Crash Course IR – Evaluation

Sebastian Hofstätter

sebastian.hofstaetter@tuwien.ac.at
/s_hofstaetter

Today

Crash Course IR – Evaluation

1. Setup
   - Evaluating ranked results
2. Binary Relevance Metrics
   - MRR & MAP
3. Graded Relevance Metrics
   - nDCG

Evaluation

- We evaluate systems to observe concrete evidence for a hypothesis
  - Is our system better than the other one?

- IR systems are hard to evaluate
  - Ambiguity – what is relevant? In which context? Humans differ a lot …
  - Collection size – explosion of query-document pairs

- Different types of result quality evaluation:
  - **Offline**: Fixed set: same collection, query set & labels
  - **Online**: Observe behavior of users (in production system)*

* Could also be a user study, beta version, etc…

Online Evaluation

- In case you have >thousands of users (Who doesn't?)
- You can run A/B test
  - A portion of users is presented with search results from the test system
- Number of concurrent tests explodes in large-scale settings
  - Complex infrastructure needed to handle changes in the pipeline
- Philosophical question: is it ok to degrade performance for some users, to be able to improve service over the long term
  - Not always possible, for example if all users pay you for your service

See also: (interesting and easy to read) Kohavi, Ron, et al. "Online controlled experiments at large scale." *Proc. of SIGKDD* 2013. https://dl.acm.org/doi/10.1145/2487575.2488217

The World of Evaluation

- Today we focus on evaluating the result quality of our own IR system
  - Does a document contain the answer for our query?
- Many other possibilities:
  - Efficiency
    - How fast can we index, return results for a query, how large becomes our index on disk?
  - Fairness, diversity, content quality, source credibility, effort, …
  - Retrieval in the context of a larger goal
    - How many products, services do we sell through search
    - How well does our website integrate with Google, Bing, etc.. (SEO)
      - Optimizing a Blackbox

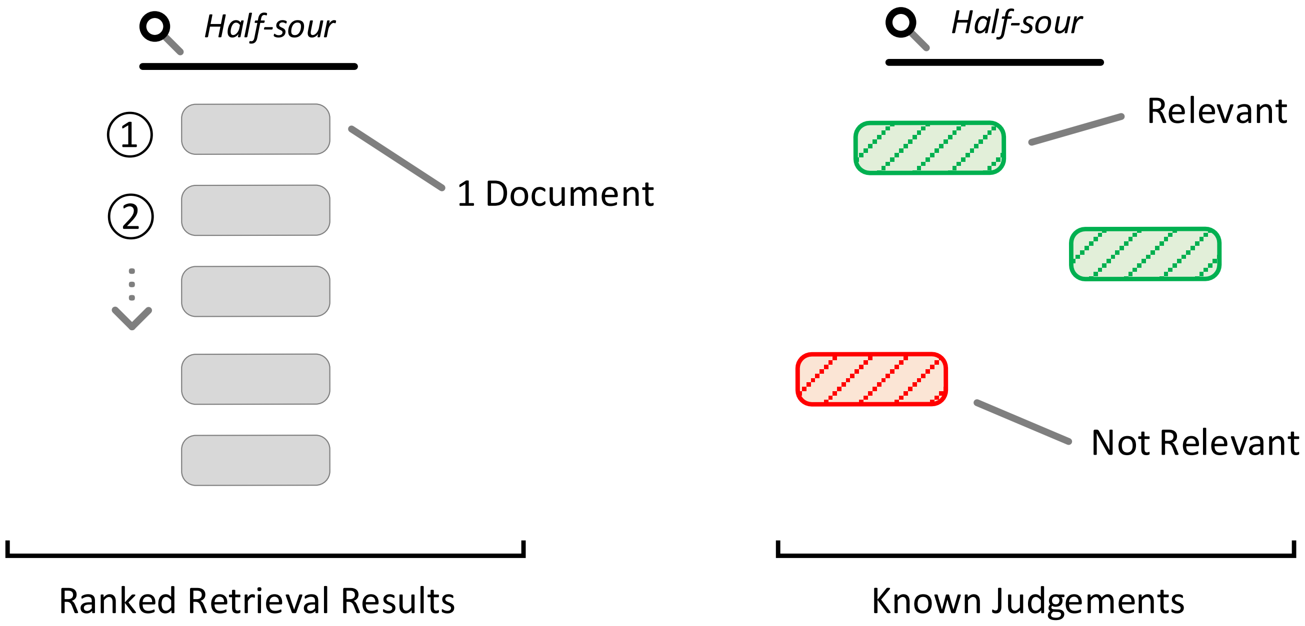
If you are interested in efficiency,
here are tips and tricks for efficient text processing: [Lecture on GitHub]

Offline Evaluation Setup

Systematically compare retrieval models

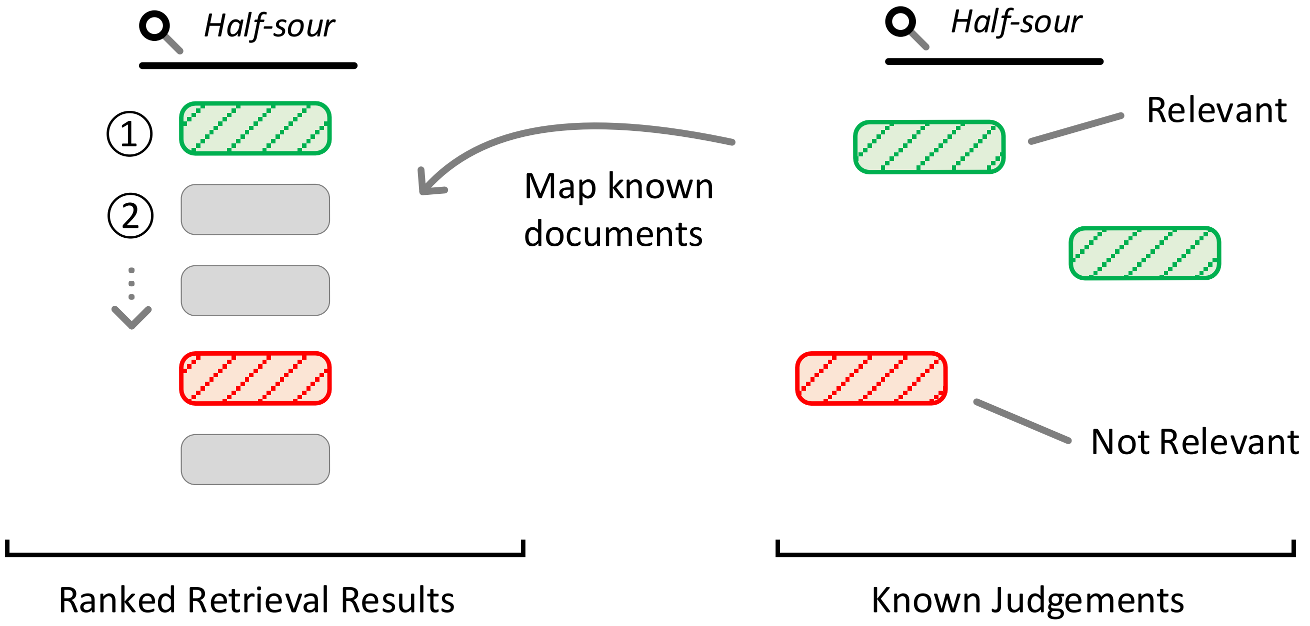
Offline Evaluation Setup

- Quality of systems, that produce ranked list of documents
- Compared by a pool of judgements (does not necessarily cover the whole list)
  - Missing judgements are often considered as non-relevant



Offline Evaluation Setup

- Quality of systems, that produce ranked list of documents
- Compared by a pool of judgements (does not necessarily cover the whole list)
  - Missing judgements are often considered as non-relevant



Test Collection

- Offline evaluation with a fixed *test collection*
  - Fixed set of documents
  - Fixed set of queries
  - Fixed set of judgements (does not cover all query-doc combinations)

- Query Source:
  - Handcrafted queries for a set of documents (Many TREC* collections)
  - Sampled queries from actual users (MS MARCO)

*TREC is an annual evaluation campaign/conference organized by NIST

More on how to create test collections in the data lecture

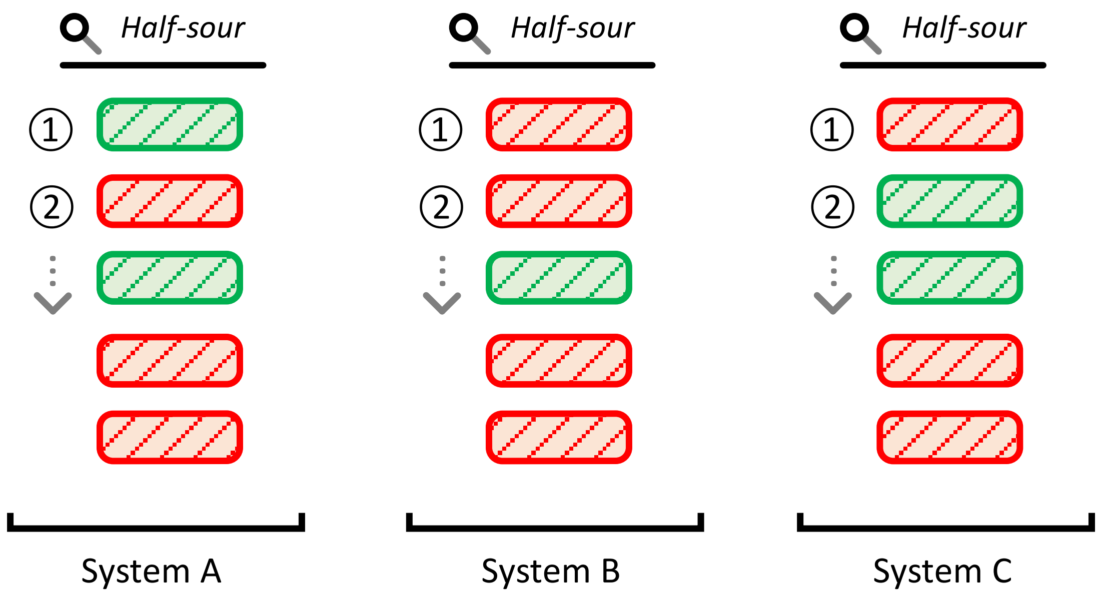
Sources & Types of Judgements

- Different sources of judgements
  - Automatic click data
  - Manual annotation

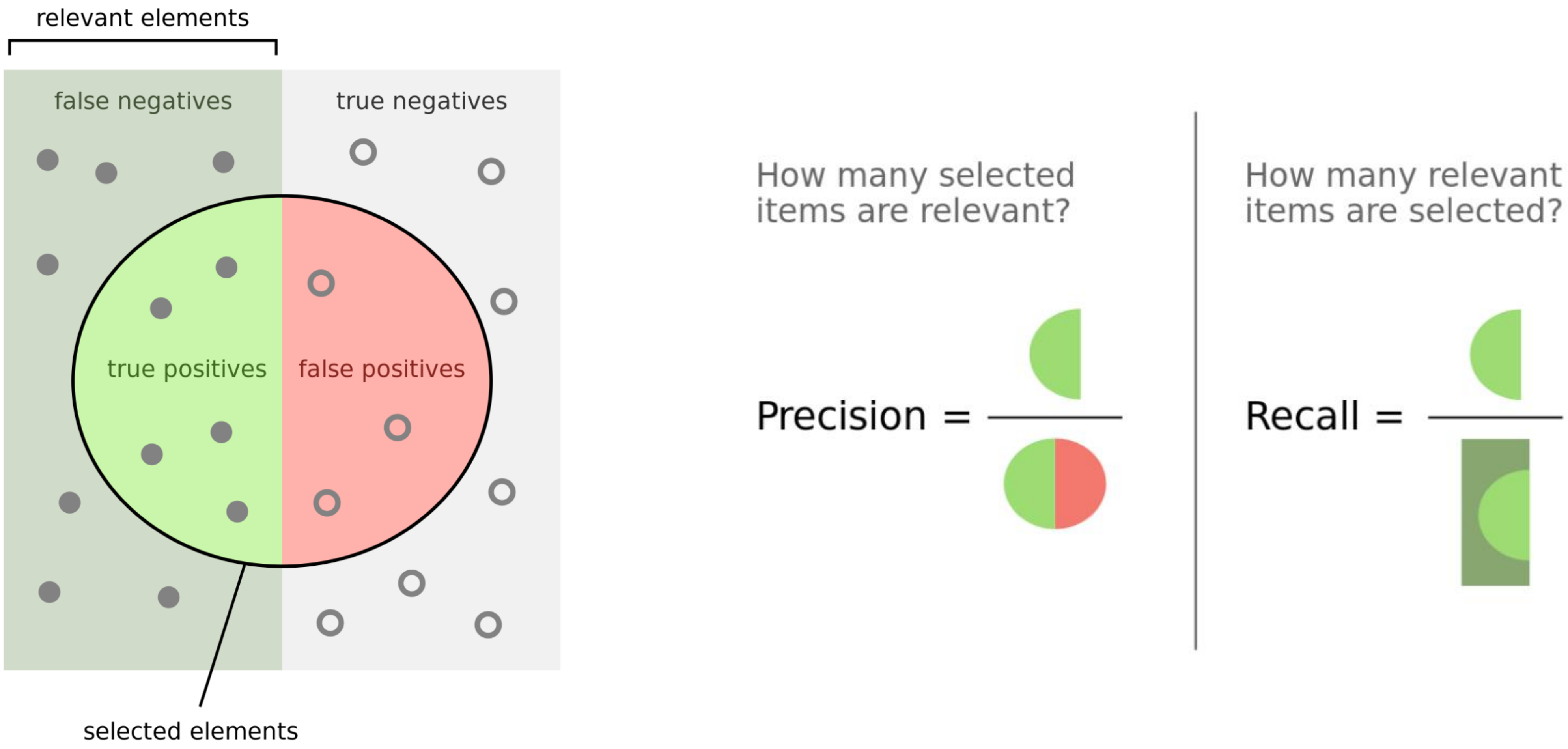
- Different types of granularity
  - **Sparse** judgements
    - ~1 judged (as relevant) per query
    - Makes sense for thousands of queries
    - Quite noisy, but covers many terms
    - MSMARCO Training & DEV sets
  - **Dense** judgements
    - >100 judged per query
    - Makes sense for 50+ queries already (better 200+)
    - Many TREC collections

Comparing Systems

- We have multiple IR systems running on the same documents & same query
- How to compare them? Evaluation metrics to the rescue!



Precision & Recall



From: Wikipedia https://en.wikipedia.org/wiki/Precision_and_recall

Evaluating Recall of Search Engines

- Recall depends on the judgement of “all” relevant items
- What happens if we don’t know that there is a relevant item
  - Test collections depend on pre-selection of candidate documents
  - Relevant items might be missing for example because of vocabulary mismatch
- Either we actively work on this problem with iterative annotation cycles
  - + potential active learning -> HiCAL is an annotation system that integrates it
- Or: If our test collection is not prepared for high recall, we should at least be aware of its limitations when interpreting results

The HiCAL tool https://hical.github.io/index.html

Ranking List Evaluation Metrics

- Binary labels
  - **MRR:** Mean Reciprocal Rank
  - **MAP**: Mean Average Precision
- Graded labels
  - **nDCG**: normalized Discounted Cumulative Gain

- Typically we measure at a cutoff @k of the top retrieved documents
  - MAP, Recall: @100, @1000
  - Precision, MRR, nDCG: @5, @10, @20

Some nice explanations: https://medium.com/swlh/rank-aware-recsys-evaluation-metrics-5191bba16832

Binary Relevance Metrics

MRR & MAP

MRR: Mean Reciprocal Rank

Users look at results from the top; gets annoyed pretty fast; stops once they found the first relevant; doesn't care about the rest

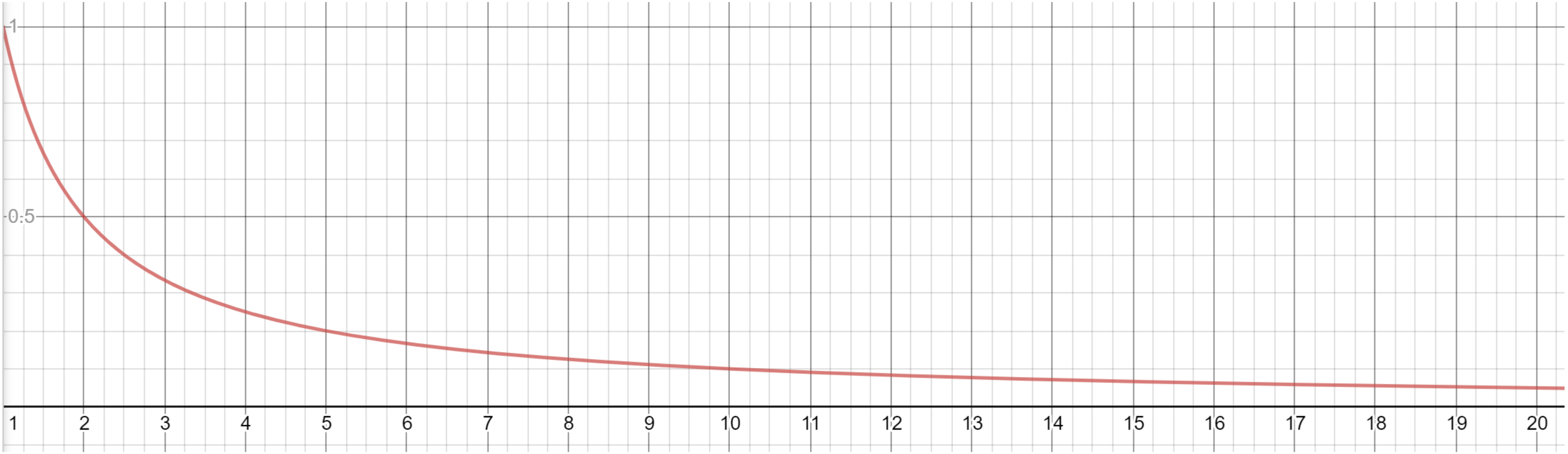
$$MRR(Q) = \overbrace{\frac{1}{|Q|} * \sum_{q \in Q}}^{\text{Mean over all queries}} \underbrace{\frac{1}{FirstRank(q)}}_{\text{Reciprocal Rank}}$$

- MRR puts the focus on the first relevant document
- Applicable with sparse judgements or assuming users are satisfied with one relevant document

Q	$\|Q\|$	$FirstRank(q)$
Query Set	Number of Queries	Returns the Rank of the first relevant document for 1 query

MRR: The Reciprocal Rank

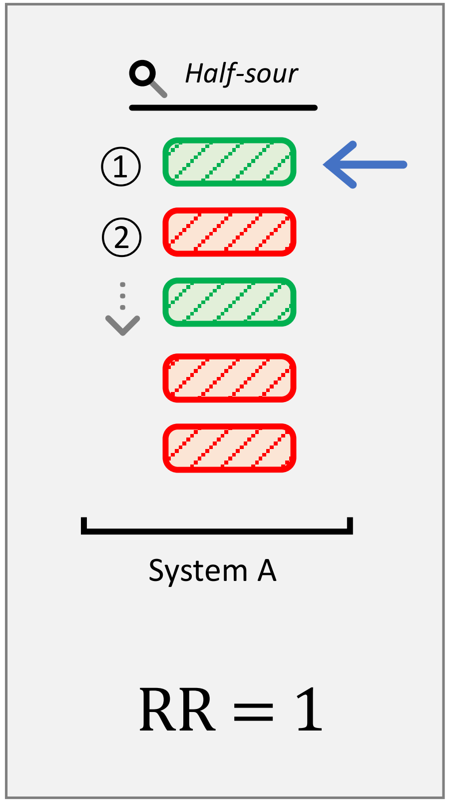
- Reciprocal Rank: $\frac{1}{x}$
- Very strongly emphasis the first position

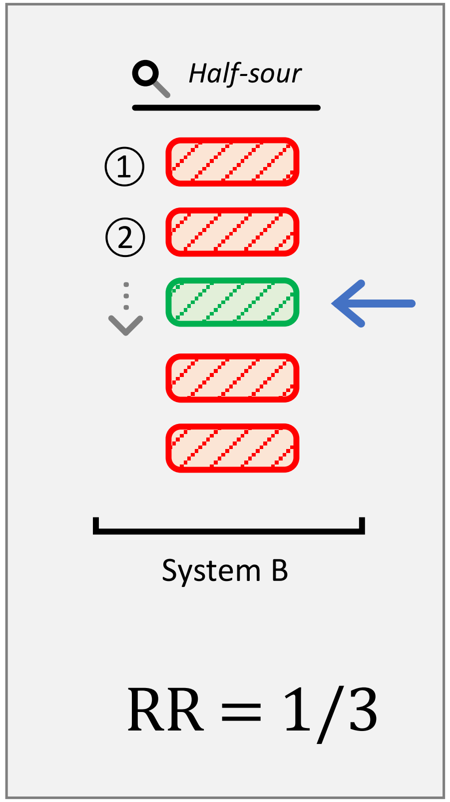


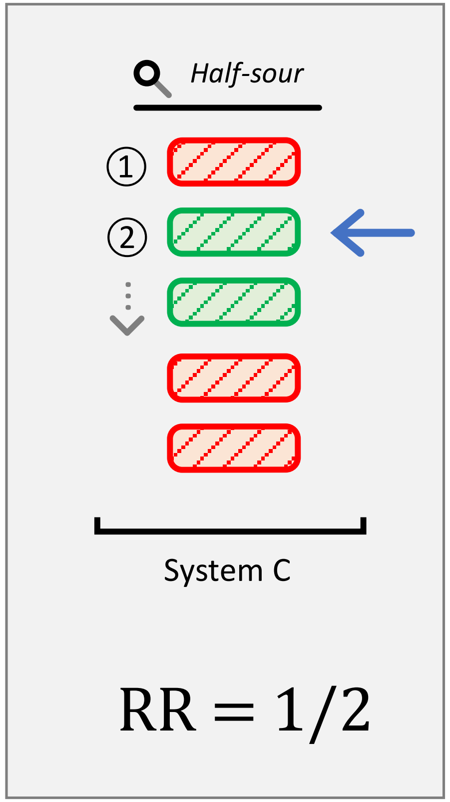
* x is plotted continuously, but in MRR x is discrete with the position in step size of 1

MRR: An Example

- Example for Reciprocal Rank:







MAP: Mean Average Precision

Users look at results closely, every time they find a new relevant document, they look at the full picture of what has been before

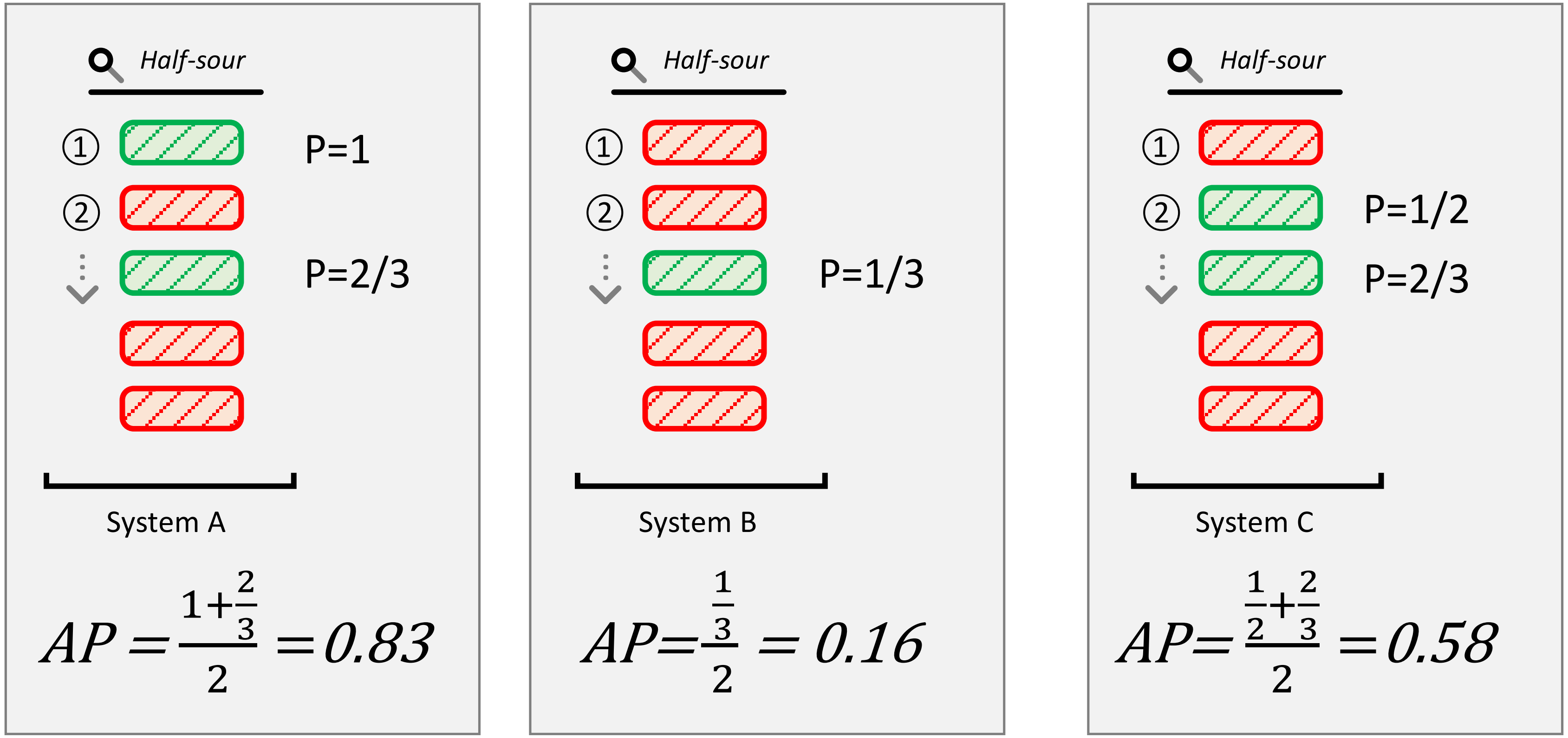
$$MAP(Q) = \underbrace{\frac{1}{|Q|}}_{\text{Mean over all queries}} * \sum_{q \in Q} \underbrace{\frac{\overbrace{\sum_{i=1}^{k} P(q)_{@i} * rel(q)_i}^{\text{Precision per relevant doc}}}{|rel(q)|}}_{\text{Average Precision}}$$

- MAP squeezes complex evaluation into a single number
- Hard to interpret
- MAP corresponds to the area under the *Precision-Recall curve*

Q	$\|Q\|$	$P(q)_{@i}$	$rel(q)_i$	$\|rel(q)\|$
Query Set	Number of Queries	Precision of query q after first i documents	Binary Relevance of doc at position i	Number of relevant documents

MAP: Mean Average Precision

- Example for Average Precision (2 relevant docs)
  - Mean is then calculated for multiple queries, for each system



Graded Relevance Metrics

nDCG

Graded Relevance

- Previous metrics all use binary relevance labels
  - Simple enough or too simple?
- Major problem: Of course there can be a difference in importance of relevance
  - Binary labels can not distinguish
- Graded relevance allows to assign different values of relevance
  - Can be floating point or fixed set of classes for manual annotation
  - Fixed set of classes for manual annotation
  - Floating point can be used when relevance inferred from logs

Common Graded TREC Relevance Labels

[3] Perfectly relevant: Document is dedicated to the query, it is worthy of being a top result in a search engine.

[2] Highly relevant: The content of this document provides substantial information on the query.

[1] Relevant: Document provides some information relevant to the query, which may be minimal.

[0] Irrelevant: Document does not provide any useful information about the query

nDCG: normalized Discounted Cumulative Gain

Users take for each document the relevance grade and position into account, normalize by best possible ranking per query

$$DCG(D) = \sum_{d \in D,\, i=1} \frac{rel(d)}{log_2(i+1)}$$

$$nDCG(Q) = \frac{1}{|Q|} * \sum_{q \in Q} \frac{DCG(q)}{DCG(sorted(rel(q)))}$$

- nDCG compares actual results with maximum per query
- Relevance is graded
- nDCG@10 most commonly used in modern offline web search evaluation

Q	$\lvert Q \rvert$	D	$rel(d)$	$rel(q)$	$sorted()$
Query Set	# of Queries	Single Doc. Result list	Relevance grade for single query-doc pair	List of all relevance grades for a query	Return graded documents by descending relevance

nDCG: A Closer Look

Discounted cumulative gain

$$DCG(D) = \sum_{d \in D,\, i=1} \frac{rel(d)}{log_2(i+1)}$$

Gain (relevance value, commonly 0 -> 3)

Position Discounting

$$nDCG(Q) = \frac{1}{|Q|} * \sum_{q \in Q} \frac{DCG(q)}{DCG(sorted(rel(q)))}$$

Actual Results

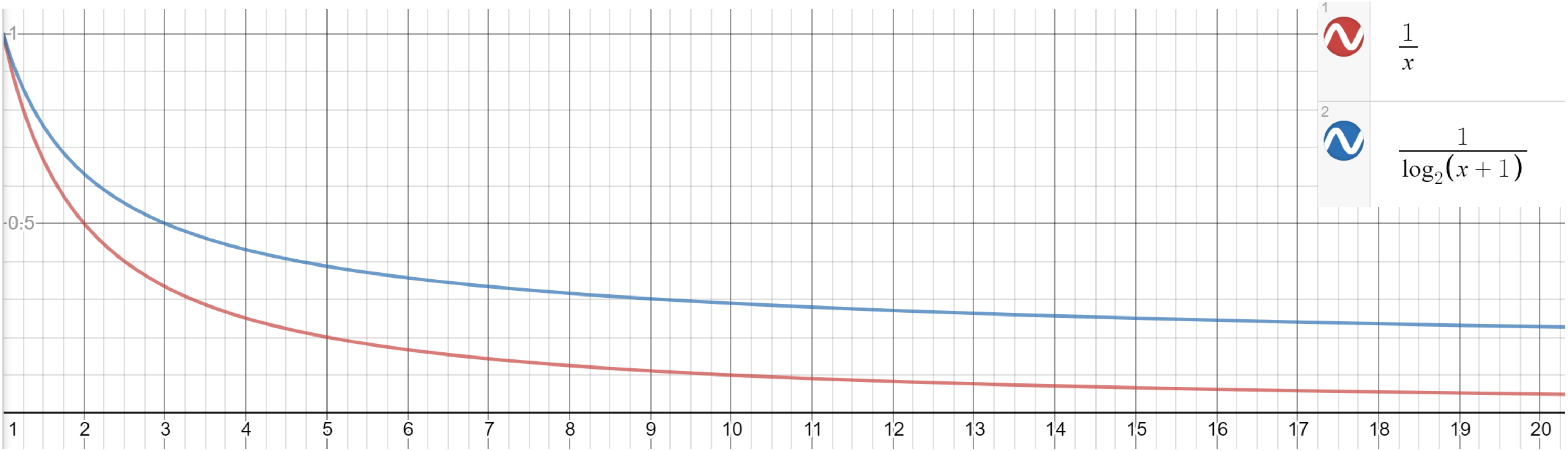
Best possible sorting (ground truth)

Mean over all queries

Q	$|Q|$	D	$rel(d)$	$rel(q)$	$sorted()$
Query Set	# of Queries	Single Doc. Result list	Relevance grade for single query-doc pair	List of all relevance grades for a query	Return graded documents by descending relevance

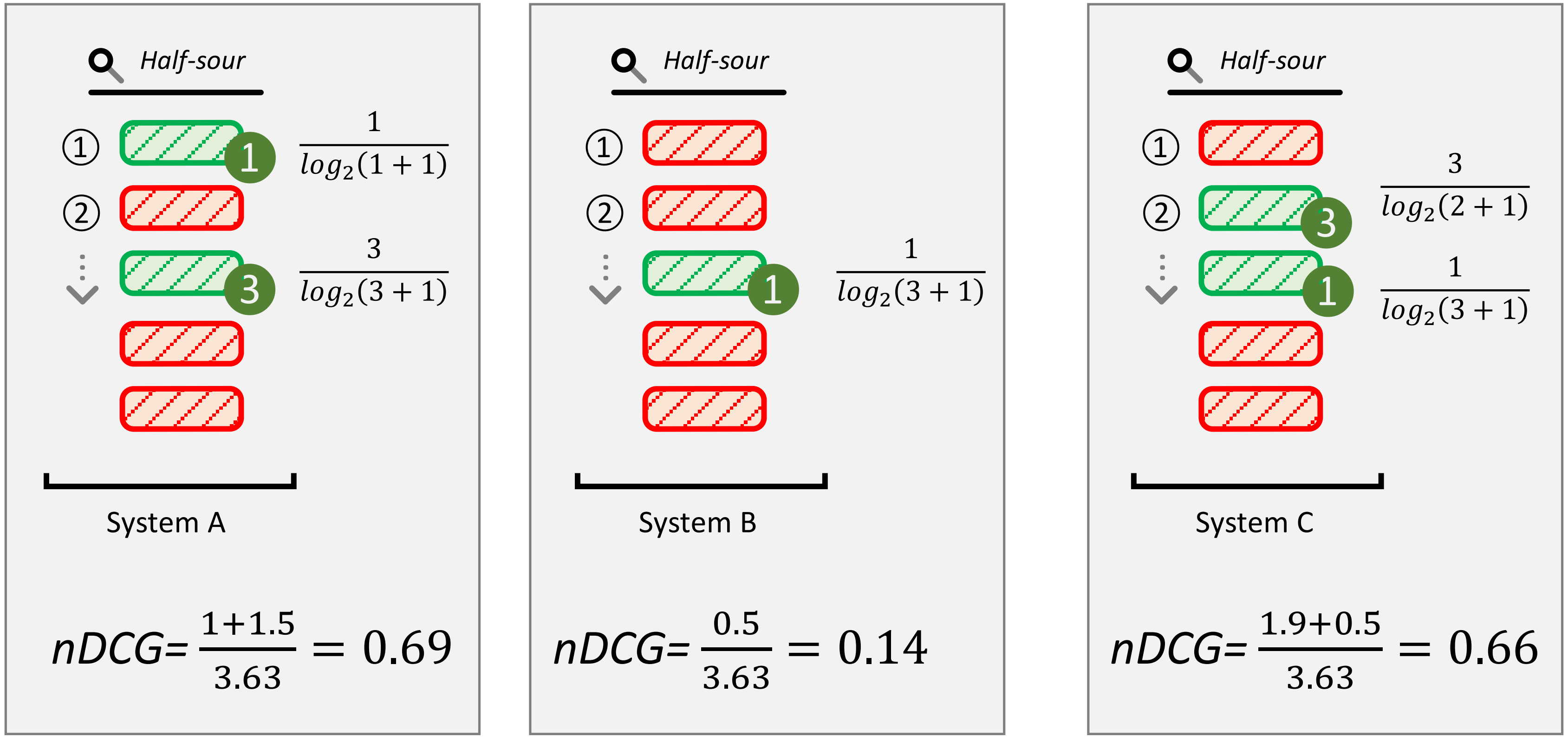
nDCG: Position Discounting

- Comparing the document position discount with reciprocal rank
  - Only for binary case rel=1
- nDCG discounts less than MRR



nDCG: Example

- Assuming two differently relevant docs (rel = 3 & 1)
  - Ideal DCG = $\frac{3}{log_2(1+1)} + \frac{1}{log_2(2+1)} = 3.63$



Bonus: Confidence in the Evaluation

Statistical Significance

Statistical Significance

- Giving meaning to the phrase *significantly different*
- We want to test whether two systems produce different rankings, that are not different by chance
  - Our hypothesis is that the systems are the same
- To test the stat. significance we compare results on a per query basis
  - Comparing AP, RR, and nDCG per query
- Common statistical test include
  - Paired: e.g., Paired Student's t-test, Wilcoxson signed-rank test, etc.
  - Non-paired: Student's t-test, Mann-Whitney U test, etc.
- We set a significance level (p-value) of how confident we are to reject our hypothesis

Statistical Significance

- The more queries we have the better does the stat. significance test work
  - Minimum of 50 queries
- And we need to be aware of the multiple testing problem
  - We often compare multiple baselines & model instances
  - If we test every model combination for statistical significance of multiple metrics – we run into cases that are significant by chance
  - A solution: Bonferroni correction -> divide the p-value by the number of comparisons
  - Best explanation: https://xkcd.com/882/ (Yes, it's the green jellybeans comic)

More information on stat. significance: https://www.ucl.ac.uk/child-health/short-courses-events/about-statistical-courses/research-methods-and-statistics/chapter-6

Evaluating Non-Deterministic Models

- Traditional retrieval (e.g. Inverted index with BM25) is deterministic
  - The same input produces the same output (also with tuned parameters)
- Every neural network that is initialized with random variables produces different outputs on different initializations
  - Could provide significant differences just by chance
- Possible solution is to run a model architecture multiple (i.e. 10x) times with different initializations and report a mean result value
  - However, this is very resource intensive
  - Next best option is to at least fix the randomness for all your experiments (!)

Very interesting further reading: **The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks**
https://arxiv.org/abs/1803.03635

Summary: Evaluation

1. We compare systems with a set of query and document relevance labels

2. Binary metrics (MRR & MAP) are a solid foundation for evaluation

3. Graded relevance allows for more fine-grained metrics (nDCG)

1. We compare systems with a set of query and document relevance labels
2. Binary metrics (MRR & MAP) are a solid foundation for evaluation
3. Graded relevance allows for more fine-grained metrics (nDCG)

Thank You