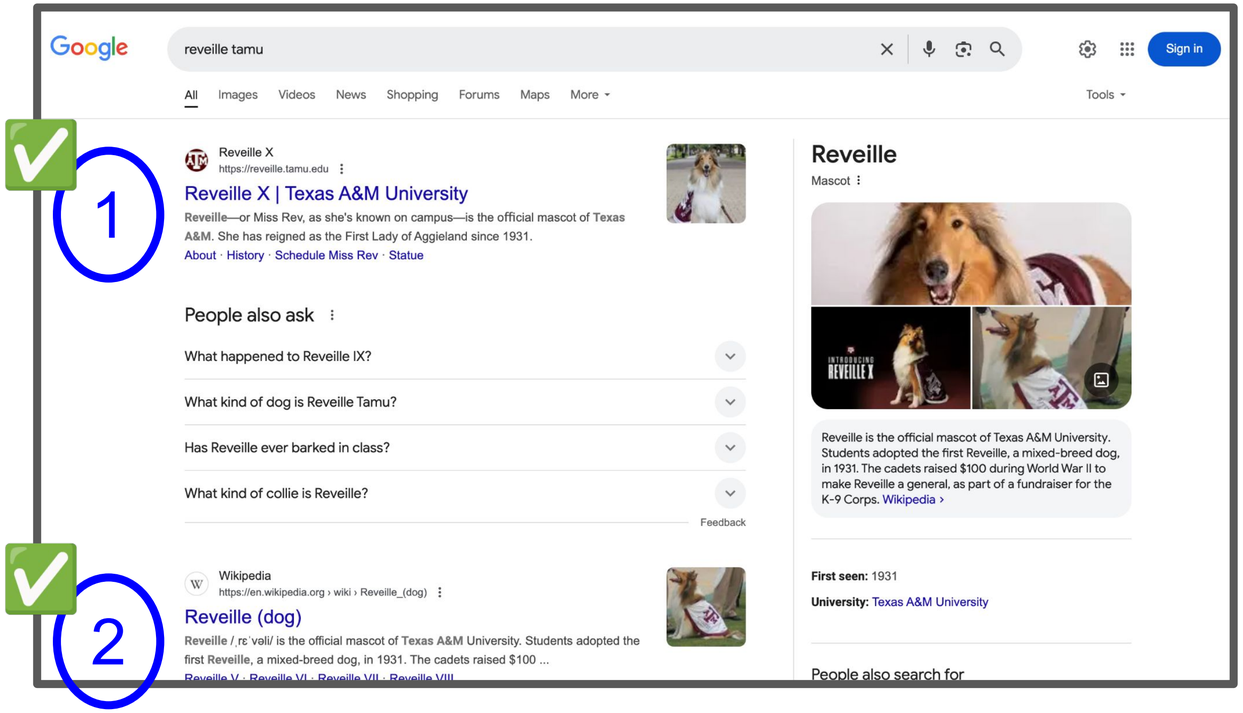
# Information Storage & Retrieval

## Class 6: Evaluation

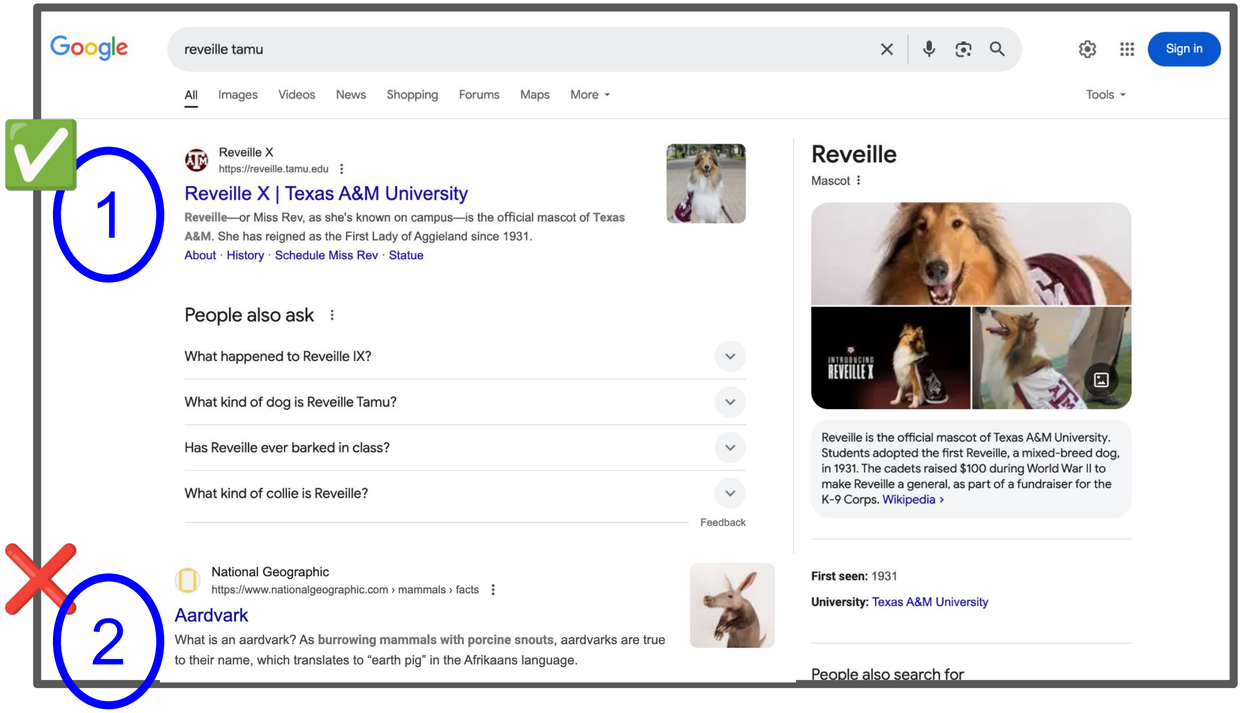
CSCE 670 :: Spring 2024
Texas A&M University
Department of Computer Science & Engineering
Prof. James Caverlee *and Maria Teleki* 🤠

# SEARCH ENGINES


Google returned
2 ✅ relevant results

# SEARCH ENGINES


Google returned:

1 ✅ relevant result

1 ❌ non-relevant result

# ✅ relevant search results =

# but more importantly,

# ✅ relevant search results =

signs of a heart attack

how to register to vote

**People getting access to the information they need.**

how to go to college

what is a 401k

how to invest

how to tell if email is a scam

how to perform CPR

# The importance of ***evaluation***

The ability to **measure differences** 📐 underlies ***experimental science*** 👩‍🔬

- How well do our systems work?
- Is *Algorithm A* better than *Algorithm B*?
- Really? Under what conditions?

Evaluation drives: ***WHAT to research***

- Identify techniques that work and that don’t

# Measuring Relevance

We need 3 things in our **BENCHMARK DATASET**:

| English | Math | Picture |
|---|---|---|
| 1) A set of documents | $D = \{(d_i, q_j, r_{ij})\}$ | D |
| 2) A set of queries | $d_i$ is a vector | |
| 3) A binary assessment of either ✅ **Relevant** or ❌ **Non-Relevant** for each query and each document | $q_j$ is a vector<br>$r_{ij} \in \{0,1\}$ | |

| Documents | Queries | Relevance |
|---|---|---|
| $d_1$ | $q_1$ | $r_{11}$ |
| $d_1$ | $q_2$ | $r_{12}$ |
| $d_1$ | $q_3$ | $r_{13}$ |
| ... | ... | ... |

# Activity

With your group,
what are some pros and cons of measuring **relevance** this way?

A binary assessment of either ✅ **Relevant** or ❌ **Non-Relevant** for each query and each document

# Activity

With your group,

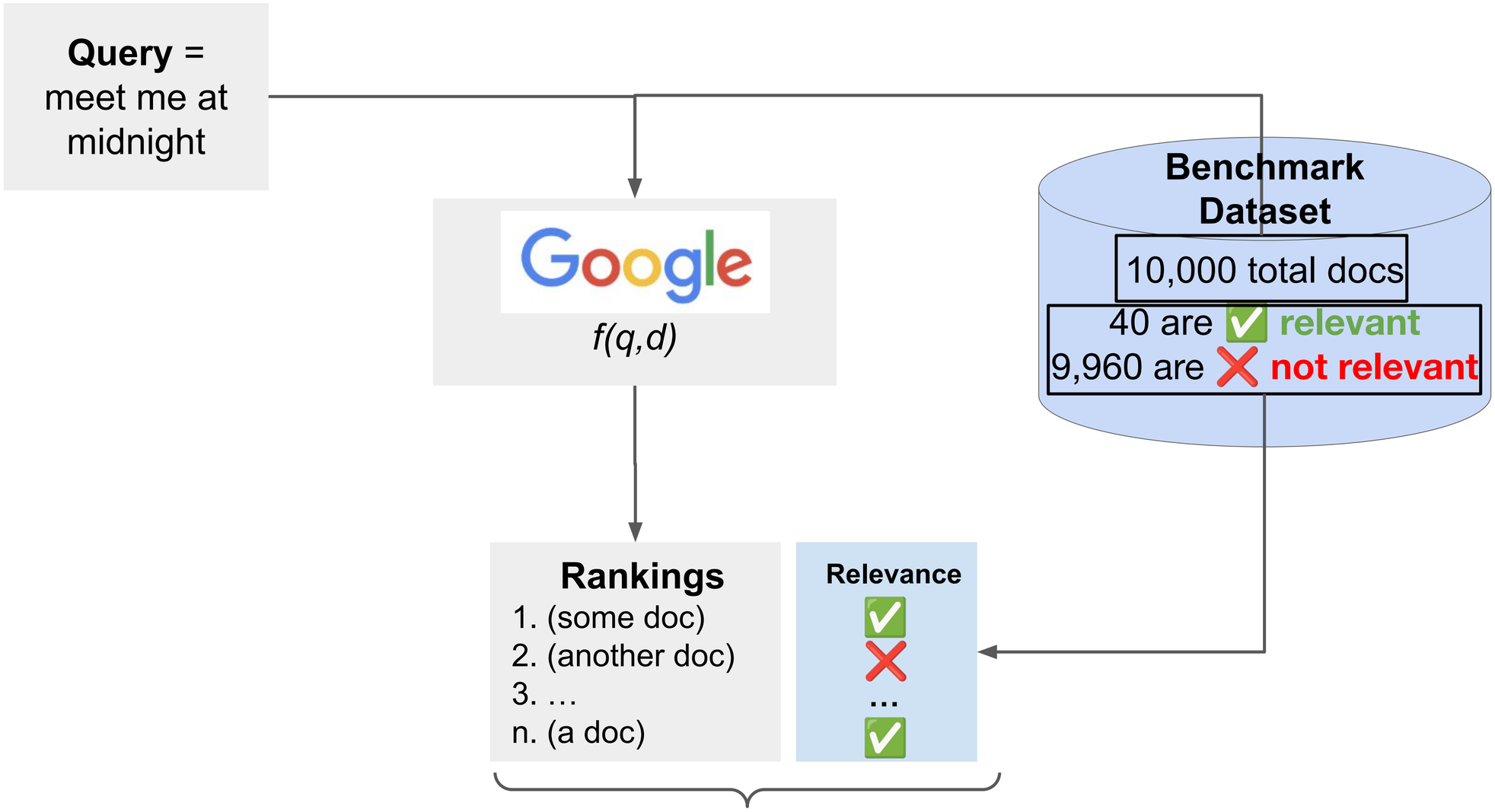
find an IR BENCHMARK DATASET online.

We need 3 things in our BENCHMARK DATASET:

| English | Math | Picture |
|---|---|---|
| 1) A set of documents | $D = \{(d_i, q_j, r_{ij})\}$<br>$d_i$ is a vector<br>$q_j$ is a vector<br>$r_{ij} \in \{0,1\}$ | D |
| 2) A set of queries | | |
| 3) A binary assessment of either ✅ **Relevant** or ❌ **Non-Relevant** for each query and each document | | |

| Documents | Queries | Relevance |
|---|---|---|
| $d_1$ | $q_1$ | $r_{11}$ |
| $d_1$ | $q_2$ | $r_{12}$ |
| $d_1$ | $q_3$ | $r_{13}$ |
| … | … | … |

# The Big Picture



# Evaluation Measures*

Precision

Recall

F aka F Score, aka F-1 Score

**UNRANKED MEASURES**

These measures don't incorporate the order of the results. They treat the results like sets.

Precision@k

Recall@k

F@k

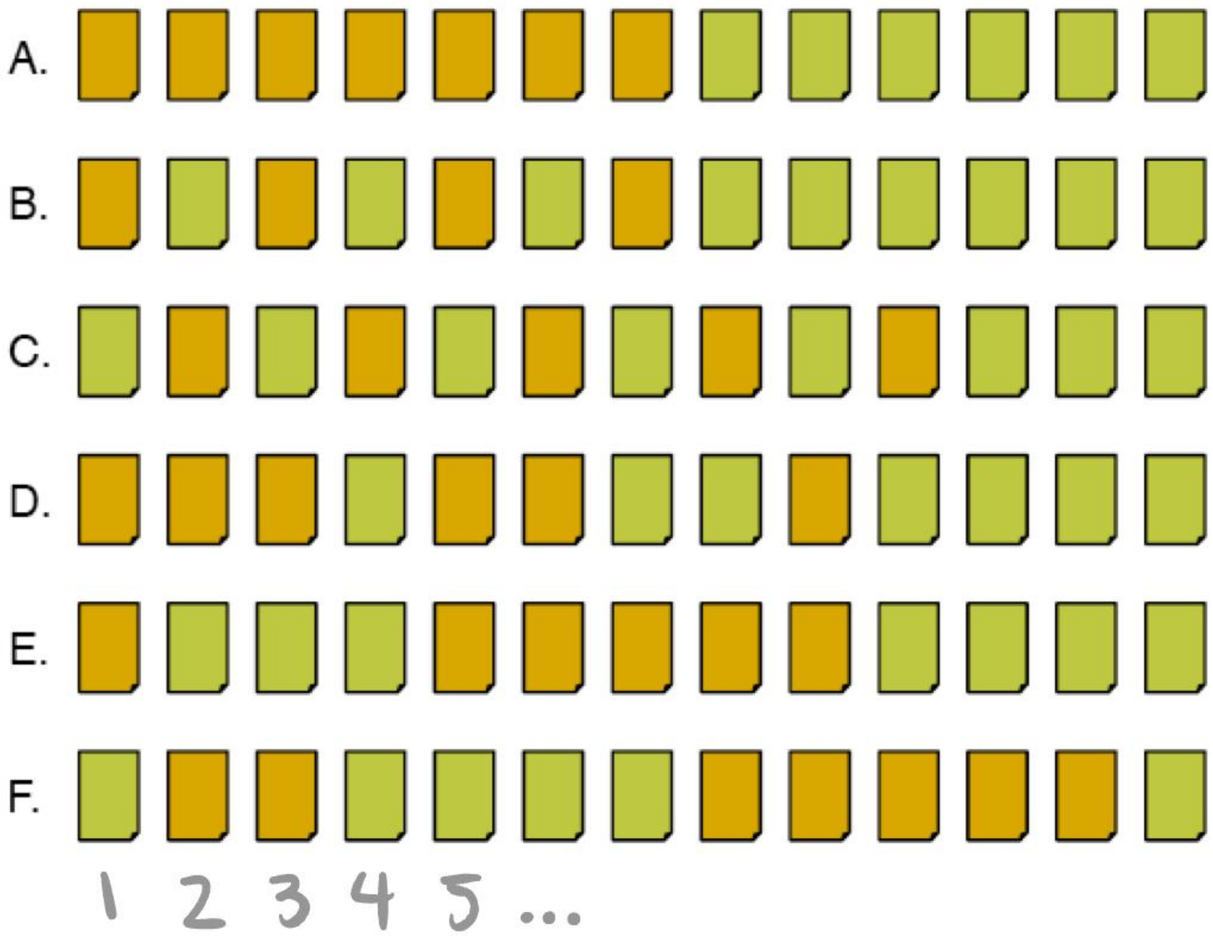
NDCG@k

**RANKED MEASURES**

These measures do (at least in some way) incorporate the order of the results.

**There are many more evaluation measures!*

# Which is the best rank order?




if orange = YES relevant

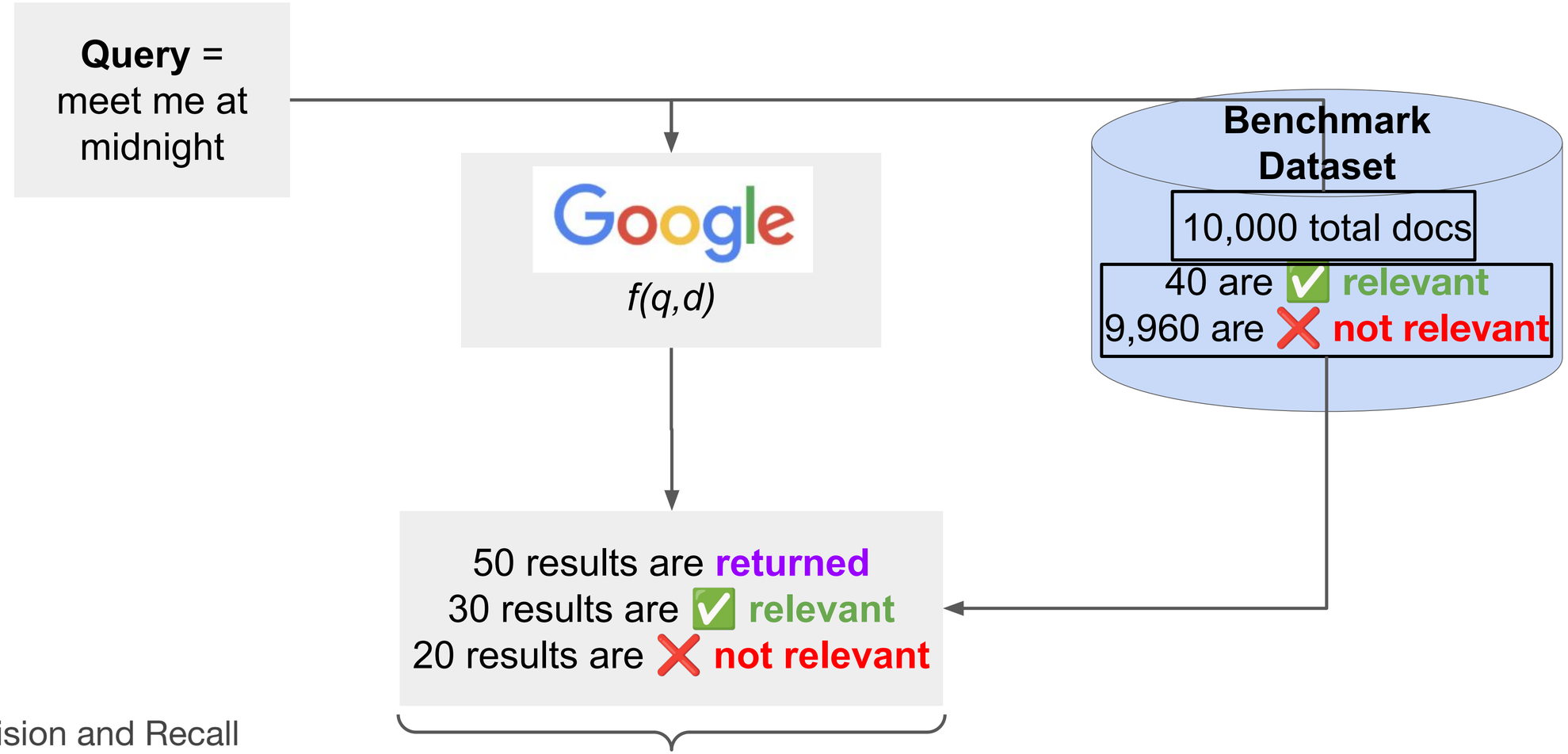
green = NOT relevant

# Precision and Recall

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

$$\text{Recall} = \frac{\textbf{\# of retrieved documents} \text{ that are } ✅ \textbf{ relevant}}{\textbf{total } \# \text{ of } ✅ \textbf{ relevant documents in the dataset}}$$

# Example 1: Calculate Precision and Recall for the following query and document set.




Precision and Recall

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

$$\text{Recall} = \frac{\text{\# of retrieved documents that are ✅ relevant}}{\text{total \# of ✅ relevant documents in the dataset}}$$

# Example 2: Calculate Precision and Recall for the following query and document set.

**Query** = always rooting for the anti-hero




*f(q,d)*

**Benchmark Dataset**

40 total docs

17 are ✅ **relevant**

23 are ❌ **not relevant**

12 results are **returned**

3 results are ✅ **relevant**

9 results are ❌ **not relevant**

P = 3/12

R = 3/17

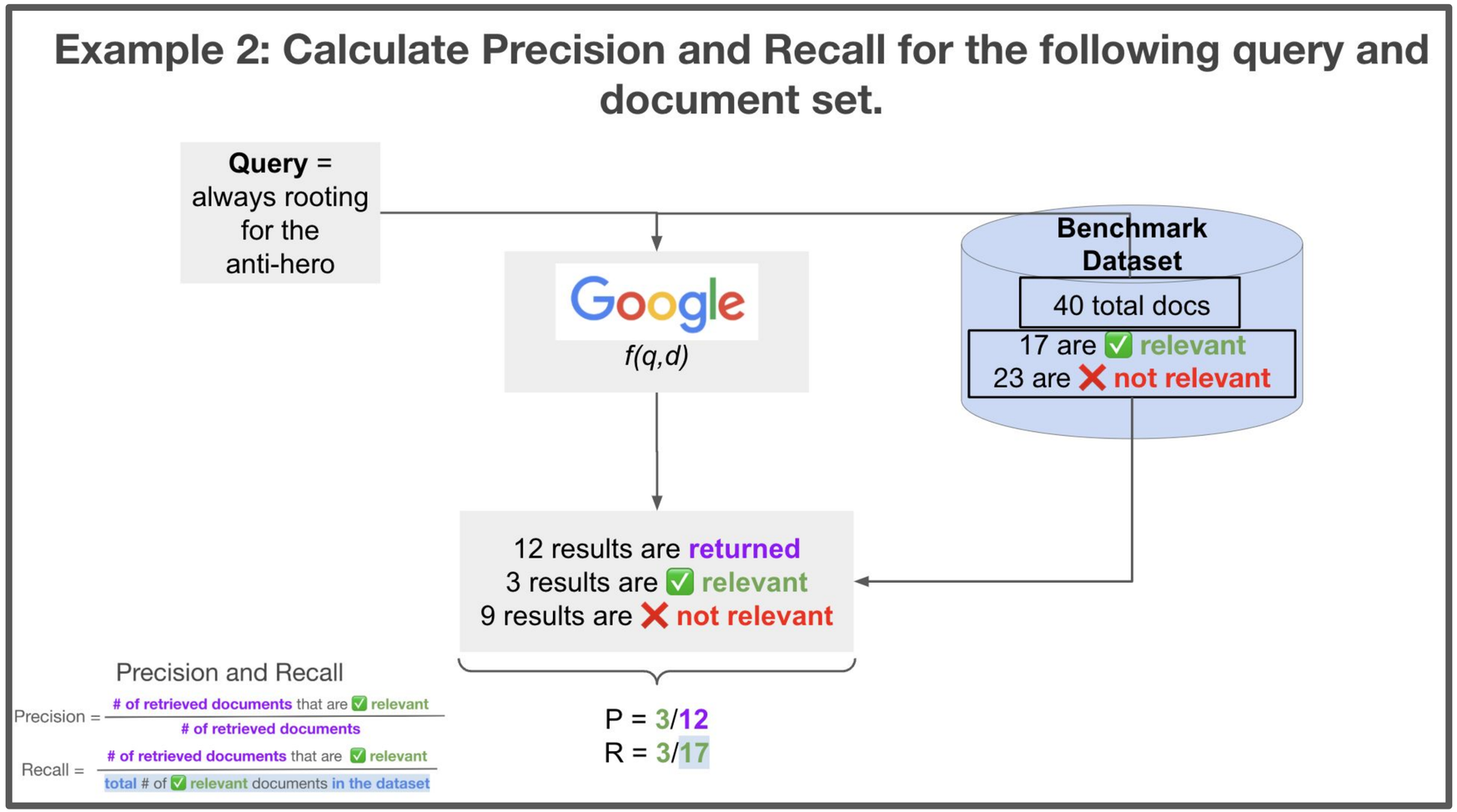
Precision and Recall

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

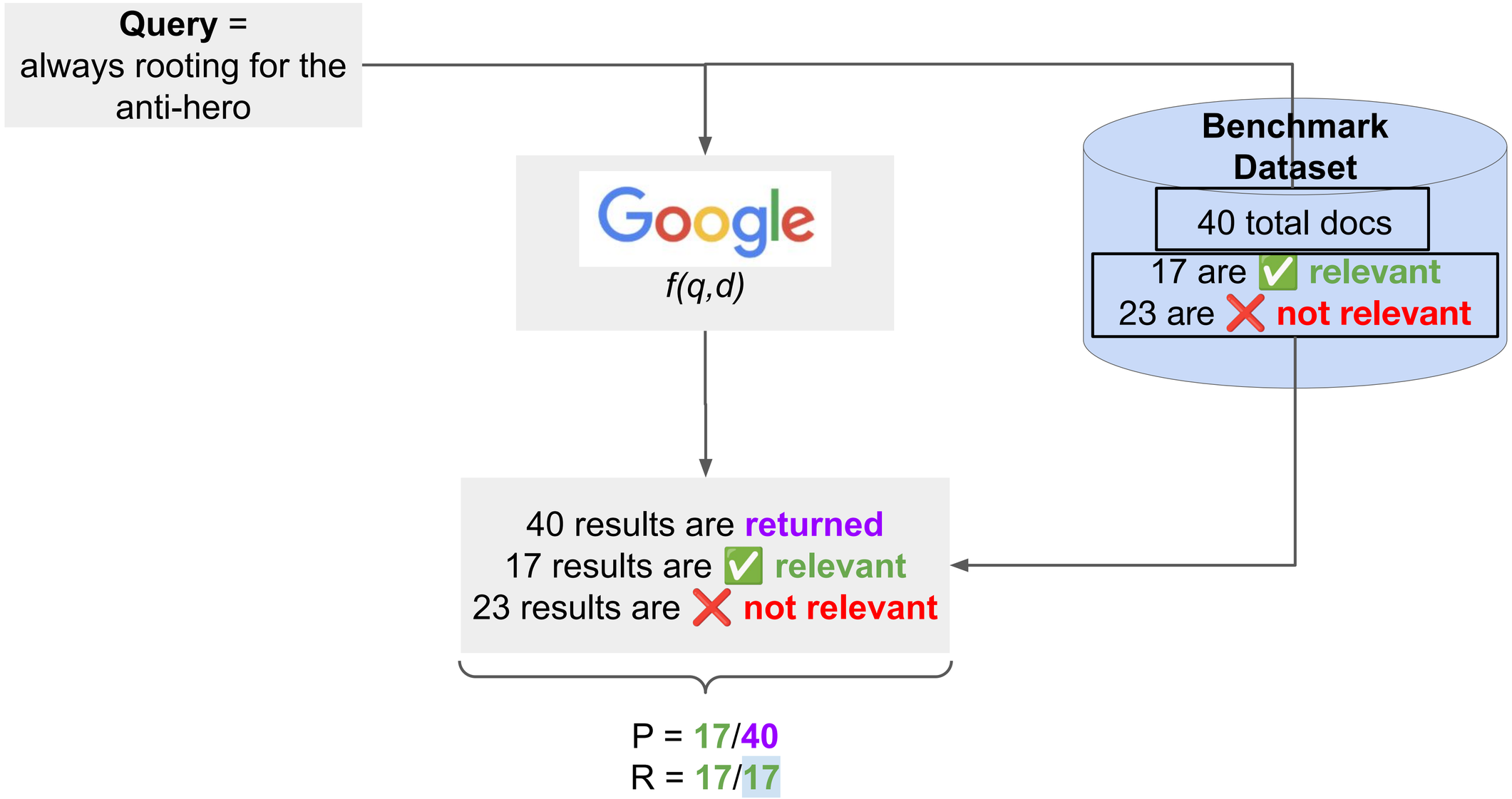
$$\text{Recall} = \frac{\textbf{\# of retrieved documents} \text{ that are } \checkmark \textbf{ relevant}}{\textbf{total \# of } \checkmark \textbf{ relevant documents in the dataset}}$$

# Activity

Can you design a search engine with perfect recall?

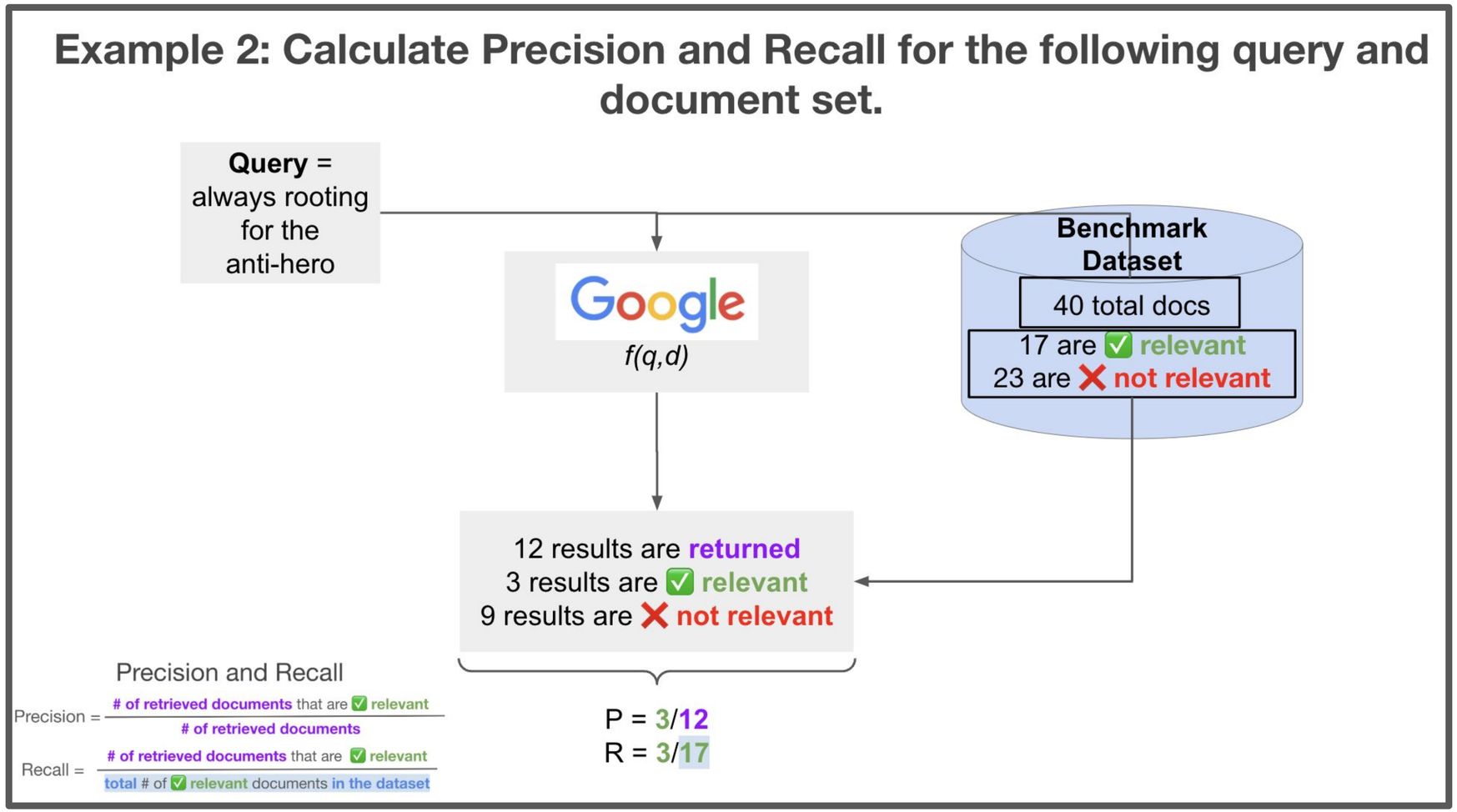


# Example: Design a search engine with perfect recall.



# Activity

What are some situations where you might ***care more about the Recall*** of your search engine? Precision?



# Combining Precision and Recall: F

*aka F1 Score*
*aka F Score*

Precision and Recall tell us *different things* about the *performance of the search engine*. So, F Score is a good way to quickly understand the *overall performance*, because it incorporates both of them!

## Measures of Central Tendency

for $x_1, x_2, x_3, \ldots, x_n$ positive #s, we can calculate a few different types of equally weighted averages:

### The 3 Classical Pythagorean Means

| Arithmetic Mean | Geometric Mean | Harmonic Mean |
|---|---|---|
| $M_A = \frac{x_1 + x_2 + \ldots + x_n}{n}$ | $M_G = \sqrt[n]{x_1 \cdot x_2 \cdot \ldots \cdot x_n}$ | $M_H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_n}}$ |

biggest mean — middle mean — smallest mean

for $x_1 = P, x_2 = R$:

$$M_H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_n}} = \frac{2}{\frac{1}{P} + \frac{1}{R}} \cdot \frac{PR}{PR} = \frac{2PR}{\frac{1}{P}\cdot\frac{PR}{1} + \frac{1}{R}\cdot\frac{PR}{1}}$$

$$= \frac{2PR}{R+P}$$

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

**Let's do some examples w/ the *arithmetic mean* vs. the *harmonic mean*:**

**If P=0.9 and R=0.1 (very different):**

$M_A$=(0.9+0.1)/2 = 0.5

$M_H$= (2*0.9*0.1)/(0.9+0.1) = 0.18

So if either precision or recall is low, the harmonic mean will also be low.

**If P=0.5 and R=0.5 (literally the same):**

$M_A$=(0.5+0.5)/2 = 0.5

$M_H$= (2*0.5*0.5)/(0.5+0.5) = 0.5

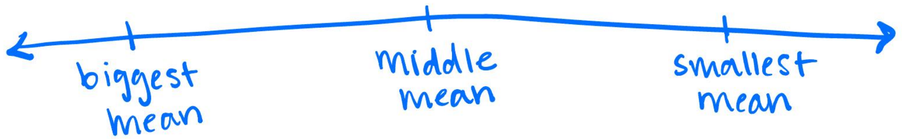
# A generalization of F

## Measures of Central Tendency

for $x_1, x_2, x_3, \ldots, x_n$ positive #s, we can calculate a few different types of equally weighted averages:

### The 3 Classical Pythagorean Means

| Arithmetic Mean | Geometric Mean | Harmonic Mean |
|---|---|---|
| $M_A = \frac{x_1 + x_2 + \ldots + x_n}{n}$ | $M_G = \sqrt[n]{x_1 \cdot x_2 \cdot \ldots \cdot x_n}$ | $M_H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_n}}$ |




for $x_1 = P$, $x_2 = R$:

$$M_H = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \ldots + \frac{1}{x_n}} = \frac{2}{\frac{1}{P} + \frac{1}{R}} \cdot \frac{PR}{PR} = \frac{2PR}{\frac{1}{P} \cdot \frac{PR}{1} + \frac{1}{R} \cdot \frac{PR}{1}}$$

$$= \frac{2PR}{R + P}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

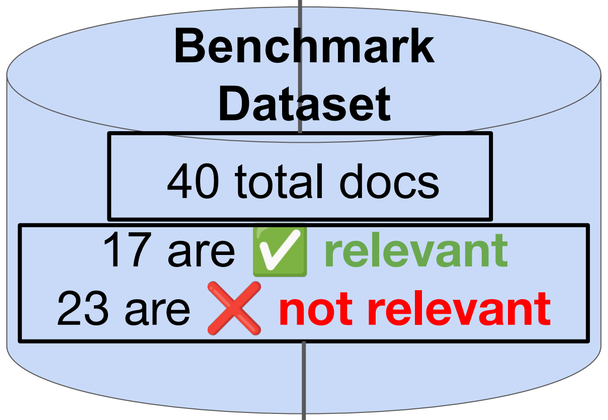
# Precision@k and Recall@k

$$\text{Precision} = \frac{\text{\# of retrieved documents that are ✅ relevant in the top k}}{\text{k}}$$

$$\text{Recall} = \frac{\text{\# of retrieved documents that are ✅ relevant in the top k}}{\text{total \# of ✅ relevant documents in the dataset}}$$

# Example: Calculate Precision@1, Precision@5, and Precision@10 for the following query and document set.

**Query** =
always rooting for the anti-hero


P@1 = 1/1
R@1 = 1/17

P@5 = 2/5
R@5 = 2/17

P@10 = 3/10
R@10 = 3/17

12 results are **returned**:

1. ✅ relevant
2. ❌ not relevant
3. ❌ not relevant
4. ✅ relevant
5. ❌ not relevant
6. ❌ not relevant
7. ❌ not relevant
8. ✅ relevant
9. ❌ not relevant
10. ❌ not relevant
11. ❌ not relevant
12. ❌ not relevant

k=1
k=5
k=10

## Precision@k and Recall@k

$$\text{Precision} = \frac{\text{\# of retrieved documents that are ✅ relevant in the top k}}{\text{k}}$$

$$\text{Recall} = \frac{\text{\# of retrieved documents that are ✅ relevant in the top k}}{\text{total \# of ✅ relevant documents in the dataset}}$$

# **Activity**

Overall, ***what do you like/not like*** about Precision and Recall?

Some questions to consider: When might they be super good/informative metrics? When are they not that helpful?

# NDCG

**Normalized Discounted Cumulative Gain**

- Sensitive to the **position** of the highest rated page
- **Log-discounting** of results
- **Normalized** for different lengths lists

**Very popular in practice**

# Measuring Relevance: NDCG Edition!

We need 3 things in our **BENCHMARK DATASET**:

| English | Math | Picture |
|---|---|---|
| 1) A set of documents | $D = \{(d_i, q_j, r_{ij})\}$ | D |
| 2) A set of queries | $d_i$ is a vector | |
| 3) An assessment of the **relevance** for each query and each document:<br>0 Not relevant<br>1 Somewhat relevant<br>2 Really relevant<br>3 Perfectly relevant | $q_j$ is a vector<br>$r_{ij} \in \{0,1,2,3\}$ | |

| Documents | Queries | Relevance |
|---|---|---|
| $d_1$ | $q_1$ | $r_{11}$ |
| $d_1$ | $q_2$ | $r_{12}$ |
| $d_1$ | $q_3$ | $r_{13}$ |
| … | … | … |

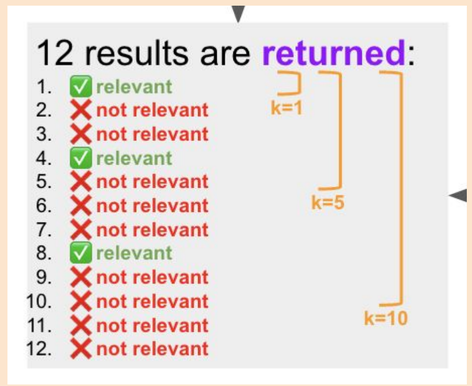
# NDCG

$$\mathrm{nDCG}_p = \frac{DCG_p}{IDCG_p}$$

n is for *normalized*

p is for *position*, and it's the same thing as **k**, the # of positions returned from the algorithm that we consider!

Remember this?



cumulative

$$\mathrm{DCG}_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

gain

discounting!

Let's analyze the numerator with an example: Let's consider what happens if the **1st position (i=1)** has different **relevance scores**:

*If* ***rel$_1$=3*** *:* $2^3$-1 = 8-1 = 7

*If* ***rel$_1$=2*** *:* $2^2$-1 = 4-1 = 3

*If* ***rel$_1$=1*** *:* $2^1$-1 = 2-1 = 1

*If* ***rel$_1$=0*** *:* $2^0$-1 = 1-1 = 0

**So, the better relevance score you have, the more points you get in the numerator!**

Let's analyze the denominator with an example: Let's consider what happens based on **which position (i)** we are calculating for:

*If* ***i=1*** *:* $\log_2(1+1) = \log_2(2) = 1$

*If* ***i=2*** *:* $\log_2(2+1) = \log_2(3) = 1.58$-ish

*If* ***i=3*** *:* $\log_2(3+1) = \log_2(4) = 2$

*If* ***i=4*** *:* $\log_2(4+1) = \log_2(5) = 2.32$-ish

**So the DCG scores are penalized based on rank!**

**Easy place to make a mistake! Just write it out :)**

$2^0$=1 and $\log_2(1)$=0

$2^1$=2 and $\log_2(2)$=1

$2^2$=4 and $\log_2(4)$=2

$2^3$=8 and $\log_2(8)$=3

**In my brain, I write out the left column, then I say "take log base 2 of both sides"**

# DCG: Example

$$\mathrm{DCG}_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

q = meet me at midnight

1.  1
2.  0
3.  0
4.  3

$$\mathrm{DCG}_4 = \frac{2^1-1}{\log_2(1+1)} + \frac{2^0-1}{\log_2(2+1)} + \frac{2^0-1}{\log_2(3+1)} + \frac{2^3-1}{\log_2(4+1)}$$

(i=1, i=2, i=3, i=4)

$$= \frac{1}{\log_2(2)} + \frac{0}{\log_2(2+1)} + \frac{0}{\log_2(3+1)} + \frac{7}{\log_2(5)}$$

$$= 1 + \frac{7}{2.32}$$

$$= 4.02$$

# Ideal DCG

For a **query**, what is the best possible **set of ranked results** (set of **docs** & their **relevance values**) we could return from our BENCHMARK DATASET?

In practice, our search engine *super-probably-most-likely CAN'T* achieve this (it would have to be literally perfect), but we can look in our BENCHMARK DATASET as an **"oracle"** to identify possible **set of ranked results** (set of **docs** & their **relevance values**)!

# Ideal DCG

Some **queries** are “easy” → there are lots of great **documents** for it in the BENCHMARK DATASET

Other **queries** are “hard” → even in the best case, there are not many good **documents** for it in the BENCHMARK DATASET

NDCG **normalizes** for these different scenarios

# IDCG: Example

$$\mathrm{DCG}_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

q = meet me at midnight

1. 3
2. 2
3. 2
4. 0

$$\mathrm{IDCG}_4 = \underset{i=1}{\frac{2^3-1}{\log_2(1+1)}} + \underset{i=2}{\frac{2^2-1}{\log_2(2+1)}} + \underset{i=3}{\frac{2^2-1}{\log_2(3+1)}} + \underset{i=4}{\frac{2^0-1}{\log_2(4+1)}}$$

$$= \frac{7}{\log_2(2)} + \frac{3}{\log_2(2+1)} + \frac{3}{\log_2(3+1)} + \frac{0}{\log_2(5)} \quad (\log_2(5) \to 0)$$

$$= \frac{7}{1} + \frac{3}{1.58} + \frac{3}{1.58}$$

$$= 10.80$$

# Putting it all together …

$$\mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$$

$$NDCG_4 = \frac{DCG_4}{IDCG_4} = \frac{4.02}{10.80} = 0.37$$