11. Evaluation in Information Retrieval

Prof. Dr. Goran Glavaš

Center for AI and Data Science (CAIDAS)
Fakultät für Mathematik und Informatik
Universität Würzburg

After this lecture, you'll...

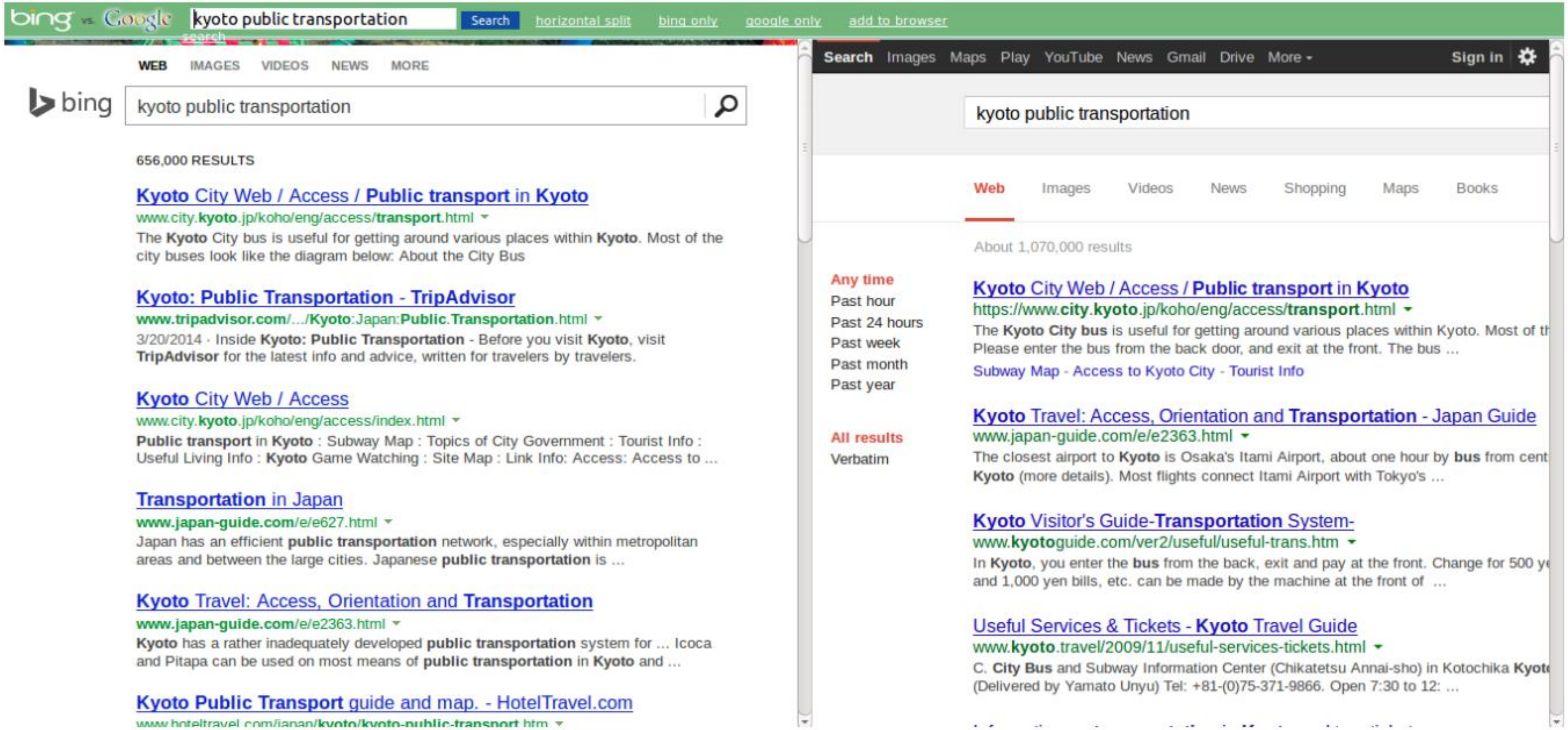
- Know different methods for evaluating IR systems
- Understand advantages and shortcomings of certain metrics
- Learn how to annotate relevance
- Understand what the pooling method is and how it is used in information retrieval evaluation

Outline

- Evaluation in IR
- Evaluation Metrics
- Relevance Judgements and Pooling

IR Evaluation

- Clash of the titans: which one is better?



IR Evaluation

- There are different aspects through which we can evaluate IR systems:
  1. **Retrieval effectiveness (standard IR evaluation)**
     - Relevance of search results
  2. **System quality**
     a) Indexing speed (e.g., how many documents per hour?)
     b) Search speed (search latency as a function of index size)
     c) Coverage (document collection size and diversity)
     d) Expresiveness of the query language
  3. **User utility**
     - User happiness based on relevance, speed, and user interface
     - User return rate, user productivity (difficult to measure)
     - A/B test: slight change on a deployed system visible to a fraction of users
       - Difference evaluated using clickthrough log analysis

Test collections in IR

- Each IR test collection is comprised of:
  1. Document collection
  2. Set of information needs (descriptions + queries)
     - A common requirement is to have at least 50 information needs
  3. Set of relevance judgements for each query-document pair
     - Binary relevance judgements (document relevant or non-relevant)
     - Graded relevance judgements (less common, more difficult for human annotators)
     - **Q:** Is it feasible to annotate all query-document pairs for relevance?
- Test collections are used for
  - Evaluating retrieval effectiveness w.r.t. different settings
    - Quantifying effects of e.g., different preprocessing methods, different ranking functions
  - Comparing performance against other systems (usually in evaluation campaigns)
  - Fine-tuning of system parameters, done on a development test collection

Test collections in IR

- Some standard test collections:
  - **Cranfield** – first IR test collection (from 1957)
    - 1,398 abstracts of aerodynamics journal articles
    - 225 queries, complete relevance judgements (1,398 x 225 annotations!)
  - **TREC collections** – NIST Text Retrieval Conferences (1992 – today)
    - Ad-hoc retrieval task: 1.89M docs, 450 inf. needs, incomplete rel. judg.
    - Many other tasks: blog track, cross-lingual track, QA track, ...
  - **CLEF collections** – Conference and Labs of the Evaluation Forum
    - Focus on European languages
    - Mono-lingual and cross-lingual ad-hoc retrieval tasks, QA tasks, ...

Outline

- Evaluation in IR
- Evaluation Metrics
- Relevance Judgements and Pooling

Evaluation metrics

- Compare retrieved documents against relevant documents
- Each document is either retrieved or not, and either relevant or not – this induces a 2x2 confusion matrix

$$\begin{array}{l c} & \begin{array}{cc} \text{relevant} & \text{not relevant} \end{array} \\ \begin{array}{l} \text{retrieved} \\ \text{not retrieved} \end{array} & \begin{pmatrix} tp & fp \\ fn & tn \end{pmatrix} \end{array}$$

- **Accuracy** is the fraction of correct decisions:

$$Acc = \frac{tp+tn}{tp+tn+fp+fn}$$

- **Q:** Is accuracy a good measure of performance of an IR system?
- **A:** No! For most queries, most documents (e.g., 98%) are irrelevant. A search engine that retrieves nothing will have accuracy of **98%** for all queries!

Precision, recall, and F-measure

- Irrelevant documents make most of collection → eliminate *true negatives*
- **Precision (P)** is a fraction of retrieved documents that are relevant

$$P = \frac{\#(relevant\ documents\ retrieved)}{\#(retrieved\ documents)} = \frac{tp}{tp+fp}$$

- **Recall (R)** is the fraction of relevant documents that are retrieved

$$R = \frac{\#(relevant\ documents\ retrieved)}{\#(relevant\ documents)} = \frac{tp}{tp+fn}$$

- **F-measure** combines precision and recall (weighted harmonic mean)

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P+R}; \quad \beta^2 = \frac{1-\alpha}{\alpha}$$

  - If P and R are equally important, we set β to 1
  - **Q:** What values for β would we use if precision is more important than recall?

Precision and recall – example

- For some query *q*, there are in total **4** relevant documents (**R**) documents in the collection, whereas all other documents are not relevant (**N**).

- Some IR system returns **6** documents for the query *q*:
  - **N**,
  - **R**,
  - **N**,
  - **R**,
  - **N**,
  - **N**

- Compute precision, recall, and F1-measure

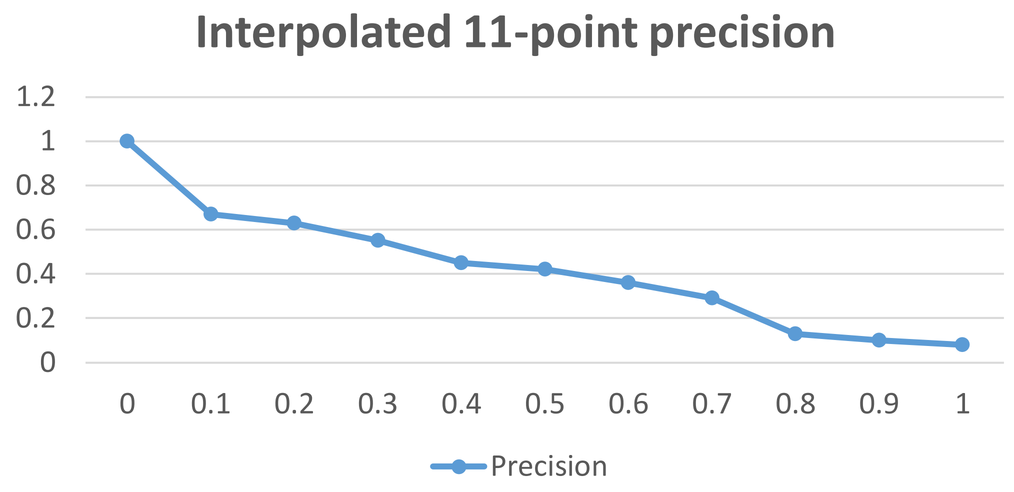
Evaluation of ranked results

- Precision, recall, and F-score are good for evaluating performance of Boolean retrieval systems, but they **cannot evaluate rankings**
  - According to P, R, AND F, ranking [**N**, **R**, **N**, **R**] is equally good as ranking [**R**, **R**, **N**, **N**]

- Most modern IR systems produce ranked results
- An ideal search engine ranks all relevant documents before all non-relevant
  - Evaluation metrics should take into account ranks of relevant documents
  - Rank-based metrics:
    - Precision-recall curve
    - 11-point precision
    - MAP
    - P@k
    - R-precision
    - nDCG

11-point precision

- **Interpolated 11-point precision** describes performance of an IR system through precision measured at 11 different levels of recall:
  - Measuring precision at ranks where recall is:
    - 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0
  - For each recall level, average precisions measured over different queries



Mean average precision

- We would like to have a single-figure measure of retrieval effectiveness across all recall levels
- **Average precision (AP)** for a query *q* with relevant documents {d_1, ..., d_m} is computed by averaging the precision scores measure at ranks of relevant docs:

$$\mathrm{AP}(q) = \frac{1}{m}\sum_{k=1}^{m} P(R_k)$$

- R_k is the rank at which we find the k-th relevant document
- **Mean average precision** is AP averaged over the set of queries ***Q***:

$$\mathrm{MAP} = \frac{1}{|Q|}\sum_{j=1}^{|Q|}\frac{1}{m_j}\sum_{k=1}^{m_j} P(R_{jk})$$

P@k and R-precision

- MAP takes into account all recall levels, even at very low ranks
  - This is inappropriate for web search:
    - Less than 6% users look at the second page of results

- **Precision at rank k (P@k)** is precision at the fixed rank k in the ranking (e.g., P@5, P@10, P@20)

- **R-precision** is the P@k where k equals to the number of relevant documents for the query
  - E.g., if there are 5 relevant documents for the query in total, then R-precision = P@5

Evaluation metrics – exercise

rank	r_1	r_2	r_3
1	d_1	d_1	d_1
2	d_2	d_2	d_2
3	d_5	d_4	d_4
4	d_6	d_5	d_5
5	d_{13}	d_6	d_9
6		d_7	d_{10}
7		d_8	d_{12}
8		d_9	d_{13}
9		d_{10}	d_{14}
10		d_{11}	d_{15}
11		d_{12}	d_{20}
12		d_{13}	
13		d_{19}	
14		d_{14}	
15		d_{17}	
16		d_3	
17		d_{15}	
18		d_{16}	
19		d_{18}	
20		d_{20}	

- You are given 3 different IR systems, r_1, r_2, and r_3, and their rankings of documents for some query q
- The collection contains 20 documents
  - Odd documents (d1, d3, ..., d19) are relevant fo the query q
  - Even documents (d2, ..., d20) are not relevant for q
- For each of the three systems compute:
  - Precision, recall, and F1-measure
  - Average Precision
  - P@4, P@7, P@12
  - R-precision

Normalized Discounted Cumulative Gain (nDCG)

- All methods so far assumed that we have binary relevance annotations
- Sometimes we have graded relevance annotations
  - E.g., from 1 (marginally relevant) to 5 (highly relevant).
- Assumptions (in order to maximize nDCG)
  - **Highly relevant** documents are more useful than **marginally relevant** documents
  - **Marginally relevant** documents are more useful than **irrelevant documents**
  - The higher the relevance of the document, the higher it should appear in the relevance ranking

- **(Normalized Discounted) Cumulative Gain** takes into account the graded relevances of documents when evaluating the ranking produced by IR systems

Normalized Discounted Cumulative Gain (nDCG)

- First try: Cumulative Gain
  - Let rel_i be the (true) relevance score of the document ranked at position i by the system
  - Cumulative gain at rank k, CG(k) is then simply

$$CG(k) = \sum_{i=1}^{k} rel_i$$

  - **Q:** What is the issue with using only CG(k) as defined above?
  - **A:** Similar as using standard precision, recall, and F1 for binary relevances – ranks at which different scores appear are not taked into account
    - Rankings: [0, 2, 4, 0, 1] and [4, 2, 1, 0, 0] will be considered equally good by CG(5)

Normalized Discounted Cumulative Gain (nDCG)

- **Discounted Cumulative Gain**
  - Idea: Normalize the relevance scores of documents at every position with the position itself
  - That way, highly relevant but low-ranked documents contribute less to the overall score, i.e., they get penalized more

$$\mathrm{DCG(k)} = \sum_{i=1}^{k} \frac{rel_i}{\log_2(i+1)}$$

  - There is an alternative formulation of DCG, that places stronger emphasis on retrieving relevant documents (and a bit less on their mutual relative ranking)

$$\mathrm{DCG(k)} = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

Normalized Discounted Cumulative Gain (nDCG)

- Different queries generally have different numbers of relevant documents
- So, the DCG scores will generally be higher for queries that have more relevant documents (and with higher relevance scores)
- To average DCG scores across different queries, we need to first **normalize** them
- **Ideal DCG (IDCG)** is the maximal DCG score any ranking can have

$$\mathrm{IDCG(k)} = \sum_{i=1}^{|relevant|} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

- Normalized nDCG is the DCG(k) score normalized with the IDCG(k), where k is the total number of relevant documents

$$nDCG = \frac{DCG(k)}{IDCG(k)}$$

- nDCG applied to binary scores (0 and 1) perfectly correlates with (M)AP

Outline

- Evaluation in IR
- Evaluation Metrics
- Relevance Judgements and Pooling

Pooling

- Annotating complete relevance judgements for larger test collections is infeasible
  - Collection of 1000 documents and 50 queries requires **50000** relevance annotations
  - It is feasible to annotate only a small subset of relevance judgements

- Luckily, for most queries, only a tiny fraction of all documents are relevant
  - Say that, on average, we expect N relevant documents per query in our collection
  - An ideal retrieval system would rank relevant documents on top positions

- **Idea:** Let's annotate for relevance only the top N results of the IR system's ranking
  - This requires only N (<< number of documents) annotations per query
  - **Shortcoming**: a real system will not rank all relevant documents on top, thus we will ignore (i.e., we will loose) some relevant documents when evaluating real IR systems

Pooling

- In most IR evaluations, we are comparing the performance of different models (or different variants of the same model)
- **Pooling** is a method for reducing the number of required relevance judgement annotations in settings where we compare different IR models
- Example: evaluating models $r_1, \ldots, r_K$ (expected N relevant docs for query q)
- Pooling involves the following steps:
  1. Rank all documents with each of the models $r_1, \ldots, r_K$
  2. In each of the rankings $R_1, \ldots, R_K$, take only the top N results: $R_{1,N}, \ldots, R_{K,N}$
  3. The documents in the union of retrieved top results are to be annotated for relevance for the given query: $R_{1,N} \cup \ldots \cup R_{K,N}$

- **Q:** Is it still possible to ignore some truly relevant document for relevance judgements? If so, is that a problem?

Now you...

- Know different methods for evaluating IR systems
- Understand advantages and shortcomings of certain metrics
- Learn how to annotate relevance
- Understand what the pooling method is and how it is leveraged in information retrieval evaluation