Information Retrieval

Evaluation

This lecture

- How do we know if our results are any good?
  - Evaluating a search engine
    - Benchmarks
    - Precision and recall
- Results presentation:
  - Making our good results usable to a user

Measures for a search engine

- How fast does it **index**
  - Number of documents/hour
  - (Average document size)
- How fast does it **search**
  - Latency as a function of index size
- Expressiveness of query language
  - Ability to express complex information needs
  - Speed on complex queries
- Nice UI
- Other features
  - Spelling, similarity, query expansion, semantics

Efficiency Metrics

Metric name	Description
Elapsed indexing time	Measures the amount of time necessary to build a document index on a particular system.
Indexing processor time	Measures the CPU seconds used in building a document index. This is similar to elapsed time, but does not count time waiting for I/O or speed gains from parallelism.
Query throughput	Number of queries processed per second.
Query latency	The amount of time a user must wait after issuing a query before receiving a response, measured in milliseconds. This can be measured using the mean, but is often more instructive when used with the median or a percentile bound.
Indexing temporary space	Amount of temporary disk space used while creating an index.
Index size	Amount of storage necessary to store the index files.

Measures for a search engine

- Many of these criteria are measurable

Metric name	Description
Elapsed indexing time	Measures the amount of time necessary to build a document index on a particular system.
Indexing processor time	Measures the CPU seconds used in building a document index. This is similar to elapsed time, but does not count time waiting for I/O or speed gains from parallelism.
Query throughput	Number of queries processed per second.
Query latency	The amount of time a user must wait after issuing a query before receiving a response, measured in milliseconds. This can be measured using the mean, but is often more instructive when used with the median or a percentile bound.
Indexing temporary space	Amount of temporary disk space used while creating an index.
Index size	Amount of storage necessary to store the index files.

- But the key measure is user happiness
  - What is this?
  - Speed of response/size of index are factors
  - But blindingly fast, useless answers won't make a user happy

- Need a way of quantifying user happiness

Measuring user happiness

- Who is the user we are trying to make happy?
  - Depends on the setting
- Web engine:
  - User finds what they want and return to the engine
    - Can measure rate of return users
  - User completes their task – search as a means, not end
- eCommerce site:
  - User finds what they want and buy
  - Is it the end-user, or the eCommerce site, whose happiness we measure?
  - Measure time to purchase, or fraction of searchers who become buyers?
- Institutional: Care about "user productivity"
  - How much time do my users save when looking for information?
  - Many other criteria having to do with breadth of access, secure access, etc.

Happiness: elusive to measure

- Most common proxy: **relevance of search results**

- Relevance measurement requires 3 elements:
  - A benchmark document collection
  - A benchmark suite of queries
  - A usually binary assessment of either Relevant or Nonrelevant for each query and each document

Evaluating an IR system

- Note: the **information need** is translated into a **query**
- Relevance is assessed relative to the **information need** *not* the **query**

- E.g., <u>Information need</u>: *I'm looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.*
- <u>Query</u>: ***wine red white heart attack effective***

- We evaluate whether the doc addresses the information need, not whether it has these words

Standard relevance benchmarks

- The Text Retrieval Conference (**TREC**)
  - co-sponsored by the National Institute of Standards and Technology (NIST) has run a large IR test bed for many years
  - https://trec.nist.gov/
- Conference and Labs of the Evaluation Forum (**CLEF**)
  - Promotes research, innovation, and development of information access systems with an emphasis on multilingual and multimodal information
  - https://www.clef-initiative.eu/
- Human experts mark, for each query and each doc, Relevant or Nonrelevant
  - or at least for a subset of docs that some system returned for that query

Standard relevance benchmarks – TREC example

Retrieval Augmented Generation (RAG)

The RAG track aims to enhance retrieval and generation effectiveness to focus on varied information needs in an evolving world. Data sources will include a large corpus and topics that capture long-form definitions, list, and ambiguous information needs.

The track will involve 2 subtasks:

Retrieval Task : Rank passages for a given queries

RAG Task : Generate answers with supporting attributed passages

The second task takes the primary focus of the track.

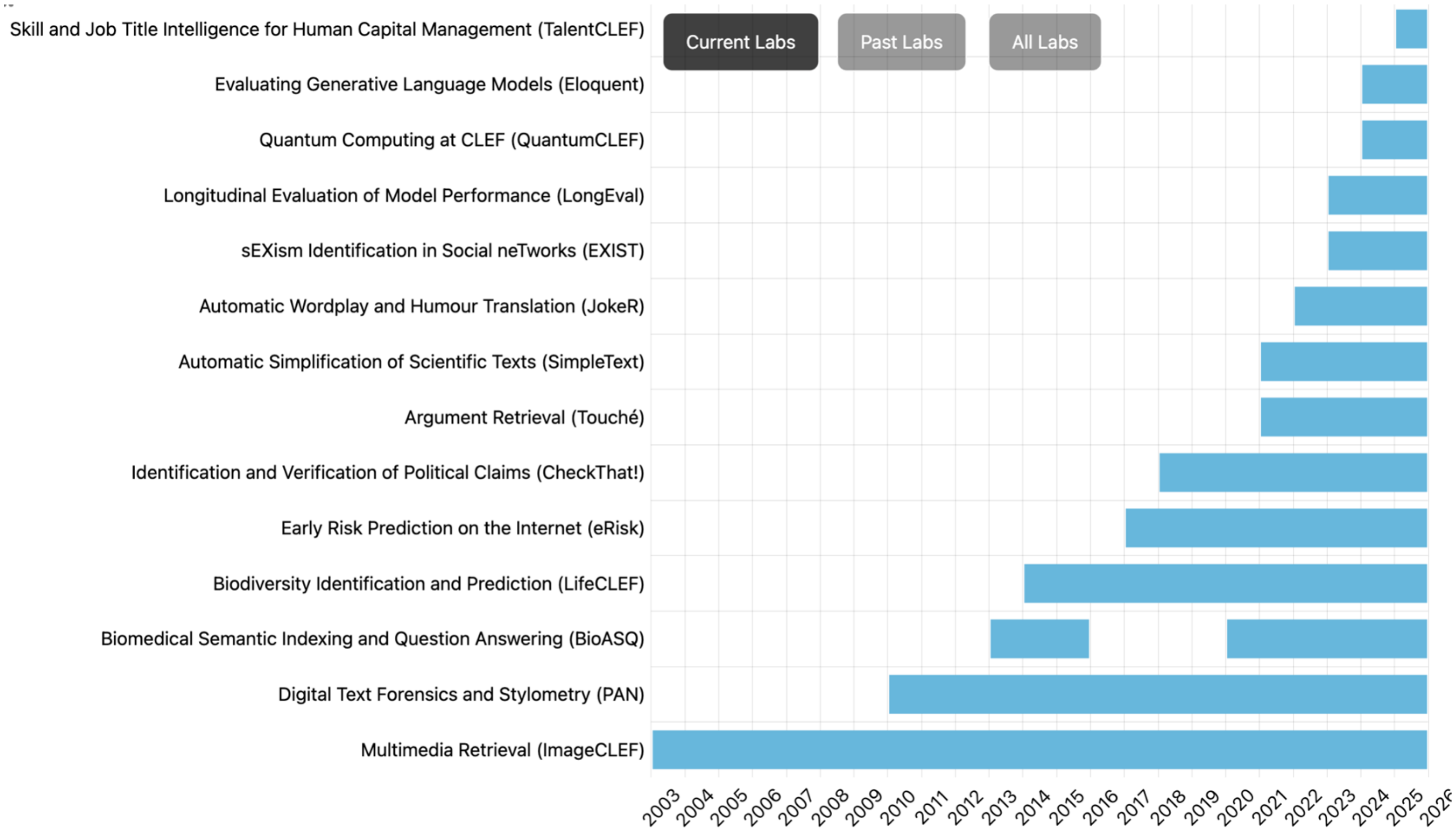
Anticipated timeline: runs due end of July.

Track coordinators:

Ronak Pradeep, University of Waterloo
Nandan Thakur, University of Waterloo
Jimmy Lin, University of Waterloo
Nick Craswell, Microsoft

Track Web Page: https://trec-rag.github.io/

Standard relevance benchmarks – CLEF



Precision and Recall

- **Precision**
  - fraction of retrieved docs that are relevant = P(relevant | retrieved)
  - Precision is used when probability that a positive result is correct is important
- **Recall**
  - fraction of relevant docs that are retrieved = P(retrieved | relevant)

	Relevant	Nonrelevant
Retrieved	tp	fp
Not Retrieved	fn	tn

Precision **P = tp/(tp + fp)**

Recall **R = tp/(tp + fn)**

Classification Errors

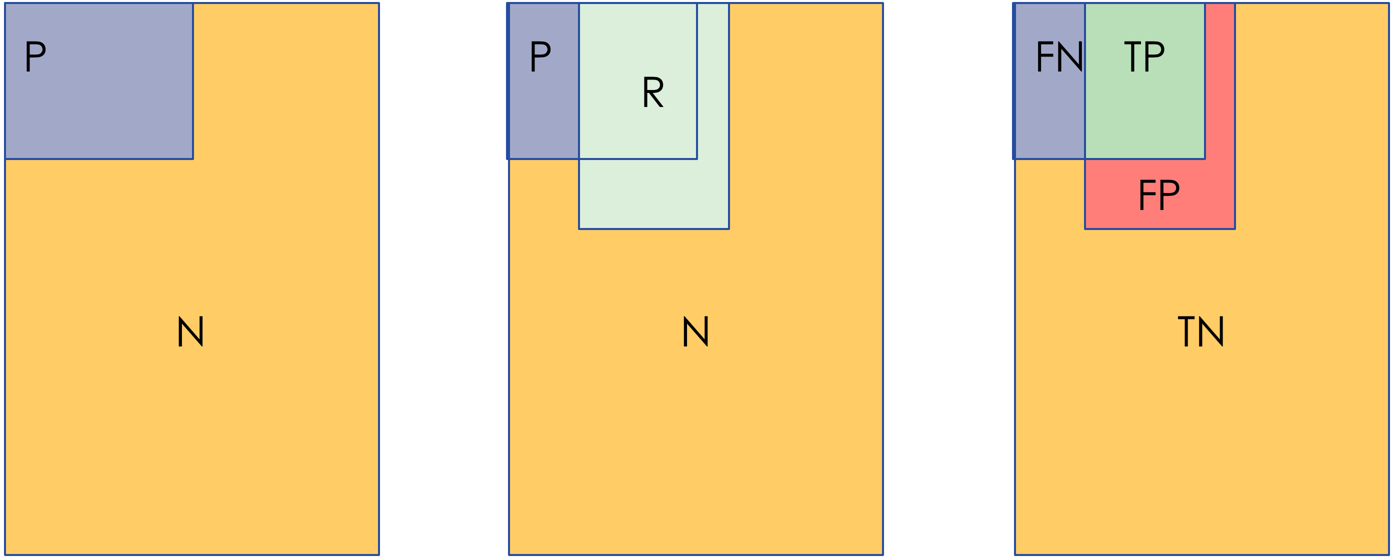
- False Positive (Type I error)
  - a non-relevant document is retrieved

Fallout = FPR = fp/(tn+fp)

- False Negative (Type II error)
  - a relevant document is not retrieved

fn/(tp+fn)
=
1- Recall
(= FNR = 1-TPR = 1-Sensitivity)

Precision and Recall



Precision P = tp/(tp + fp)
Recall R = tp/(tp + fn)

Binary classification measures

		Condition (as determined by "Gold standard")		
		Condition Positive	Condition Negative	Precision
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)	Positive predictive value = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Test Outcome Positive}}$
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**	Negative predictive value = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Test Outcome Negative}}$
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	**Specificity** = $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$	
		Recall		

Source: http://en.wikipedia.org/wiki/Sensitivity_and_specificity

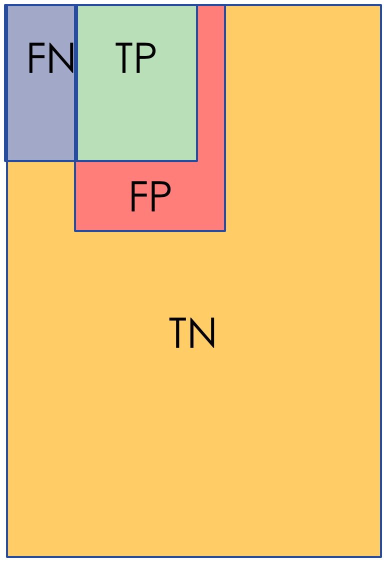
Accuracy

- Given a query, an engine classifies each doc as "Relevant" or "Nonrelevant"
- The accuracy of an engine: the fraction of these classifications that are correct
  - (tp + tn) / (tp + fp + fn + tn)
- Accuracy is a commonly used evaluation measure in machine learning classification work
- Why is this not a very useful evaluation measure in IR?

Why not just use accuracy?

- How to build a highly accurate search engine on a low budget?

- Typically :
  - (tp + tn) / (tp + fp + fn + tn)
  - tn >> tp



Precision/Recall

- We can get high recall (but low precision) by retrieving all docs for all queries!
- Recall is a non-decreasing function of the number of docs retrieved

- In a good system, precision decreases as either the number of docs retrieved or recall increases
  - This is not a theorem, but a result with strong empirical confirmation

- **Q**: How to adequately combine the Precision and Recall measures?

F Measure

- The harmonic mean of recall and precision (**F1**)

$$F_1 = \frac{1}{\frac{1}{2}\left(\frac{1}{R}+\frac{1}{P}\right)} = \frac{2RP}{(R+P)}$$

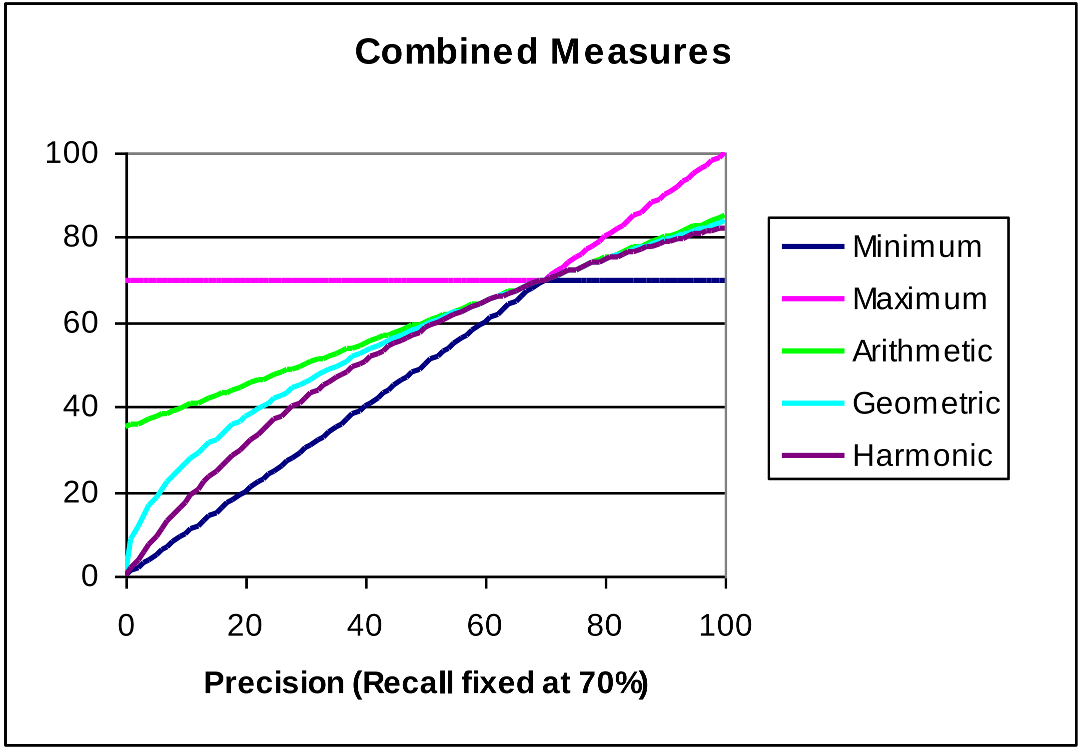
  – harmonic mean emphasizes the importance of small values, whereas the arithmetic mean is affected more by outliers that are unusually large

- More general form

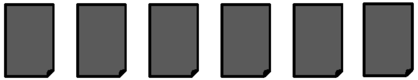
  – β is a parameter that determines relative importance of recall and precision

$$F_\beta = (\beta^2 + 1)RP/(R + \beta^2 P)$$

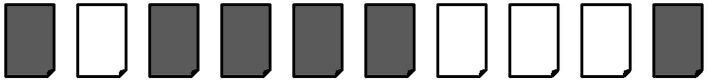
F1 and other averages



Ranking Effectiveness

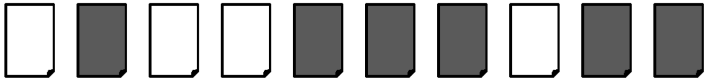
 = the relevant documents

Ranking #1



Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

Ranking #2



Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6

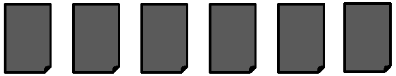
Summarizing a Ranking

- Calculating recall and precision at fixed rank positions

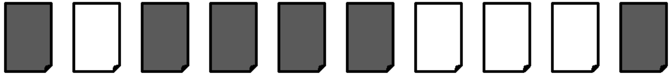
- Calculating precision at standard recall levels, from 0.0 to 1.0
  - requires *interpolation*

- Averaging the precision values from the rank positions where a relevant document was retrieved

Average Precision

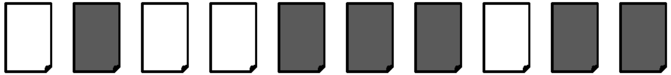
 = the relevant documents

Ranking #1



Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

Ranking #2



Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6

$$\text{Ranking } \#1\text{: } (1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6)/6 = 0.78$$

$$\text{Ranking } \#2\text{: } (0.5 + 0.4 + 0.5 + 0.57 + 0.56 + 0.6)/6 = 0.52$$

MAP – Averaging Across Queries

- **Mean Average Precision** (MAP)
  - summarize rankings from multiple queries by averaging the average precision
  - most commonly used measure in research papers
  - assumes the user is interested in finding many relevant documents for each query
  - requires many relevance judgments in a text collection

- Recall-precision graphs are also useful summaries

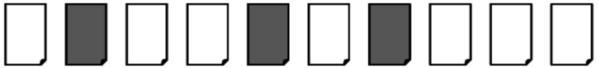
MAP – Averaging Across Queries

 = relevant documents for query 1

Ranking #1 

Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

 = relevant documents for query 2

Ranking #2 

Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

$$\begin{aligned} \textit{average precision query 1} &= (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62 \\ \textit{average precision query 2} &= (0.5 + 0.4 + 0.43)/3 = 0.44 \end{aligned}$$

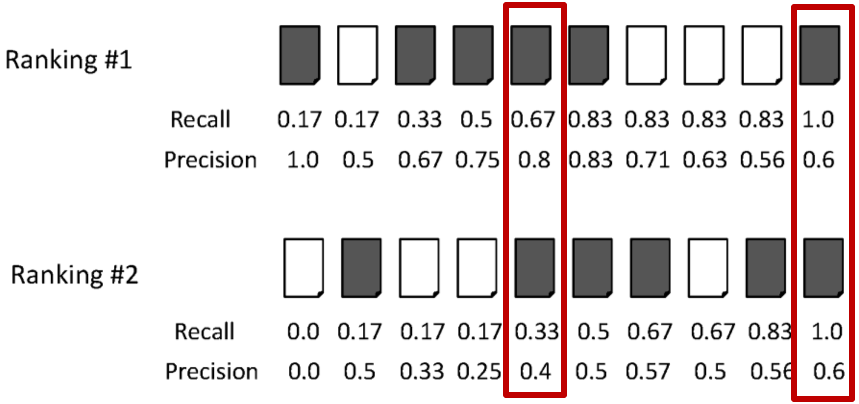
$$\textit{mean average precision} = (0.62 + 0.44)/2 = 0.53$$

Focusing on Top Documents

- Users tend to look at only the top part of the ranked result list to find relevant documents
- Some search tasks have only one relevant document
  - e.g., navigational search, question answering
- Recall not appropriate
  - instead need to measure how well the search engine does at retrieving relevant documents at very high ranks

Focusing on Top Documents

- Precision at Rank R
  - R typically 5, 10, 20
  - easy to compute, average, understand
  - not sensitive to rank positions less than R



- Reciprocal Rank (RR)
  - reciprocal of the rank at which the first relevant document is retrieved

$$\mathbf{d_p},\ d_n,\ \mathbf{d_p},\ \mathbf{d_p},\ \mathbf{d_p} \rightarrow (RR = 1)$$

$$d_n,\ \mathbf{d_p},\ d_n,\ d_n,\ \mathbf{d_p} \rightarrow (RR = 0{,}5)$$

  - very sensitive to rank position

MRR

- Mean Reciprocal Rank (MRR) is the average of the reciprocal ranks over a set of queries

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{Q} \frac{1}{\text{rank}_i}.$$

Query	Results	Correct response	Rank	Reciprocal rank
cat	catten, cati, **cats**	cats	3	1/3
torus	torii, **tori**, toruses	tori	2	1/2
virus	**viruses**, virii, viri	viruses	1	1

Discounted Cumulative Gain

- Popular measure for evaluating web search and related tasks
  - Uses *graded relevance* as a measure of the usefulness, or *gain,* from examining a document
- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant document
  - 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
- Gain is accumulated starting at the top of the ranking and may be reduced, or *discounted*, at lower ranks
- Typical discount is 1/*log (rank)*
  - With base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3

Discounted Cumulative Gain

- DCG is the total gain accumulated at a particular rank p:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

Example:

- 10 ranked documents judged on 0-3 relevance scale:

  3, 2, 3, 0, 0, 1, 2, 2, 3, 0

- discounted gain:

  3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0

  = 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0

- DCG (p = [1..10]):

  3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

Normalized DCG (nDCG)

- DCG numbers are averaged across a set of queries at specific rank values
  - e.g., DCG at rank 5 is 6.89 and at rank 10 is 9.61
- 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 Example

- Perfect ranking:
  3, 3, 3, 2, 2, 2, 1, 0, 0, 0
- ideal DCG values:
  3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88
- Real DCG values (previous example):
  3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61
- NDCG values (divide actual by ideal):
  1, 0.83, 0.87, 0.76, 0.71, 0.69, 0.73, 0.8, 0.88, 0.88
  - NDCG $\leq$ 1 at any rank position

This lecture

- How do we know if our results are any good?
  - Evaluating a search engine
    - Benchmarks
    - Precision and recall
- Results presentation:
  - Making our good results usable to a user

Result Summaries

- Having ranked the documents matching a query, we wish to present a results list
- Most commonly, a list of the document titles plus a short summary, aka “10 blue links”

Summaries

- The title is often automatically extracted from document metadata. What about the summaries?
  - This description is crucial.
  - User can identify good/relevant hits based on description.

- Two basic kinds:
  - A **static summary** of a document is always the same, regardless of the query that hit the doc
  - A **dynamic summary** is a *query-dependent* attempt to explain why the document was retrieved for the query at hand

Static summaries

- In typical systems, the static summary is a subset of the document
- Simplest heuristic: the first 50 (or so – this can be varied) words of the document
  - Summary cached at indexing time
- More sophisticated: extract from each document a set of “key” sentences
  - Simple NLP heuristics to score each sentence
  - Summary is made up of top-scoring sentences

Dynamic summaries

- Dynamic summaries are tailored in real-time to reflect the most relevant information based on user needs.

- Present one or more "windows" within the document that contain several of the query terms

Quicklinks

- For a navigational query such as ***camões instituto*** user's need is likely satisfied on instituto-camoes.pt/
- Quicklinks provide navigational cues on that home page

This lecture

- How do we know if our results are any good?
  - Benchmarks
  - Precision, Recall, F-measure, DCC, nDCC
- Results presentation:
  - Clear Layout: Organizing results in a visually appealing and easy-to-navigate format.
  - Summarization: Providing concise summaries or snippets of each result to help users quickly identify relevance.
  - Filtering Options: Allowing users to filter results by date, type, or relevance.
  - Ranking: Displaying the most relevant results first, based on the evaluation metrics.
  - User Feedback Mechanism: Implementing features for users to rate or provide feedback on results, helping to refine future searches.