Cluster of Excellence
The Politics of Inequality

Steinbeis Transfer Center
Linguistic Data Analysis

UNIVERSITÄT PASSAU

Information Retrieval & Natural Language Processing
Week 6: Evaluation in IR

Annette Hautli-Janisz, Prof. Dr.

5 December 2024

IR & NLP: Course schedule winter 2024/25

	When?	What?
Week 1	17 October 2024	Introduction
Week 2	24 October 2024	Indexing, Boolean IR
Week 3	31 October 2024	--
Week 4	7 November 2024	--
Week 5	14 November 2024	Scoring, term weighting, the vector space model
Week 6	21 November 2024	Relevance and probabilistic IR
Week 7	28 November 2024	Tolerant retrieval and index compression
Week 8	5 December 2024	Evaluation in IR
Week 9	12 December 2024	Distributed word representations for IR

IR & NLP: Course schedule winter 2024/25

	When?	What?
Week 10	19 December 2024	Natural Language Processing
Week 11	9 January 2025	NLP with Python
Week 12	16 January 2025	Personalization in IR systems
Week 13	23 January 2025	AI & ethics
Week 14	30 January 2025	Recap
Week 15	6 February 2025	No class (conference trip)

(also on stud.IP)

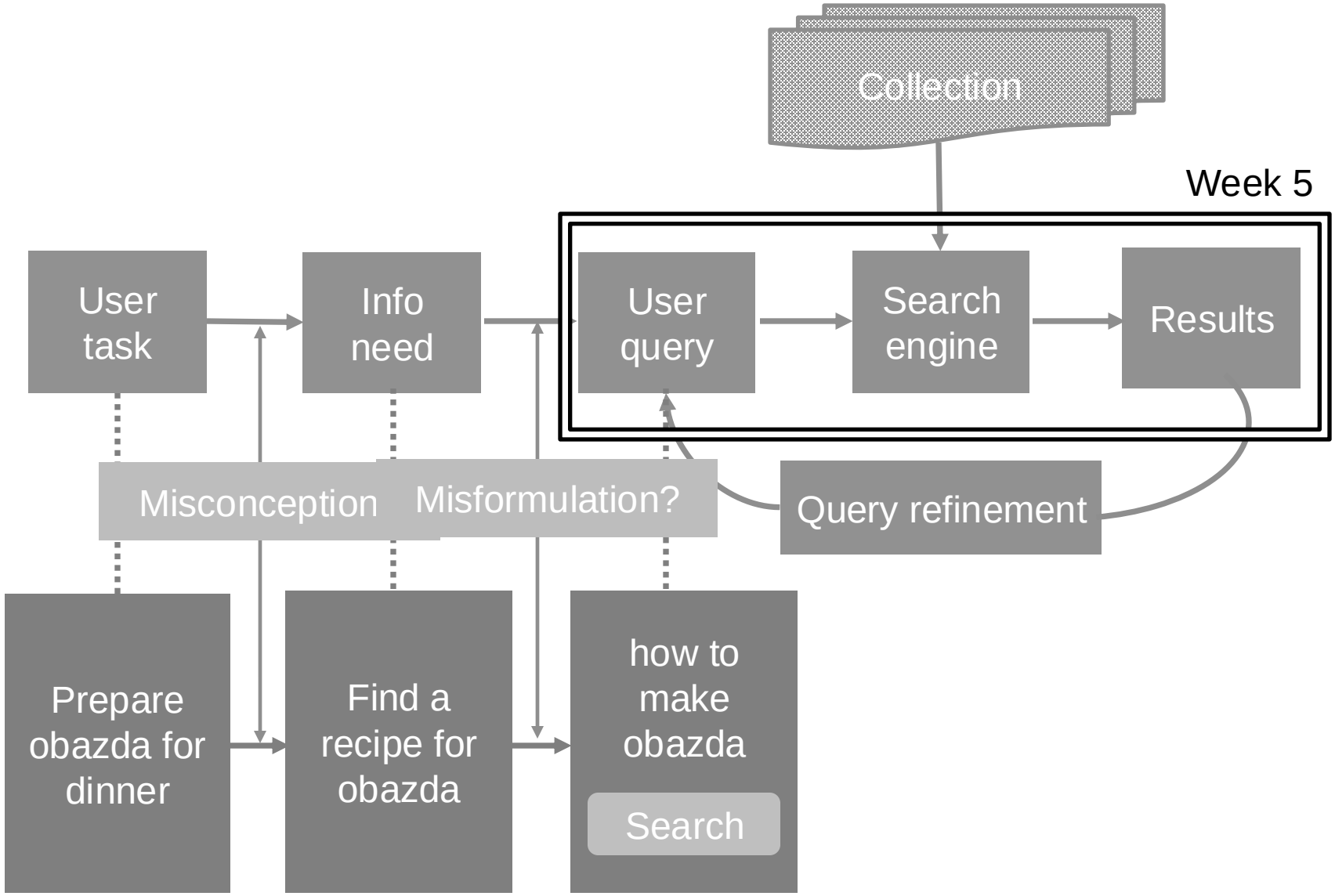
Registration to exam now possible!

Information Retrieval (IR): Today

(IIR): Manning, Raghavan and Schütze, Introduction to IR, 2008

Chapter 8: Evaluation in information retrieval

The classic search model



How can you tell if users are happy?

Search returns products relevant to users. But how do you assess this at scale?

Search results get clicked a lot

- Misleading titles/summaries can cause users to click.
- Vaguely relevant documents, user browses.

Users buy after using the search engine (or spend a lot of money after using the search engine)

Repeat visitors/buyers

- Do they leave soon after searching?
- Do they come back within a week/month/…?

Happiness: elusive to measure

Most common proxy: relevance of search results.

Pioneer: Cyril Cleverdon in the Cranfield experiments.

How exactly do we measure relevance?

Measuring relevance

Three elements:

- A benchmark document collection.
- A benchmark suite of queries.
- An assessment of either ‘relevant’ or ‘non-relevant’ for each query and each document.

Measuring relevance

In the case of an online retailer:

- Benchmark documents: the retailer's products
- Benchmark query suite: more on this
- Judgements of document relevance for each query

Relevance judgements

Binary (relevant versus non-relevant) in the simplest case. More nuanced relevance levels are also used.

What are some issues already?

The online retailer: 5 million product x 50k queries takes us into a range of a quarter trillion judgements.

- If each judgment took a human 2.5 seconds, we'd still need 10^{11} seconds, or nearly $300 million if you pay people $10 per hour to assess
- 10K new products per day

Relevance judgements

Crowdsource them?

Present query-document pairs to low-cost labor on online crowdsourcing platforms (Amazon Mechanical Turk, Prolific).

Let’s hope that it is cheaper!

A lot of literature on using crowdsourcing for such tasks.

In general: fairly good signal, but the variance in the judgements is quite high.

What else?

Still need test queries

- must be connected in a significant way to the available documents
- must be representative of actual user needs
- random query terms from the documents are not a good idea
- sample from query logs if available

Classically (no-web IR systems):

- low query rates – not enough query logs
- experts manually craft information needs and queries

Standard benchmark datasets

The Cranfield collection (pioneer): 1.398 journal articles, 225 queries (relevance judgements of all query-document pairs).

TREC (Text Retrieval Conference): 1,89 million documents, relevance judgement for 450 information needs (“topics”) on the top k documents of some TREC evaluation.

GOV_2:25-million web page collection

CLEF (Cross Language Evaluation Forum): cross-language information retrieval (query is in different language than the document)

Standard benchmark datasets

The user need is translated into a query.

The relevance is assessed relative to the user need, NOT the query.

Example:

- Information need: My swimming pool bottom is becoming black and needs to be cleaned.
- Query: pool cleaner
- Assess whether the docs address the underlying need, not whether they contain the query terms.

Unranked retrieval evaluation

Binary assessment: relevant or non-relevant.

Precision: fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved})$$

Recall: fraction of relevant documents that are retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant})$$

Unranked retrieval evaluation

	Relevant	Non-relevant
Retrieved	tp	fp
Not Retrieved	fn	tn

Precision P =

Recall R =

F-measure: weighted harmonic mean F =

Unranked retrieval evaluation

	Relevant	Nonrelevant
Retrieved	5	10
Not Retrieved	3	7

Calculate P, R and F-measure.

Unranked retrieval evaluation

	Relevant	Nonrelevant
Retrieved	tp	fn
Not Retrieved	fp	tn

How is accuracy, i.e., the fraction of classifications that are correct, calculated? What is accuracy in the example before?

Is accuracy an appropriate measure for an IR system?

Rank-based measures

Precision, recall and F-measure are computed using unordered sets of documents.

→ We need to extend those measures if we evaluate ranked retrieval results.

Precision@k

Set a rank threshold at k.

Compute ratio of relevant docs in the top k (ignore documents ranked lower than k) with precision.

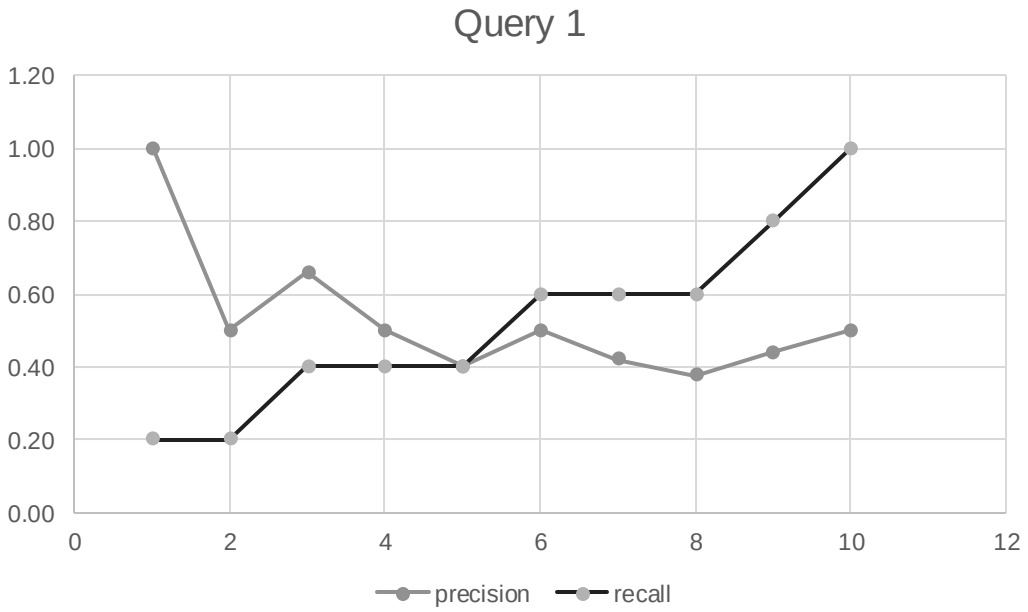
Example:

What is P@1 - P@10 here?

In a similar fashion we calculate R@3, R@4, R@5.

Precision/recall and F1 versus the rank

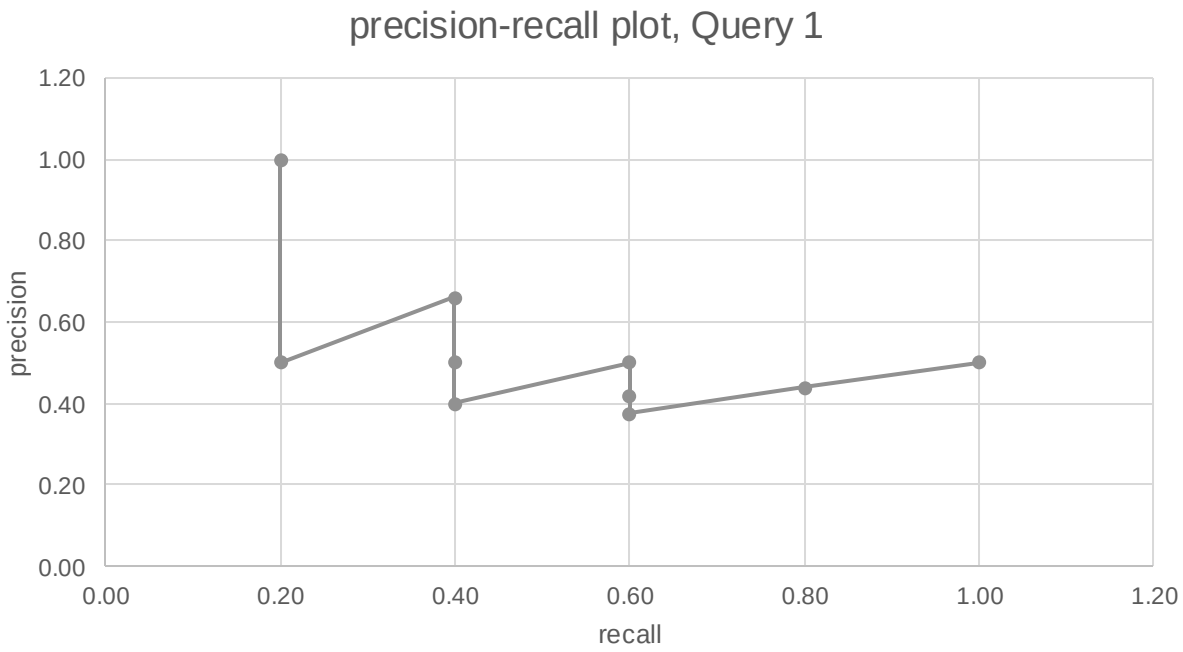
As a function of rank: precision will go down, recall will go up.



The tendency is not particularly interesting. Let's plot one against the other.

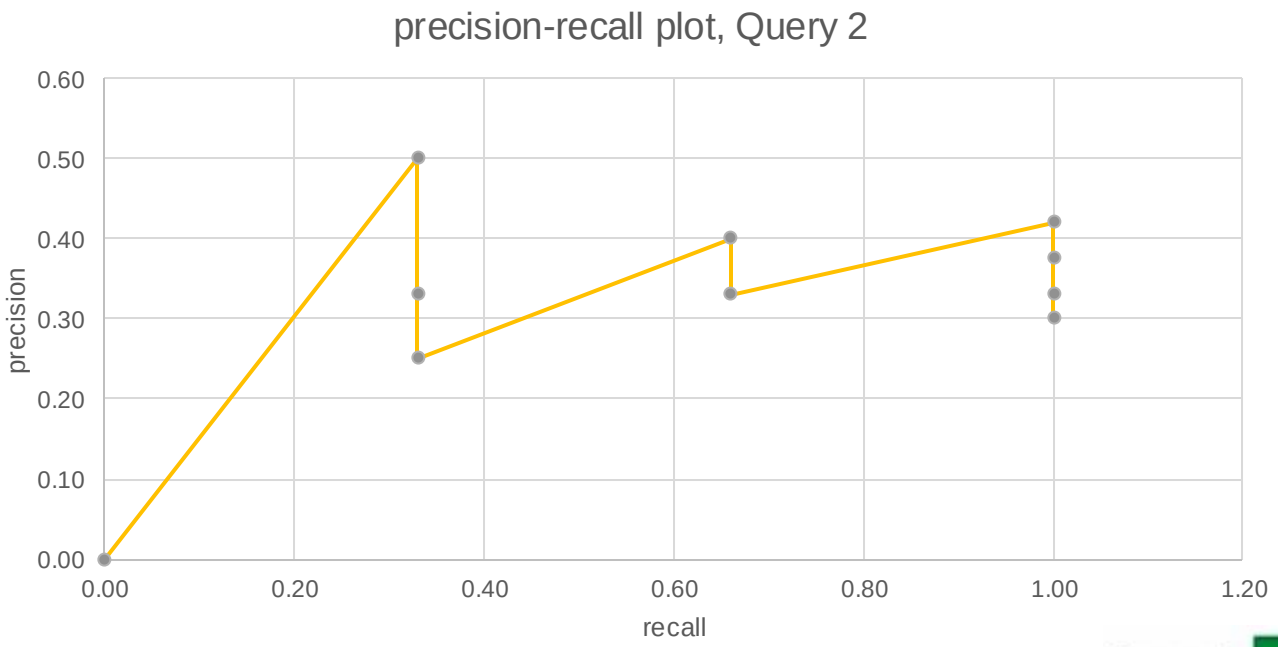
Precision-recall curve

Sawtooth shape of the precision-recall curve.



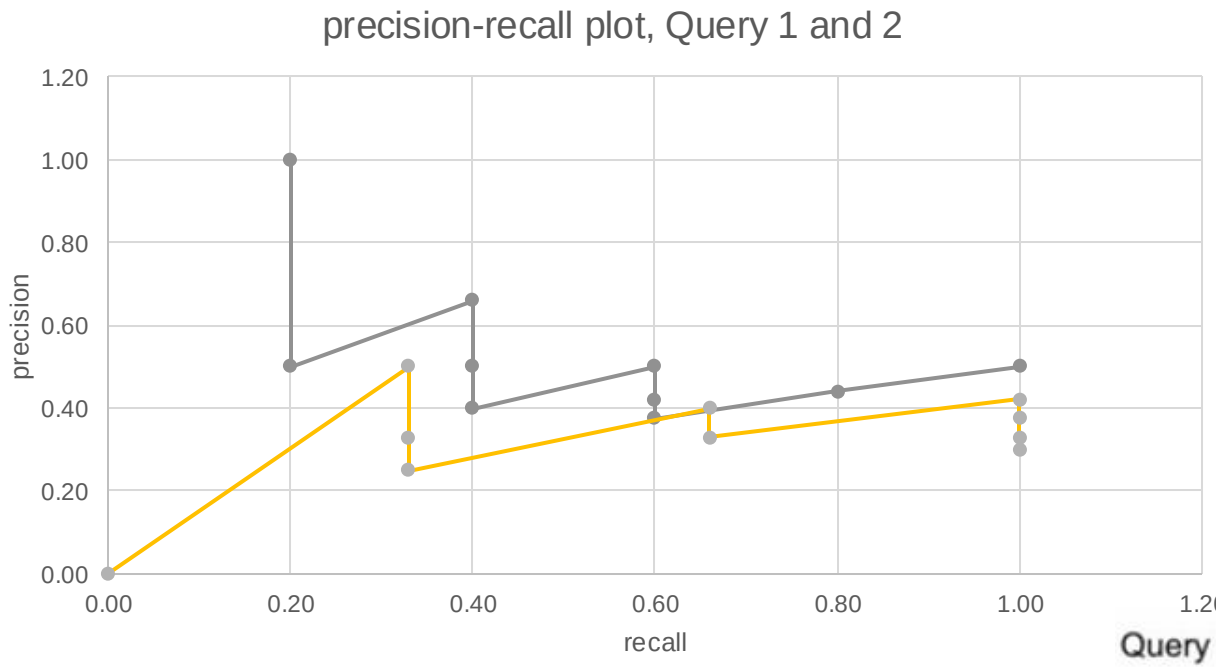
Precision-recall curve

One curve per query/result set.



Precision-recall curve

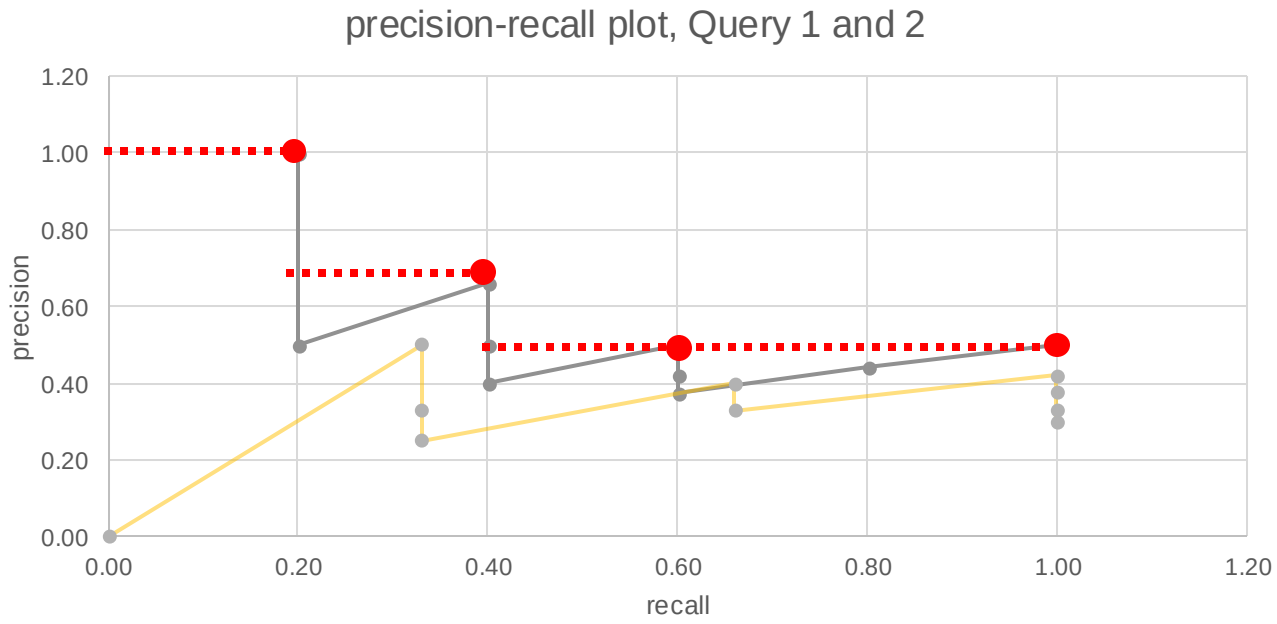
Detailed picture, but erratic behavior → need to “average” the curves across different queries.



Precision-recall curve

Issue: What is precision at recall 0.5? We need to interpolate, aka. infer some value based on other precision values for query 1 and 2.

Standard averaging at fixed recall levels: 0, 0.1, 0.2, 0.3, …



Precision-recall curve: interpolation

On average, precision drops as recall increases. Define interpolation to preserve this monotonicity.

Interpolated precision: find the highest precision for any recall level r' ≥ r.

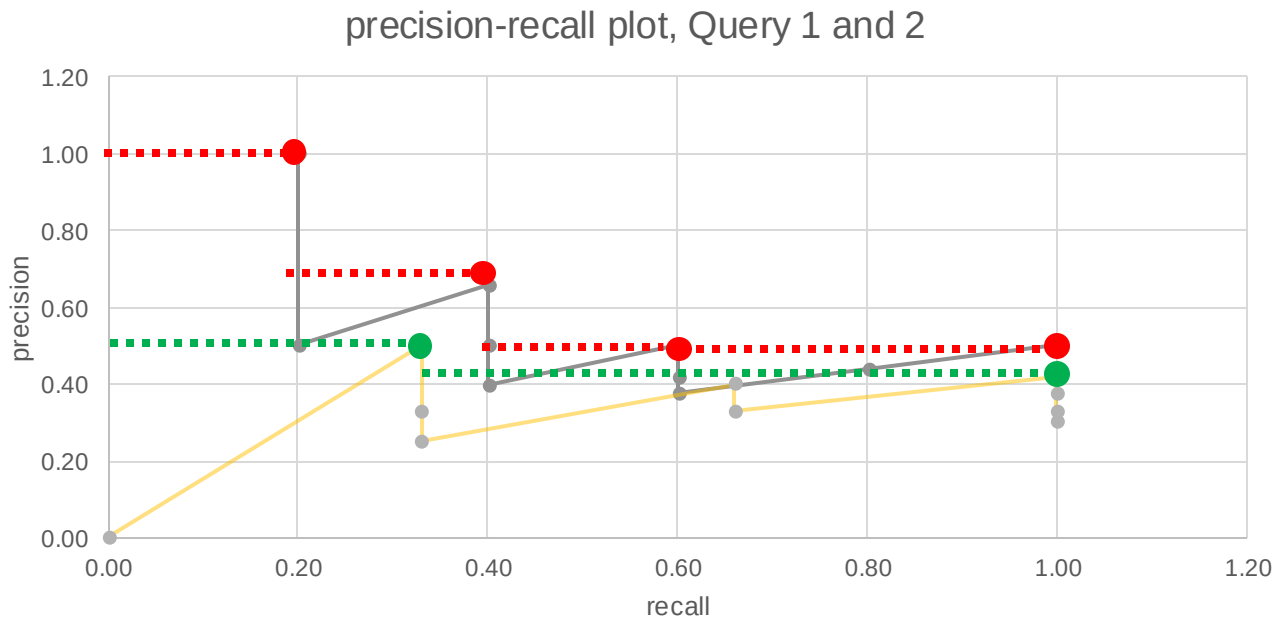
$$p_{interp}(r) = \max_{r' \geq r} p(r')$$

Optimistic interpolation: upper bound of the original precision-recall curve.

Standard way to interpolate in these IR settings.

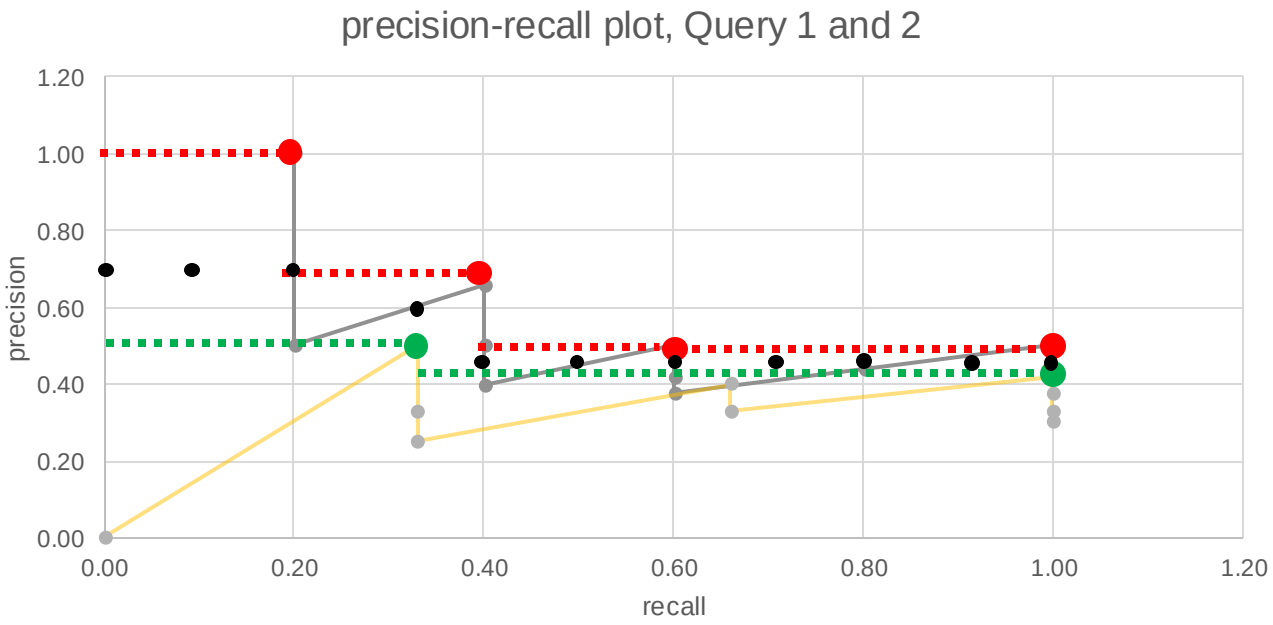
Precision-recall curve: interpolation

Take the average of both curves

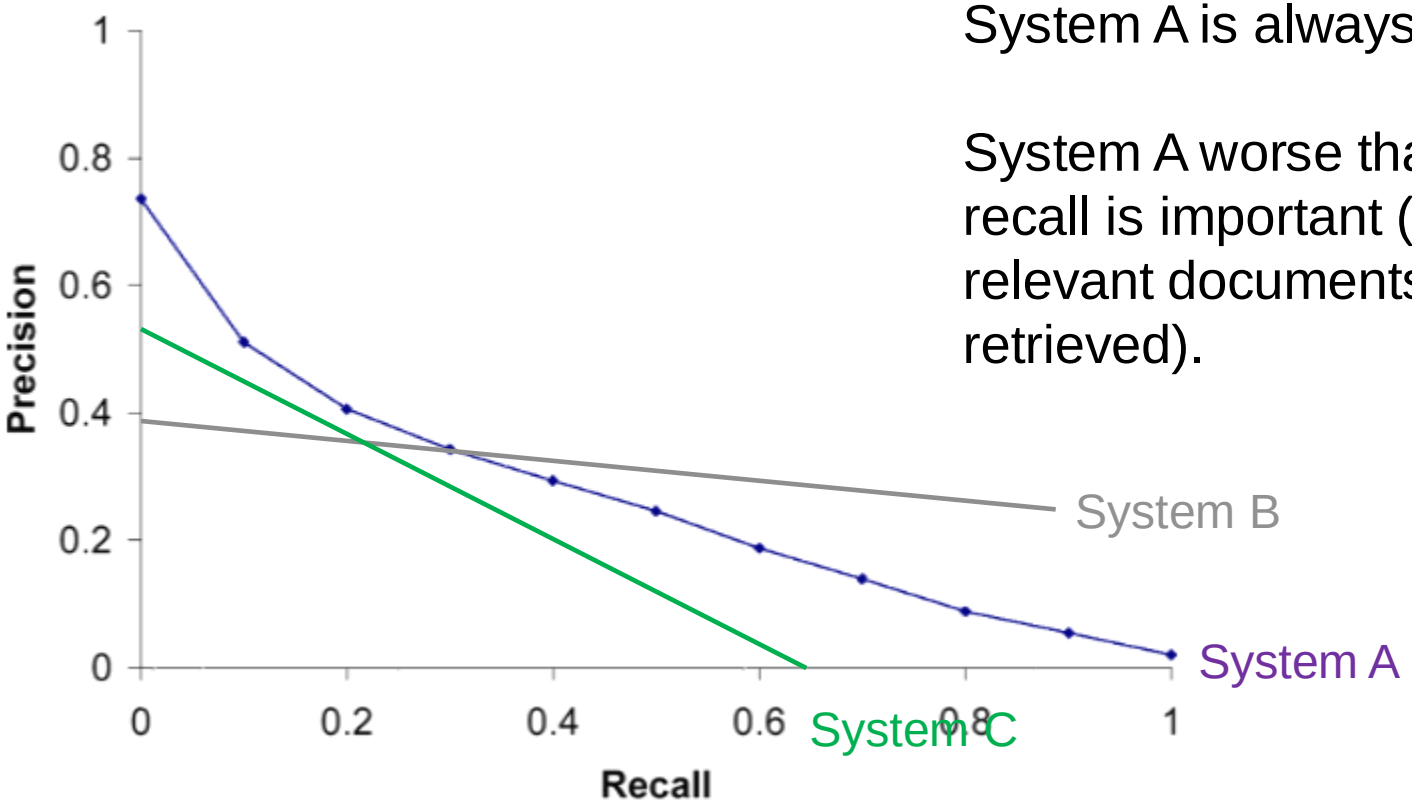


Precision-recall curve: interpolation

Take the average of both curves over 11 points: 0, 0.1, 0.2, 0.3, …, 1,0



Averaged eleven-point precision-recall graph



System A is always better than C.

System A worse than System B if recall is important (fraction of the relevant documents that are retrieved).

Average across 50 queries for representative TREC system.

Mean average precision (MAP)

Another way to measure binary relevance.

Consider the rank position of each *relevant* document.

Compute P@K for each K.

Average precision = average P@K

What's the average precision of Query 1 and Query 2?

MAP is average precision across multiple queries/rankings.

MAP across Query 1 and Query 2 =

Mean average precision (MAP)

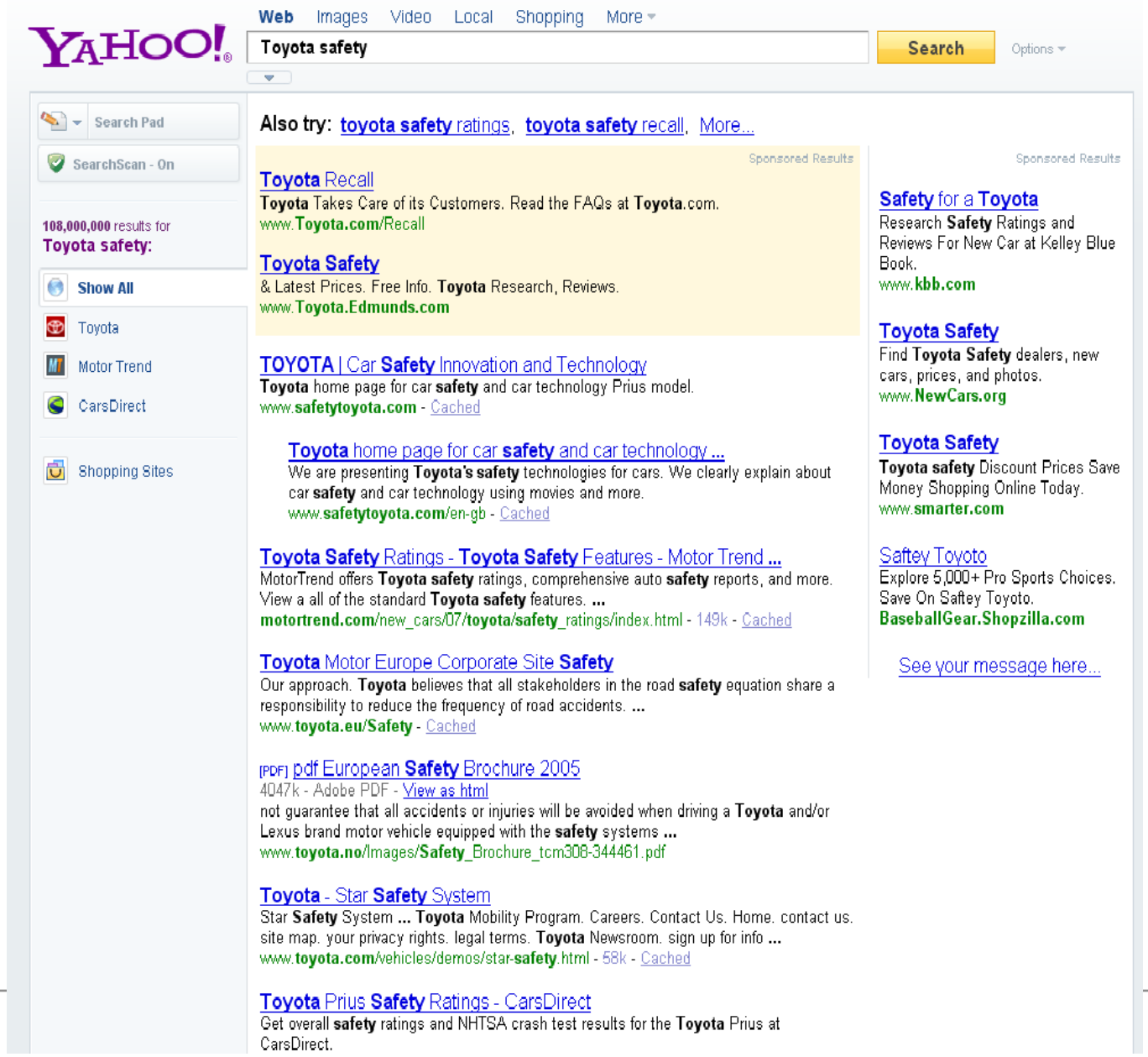
Now perhaps the most commonly used measure in research papers.

- If a relevant document never gets retrieved, we assume the precision corresponding to that relevant doc to be zero.
- MAP is macro-averaging: each query counts equally.

Good for web search?

- MAP assumes the user is interested in finding many relevant documents for each query.
- MAP requires many relevance judgements in text collections.

Beyond binary relevance



Discounted cumulative gain

Popular measure for evaluating web search and related tasks.

Two assumptions:

1. Highly relevant documents are more useful than marginally relevant documents.
2. The lower the ranked position of a relevant document, the less useful it is for the user, since it's less likely to be examined.

Used by some web search companies.

Focus on retrieving highly relevant documents.

Discounted cumulative gain

Uses graded relevance as a measure of usefulness, or gain, from examining a document.

Gain is accumulated starting at the top of the ranking and may be reduced, or discounted, at lower ranks.

Like P@K, it is evaluated over some number of top K results.

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

Summarize a ranking:

- Imagine the relevance judgements are on a scale of [0, r], with r > 2.
- Cumulative gain (CG) at rank $r_p = r_1 + r_2 + r_3 + \ldots + r_p$
- Discounted Cumulative Gain (DCG) at rank r_p
  - DCG = $r_1 + r_2/\log_2 2 + r_3/\log_2 3 + \ldots + r_n/\log_2 p$

→ DCG_p is the total gain accumulated at a particular rank (written differently):

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

Normalized discounted cumulative gain

$NDCG_n$ = Normalized DCG at rank *n* by the DCG value at rank *n* of the ideal, ground truth ranking.

The ideal ranking first returns the documents with the highest relevance level, then the next highest relevance level, etc.

Normalized discounted cumulative gain

i	Ground Truth		Ranking Function$_1$		Ranking Function$_2$	
	Document Order	r_i	Document Order	r_i	Document Order	r_i
1	d4	2	d3	2	d3	2
2	d3	2	d4	2	d2	1
3	d2	1	d2	1	d4	2
4	d1	0	d1	0	d1	0
	$NDCG_{GT}$=1.00		$NDCG_{RF1}$=1.00		$NDCG_{RF2}$=0.9203	

$$DCG_{GT} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF1} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF2} = 2 + \left(\frac{1}{\log_2 2} + \frac{2}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.2619$$

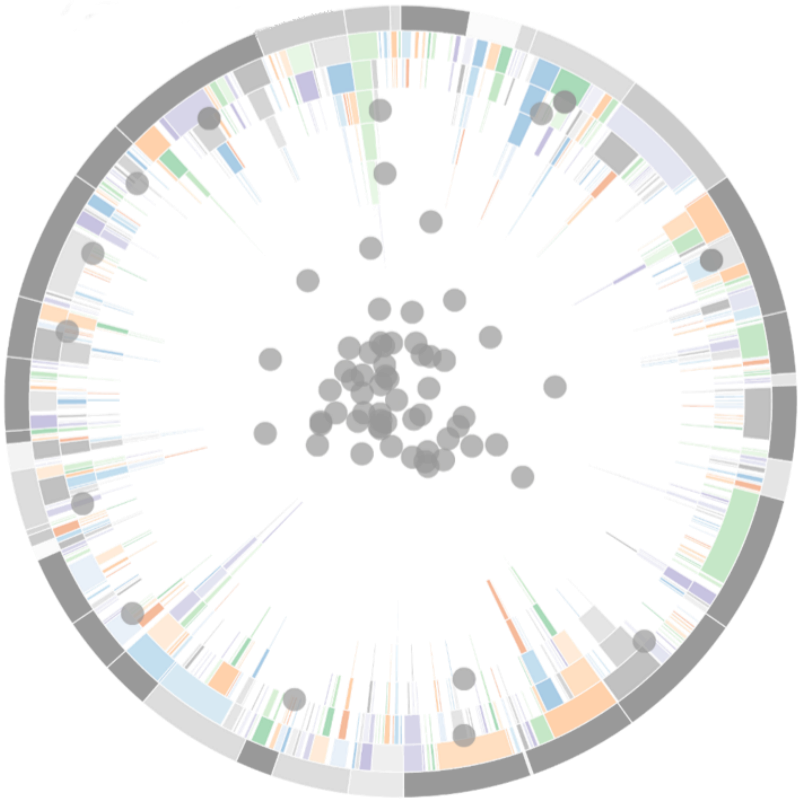
$$MaxDCG = DCG_{GT} = 4.6309$$

Cluster of Excellence
The Politics of Inequality

Steinbeis Transfer Center
Linguistic Data Analysis

UNIVERSITÄT PASSAU

Thank you.
Questions?
Comments?



Annette Hautli-Janisz, Prof. Dr.
cornlp-teaching@uni-passau.de