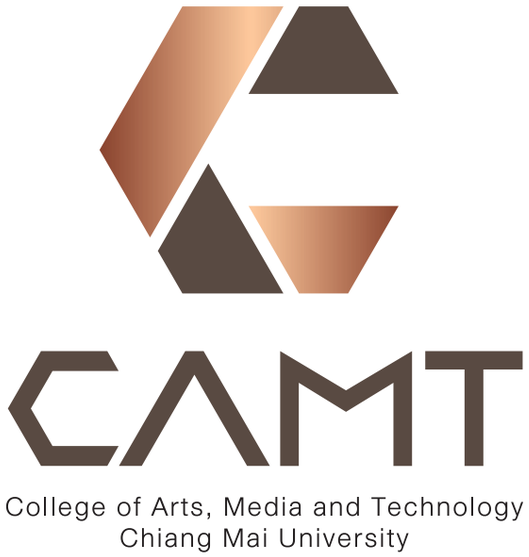
SE 481 Introduction to Information Retrieval (IR for SE)

Module #5 — Evaluation



Passakorn Phannachitta, D.Eng.

passakorn.p@cmu.ac.th

College of Arts, Media and Technology

Chiang Mai University, Chiangmai, Thailand

Agenda

- Evaluation metrics in IR

Determine if search results align with the user's needs

- Monitor the frequency and depth of user interactions with search results.
- Evaluate revenue generated following product searches.
- Measure the time taken for users to find satisfactory results.

= Is the user happy?

Is the user happy? — a challenging measurement

- It is elusive but not immeasurable
  - Analogous to unit test versus acceptance test in software

- In IR, the **unit test** equates to measuring **relevance**

Measuring relevance

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

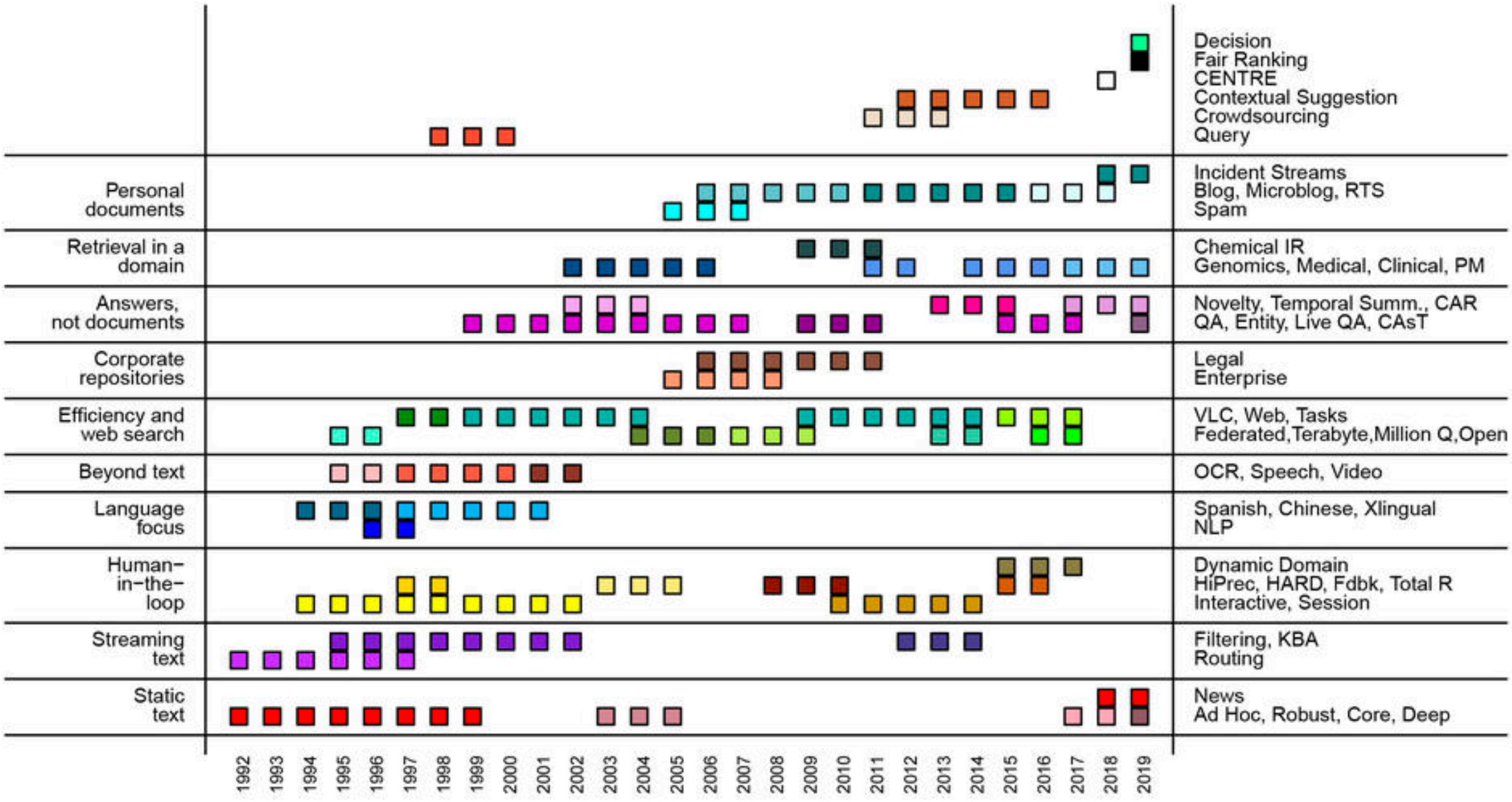
Relevance judgement

- Issue?
  - Document sets are large, e.g., web pages on the internet
  - Rapid increase in document volume daily
- Crowd-sourcing is commonly used
  - Potential issues with variance and quality
- Testing on standard benchmarking collections is crucial
  - Includes difficult and representative questions

Standard benchmark example

- Text REtrieval conference — TREC datasets
  - Series of workshops focusing on a list of different IR research areas
  - Start in 1992 (~30 years ago)

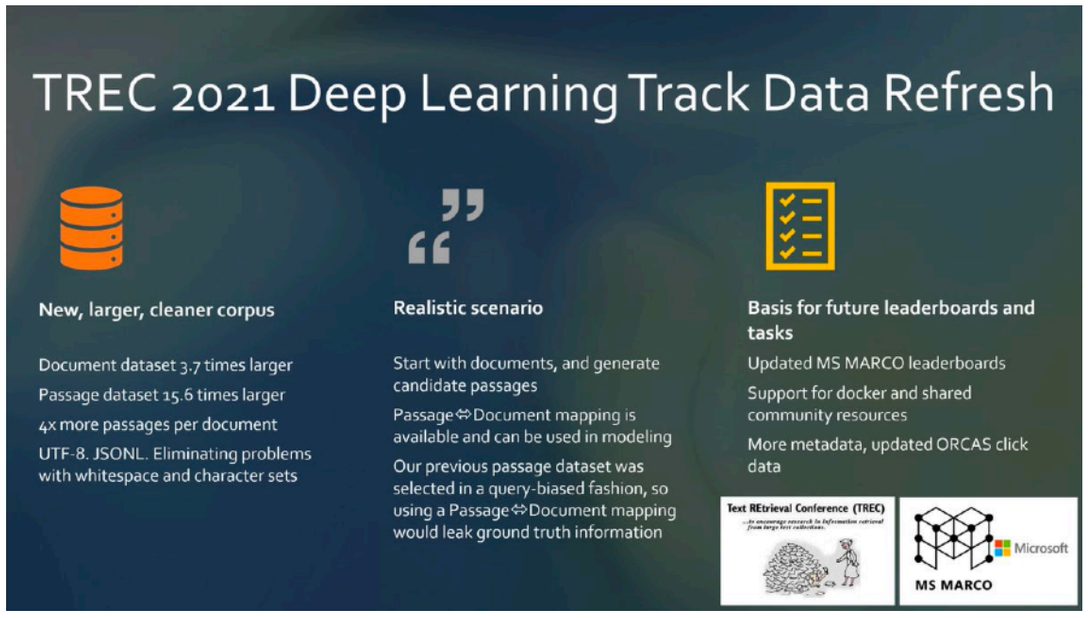
TREC



Ref: https://www.nist.gov/image/tracksjpg

TREC

- Some recent interesting tracks



Evaluating an IR system

- A user's need is translated into a query before assessment
- Relevance is assessed relative to the user's need, **not the query**

- E.g.,
  - **Information need** — I want to extract all the textual information from a PDF file using Java
  - **A possible query** — *get text java pdf*
  - **A potentially better query** — *pdf parsing java*
  - **What to assess** — whether the returned documents help the user find a solution, not just if they contain the query words

Standard evaluation metric family

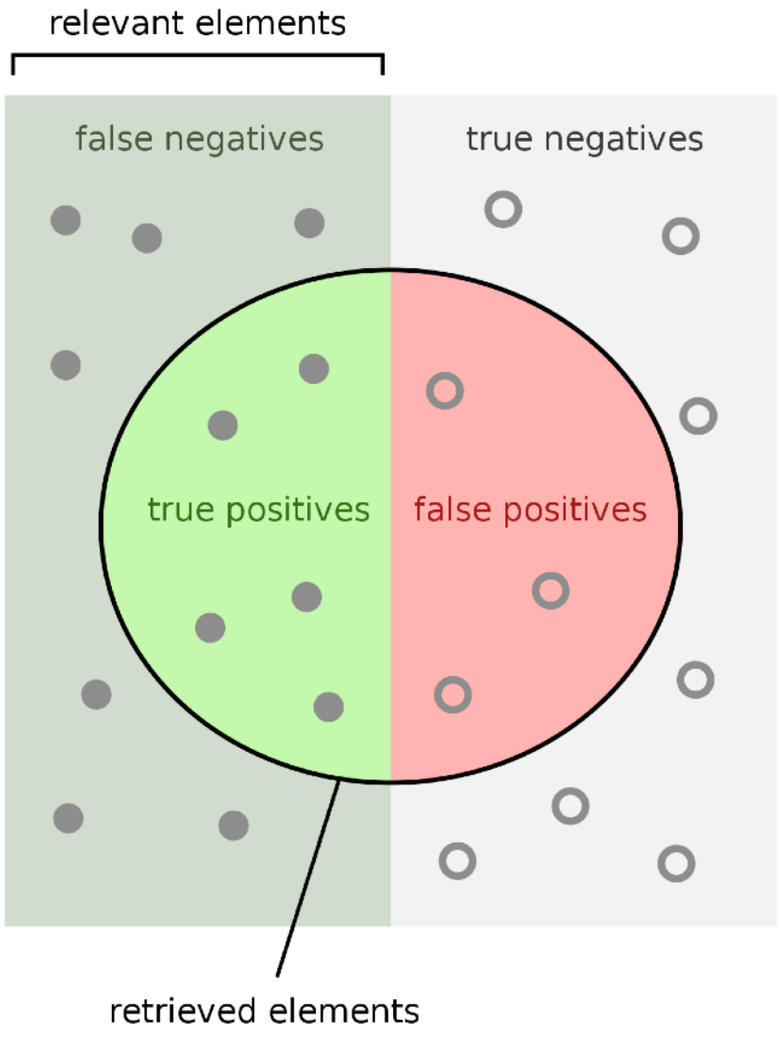
- Precision (for ranking)
- Recall (for ranking)
- Discounted cumulative gain (DCG)

Precision and Recall

- In unranked binary assessment, the metrics are defined as:
  - **Precision** answers: Of all the items labeled as positive, how many are actually positive?.
  - **Recall** answers: Of all the positive items that exist, how many did the model correctly identify?

- Understanding the terms:
  - **True Positives (TP)**: #correctly predicted positives out of all actual positives.
  - **True Negatives (TN):** #correctly predicted negatives out of all actual negatives.
  - **False Positives (FP):** #incorrectly predicted positives out of all actual negatives.
  - **False Negatives (FN):** #incorrectly predicted negatives out of all actual positives.

Precision and Recall



How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{retrieved elements}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Ref: https://en.wikipedia.org/wiki/Precision_and_recall

Precision and recall in unranked problems

- Precision = $\frac{TP}{TP + FP}$
- Recall = $\frac{TP}{TP + FN}$

- Example

	Predicted positive	Predicted negative
Actual positive	80 (TP)	40 (FN)
Actual negative	20 (FP)	100 (TN)

- Precision = (80) / (80 +20) = 0.8
- Recall = (80) / (80 +40) = 0.667

Precision and recall in unranked problems

- **Map to a familiar problem: e.g.,**
  - To automatically identify critical bugs from a pool of reported issues
  - Precision: Out of 100 bugs retrieved, if 80 are genuinely critical bugs related to the module, and 20 are not, the precision is 80/100 = 80%.
  - Recall: If there are 120 actual critical bugs in the module, and the system identifies 80, the recall is 80/120 = 66.67%

	Predicted positive	Predicted negative
Actual positive	80 (TP)	40 (FN)
Actual negative	20 (FP)	Any (TN)

Precision and Recall in unranked problems

- **Implication**
  - Precision: Higher precision means **fewer irrelevant bugs are presented** to the team, **saving time and effort**.
  - Recall: Higher recall means **more of the actual critical bugs are captured**, ensuring **important issues are not missed**.

- **In practice**
  - Balancing precision and recall

Standard evaluation metrics for ranking

- Binary relevance
  - Precision@k
  - Recall@k
  - Mean average precision (mAP)
  - Mean reciprocal rank (mRR)
- Multiple levels of relevance
  - Normalized discounted cumulative gain (NDCG)

Precision@k

- In binary classification, all positive instances are considered equally,
  - However, in the context of ranking, the order in which results appear (i.e., their placement in the list) is important.
- In other words, precision@k focuses on the quality of the top part of the ranked list.
- Precision@k is especially useful in scenarios where the user is only interested in the top few results (like web search engines).

Precision@K

- Choose a specific number **k**
- Calculate the percentage of relevant items in the top K results.
- Ignore results ranked below K.
- E.g.,

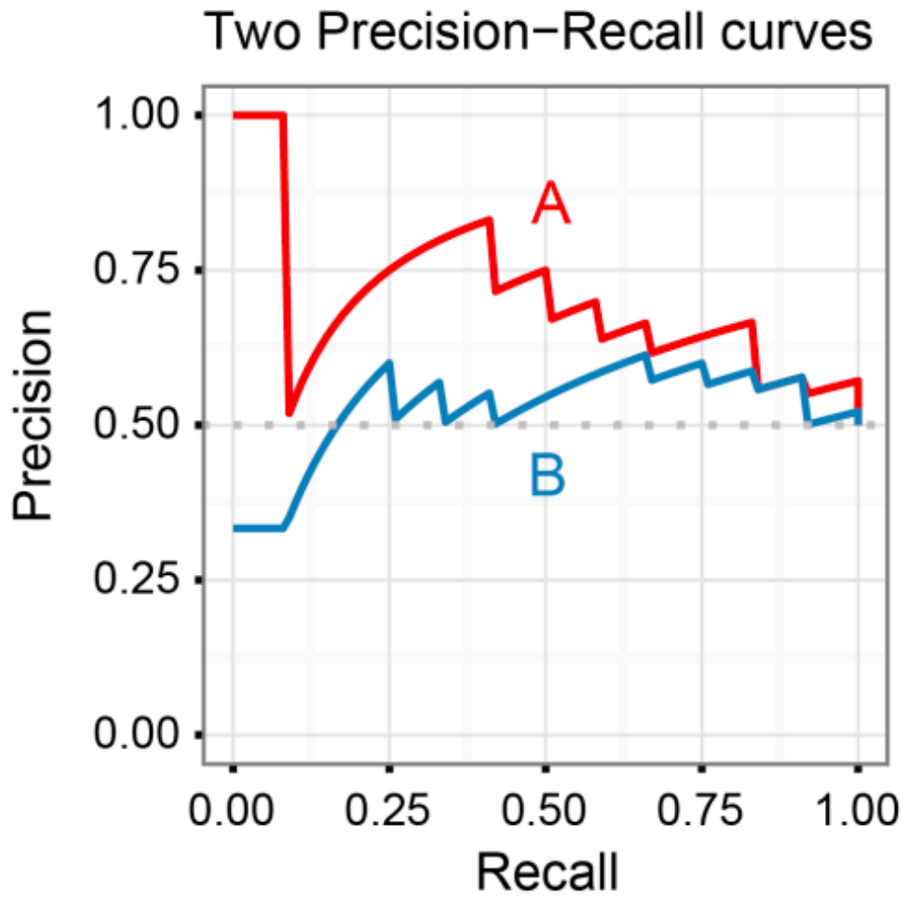
- Precision@3 = 1/3
- Precision@4 = 2/4
- Precision@5 = 3/5

Recall@k

- While precision@k evaluates the top part of a ranked list, recall@k assesses how many relevant items are included in these top k results compared to all relevant items in the dataset.
- Recall@k can be more complex to evaluate because it requires knowledge of all relevant items in the dataset, not just those in the top
- e.g., In many real-world scenarios, especially in large datasets, knowing all relevant items is not feasible, making recall@k harder to measure accurately.

Precision-recall curve

- Utilized to compare the effectiveness of two or more information retrieval systems.
- Plots the trade-off between precision and recall for different threshold settings.
- A higher area under the curve (AUC) indicates a better performance of the IR system.



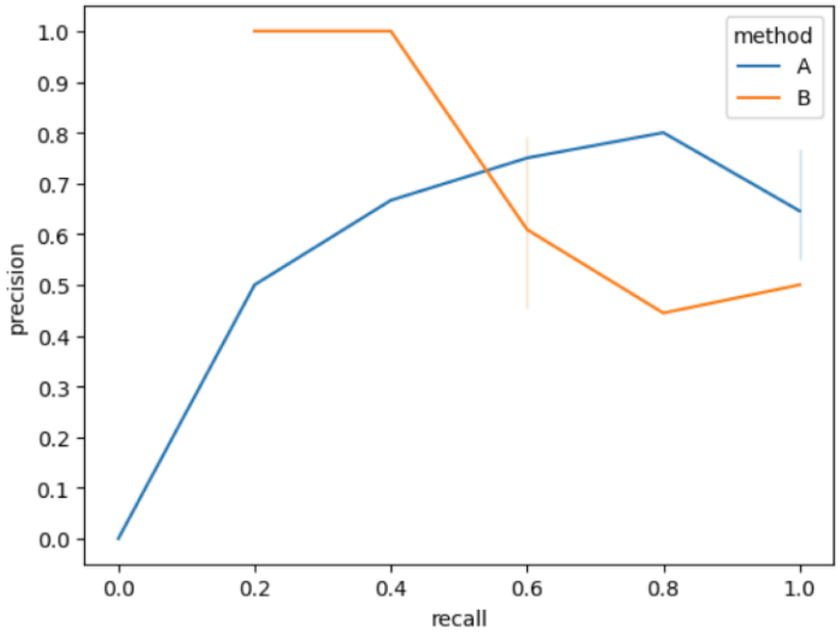
Ref — https://stackoverflow.com/questions/40865645/confusion-about-precision-recall-curve-and-average-precision

Quick workout

- Given 5 relevant documents.
  - Recommendation system A provides: FTTTTTFFFF.
  - Recommendation system B provides: TTTFFFFFTT.
  - To compare the two, manually create a visualization, such as a precision-recall curve.

A quick workout #1

- Given 5 relevant documents.
  - Recommendation system A provides: FTTTTTFFFF.
  - Recommendation system B provides: TTTFFFFFTT.
  - To compare the two, manually create a visualization, such as a precision-recall curve.
- E.g.,



Mean average precision (mAP)

- **AP (Average precision)** — Measures precision at each point a relevant document is retrieved, then averaged over all relevant documents.
- **mAP (Mean AP)** — Calculates the average of AP scores across all queries in a set.
    - Macro Average: Each query's AP contributes equally to the final mAP, regardless of the number of relevant documents for each query.

- Precision is considered zero for relevant documents that the system fails to retrieve.

Average precision

There are 5 documents relevant to query #1

Ranking for query #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

Average precision for query #1 = (1.0 + 0.67 + 0.5 + 0.44 + 0.5) / 5 = 0.62

There are 3 documents relevant to query #2

Ranking for query #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

Average precision for query #2 = (0.5 + 0.4 + 0.43) / 3 = 0.44

Mean average precision

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

Average precision for query #1 = (1.0 + 0.67 + 0.5 + 0.44 + 0.5) / 5 = 0.62

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

Average precision for query #2 = (0.5 + 0.4 + 0.43) / 3 = 0.44

Mean average precision = (0.62 + 0.44) / 2 = 0.53

A quick workout #2

- Given 5 relevant documents
  - Recommendation system A provides: FTTTTTFFFF.
  - Recommendation system B provides: TTTFFFFFTT.
  - Provide the Python code snippet for comparing AP values between the two systems.

Mean reciprocal rank

- **RR (Reciprocal Rank)** — The inverse of the rank at which the first relevant document is retrieved, or 0 if no relevant document is retrieved.
- **mRR (Mean RR)** — The average of the RR scores across multiple queries.

- Emphasizes the importance of the rank of the first relevant item retrieved.

Reciprocal rank

Ranking for query #1

Reciprocal Rank = 1/5

Ranking for query #2

Reciprocal Rank = 1/2

Ranking for query #3

Reciprocal Rank = 1/1

Mean reciprocal rank

Ranking for query #1

Reciprocal Rank = 1/5

Ranking for query #2

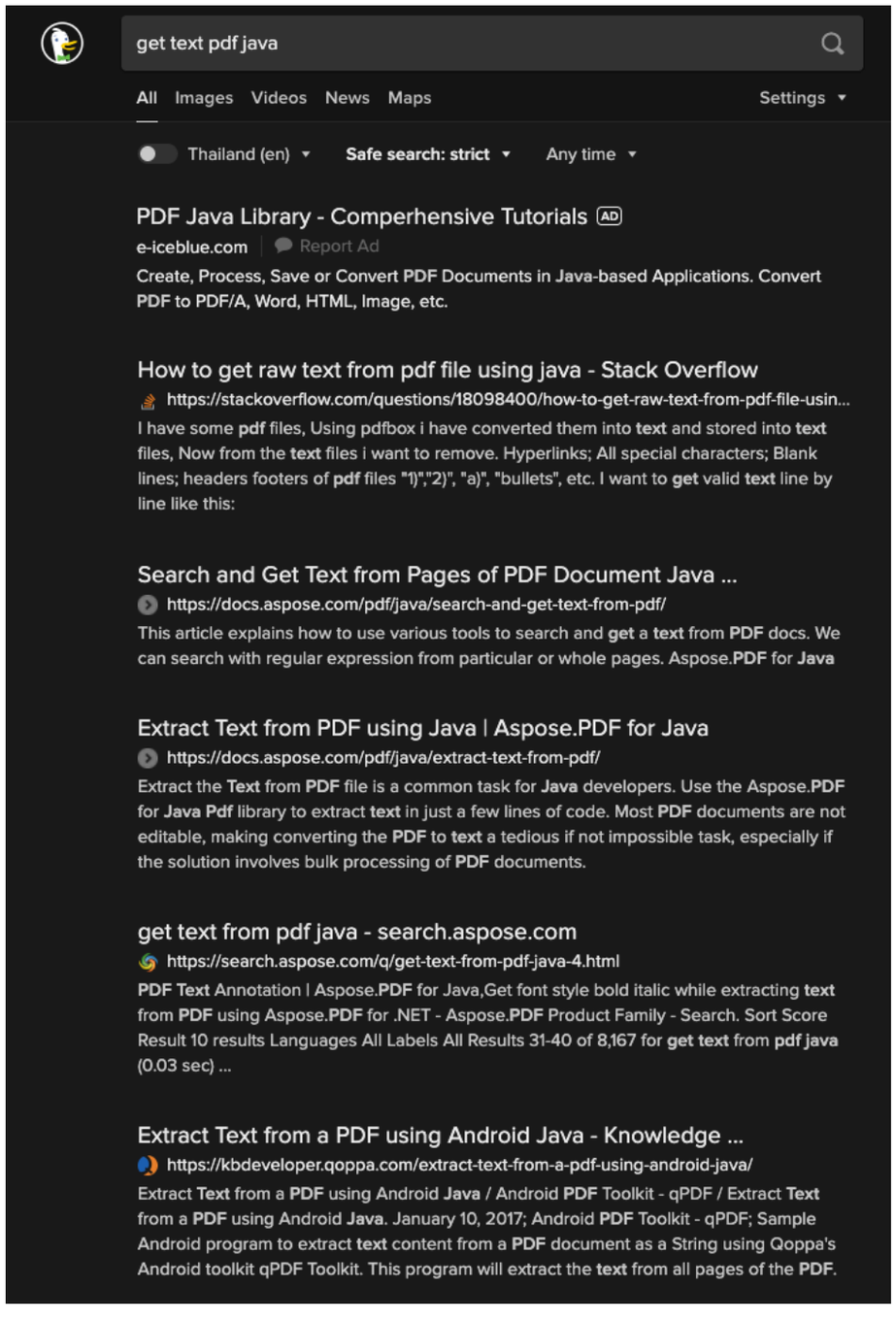
Reciprocal Rank = 1/2

Ranking for query #3

Reciprocal Rank = 1/1

Mean reciprocal rank = (1/5 + 1/2 + 1/1) = 1.7

Beyond binary relevance



User #1's judgement

★★★★

★★★

★★

★★

★★★★

User #2's judgement

★★★★★

★

★

★

★★

Beyond binary relevance

- **Utilizing ranking scores** — With this, we can approximate a preferred rank list tailored to each user.To build a personalized search engine
  - A step towards constructing a personalized search engine.
- **Learning to rank foundation** — Essential for developing machine-learned ranking models.
- **Preparatory knowledge** — Understanding common evaluation metrics is crucial.
  - Example metric: Discounted Cumulative Gain (DCG).

Discounted cumulative gain (DCG)

- Multiple levels of document relevance.

- **Main assumptions**
  - Greater value is placed on highly relevant documents over marginally relevant ones.
  - Documents at lower ranks are deemed less useful, reflecting the decreasing likelihood of user review.

Discounted cumulative gain (dcg)

- Multiple levels of document relevance.
- Value decreases as document rank lowers.
- **Discount Function:**
  - Typical discount function is $1/\log_2(\text{rank})$
  - e.g., the discount at rank 4 and rank 8 are 1/2 and 1/4, respectively.

Discounted cumulative gain

- **Relevance scale** — Judgments range from [0, r] where r > 2.
- CG at rank n = $r_1 + r_2 + \ldots + r_n$
- DCG at rank n = $r_1/\log_2 2 + r_2/\log_2 3 + r_3/\log_2 4 + \ldots + r_n/\log_2(n+1)$

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)}$$

Discounted cumulative gain — example

- 10 ranked document in 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:

3	5	6.89	6.89	6.89	7.28	7.99	8.66	9.61	9.61

Normalized DCG

- **Normalization of DCG** — DCG at rank n is normalized by the ideal DCG at the same rank.
  - **Ideal DCG** — Documents are sorted in perfect order of relevance.

- Facilitates comparison across queries with different amounts of relevant documents.

Normalized DCG

i	Ideal		Ranking function 1		Ranking function 2	
	Order	r_i	Order	r_i	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
	$NDCG_{ideal}$ = 1.00		$NDCG_{function1}$ = 1.00		$NDCG_{function2}$ = 0.9652	

$$DCG_{ideal} = \frac{2}{log_2 2} + \frac{2}{log_2 3} + \frac{1}{log_2 4} + \frac{0}{log_2 5} = 3.7618$$

$$DCG_{function_1} = \frac{2}{log_2 2} + \frac{2}{log_2 3} + \frac{1}{log_2 4} + \frac{0}{log_2 5} = 3.7618$$

$$DCG_{function_2} = \frac{2}{log_2 2} + \frac{1}{log_2 3} + \frac{2}{log_2 4} + \frac{0}{log_2 4} = 3.6309$$

$$MaxDCG = DCG_{ideal} = 3.7618$$

In practice

- The integrated use of precision@k, recall@k, and NDCG@k at different k levels, e.g., k is set as 5, 10, or 20, are commonly used.
  - enables an analysis that captures each technique's performance from various perspectives.
  - includes assessing the effectiveness in ranking at the highest ranks and the capability to identify the maximum number of relevant entries.
  - also incorporates partial matches, enhancing our understanding of which technique is most likely to meet user expectations and satisfaction.

A quick workout #3

- Given an automated system used to rank reported bugs, where the most critical bugs should be addressed first.
  - Suppose that there are 5 bugs, which have all been deemed critical.
  - Bug ranking system A outputs the following order: Minor, Critical, Critical, Critical, Critical, Minor, Critical, Minor, Minor, Minor.
  - Bug ranking system B outputs the following order: Critical, Critical, Critical, Minor, Minor, Minor, Minor, Critical, Critical, Minor.
  - Relevance scores are defined as: Critical = 3, Major = 2, Minor = 1.
  - Using python, calculate the NDCG@5 for both ranking systems A and B.

Determine if search results align with the user's needs

- Monitor the frequency and depth of user interactions with search results.
- Evaluate revenue generated following product searches.
- Measure the time taken for users to find satisfactory results.

= Is the user happy?

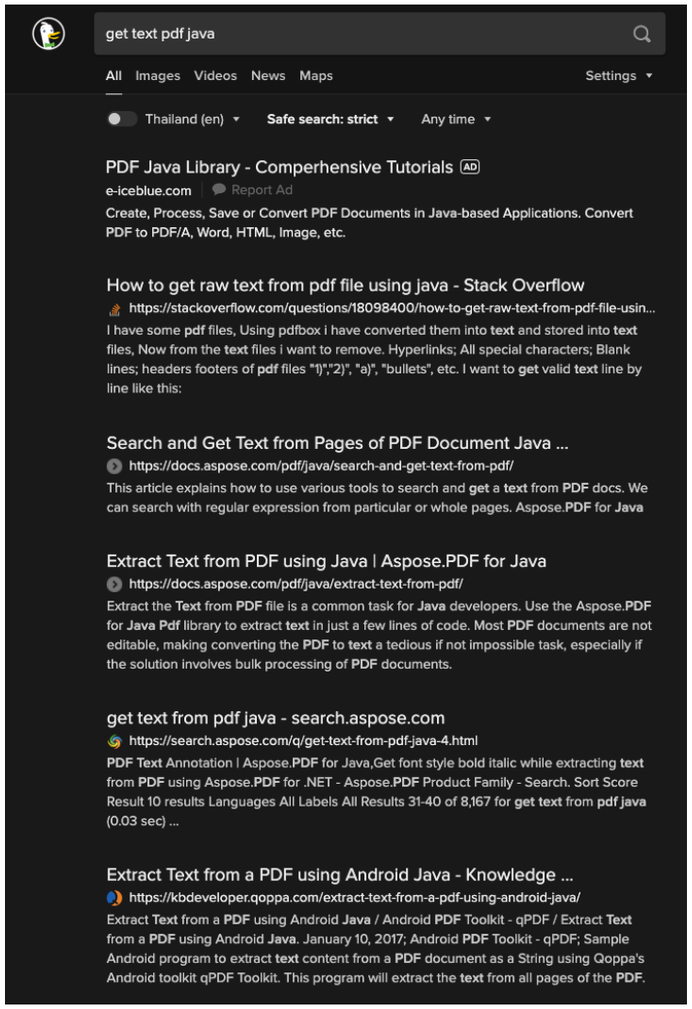
Challenges in directly measuring user happiness

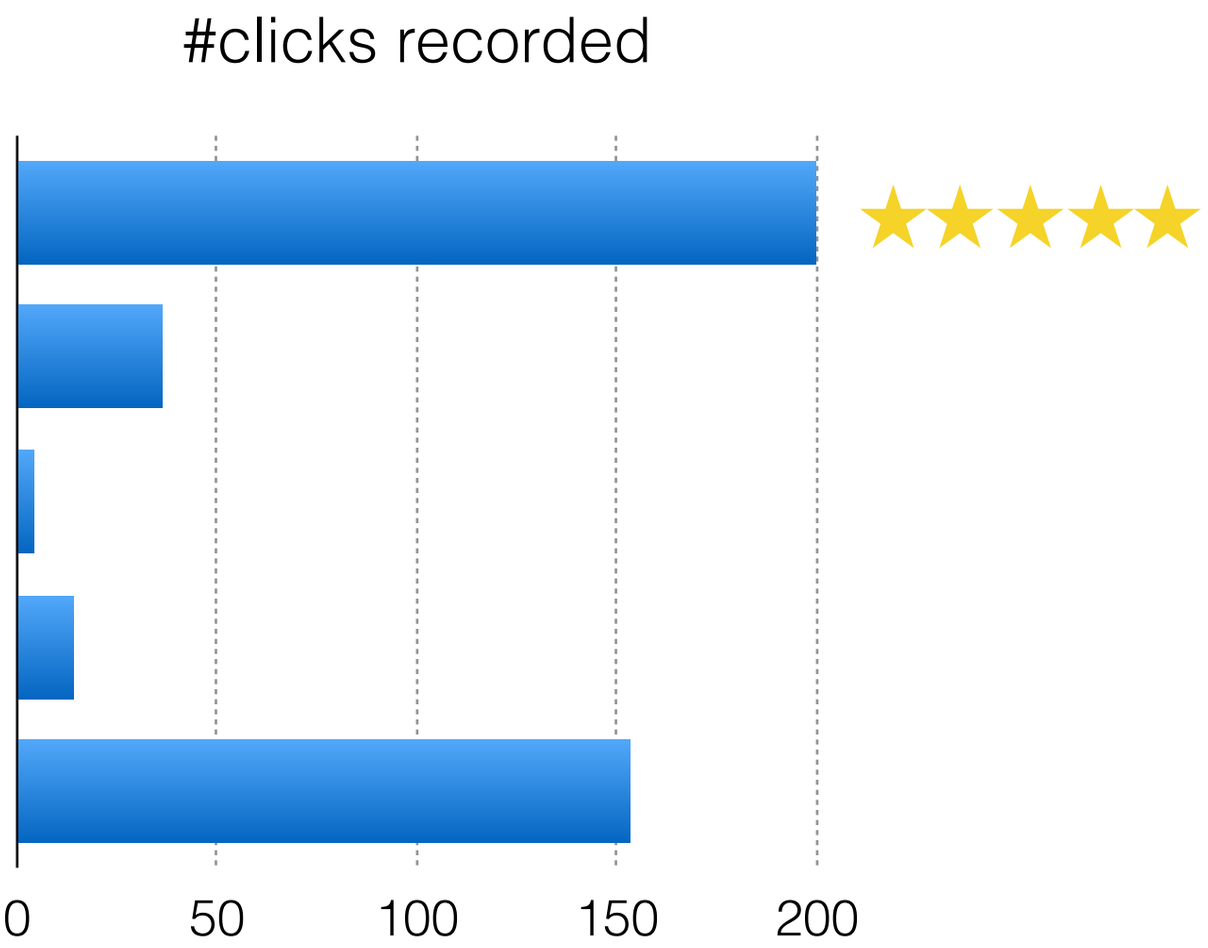
- Human Judgments:
  - Are inherently subjective.
  - Often show inconsistency among different raters and across time.
  - May decrease in relevance as documents and queries evolve.
  - Struggle to accurately represent the diversity of real users.

- Despite difficulties, attempting to gauge user satisfaction is valuable.

Measuring user clicks

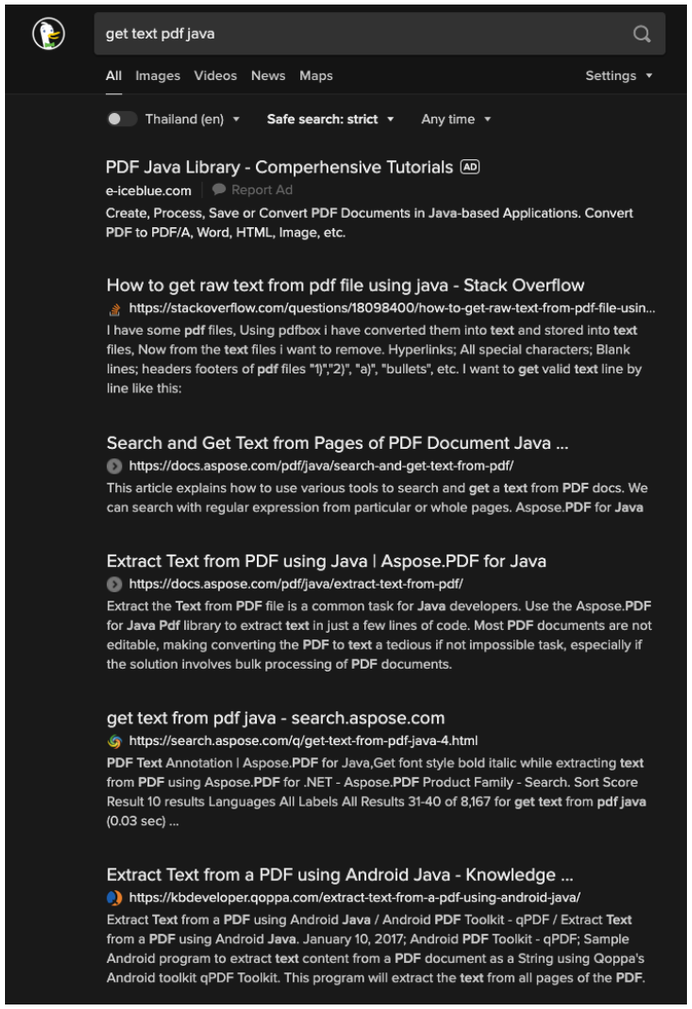
- Click data is a direct approach to infer user behavior.

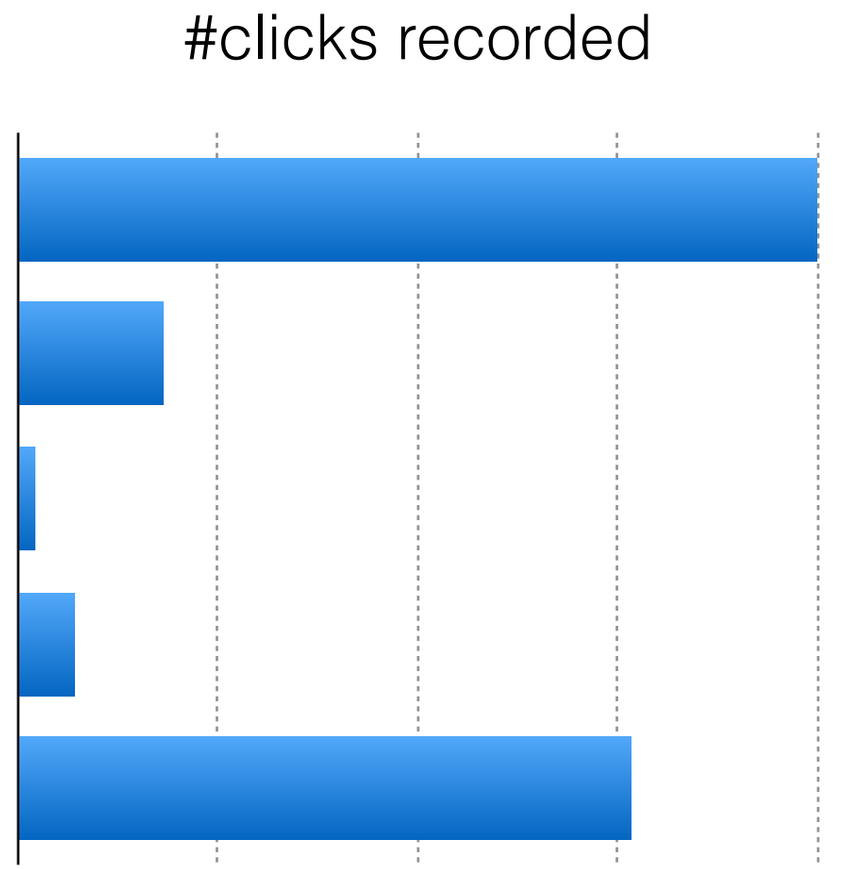




Exploiting the #user clicking information

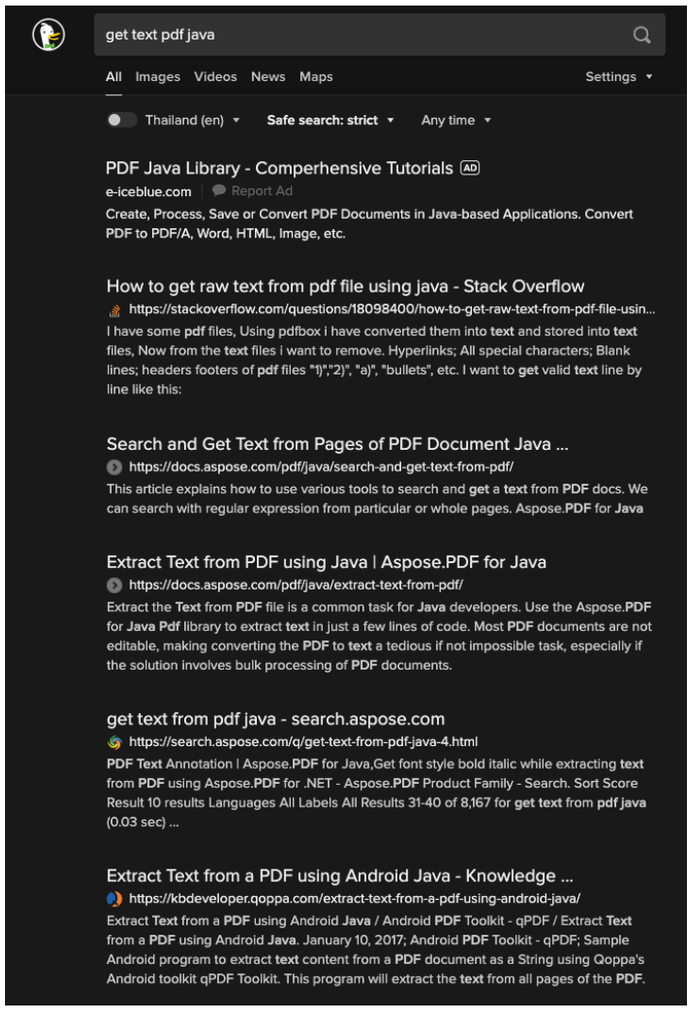
- Adapt ranking to user clicks

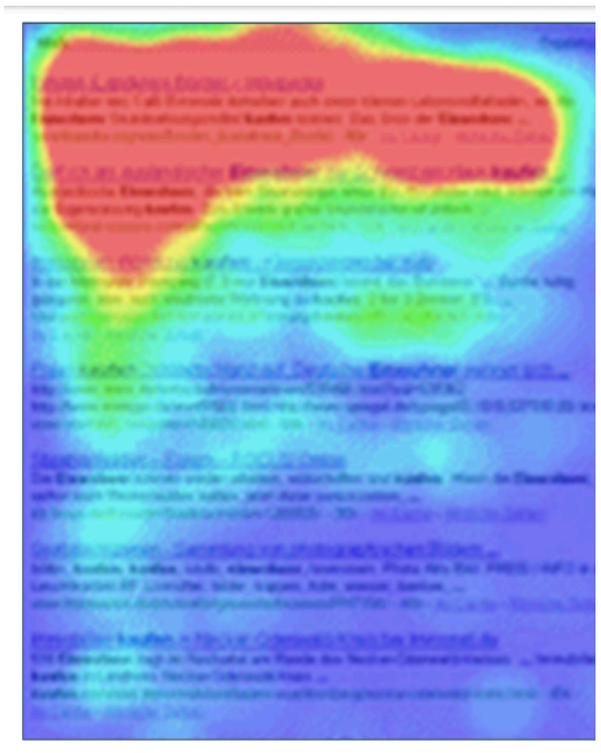




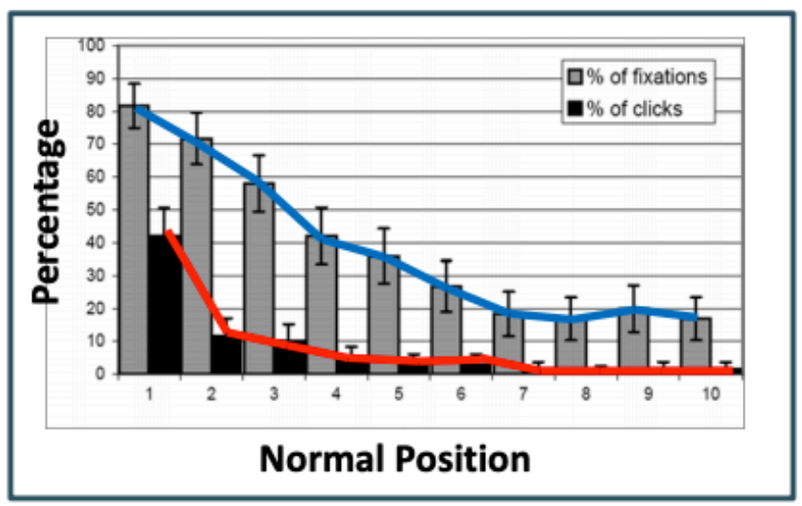
Eye-tracking

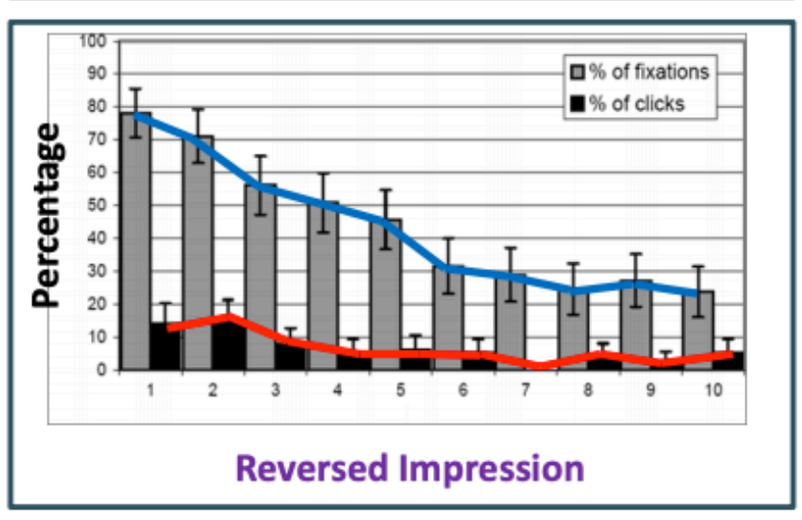
- Click data is a direct approach to infer user behavior.





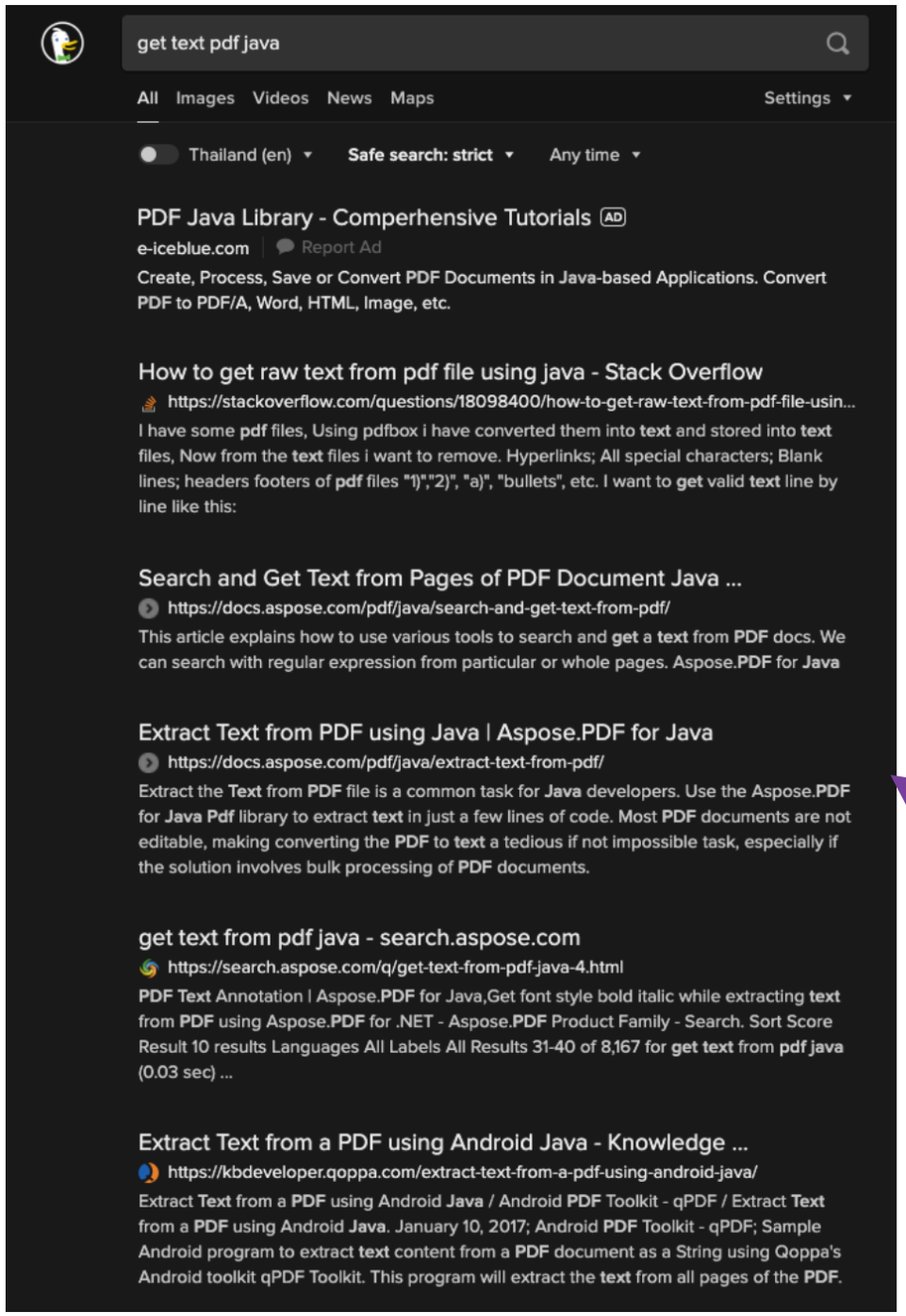
Comes with inherent biases





- Items at higher positions naturally attract more clicks.
- This pattern persists even when result orders are reversed.

Relative rating



User's click sequence

- Hard to say Click #1 > Click #3
- Probably Click #3 > Click #2

Evaluating pairwise relative ratings

- Utilizing pairs such as **Doc_a** is preferred over **Doc_b** for a query **q**

- **Assessment focus**
  - Not solely about rank-ordering according to individual document relevance.
  - Aimed at alignment with historical pairwise preferences as indicated by user click patterns.

E.g., comparing two rankings via clicks

- Query: DevOps

Ranking #1	Ranking #2
Wikipedia	Docker
Ultimate guide by AWS	Jenkins
CD&CI	Intro to DevOps
Azure’s service	CD&CI
Docker	Youtube’s clip

E.g., comparing two rankings via clicks

- Interleaved ranking, this example start with #2

Docker

Wikipedia

Jenkins

Ultimate guide by AWS

Intro to DevOps

CD&CI

CD&CI

Azure's service

Wikipedia

Docker

E.g., comparing two rankings via clicks

- Remove duplicate results

Docker

Wikipedia

Jenkins

Ultimate guide by AWS

Intro to DevOps

CD&CI

CD&CI

Azure’s service

Wikipedia

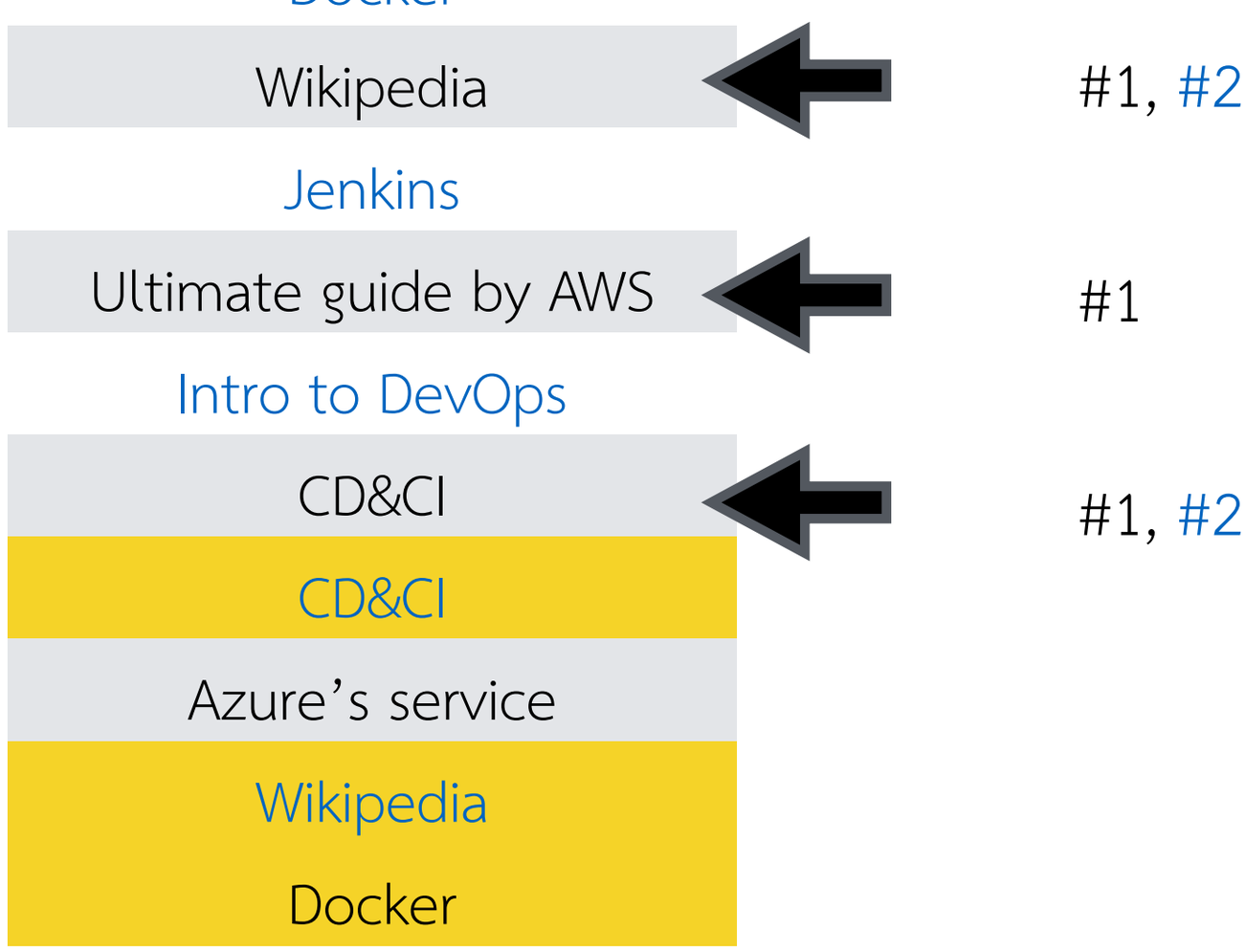
Docker

E.g., comparing two rankings via clicks

- Count user clicks

Ranking #1: 3

Ranking #2: 2



Interleaved ranking

- **User Interaction** — Show users an interleaved combination of results from two different rankings.

- **Click Tracking** — Monitor clicks on results originating from System A versus System B.

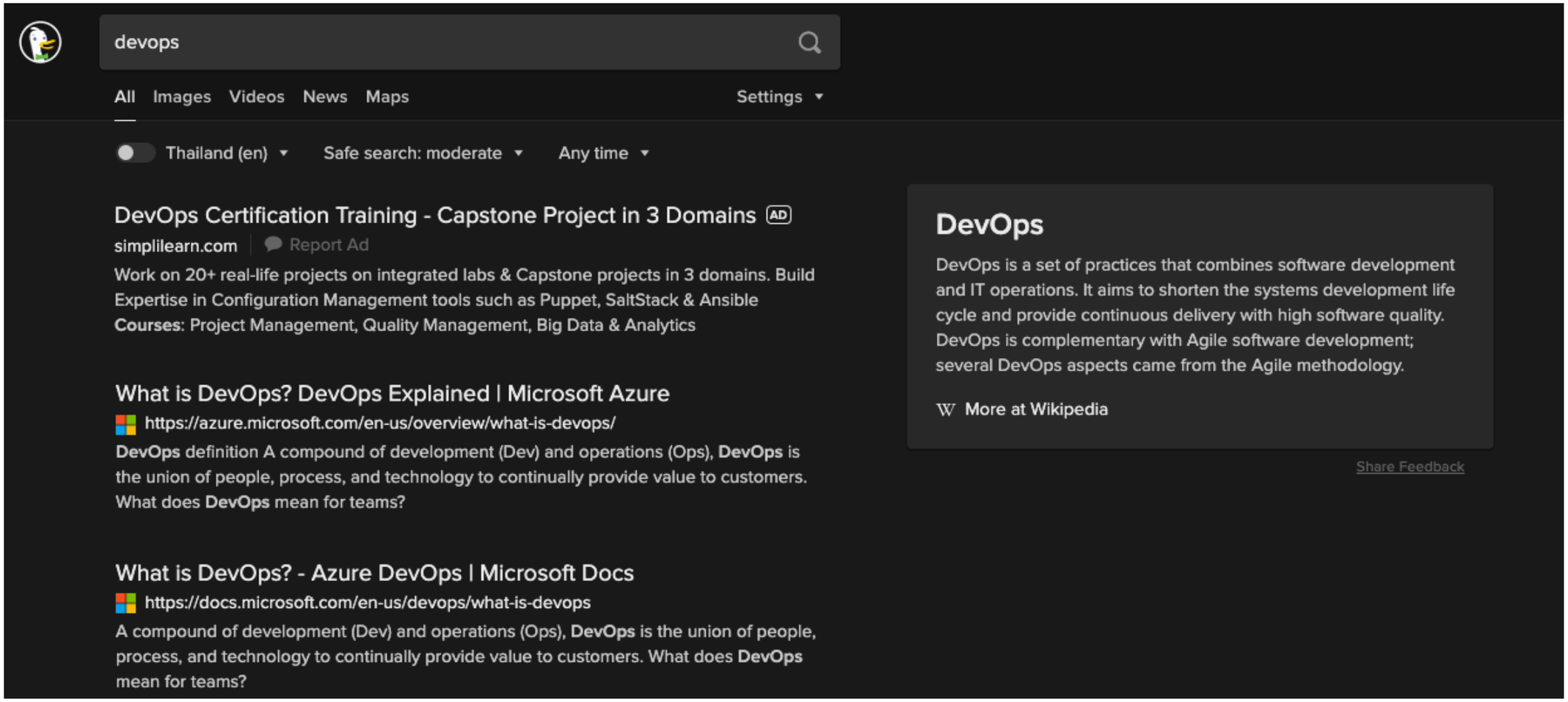
- **Click Indicators** — The system with more clicks on its results, on average, indicates a better ranking preference.

A/B testing at web search engines

- Employed to evaluate new hypotheses about search results.
- Prerequisite:
  - An established search engine with a significant user base.
  - Majority of users are on the existing system.

- Testing approach —
  - Redirect a minor segment of the user traffic to a variant system for comparative analysis.
  - Experimentation can be conducted using either interleaved results or separate full result sets.

What can be achieved from click profiling



User behavior

- User interactions with search engines can provide informed indications of relevance.
- Search logs potentially contain extensive data from user activities.
- Challenges
  - Data Quality — User behavior data may contain a high level of noise.
  - Interpretation — Deciphering the true intent behind user actions can be complex.
  - Spam — Malicious activities can distort the data.
  - Coverage — Not every query may have associated user behavior data.

Time for questions

A reading assignment

- Your assignment involves reading the research paper named **A Learn-to-Rank Approach for Predicting Road Cycling Race Outcomes** (attached in the MSTeams).
- After reviewing the paper, please answer to the following questions:
  - What are the main themes of the paper, and in what capacity is ranking utilized?
  - Could you explain the concept of Learning-to-Rank through this paper?
  - In what manner was the NDCG metric applied?
  - Does the model demonstrate sufficient reliability for predicting future outcomes? And why?