SE 481 Introduction to Information Retrieval

Module #5 — Evaluation

College of Arts, Media and Technology
Chiang Mai University

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

It is all about telling whether users are happy

- Is the returned search products relevant to users?
- Do the users click the search results a lot?
- How much users spend a lot of money after searching for products using the search engine?
- How long does it take for a user to reach the solutions after starting the search

Happiness — too elusive to measure

- It does not mean it cannot be measured
  - Similar to unit test vs acceptance test

- The unit-level test in IR is the **degree of relevancy** or **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 in the internet
  - Rapid increasing rates of documents everyday

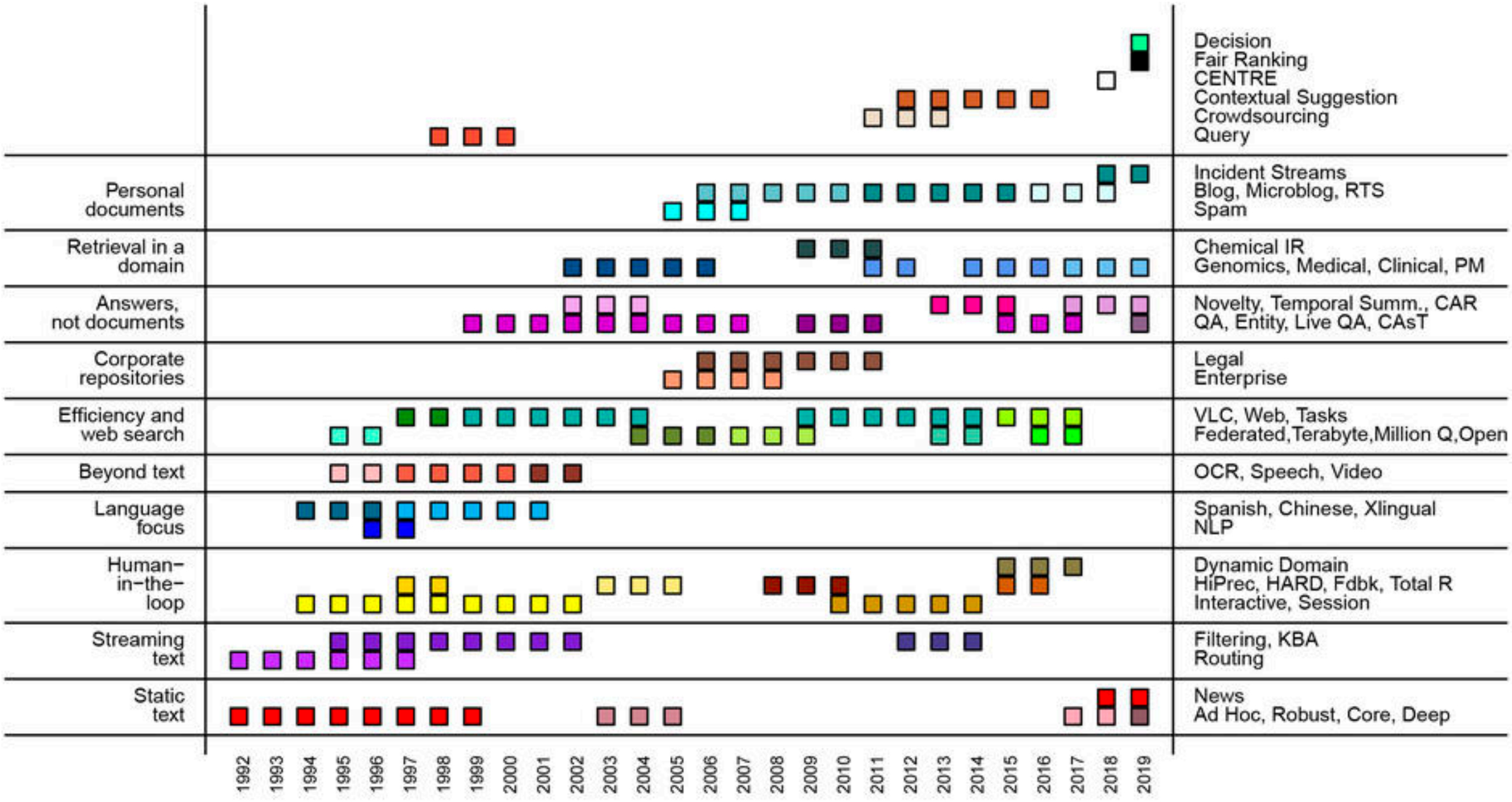
- Crowd-sourcing is commonly used
  - variance and quality might not be sufficiently high

- Testing on standard benchmarking collection is important
  - Difficult and representative questions

Standard benchmark

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

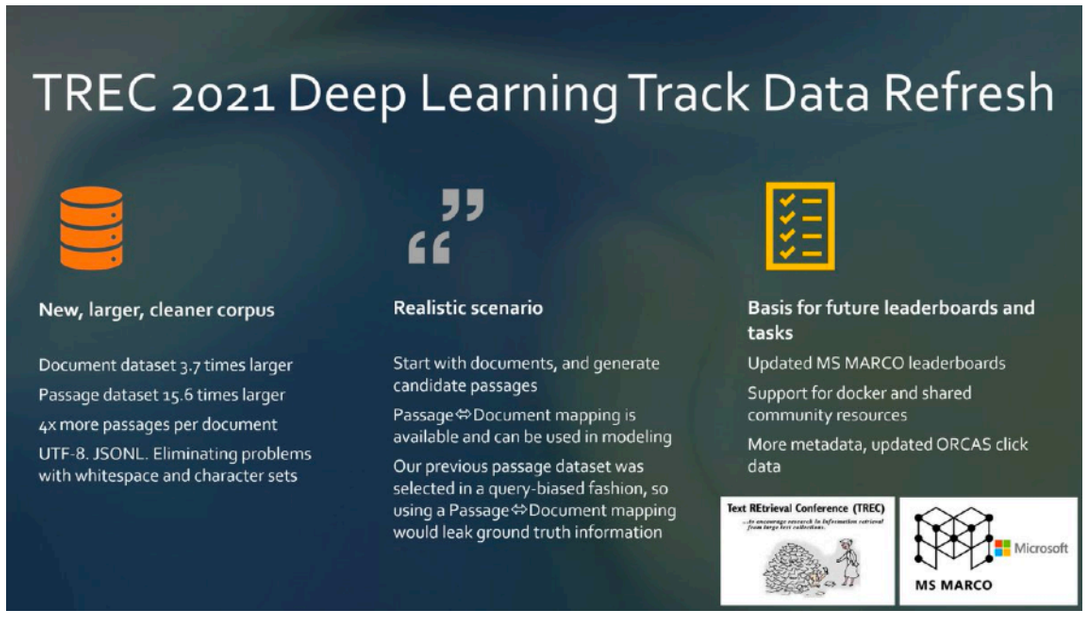
TREC



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

TREC

- Some recent interesting tracks



Evaluating an IR system

- User's need is translated into a query prior to any assessment
- Relevance is assessed relative to the user 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 **maybe** better query: **pdf parsing java**
  - **What to access:** Whether the returned documents can help the user to find a solution, not whether they contain these words.

Standard evaluation metric family

- Precision (for ranking)
- Recall (for ranking)
- DCG

Precision and Recall

- In unranked binary assessment they are defined as—
  - Precision — ratio between the correct prediction and total case of predicting true.
  - Recall — ratio between the correct prediction and total correct case.

- Let’s discuss some terms first
  - True Positives (TP) — number of predicted positive in the actual positive group
  - True Negatives (TN) — number of predicted negative in the actual negative group
  - False Positives (FP) — number of predicted positive in the actual negative group
  - False Negatives (FN) — number of predicted negative in the actual positive group

Precision and Recall in unranked problems

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

- Example

	Predicted Positive	Predicted Negative
Actual Positive	10 (TP)	15 (FN)
Actual Negative	25 (FP)	100 (TN)

- Precision = (10) / (10 +25) = 0.29
- Recall = (10) / (10 +15) = 0.4

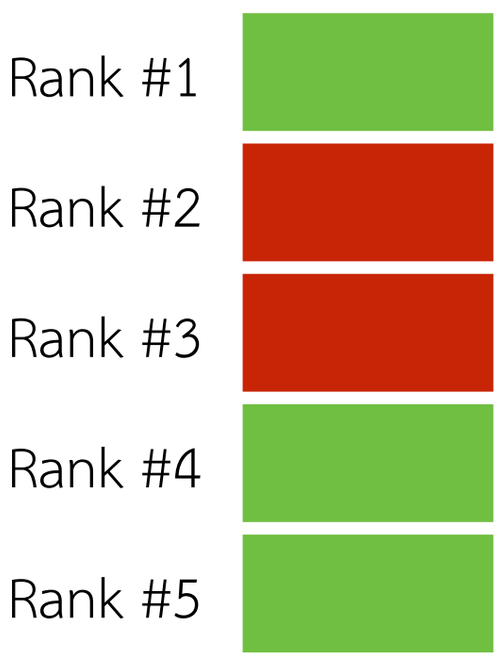
Standard evaluation metrics for ranking

- Binary relevance
  - Precision@K (P@K)
  - Recall@K (R@K)
  - Mean average precision (mAP)
  - Mean reciprocal rank (mRR)

- Multiple levels of relevance
  - Normalized discounted cumulative gain (NDCG)

Precision@K

- Set a rank threshold K
- Compute % relevant in top K
- Ignores documents ranked lower than K
- E.g.,

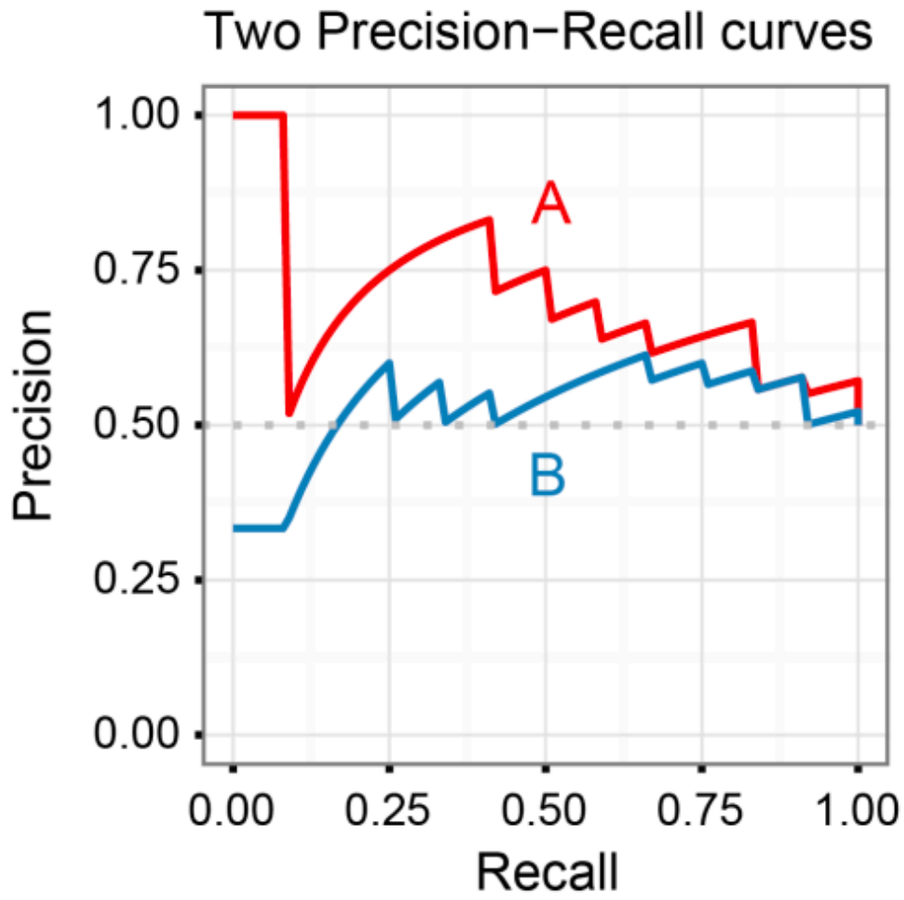


  - Precision@3 = 1/3
  - Precision@4 = 2/4
  - Precision@5 = 3/5

- Recall@K is in similar fashion

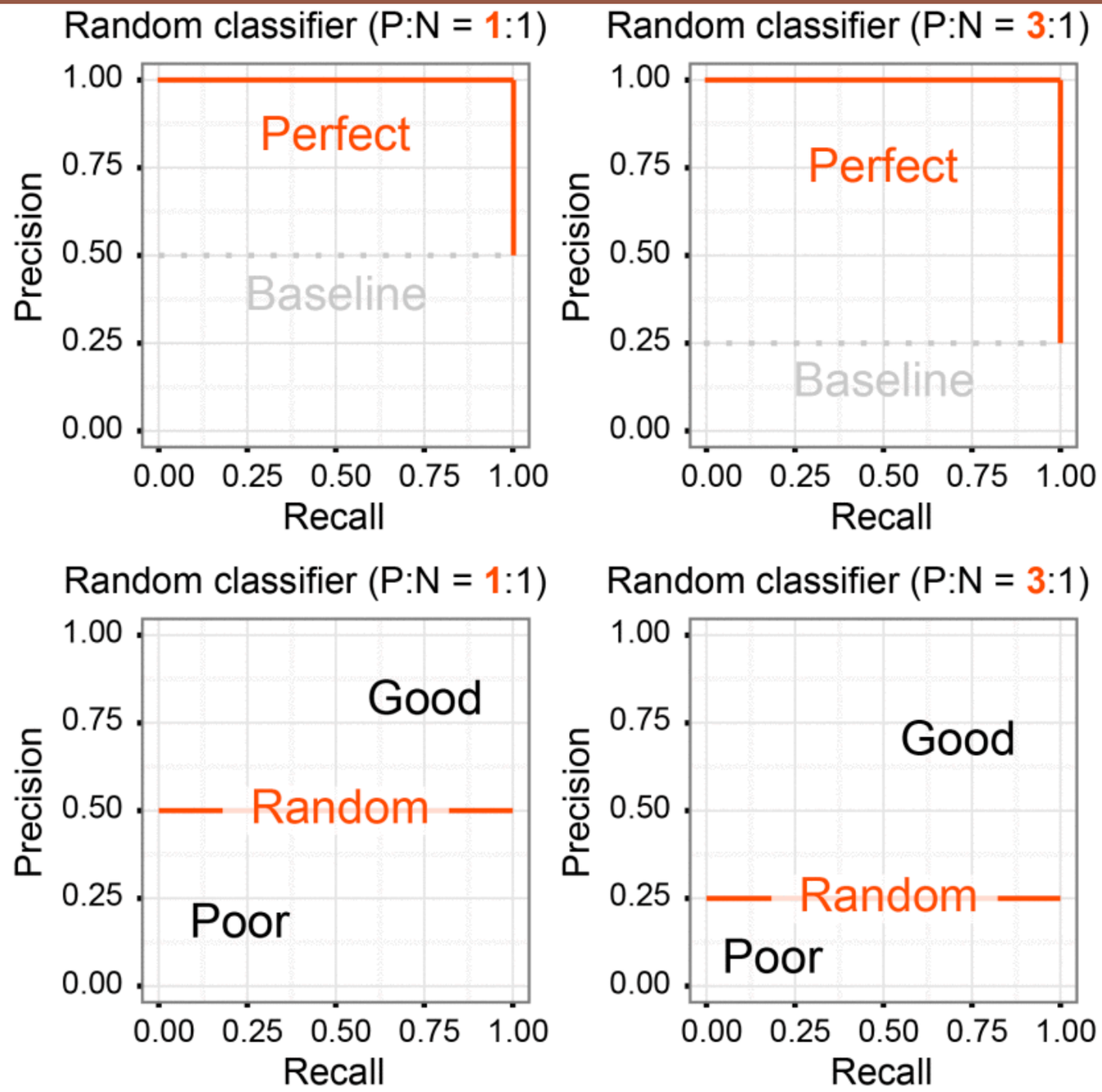
Precision-recall curve

- For comparing two or more information retrieval systems



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

Precision-recall curve



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

Mean average precision

- AP — the average of the precision scores at the rank locations of each relevant document
- mAP — the mean of the average precision scores for a group of queries
  - Macro average — All queries are considered equal

- If a relevant document never gets retrieved, we assume its corresponding precision as zero.

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

Mean reciprocal rank

- RR — reciprocal of the rank at which first correct response returned (or 0 if none)
- mRR — the mean of the RR scores for a group of queries

- Only cares about the single highest-ranked relevant item.

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

- With access to this kind of ranking scores, we will be able to roughly create the preferred rank list for each user
  - To build a personalized search engine

- Fundamental for the **learning to rank** approach

- Before that, we have to know more about the common evaluation metric for the case.
  - Discounted cumulative gain (DCG)

Discounted cumulative gain

- Multiple levels of relevance

- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant documents.
  - 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.

Discounted cumulative gain

- **Graded relevance** is the measure of usefulness from examining a document

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

- 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

- Let the relevance judgments be in a scale of [0, r] where r>2
- CG at rank n = $r_1 + r_2 + \ldots + r_n$
- DCG at rank n = $r_1 + r_2/\log_2 2 + r_3/\log_2 3 + \ldots + r_n/\log_2 n$

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

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

- Normalize DCG at rank n by the DCG value at rank n of the known-to-be best ranking
  - i.e., the ideal ranking — the ranking that perfectly order the documents by the degree of relevance

- Normalization is useful for demonstrating queries with varying numbers of relevant results

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		$NDCG_{function2}$ = 0.9203	

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

$$DCG_{function_1} = 2 + \left(\frac{2}{log_2 2} + \frac{1}{log_2 3} + \frac{0}{log_2 4}\right) = 4.6309$$

$$DCG_{function_2} = 2 + \left(\frac{1}{log_2 2} + \frac{2}{log_2 3} + \frac{0}{log_2 4}\right) = 4.2619$$

$$MaxDCG = DCG_{ideal} = 4.6309$$

It is all about telling whether users are happy

- Is the returned search products relevant to users?
- Do the users click the search results a lot?
- How much users spend a lot of money after searching for products using the search engine?
- How long does it take for a user to reach the solutions after starting the search

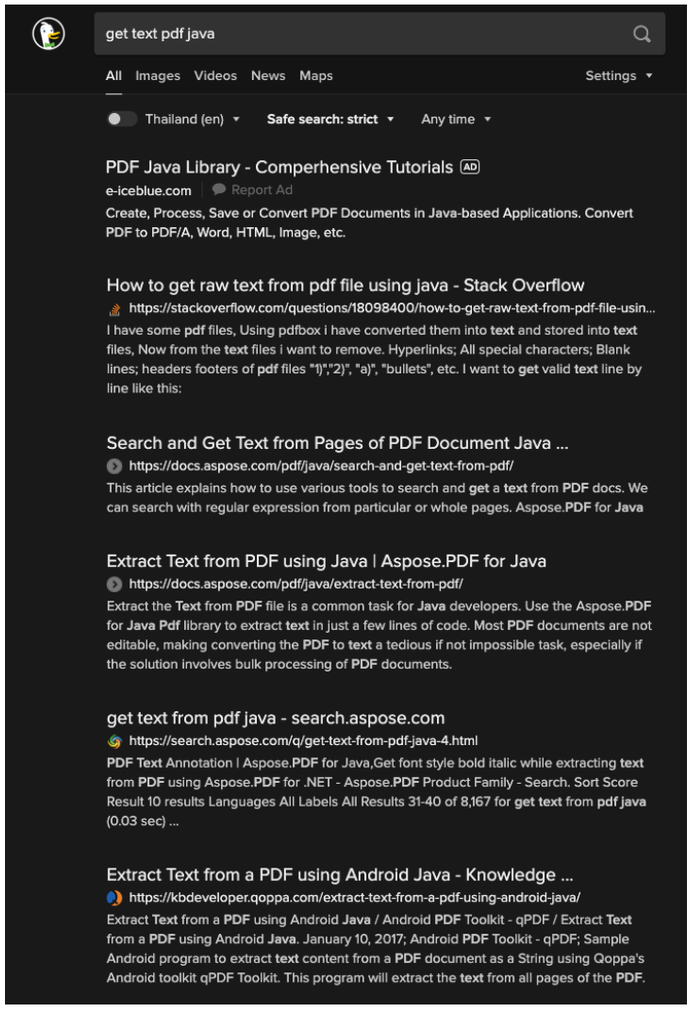
Why don’t we measure user happiness directly

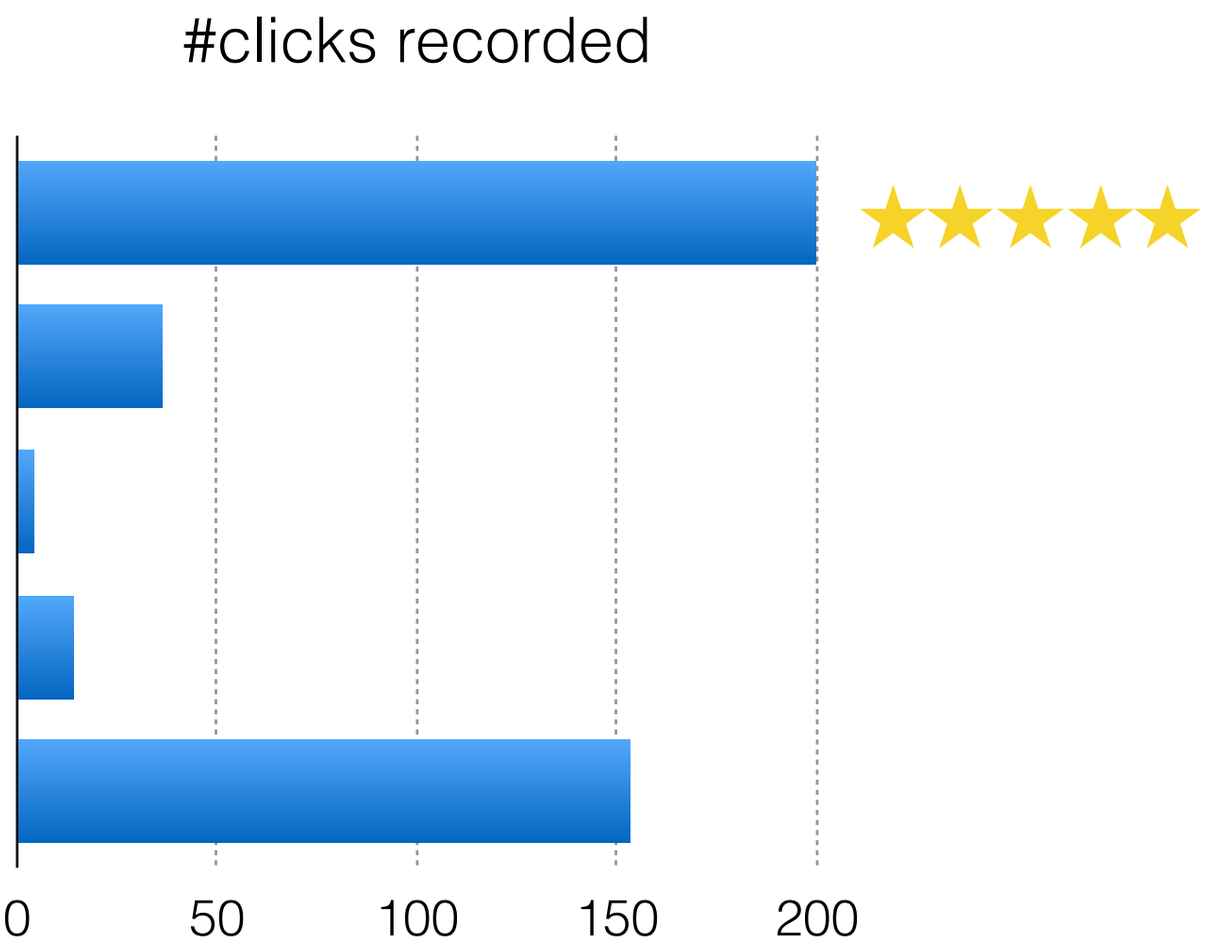
- Human judgments are
  - Subjective
  - Inconsistent — between raters and over time
  - Decay in value as documents | query evolves, i.e., how we answer lengthily questionnaire
  - Not easy to represent all real users

- Anyway, it is worth to give it a try

Measuring user clicks

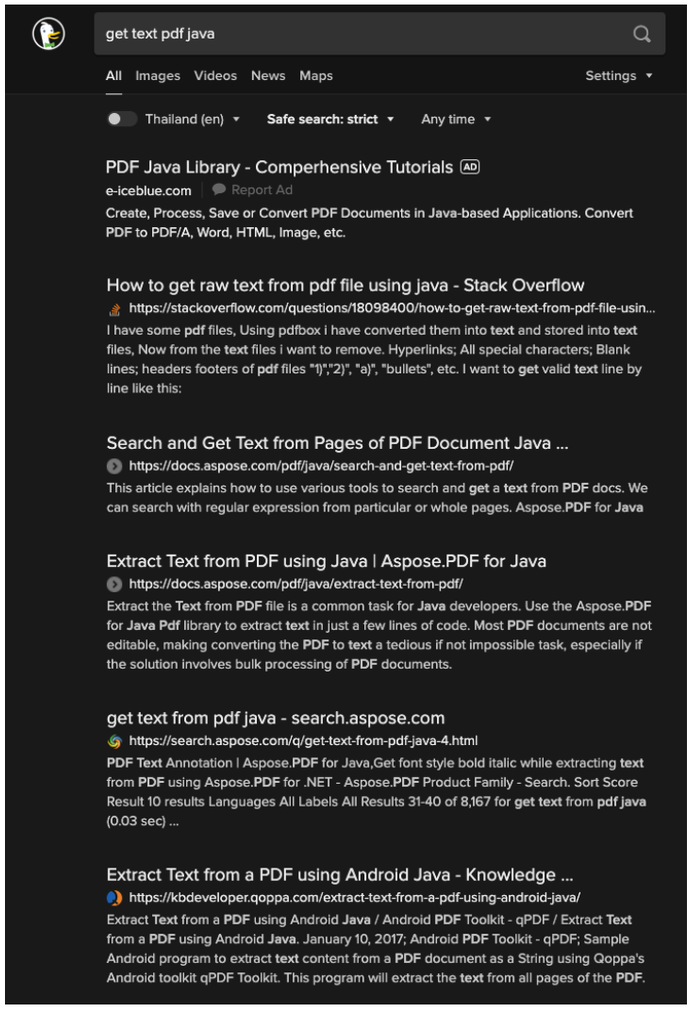
- One of the most straightforward approaches to imply a user behavior learning

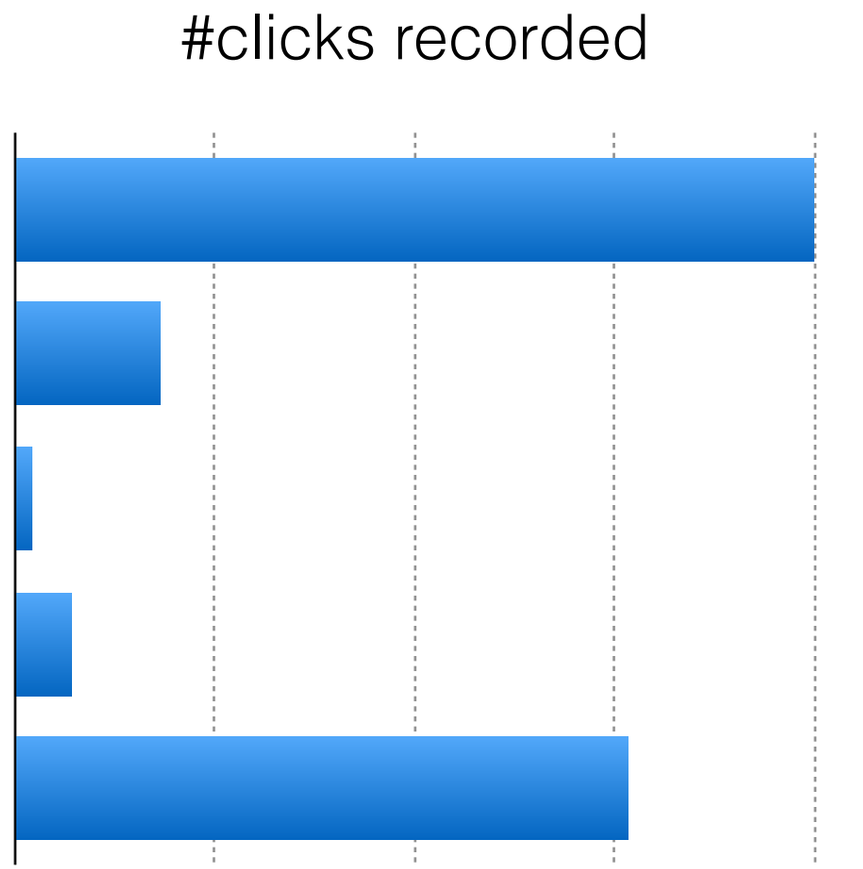




Exploiting the #user clicking information

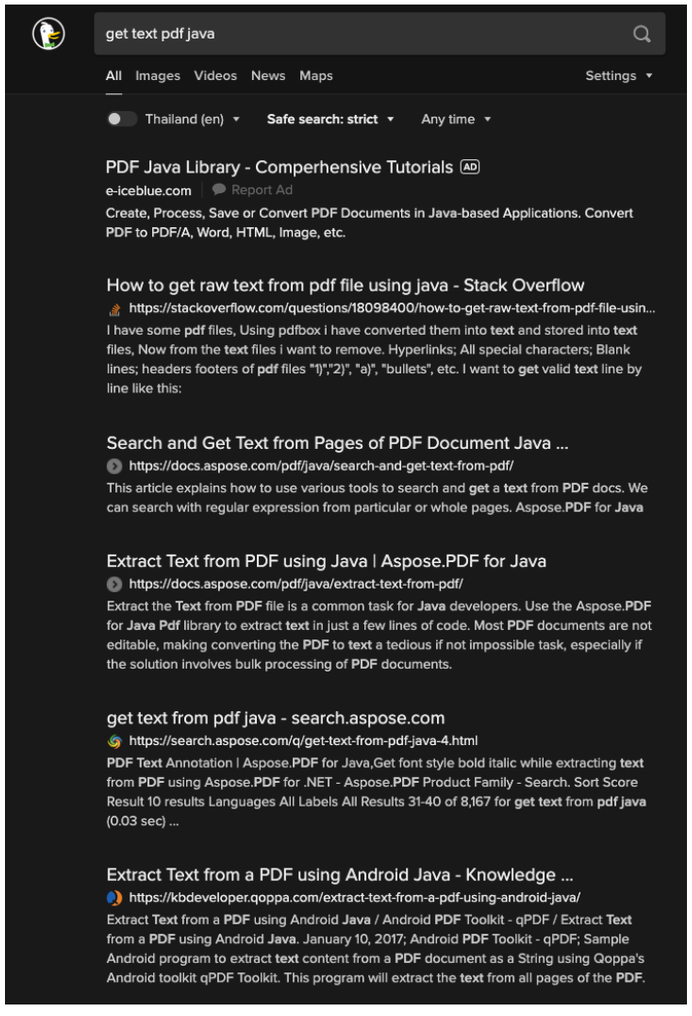
- Adapt ranking to user clicks

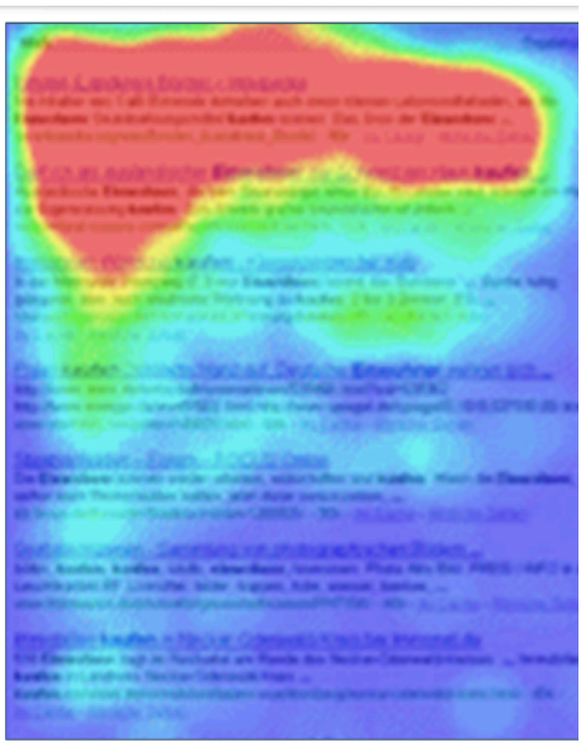




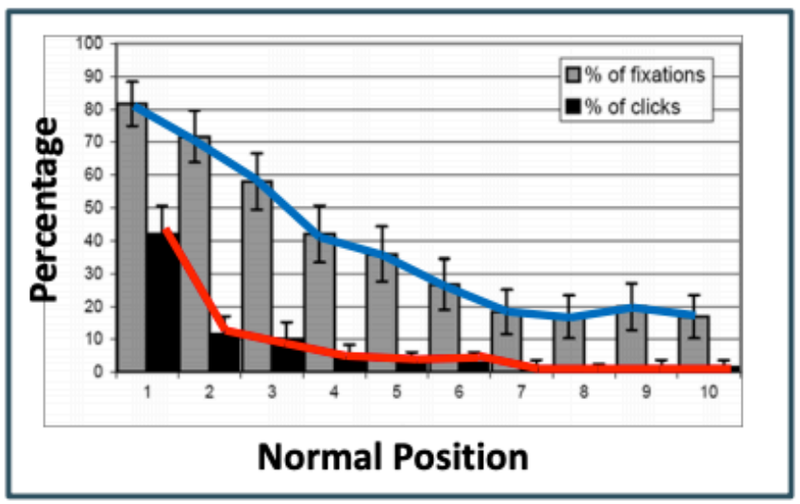
Eye-tracking

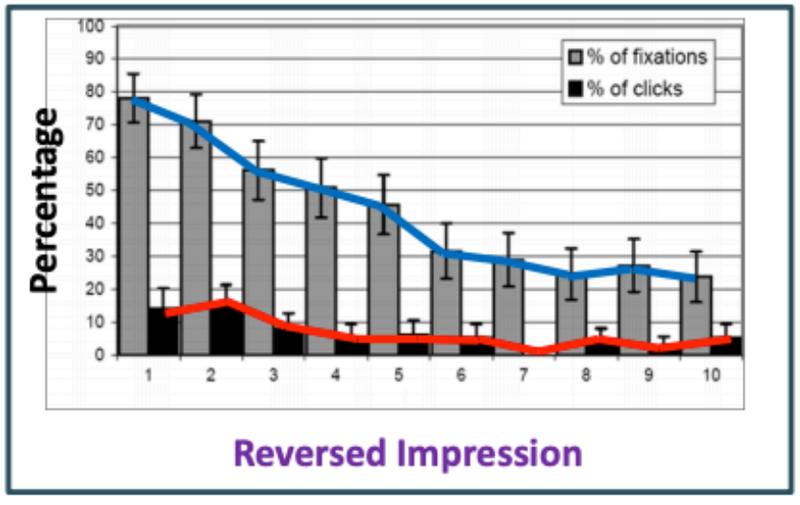
- One of the most straightforward approaches to imply a user behavior learning





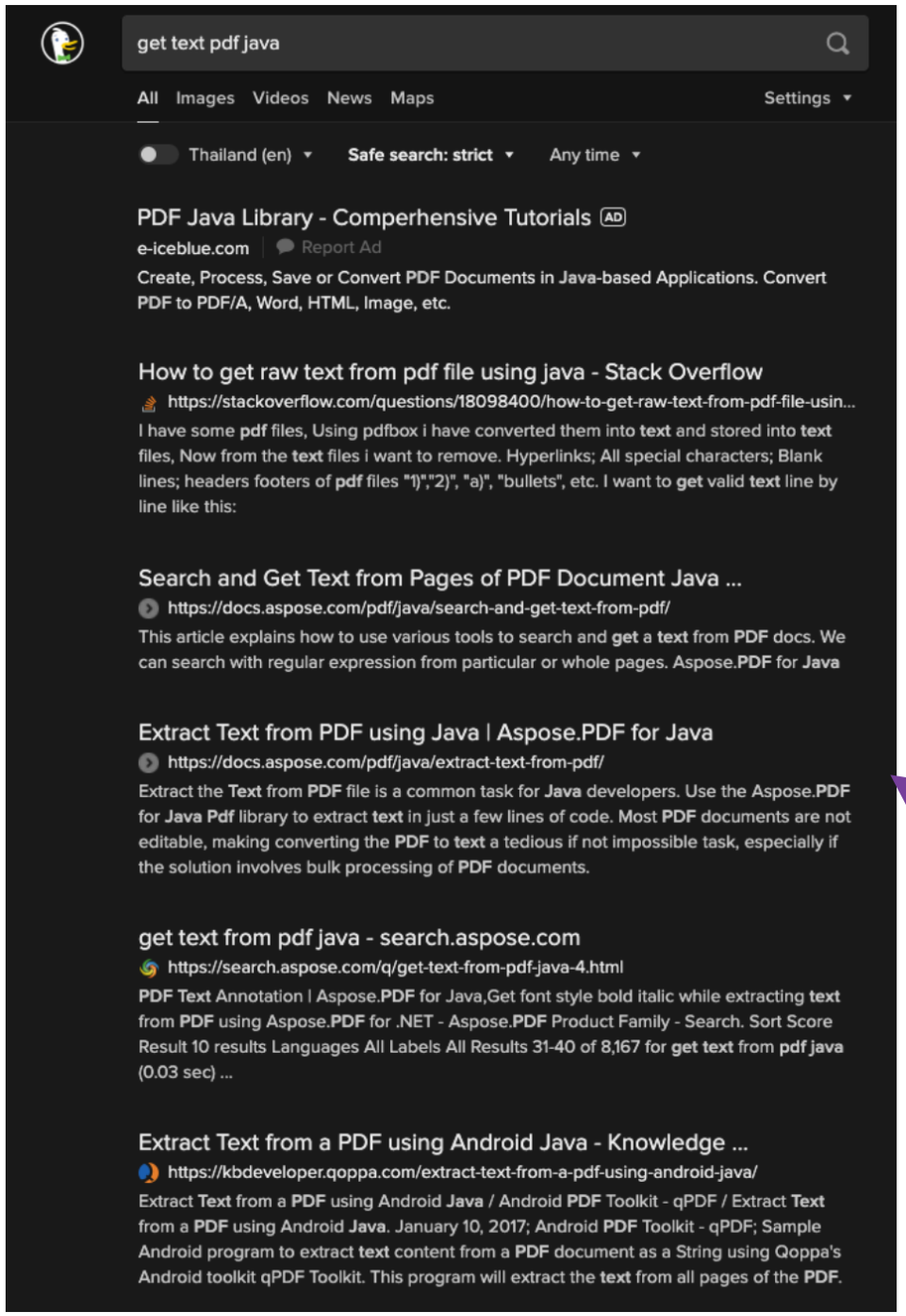
Informative but biased





- Higher positions receive more attention and more clicks
- Even the order is reversed

Relative rating



User's click sequence

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

Evaluating pairwise relative ratings

- Pairs of the form: **Doc_a** is better than **Doc_b** for a query **q**
- Not to assess a rank-ordering wrt per-doc relevance assessments but
  - to assess in terms of conformance with historical pairwise preferences recorded from user clicks

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

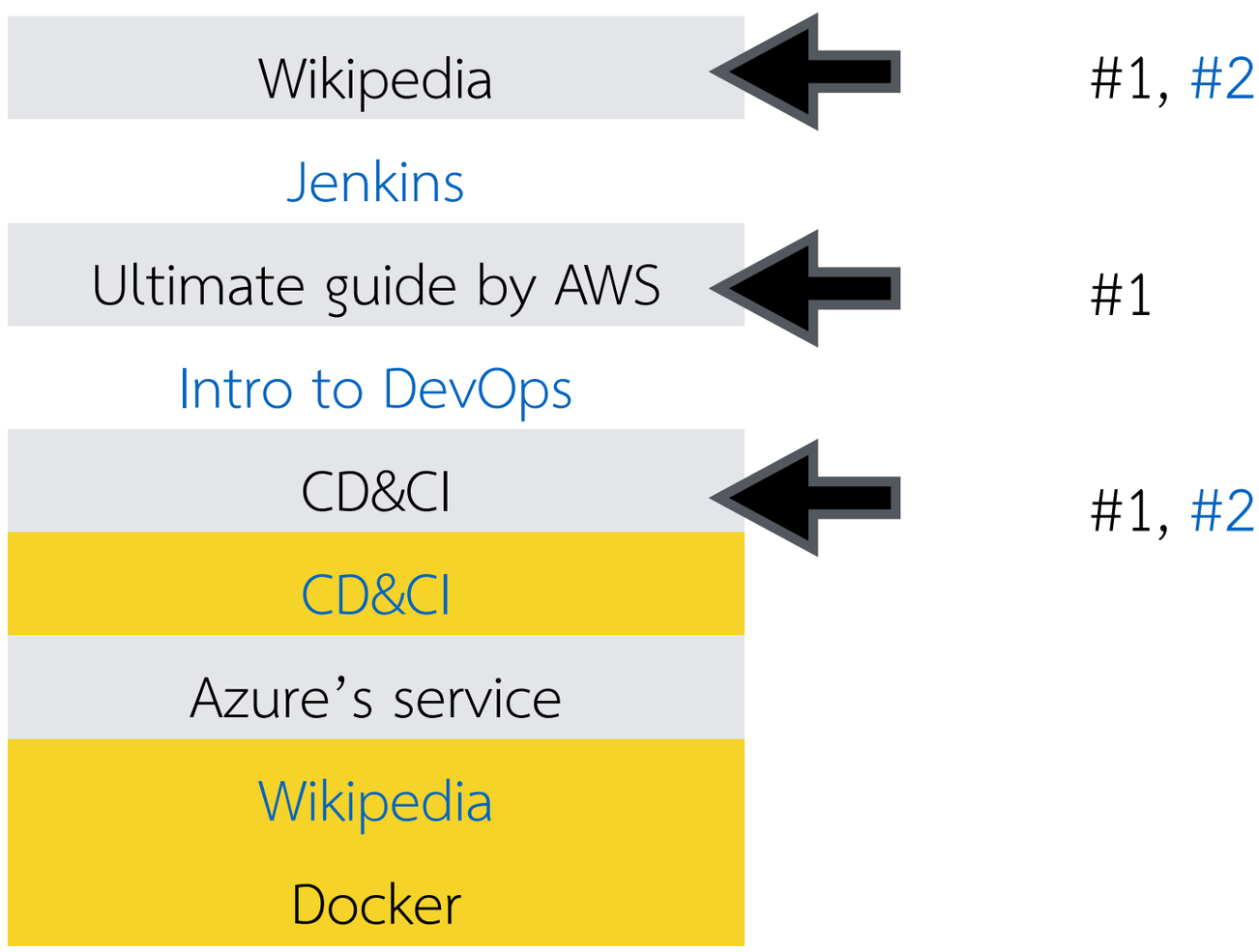
Docker

E.g., comparing two rankings via clicks

- Count user clicks

Ranking #1: 3

Ranking #2: 2



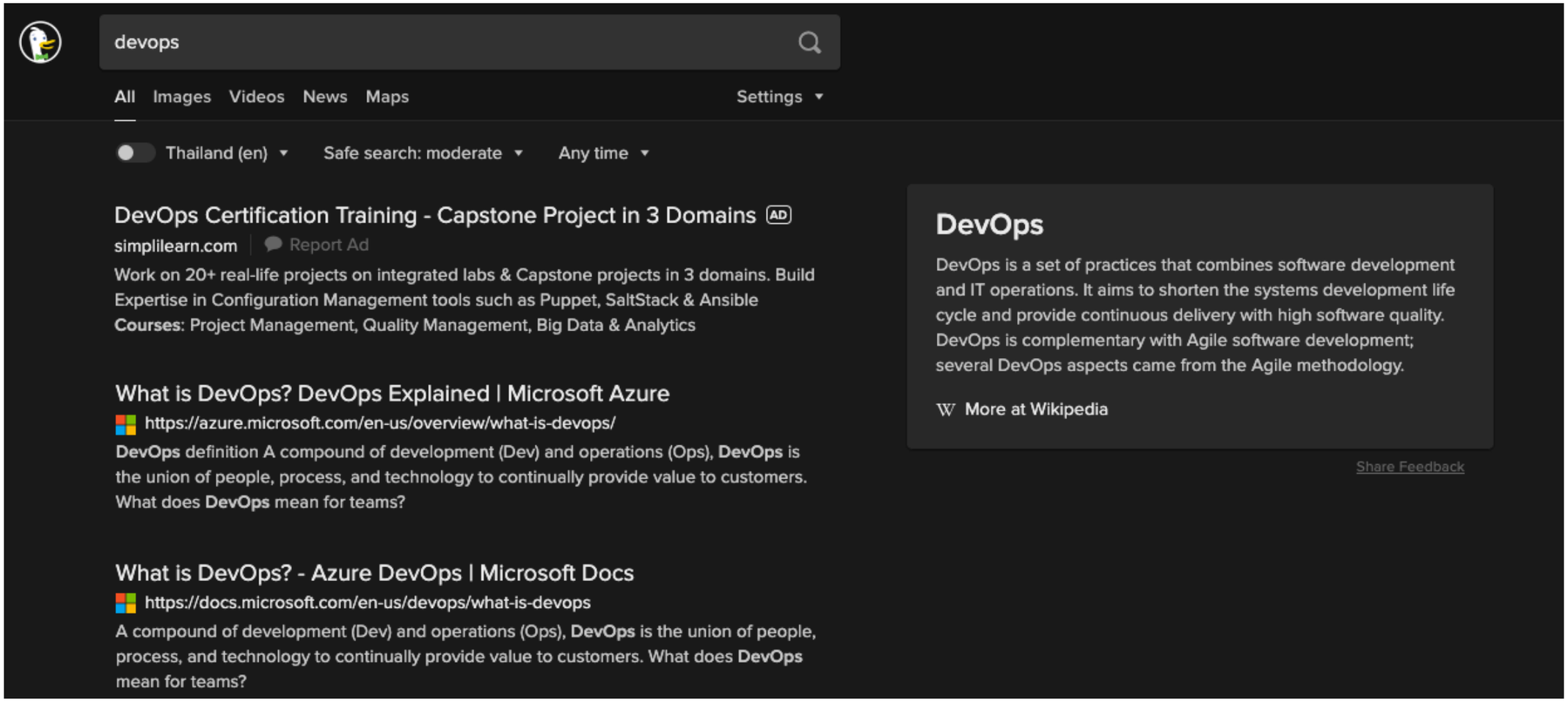
Interleaved ranking

- Present interleaved ranking to users
- Count clicks on results from #1 versus results from #2B
- Better ranking will (on average) get more clicks

A/B testing at web search engines

- To test new assumption about search results
- Prerequisite:
  - A large search engine is already up and running
  - Most users are using the old system
- What to do —
  - Divert a small portion of traffic to an experimental site and evaluate it.
  - Test can be done on either interleaved or full result.

What can be achieved from click profiling



User behavior

- User behavior is an intriguing source of relevance data:
  - Users make (somewhat) informed choices when they interact with search engines
  - Potentially a lot of data available in search logs
- Challenges
  - User behavior data can be very noisy
  - Interpreting user behavior can be tricky
  - Spam can be a significant problem
  - Not all queries will have user behavior

Time for questions