CSCE 670 - Information Storage and Retrieval

Week 4: Evaluation of Search Engines

Yu Zhang
yuzhang@tamu.edu

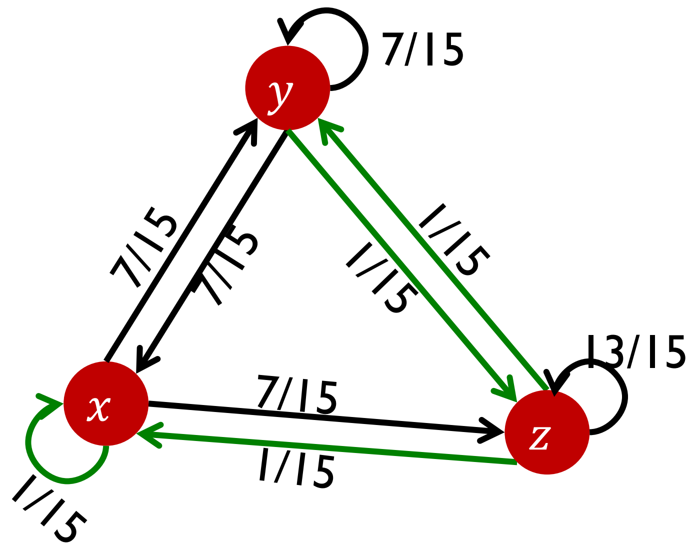
Course Website: https://yuzhang-teaching.github.io/CSCE670-S26.html

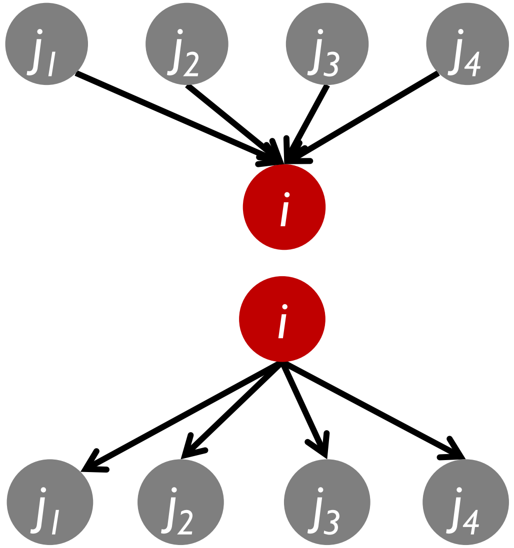
Adapted from the slides by Prof. James Caverlee

Recap: PageRank and HITS

- How to identify important pages given the hyperlink graph of webpages?
  - PageRank $(\beta \boldsymbol{A} + (1-\beta)\frac{\mathbf{1}}{N})$
  - HITS ($\boldsymbol{A}^T\boldsymbol{A}$ and $\boldsymbol{A}\boldsymbol{A}^T$)

- Variant of PageRank
  - Topic-Sensitive PageRank: only teleport into a topic-specific set of pages
  - Combating Link Farming: only teleport into trusted pages





Our Plan: Ranking

- ✅ Why is ranking important?
- ✅ What factors impact ranking?
- ✅ Two foundational text-based approaches: TF-IDF and BM25
- ✅ Two foundational link-based approaches: PageRank and HITS
- Evaluation
  - How do we know if we are doing a good job?
- Combining scoring functions (BM25, PageRank, …)
  - By hand
  - Using machine learning – "Learning to rank"

The Importance of Evaluation

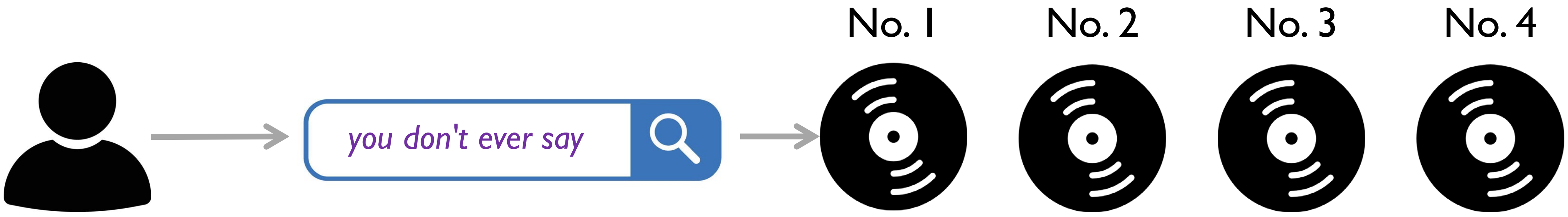
- Critical step for understanding if our algorithm actually does anything net positive

- The ability to measure differences underlies experimental science
  - How well does an algorithm work? (E.g., *provide performance metrics for the BM25 algorithm*)
  - Is Algorithm A better than Algorithm B? (E.g., *BM25 vs. TF-IDF*)
  - Under what conditions? (*longer documents? longer queries? …*)
  - To what extent? (*by 5%? 1%? 0.001%?*)

- Evaluation drives what to research
  - Identify techniques that work and that do not

Evaluating a Search Engine

- Evaluation frameworks should be targeted to the application scenario:
  - Typically, different metrics and approaches for ranking, classification, recommender systems, …
- Today: Evaluating a search engine



A ranked list of albums.

User Happiness

- Often, we would like to measure "user happiness" for a search engine
- Any ideas?

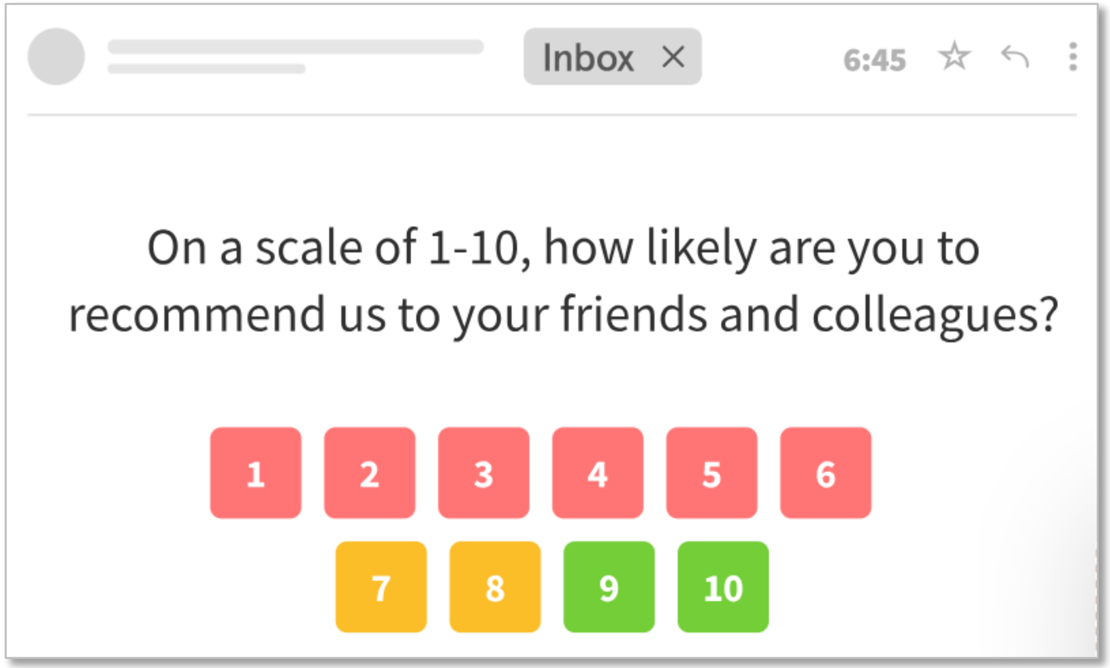
- Examples that are easy to measure but (possibly) NOT important
  - How might we "optimize" the following metrics while leading to worse results for our customers?
  - Example 1: Time spent on website (Objective: MAX)
  - Example 2: Time until purchase (Objective: MIN)

- Cranfield Experiments (1957-1966)
  - Led by Cyril Cleverdon from Cranfield University
  - Conclusion: user happiness ≅ relevance of search results

Measuring Relevance

- Suppose you have invented a new ranking algorithm, *SuperRank*, for our record store
- You believe *SuperRank* performs exceptionally well (even better than BM25). How would you go about proving that?

- Online Evaluation
  - Implement BM25 and *SuperRank* on our store website
  - Ask users to rate the ranking results
  - Compare the average user ratings to see which algorithm performs better

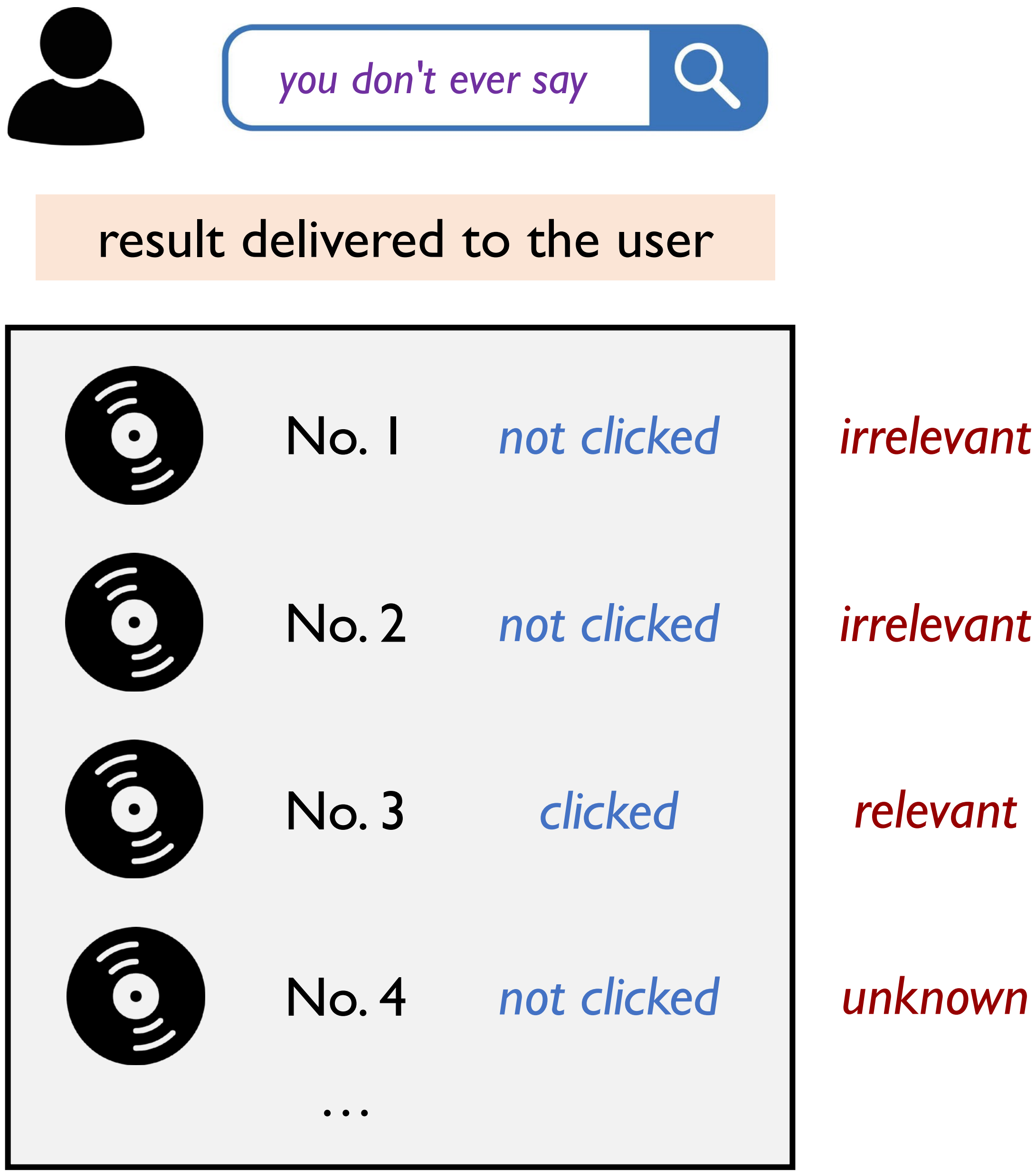


Measuring Relevance

- Drawbacks of Online Evaluation?
  - What if *SuperRank* performs quite poorly?
  - We will lose potential customers because of this experiment!
- Offline Evaluation
  - Simulate an online experiment
  - A benchmark document collection
    - No need to use every CD in the store, but we should select a sufficiently large and representative sample to cover all categories
  - A benchmark suite of queries
    - Do our best to create/collect a sufficiently large and representative set of queries

Measuring Relevance

- Offline Evaluation
  - Simulate an online experiment
  - A benchmark document collection
  - A benchmark suite of queries
  - A binary assessment of either Relevant or Irrelevant for each query and each document
    - Human annotations OR previous user queries and clickthrough data
- Start the online experiment only after offline experiments have confirmed that *SuperRank* outperforms BM25



Offline Evaluation for Different Domains

- BEIR benchmark (NeurIPS 2021): https://github.com/beir-cellar/beir

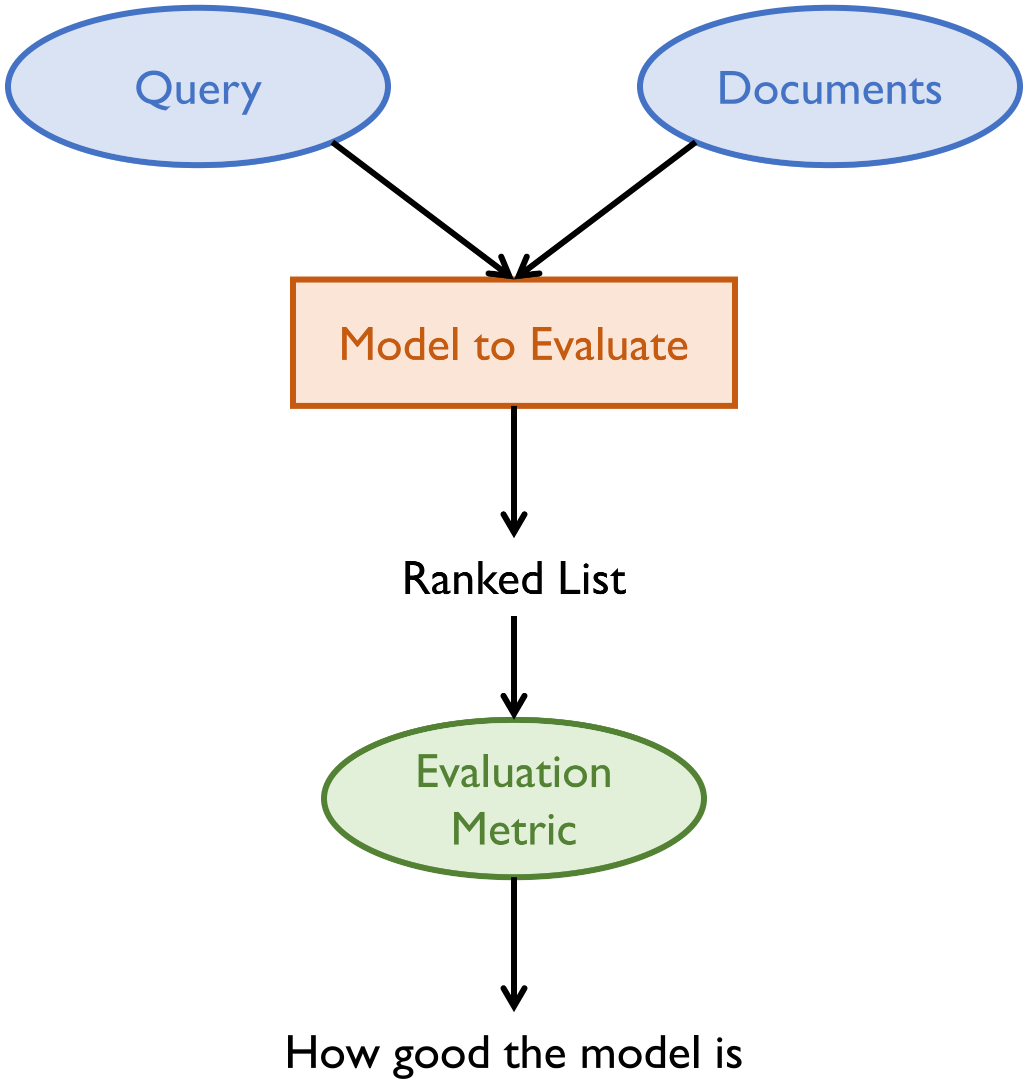
Split (→)					Train	Dev	Test			Avg. Word Lengths	
Task (↓)	Domain (↓)	Dataset (↓)	Title	Relevancy	#Pairs	#Query	#Query	#Corpus	Avg. D / Q	Query	Document
Passage-Retrieval	Misc.	MS MARCO [45]	✗	Binary	532,761	—	6,980	8,841,823	1.1	5.96	55.98
Bio-Medical Information Retrieval (IR)	Bio-Medical	TREC-COVID [65]	✓	3-level	—	—	50	171,332	493.5	10.60	160.77
	Bio-Medical	NFCorpus [7]	✓	3-level	110,575	324	323	3,633	38.2	3.30	232.26
	Bio-Medical	BioASQ [61]	✓	Binary	32,916	—	500	14,914,602	4.7	8.05	202.61
Question Answering (QA)	Wikipedia	NQ [34]	✓	Binary	132,803	—	3,452	2,681,468	1.2	9.16	78.88
	Wikipedia	HotpotQA [76]	✓	Binary	170,000	5,447	7,405	5,233,329	2.0	17.61	46.30
	Finance	FiQA-2018 [44]	✗	Binary	14,166	500	648	57,638	2.6	10.77	132.32
Tweet-Retrieval	Twitter	Signal-1M (RT) [59]	✗	3-level	—	—	97	2,866,316	19.6	9.30	13.93
News Retrieval	News	TREC-NEWS [58]	✓	5-level	—	—	57	594,977	19.6	11.14	634.79
	News	Robust04 [64]	✗	3-level	—	—	249	528,155	69.9	15.27	466.40
Argument Retrieval	Misc.	ArguAna [67]	✓	Binary	—	—	1,406	8,674	1.0	192.98	166.80
	Misc.	Touché-2020 [6]	✓	3-level	—	—	49	382,545	19.0	6.55	292.37
Duplicate-Question Retrieval	StackEx.	CQADupStack [25]	✓	Binary	—	—	13,145	457,199	1.4	8.59	129.09
	Quora	Quora	✗	Binary	—	5,000	10,000	522,931	1.6	9.53	11.44
Entity-Retrieval	Wikipedia	DBPedia [21]	✓	3-level	—	67	400	4,635,922	38.2	5.39	49.68
Citation-Prediction	Scientific	SCIDOCS [9]	✓	Binary	—	—	1,000	25,657	4.9	9.38	176.19
Fact Checking	Wikipedia	FEVER [60]	✓	Binary	140,085	6,666	6,666	5,416,568	1.2	8.13	84.76
	Wikipedia	Climate-FEVER [14]	✓	Binary	—	—	1,535	5,416,593	3.0	20.13	84.76
	Scientific	SciFact [68]	✓	Binary	920	—	300	5,183	1.1	12.37	213.63

Evaluation Metrics

- Precision
- Recall
- F1 Score

- Precision@*k*
- MAP
- NDCG

- There are many more metrics!



Precision and Recall

$$\text{Precision} = \frac{\#\ \text{retrieved documents that are relevant}}{\#\ \text{retrieved documents}}$$

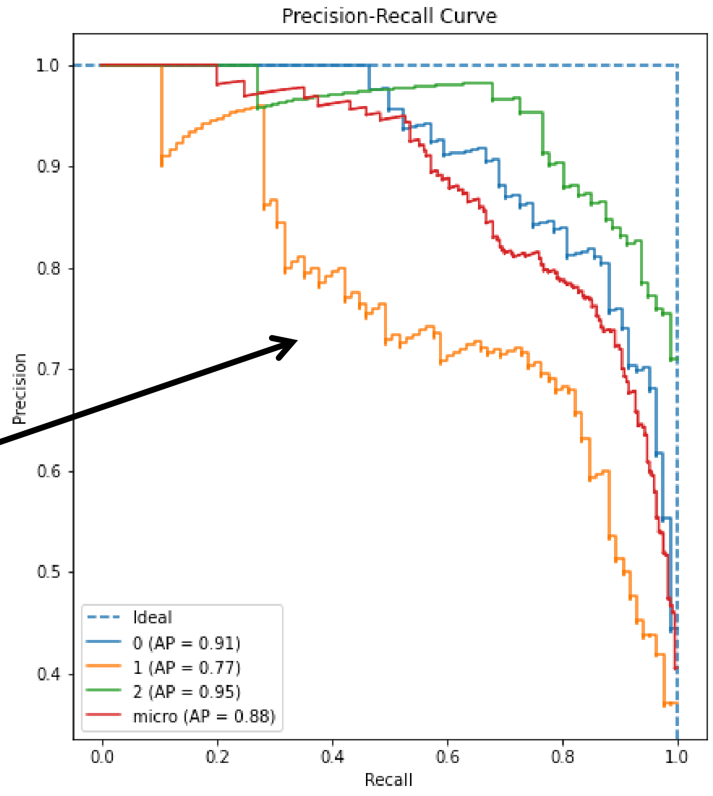
$$\text{Recall} = \frac{\#\ \text{retrieved documents that are relevant}}{\#\ \text{relevant documents}}$$

- Example
  - There are 10,000 candidate documents. Given the query "*meet me at midnight*", 100 documents are labeled as Relevant, the other 9,900 are labeled as Irrelevant.
  - Your SuperRank algorithm retrieves 20 documents for the query "*meet me at midnight*", among which 12 are Relevant and 8 are Irrelevant.
  - Precision $= \frac{12}{20} = 0.60$
  - Recall $= \frac{12}{100} = 0.12$

Trade-off Between Precision and Recall

- *"A Recall of 0.12 is too low. How can we improve it?"*
- *"Retrieving only 20 documents is too limited. Even the results were perfect, we would cap the Recall at 0.2. Why not relax the constraints (e.g., lower the BM25 score threshold) to retrieve more documents?"*

- Typically, when you retrieve more documents, Recall increases, but Precision tends to decrease.
  - Because the additional documents you retrieve are ones the ranking model is increasingly uncertain about in terms of relevance.
  - Examples of Precision-Recall curves



Trade-off Between Precision and Recall

- *SuperRank* ranking over all 6 candidate CDs:

✓ score: 0.96

✗ score: 0.93

cutoff: 0.9
precision: 0.50, recall: 0.33

✓ score: 0.85

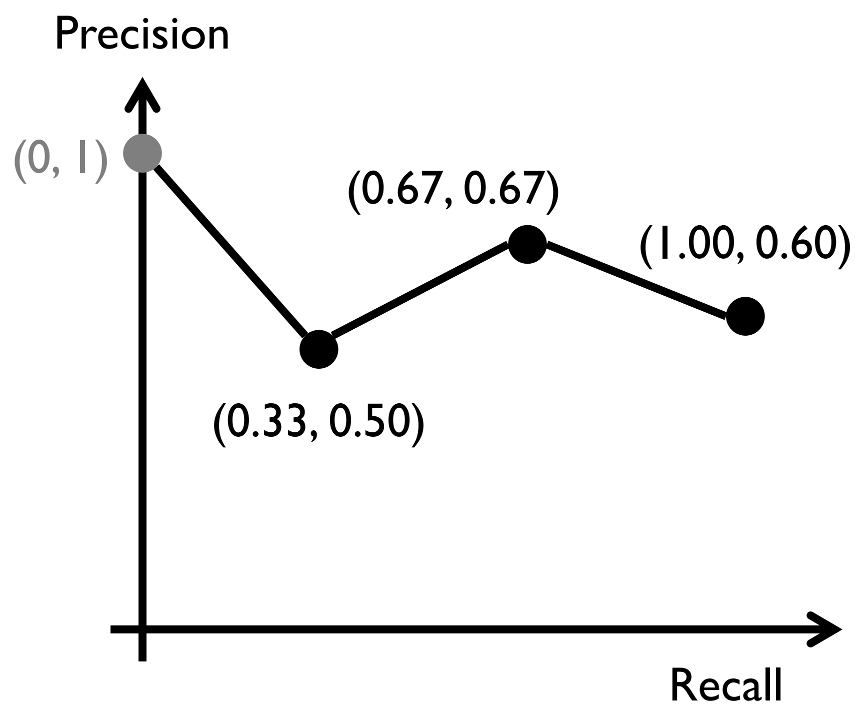
cutoff: 0.8
precision: 0.67, recall: 0.67

✗ score: 0.76

✓ score: 0.73

cutoff: 0.7
precision: 0.60, recall: 1.00

✗ score: 0.55



F1 Score: Combining Precision and Recall

$$\text{F1} = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- F1 is the harmonic mean of Precision and Recall.
- To make F1 large, both Precision and Recall need to be large. Even a very large Precision cannot make up for a very small Recall.

- Example
  - If Precision = 0.60 and Recall = 0.12, what is the F1 score?
  - F1 $= \frac{2 \times 0.60 \times 0.12}{0.60 + 0.12} = 0.20$ (far away from 0.60, close to 0.12)

  - How would you optimize the F1 score if we know the Precision-Recall curve is Precision + Recall = 0.72?

Questions?

Position-Aware Evaluation Metrics

- Given a query, suppose two algorithms, *A* and *B*, each retrieve 4 documents.
- Below are the relevance labels (1 = relevant, 0 = irrelevant) for the 4 documents, listed in order from the top-ranked to the lowest-ranked document by each algorithm:
    - Algorithm *A*: [1, 1, 0, 0]
    - Algorithm *B*: [0, 0, 1, 1]
- Which algorithm is better?
- By default, in an IR system, we always assume that users read the ranking results from top to bottom. Therefore, if Algorithm *A* allows users to find relevant documents more quickly, it should be considered better than Algorithm *B*.
- However, both sets of results have identical Precision, Recall, and F1 scores.
    - We need some other metrics that can distinguish *A* from *B*.

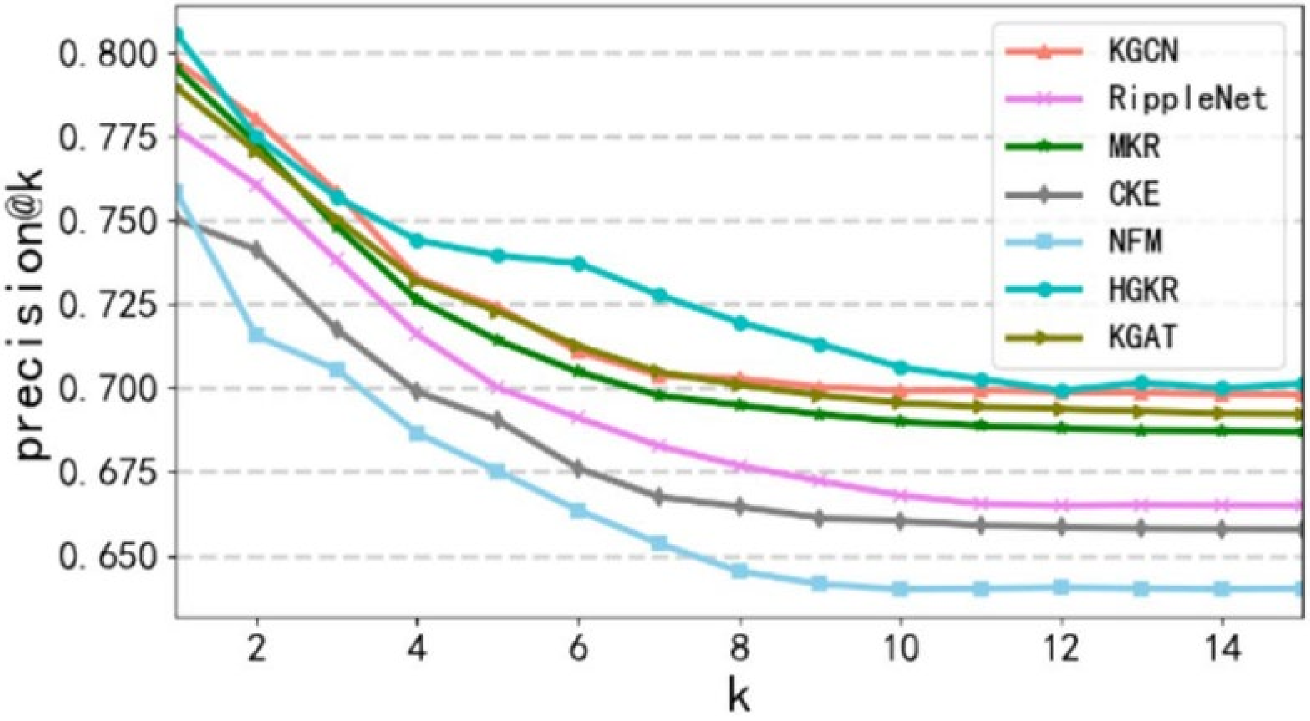
Precision@*k* (a.k.a., P@*k*)

$$\text{P@}k = \frac{\#\text{ retrieved documents that are relevant in the top } k}{k}$$

- Example
  - Algorithm *A*: [1, 1, 0, 0]
    - P@1 = 1/1 = 1.00
    - P@2 = 2/2 = 1.00
    - P@3 = 2/3 = 0.67
    - P@4 = 2/4 = 0.50
  - Algorithm *B*: [0, 0, 1, 1]
    - P@1 = 0/1 = 0.00
    - P@2 = 0/2 = 0.00
    - P@3 = 1/3 = 0.33
    - P@4 = 2/4 = 0.50
- Except for P@4 (i.e., Precision), Algorithm *A* is always better.

Precision@*k* (a.k.a., P@*k*)

- Examples of Precision@*k* curves



Although HGKR and KGCN are very close at P@14, HGKR's curve is generally above KGCN's and should therefore be considered the better performer.

- How can we summarize the height of a curve into a single metric (a numerical value)?

Mean Average Precision (MAP)

- Assume there are only 2 relevant documents in total.
- Algorithm *A*'s retrieval result: [1, 1, 0, 0]

- Step 1: Get the positions of all the relevant documents
  - k = 1 and k = 2

- Step 2: Compute P@k at each of those positions
  - P@1 = 1.00 and P@2 = 1.00

- Step 3: Take the average of these P@k values
  - MAP = (P@1 + P@2)/2 = 1.00
  - The only 2 relevant documents are ranked in the top 2 positions, so the algorithm deserves a perfect score.

Mean Average Precision (MAP)

- Assume there are only 2 relevant documents in total.
- Algorithm *B*'s retrieval result: [0, 0, 1, 1]

- Step 1: Get the positions of all the relevant documents
  - $k = 3$ and $k = 4$

- Step 2: Compute P@k at each of those positions
  - P@3 = 0.33 and P@4 = 0.50

- Step 3: Take the average of these P@k values
  - MAP = (P@3 + P@4)/2 = 0.42

Mean Average Precision (MAP)

- Assume there are 3 relevant documents in total.
- Algorithm *A*'s retrieval result: [1, 1, 0, 0]

- Step 1: Get the positions of all the relevant documents
  - k = 1 and k = 2

- Step 2: Compute P@k at each of those positions. When a relevant document is not retrieved at all, its corresponding "P@k" should be 0.
  - P@1 = 1.00 and P@2 = 1.00
  - The 3rd relevant document is not retrieved at all, so P@k = 0.

- Step 3: Take the average of these P@k values
  - MAP = (P@1 + P@2 + 0)/3 = 0.67

Mean Average Precision (MAP)

- Assume there are 4 relevant documents in total.
- Algorithm *A*'s retrieval result: [1, 1, 0, 0]

- Step 1: Get the positions of all the relevant documents
  - k = 1 and k = 2

- Step 2: Compute P@k at each of those positions. When a relevant document is not retrieved at all, its corresponding "P@k" should be 0.
  - P@1 = 1.00 and P@2 = 1.00
  - The 3rd and 4th relevant documents are not retrieved at all, so P@k = 0.

- Step 3: Take the average of these P@k values
  - MAP = (P@1 + P@2 + 0 + 0)/4 = 0.50

Questions?

Discounted Cumulative Gain (DCG)

- Idea: Retrieving a relevant document at the top position earns the highest reward, with the reward gradually decreasing for lower-ranked positions.

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

$\mathrm{rel}(i)$: the relevance of the document ranked at position i

- Example:
  - Algorithm *A*'s retrieval result: [1, 1, 0, 0]
  - $\mathrm{DCG@}4 = \frac{1}{\log_2(1+1)} + \frac{1}{\log_2(2+1)} + \frac{0}{\log_2(3+1)} + \frac{0}{\log_2(4+1)} = \frac{1}{1} + \frac{1}{1.58} = 1.63$

Discounted Cumulative Gain (DCG)

- Idea: Retrieving a relevant document at the top position earns the highest reward, with the reward gradually decreasing for lower-ranked positions.

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

$\mathrm{rel}(i)$: the relevance of the document ranked at position i

- Example:
  - Algorithm *B*'s retrieval result: [0, 0, 1, 1]
  - $\mathrm{DCG}@4 = \frac{0}{\log_2(1+1)} + \frac{0}{\log_2(2+1)} + \frac{1}{\log_2(3+1)} + \frac{1}{\log_2(4+1)} = \frac{1}{2} + \frac{1}{2.32} = 0.93$
  - Although both *A* and *B* retrieve 2 relevant documents, they appear at lower ranks in *B*'s results, leading to a lower score.

Discounted Cumulative Gain (DCG)

$$\text{DCG@}k = \sum_{i=1}^{k} \frac{2^{\text{rel}(i)} - 1}{\log_2(i+1)}$$

- Rather than binary relevance, we can think of documents with multiple values of relevance.
  - 0 Not relevant
  - 1 Somewhat relevant
  - 2 Really relevant
  - 3 Perfectly relevant

- Example:
  - Algorithm *D*'s retrieval result: [1, 3, 2, 1, 0]
  - $\text{DCG@}5 = \frac{1}{\log_2(1+1)} + \frac{7}{\log_2(2+1)} + \frac{3}{\log_2(3+1)} + \frac{1}{\log_2(4+1)} + \frac{0}{\log_2(5+1)} = 7.35$

Ideal DCG?

- For a query, what is the best possible set of ranked results we could return?
- In practice, our search engine cannot achieve this, but we look in our dataset as an "oracle" and identify the best documents

- Some queries are "easy" … there are lots of great documents
- Other queries are "hard" … even in the best case, there are not many good documents

- We should normalize DCG for these different scenarios

Ideal DCG (IDCG)

- Algorithm *D*'s retrieval result: [1, 3, 2, 1, 0]
- $\text{DCG@5} = \frac{1}{\log_2(1+1)} + \frac{7}{\log_2(2+1)} + \frac{3}{\log_2(3+1)} + \frac{1}{\log_2(4+1)} + \frac{0}{\log_2(5+1)} = 7.35$

- Assume that in the entire collection, there are:
  - 2 documents with a relevance score of 3
  - 1 document with a relevance score of 2
  - 20 documents with a relevance score of 1
  - and all remaining documents have a relevance score of 0
- What is the best possible set of ranked results we could return (if we are allowed to return only 5 documents)?

- Ideal result: [3, 3, 2, 1, 1]
- $\text{IDCG@5} = \frac{7}{\log_2(1+1)} + \frac{7}{\log_2(2+1)} + \frac{3}{\log_2(3+1)} + \frac{1}{\log_2(4+1)} + \frac{1}{\log_2(5+1)} = 13.73$

Normalized DCG (NDCG)

$$\mathrm{NDCG@}k = \frac{\mathrm{DCG@}k}{\mathrm{IDCG@}k}$$

- NDCG@5 for Algorithm *D*'s retrieval result:
  - NDCG@5 = 7.35 / 13.73 = 0.54

- We have only demonstrated how to compute NDCG (and other metrics) for a single query.
- In practice, benchmark datasets always contain multiple queries, so we simply calculate the metric for each query and then take the average.

Summary: Offline Evaluation

- Hypothesis: A new search engine (e.g., based on *SuperRank*) is better than an old one (e.g., based on BM25)
- What we need:
  - Documents (representative of our collection),
  - Queries (that we hope are representative of what our users will ask), and
  - Relevance judgments (can be expensive to collect and noisy)
- Metrics:
  - Precision, Recall, F1
  - P@*k*, MAP, NDCG@*k*
- Challenge: Do the results generalize to the online scenario?

Types of Evaluation

- Offline: Usually with a standard dataset or using historical interactions from a production system (e.g., at Google)

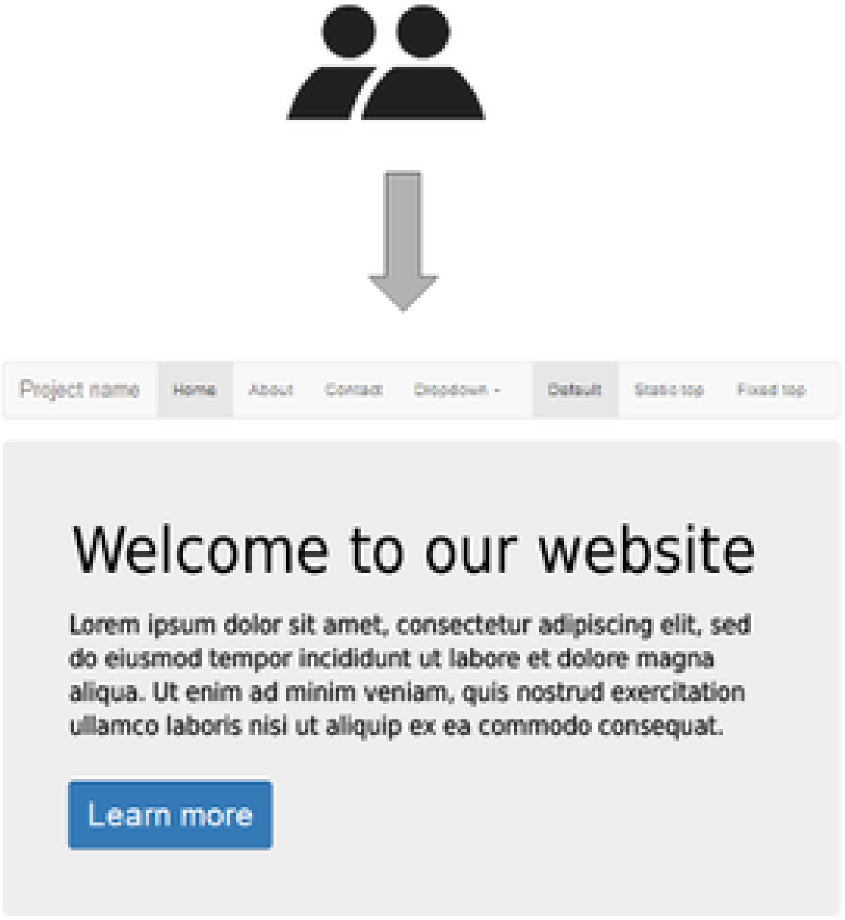
↓

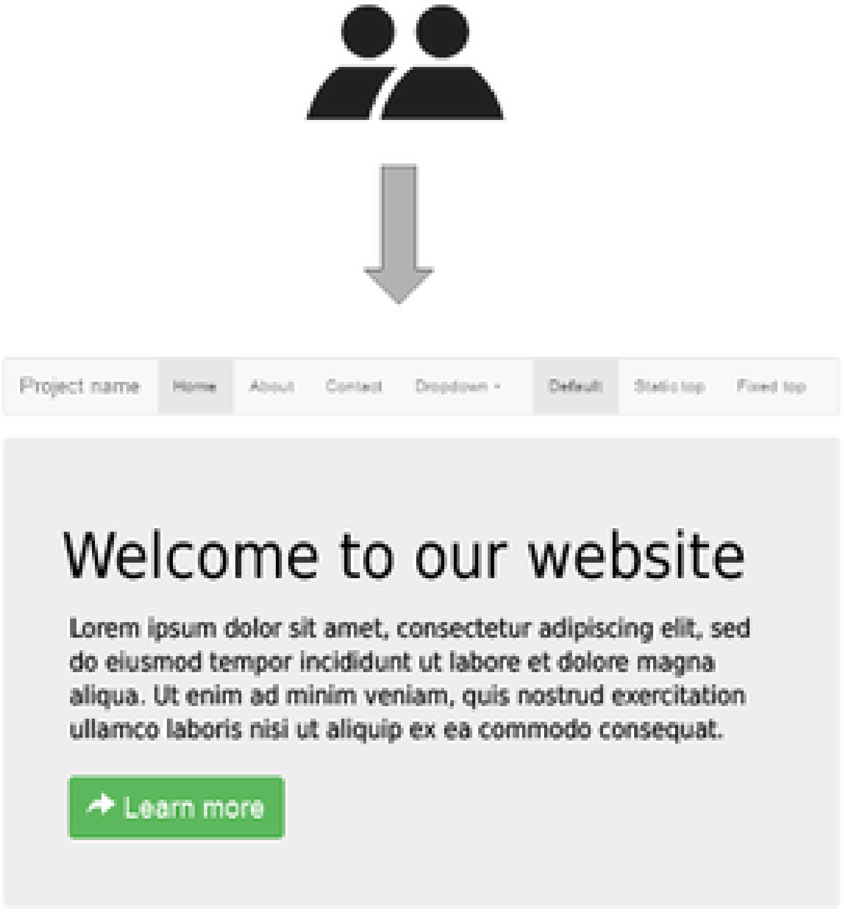
- User Studies: Present search interface to a group of users (say 10-100), often in person or using a system like Amazon Mechanical Turk (can scale to 100s)

↓

- Online: Typically requires access to a production system with existing users (challenging for a class project!)
  - A/B tests (e.g., to measure click through rate)

A/B Testing



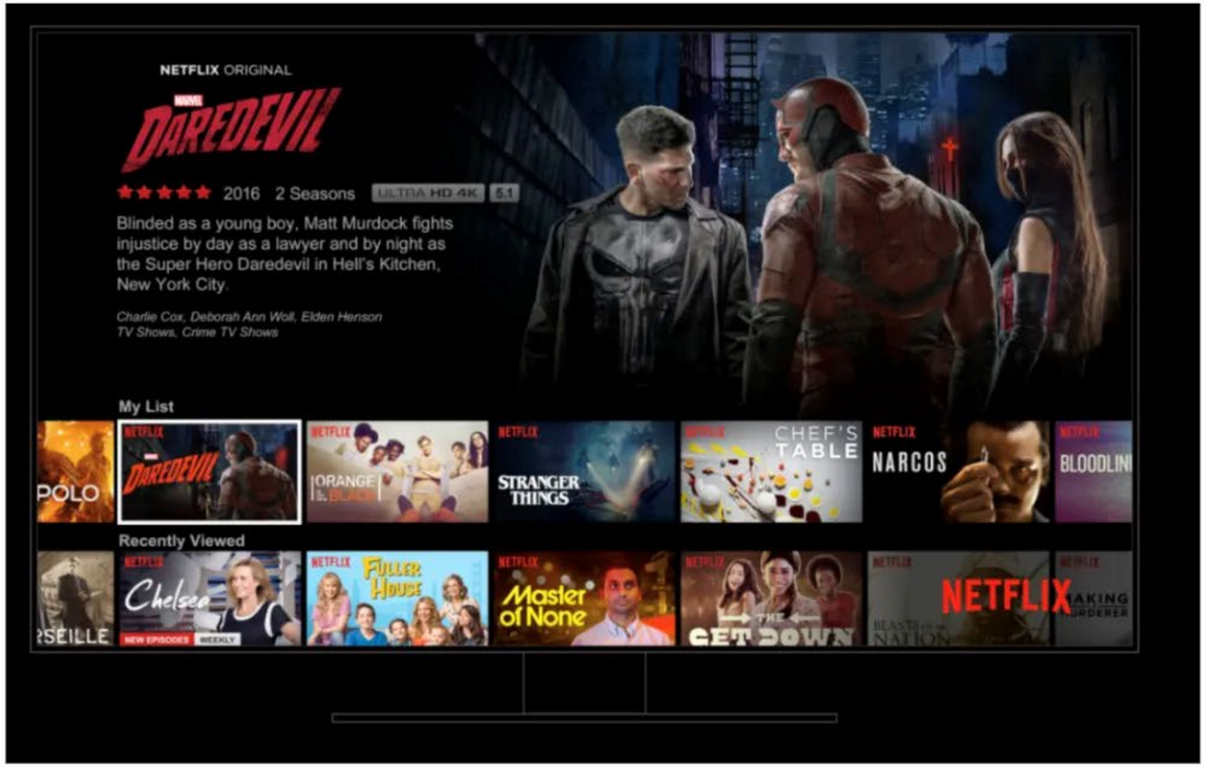


Click rate: 52 % 72 %

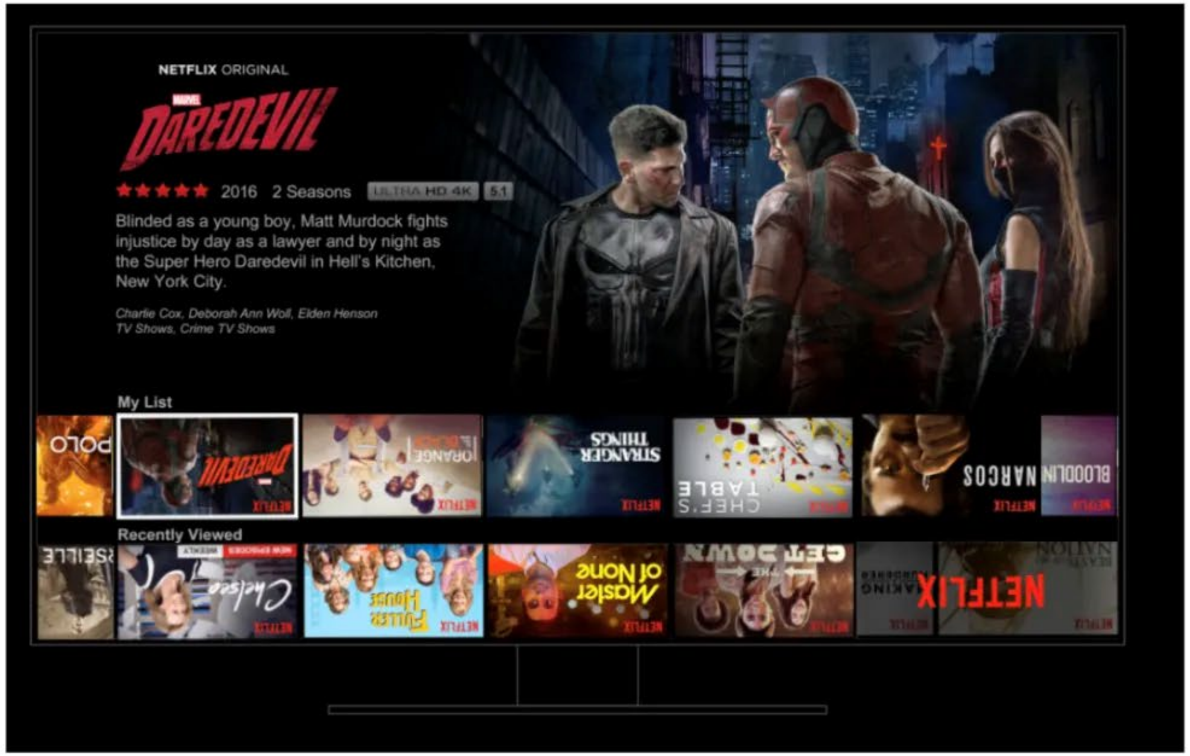
A/B Testing

- https://netflixtechblog.com/what-is-an-a-b-test-b08cc1b57962

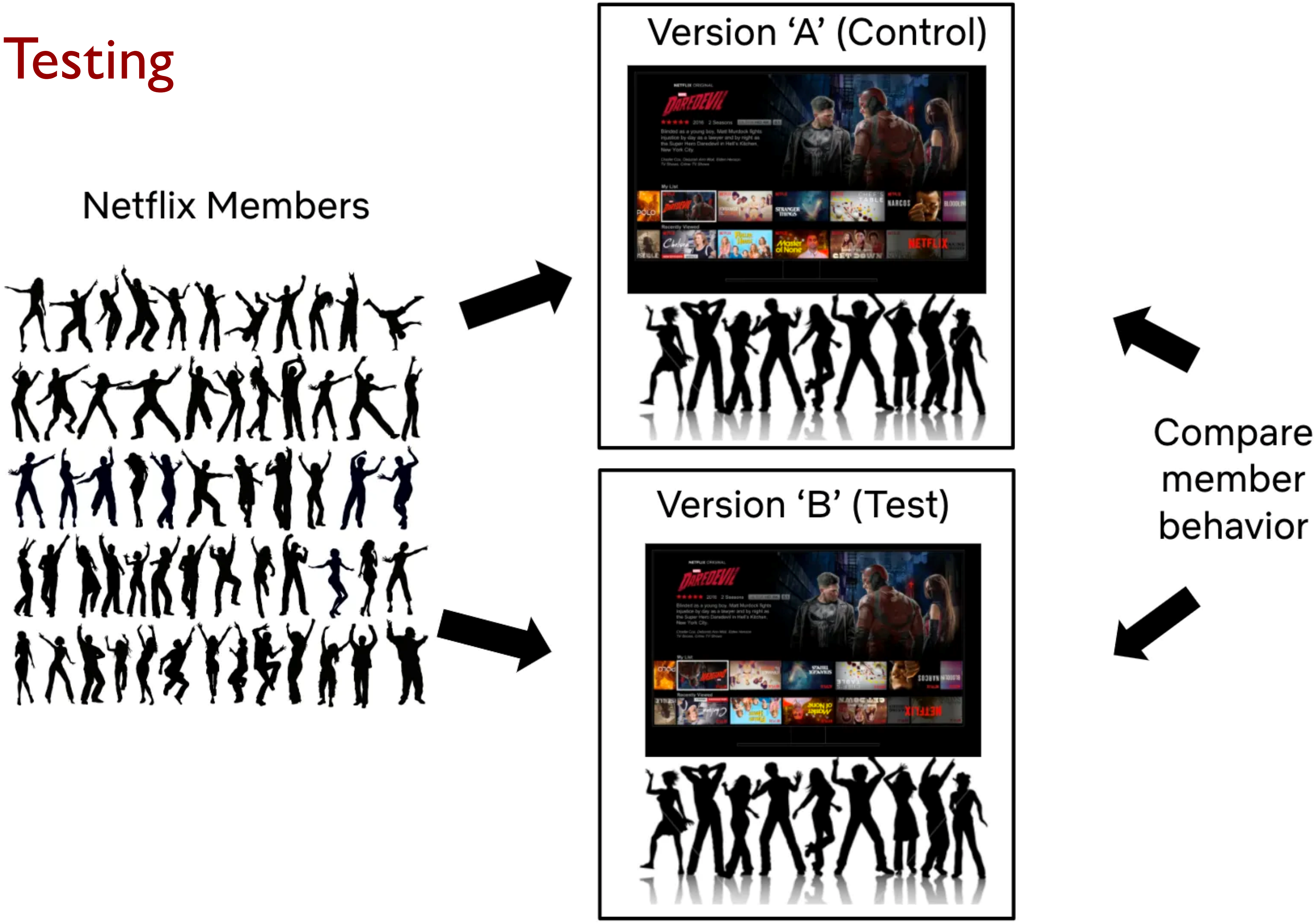
Product A: Standard box art



Product B: Upside-down box art



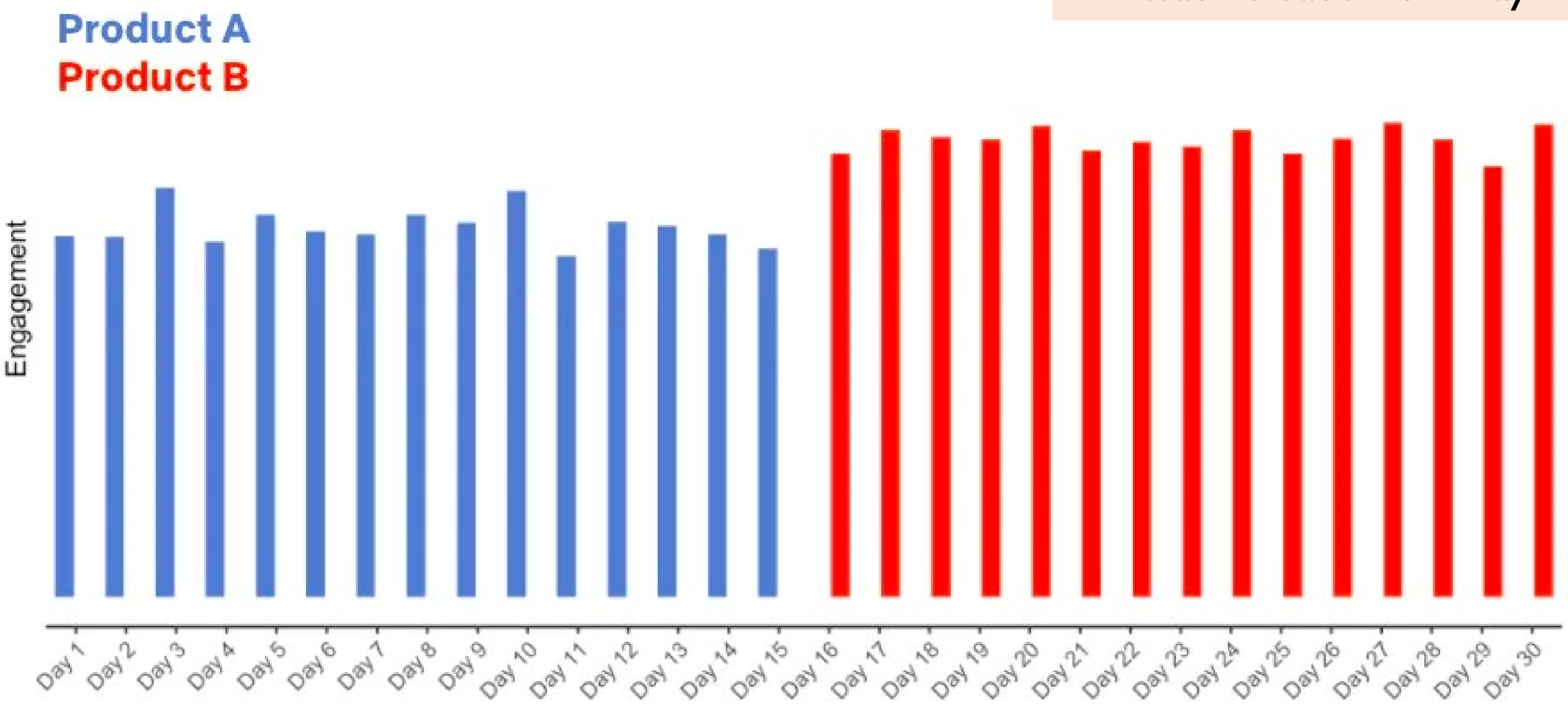
A/B Testing


Netflix Members
Version 'A' (Control)
Version 'B' (Test)
Compare member behavior

Controlling Variables as Much as Possible

- Is this enough to conclude that Product B is better?

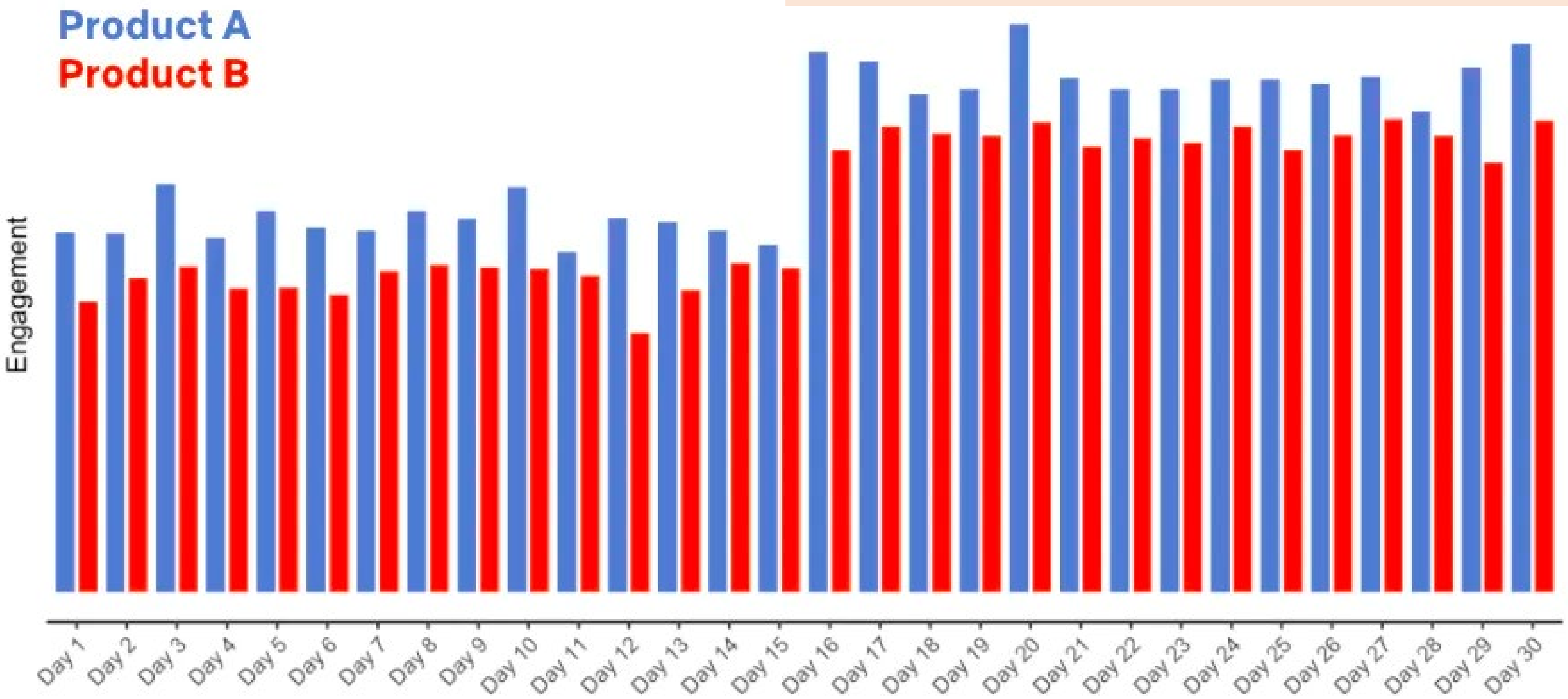
What if a hit title or a hit movie was released on Day 16?



Controlling Variables as Much as Possible

- A more controlled A/B test

the Upside-Down product results in generally lower engagement (not surprisingly)



How can we know that this difference is not (very likely) due to randomness?

True Merit vs. Randomness

- Can we conclude from this offline test that Algorithm *B* outperforms Algorithm *A*?

	NDCG@5
Algorithm *A*	0.7000
Algorithm *B*	0.7001

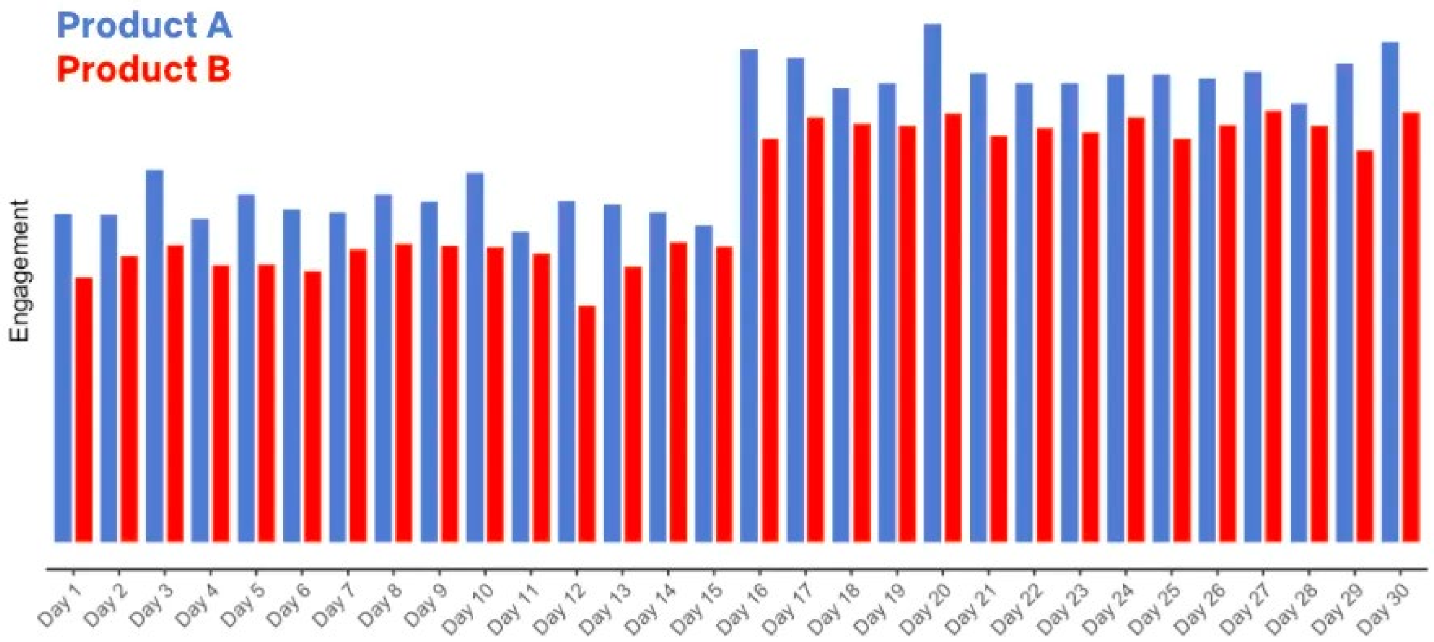
- Can we conclude from this online test that Algorithm *B* outperforms Algorithm *A*?

	User Click-Through Rate
Algorithm *A*	0.3000
Algorithm *B*	0.3100

- We need statistical significance tests!

Statistical Significance Tests for Evaluating a Search Engine

- Step 1: Evaluate Algorithms *A* and *B* under different experimental conditions
  - Query types (offline)
  - Time of experiment (online)
  - Random seeds (if the algorithm involves randomness)
  - …



Statistical Significance Tests for Evaluating a Search Engine

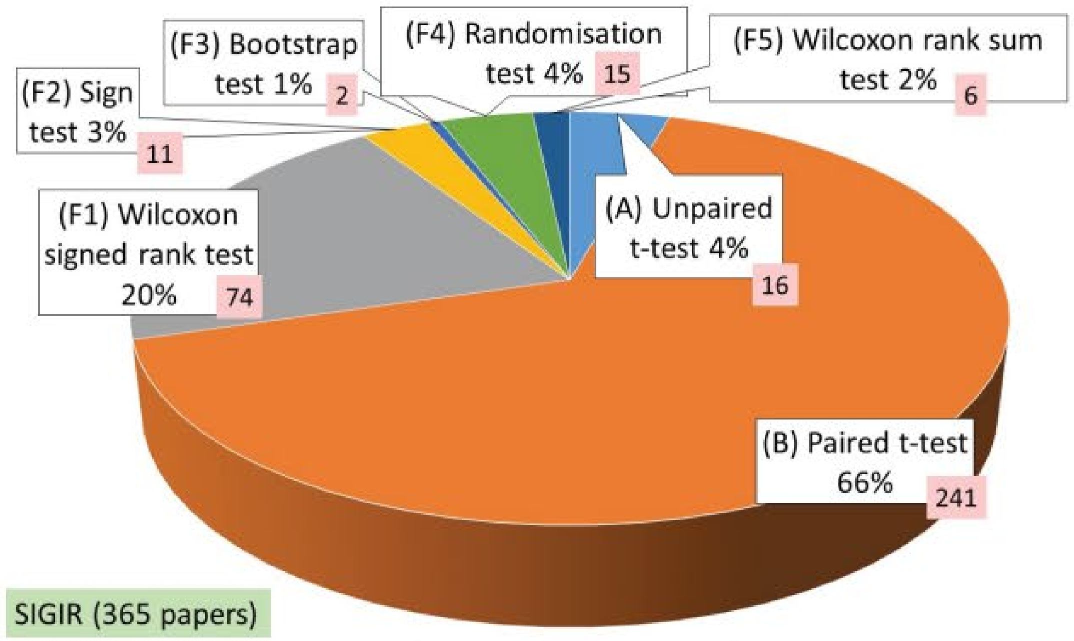
- Step 2: Compare the metrics of Algorithms *A* and *B* and examine whether they are likely drawn from different probability distributions

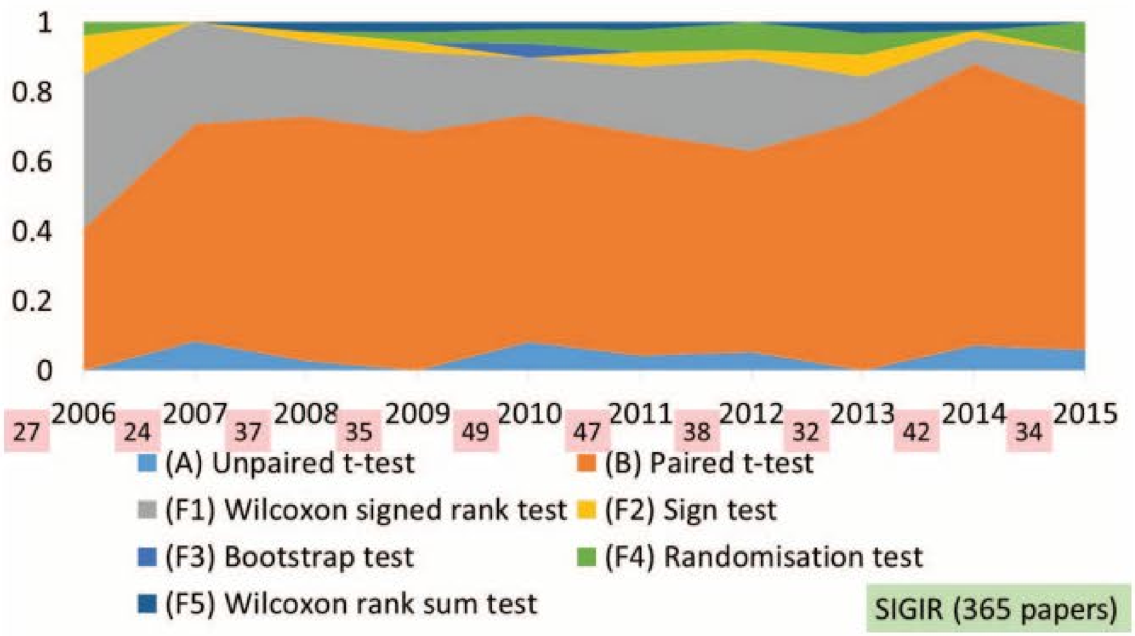
	Algorithm *A*	Algorithm *B*	Difference
Condition 1	x_1	y_1	$y_1 - x_1$
Condition 2	x_2	y_2	$y_2 - x_2$
…	…	…	…
Condition *N*	x_N	y_N	$y_N - x_N$

- Null Hypothesis: The case you hope to rule out
  - $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ are drawn from two distributions with the same mean, OR
  - $\{y_i - x_i\}_{i=1}^N$ are drawn from a distribution with mean 0
- Statistical Significance Test: Using probability theory to show that the likelihood of the null hypothesis being true is very small (e.g., < 0.01).

Statistical Significance Tests for Evaluating a Search Engine

- *Statistical Significance, Power, and Sample Sizes: A Systematic Review of SIGIR and TOIS, 2006-2015.* SIGIR 2016.
  - The most commonly used tests in IR: Paired t-test (66%), Wilcoxon signed rank test (20%), and Unpaired t-test (4%)





Paired t-test

- Null Hypothesis: $\{y_i - x_i\}_{i=1}^N$ are drawn from a distribution with mean 0
- Step 1: Calculate the t-statistic

$$t = \frac{\text{mean of } \{y_i - x_i\}_{i=1}^N}{\left(\text{standard deviation of } \{y_i - x_i\}_{i=1}^N\right)/\sqrt{N}}$$

- Step 2: Calculate the "*degrees of freedom*": $N - 1$
- Step 3: Look up the t-statistic in a t-distribution table (you need to know $N - 1$ to find the correct row) to obtain the p-value
    - p-value = Prob[the difference between Algorithms A and B is due to randomness]
    - If p-value < 0.05, then Prob[the difference between Algorithms A and B is due to true merit] > 0.95, and we say the difference is statistically significant.

Paired t-test

- We can also do this in Python:

```python
from scipy.stats import ttest_rel

# Sample data
X = [0.5, 0.4, 0.6, 0.3, 0.2, 0.4, 0.5, 0.3, 0.2, 0.5]
Y = [0.3, 0.2, 0.5, 0.2, 0.1, 0.3, 0.4, 0.2, 0.1, 0.4]

# Calculate t-statistic and p-value
t, p = ttest_rel(X, Y)

# Print p-value
print(p)
```

  - In this example, p-value = 8.538e-06

Wilcoxon Signed Rank Test

- Paired t-test assumes that $\{y_i - x_i\}_{i=1}^N$ are drawn from a normal distribution
- Wilcoxon signed rank test has a much weaker assumption: $\{y_i - x_i\}_{i=1}^N$ are drawn from a symmetric distribution around the mean

- Null Hypothesis: $\{y_i - x_i\}_{i=1}^N$ are drawn from a distribution with mean 0

- Example: $\{y_i - x_i\}_{i=1}^N$ = {0.20, -0.10, 0.30, -0.05}

- Step 1: Compute $|y_i - x_i|$
  - 0.20, 0.10, 0.30, 0.05

- Step 2: Sort these values and assign ranks
  - 0.05 (rank=1), 0.10 (rank=2), 0.20 (rank=3), 0.30 (rank=4)

Wilcoxon Signed Rank Test

- Null Hypothesis: $\{y_i - x_i\}_{i=1}^{N}$ are drawn from a distribution with mean 0
- Example: $\{y_i - x_i\}_{i=1}^{N}$ = {0.20, -0.10, 0.30, -0.05}
- Step 1: Compute $|y_i - x_i|$
  - 0.20, 0.10, 0.30, 0.05
- Step 2: Sort these values and assign ranks
  - 0.05 (rank=1), 0.10 (rank=2), 0.20 (rank=3), 0.30 (rank=4)
- Step 3: Calculate the signed-rank sum
  - *T* = (-1) + (-2) + (+3) + (+4) = 4
  - Intuition: If the Null Hypothesis holds, *T* should be close to 0.
- Step 4: Look up *T* in the table to obtain the p-value

Wilcoxon Signed Rank Test

- We can also do this in Python:

```python
from scipy.stats import wilcoxon

# Sample data
X = [0.5, 0.4, 0.6, 0.3, 0.2, 0.4, 0.5, 0.3, 0.2, 0.5]
Y = [0.3, 0.2, 0.5, 0.2, 0.1, 0.3, 0.4, 0.2, 0.1, 0.4]

# Perform Wilcoxon Signed-Rank Test
stat, p = wilcoxon(X, Y)

# Print p-value
print(p)
```

  - In this example, p-value = 0.00195

Unpaired t-test

- What if a paired comparison is NOT feasible?
  - E.g., when the two IR models use entirely different architectures with different hyperparameter settings, and we are conducting an offline evaluation

- Null Hypothesis: $\{x_i\}_{i=1}^{M}$ and $\{y_j\}_{j=1}^{N}$ are drawn from two distributions with the same mean

- If we can assume the two distributions have the same variance:

$$t = \frac{\bar{x} - \bar{y}}{s_p\sqrt{\frac{1}{M} + \frac{1}{N}}}, \qquad \text{where } \bar{x} = \frac{x_1 + \cdots + x_M}{M}, \qquad \bar{y} = \frac{y_1 + \cdots + y_N}{N}$$

$$\text{and } s_p = \sqrt{\frac{(M-1)s_X^2 + (N-1)s_Y^2}{M+N-2}}$$

Unpaired t-test

- If we cannot assume the two distributions have the same variance (Welch's t-test):

$$t = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{s_X^2}{M} + \frac{s_Y^2}{N}}}, \qquad \text{where } \bar{x} = \frac{x_1 + \cdots + x_M}{M}, \qquad \bar{y} = \frac{y_1 + \cdots + y_N}{N}$$

python Copy Edit

```python
from scipy.stats import ttest_ind

# Sample data
X = [0.5, 0.4, 0.6, 0.3, 0.2, 0.4, 0.5, 0.3, 0.2, 0.5]
Y = [0.3, 0.2, 0.5, 0.2, 0.1, 0.3]  # 4 elements removed

# Unpaired t-test (equal variance)
t_equal, p_equal = ttest_ind(X, Y, equal_var=True)

# Welch’s t-test (unequal variance)
t_unequal, p_unequal = ttest_ind(X, Y, equal_var=False)
```

Quiz 1 (5%)

- Will be held in the next class (Feb 6)
  - 45 minutes, but designed to only take 30-35 minutes
- 7 multiple-choice questions covering content from Week 1 to Week 4, including this lecture, as well as Homework 0.
  - Most (e.g., 5) of them will come from numerical examples on the slides.
  - One will be from homework.
  - The remaining will be on a deeper understanding of the techniques introduced.
  - No rote memorization
- Answering 5 questions correctly will earn you full credit (5%).

# correct answers	0	1	2	3	4	5	6	7
credit	0%	1%	2%	3%	4%	5%	5%	5%

Quiz 1 (5%)

- Closed book
  - Laptops, books, and notes are NOT allowed.

- Calculators are NOT required, and the questions will NOT involve calculations (such as square roots or logarithms) that cannot be done easily by hand.

- Please refer to Student Rule 7 (https://student-rules.tamu.edu/rule07/) about excused absences, including definitions, and related documentation and timelines.
  - For students who miss the quiz due to an excused absence, your quiz score will be counted as part of the final exam.
  - Specifically, your final exam weight will increase by 5% for each quiz missed with an excused absence (i.e., 30% + 5% × number of excused quiz absences).

Thank You!

Course Website: https://yuzhang-teaching.github.io/CSCE670-S26.html