1
2	
3	1
4	1
5	
6	1
7	
8	
9	1
10	
11	
12	
13	
14	
15	1
16	
17	
18	
19	
20	





- As recall is increased, precision generally (but not always) decreases
- Interpolated precision fixes this anomaly
- Many ways to reduce to single number
  - R, P, F1
  - Break-even point
  - MRR
  - AUC
  - MAP
  - NDCG

② Incomplete relevant set known

- More accurately, (e.g. TREC) competitors report top-K lists
- These are merged ("pooled") and rated by humans
- Docs outside the pool are not rated and may content more relevant docs
- Recall cannot be measured and is not the concern
- Precision at top K ranks is the focus
- Recall the eye tracking heatmaps
  - Walking down the list has cognitive cost
  - Relevant doc offers a reward, decreasing with rank
- Normalized discounted cumulative gain (NDCG) for one query

$$\frac{1}{Z}\sum_{k=1}^{K}\frac{[\![\text{doc at } k \text{ is relevant}]\!]}{\text{discount}(k)}$$

0/1 judgment may be replaced with a few grades of relevance

nt is usually
$_2\,(K+1)$

Divide by maximum DCG achievable given the pool, to ensure each query has equal say in evaluation

Cumulative

Average over queries

③ Incomplete pairwise preferences

- Most realistic data collection scenario
  - Tell editor to compare two docs wrt query
  - (More common) collect noisy preferences from views and clicks
- Skip rank 1 and click (and dwell) on 2 strong evidence that 2 is better than 1
  - Denoted $2 \succ 1$, or $d_2 \succ d_1$
  - Reverse event (dwell on 1 more than 2) is weaker signal because of presentation bias
- Perfectly possible that
  - Different users find $d_2 \succ d_1$ and $d_1 \succ d_2$ even for same query
  - Same user finds $d_3 \succ d_2 \succ d_1 \succ d_3$ (cycle)

Evaluation wrt pairwise preference set $\mathcal{P}_q$

- Search system will generally provide a single (total) order by scoring all docs in response to a fixed query
  - $s(d_i|q) > s(d_j|q)$ or $s(d_i|q) < s(d_j|q)$
  - Not really: personalization, generative AI, …
- If $s(d_i|q) > s(d_j|q)$ but $i \prec j \in \mathcal{P}_q$, assess one unit of loss

$$\frac{1}{|\mathcal{P}_q|} \sum_{i \prec j \in \mathcal{P}_q} [\![s(d_i|q) > s(d_j|q)]\!]$$

- Fraction of pair-preferences that are violated, in $[0,1]$
  - Compare with area under curve (AUC)
- Now average over queries in workload
- More practical and reliable than asking for absolute relevance judgments
- Flipping 1 and 11 vs 18 and 19 have same penalty ☹

Eval recap

- Square error, ordinal error
- Recall, precision, F1, break-even
- MRR, AP, MAP, NDCG
- Smooth vs. non-smooth
  - Many of the eval measures are not smooth wrt typical ranking model params
- Next: design smooth surrogates to train learning-to-rank models