# Ciencia de Datos

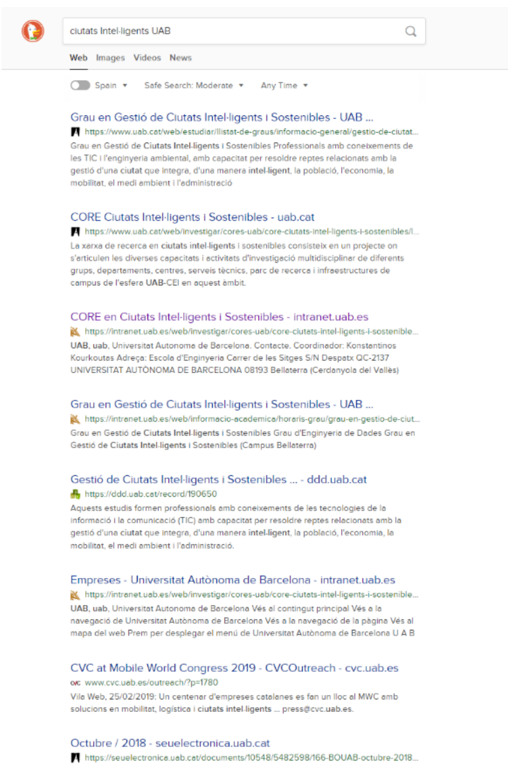
Lecture 10:
Information Retrieval and
Recommender Systems

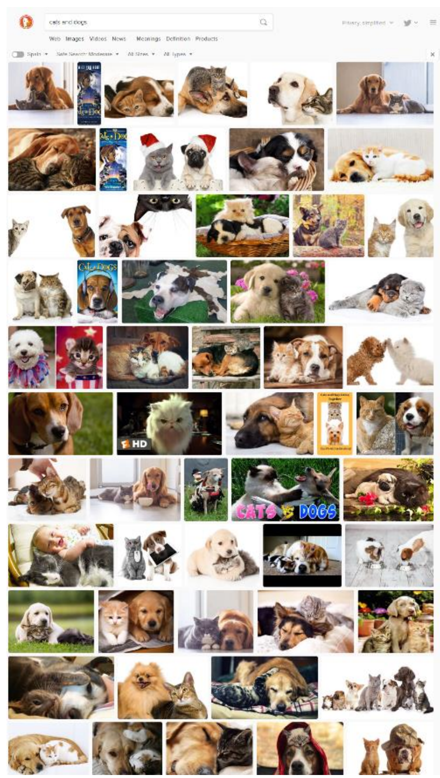
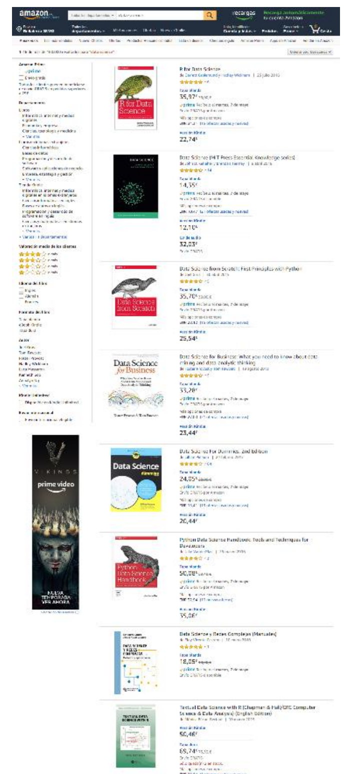
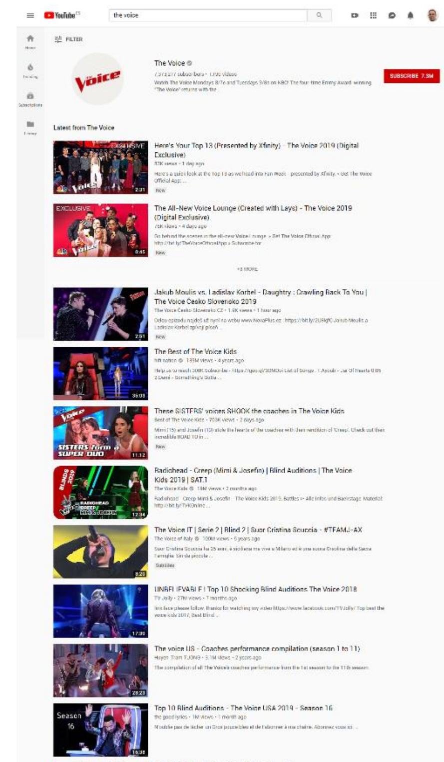
Dimosthenis Karatzas (dimos@cvc.uab.es)

# INFORMATION RETRIEVAL

# Information Retrieval

**Information Retrieval** is finding material [...] of an unstructured nature [...] that satisfies an information need, from within large collections [...] [1]



(1) *Manning, Raghavan, and Schütze, Introduction to information retrieval, Cambridge University Press, 2008*

# What makes a good Search Engine?

## User Satisfaction

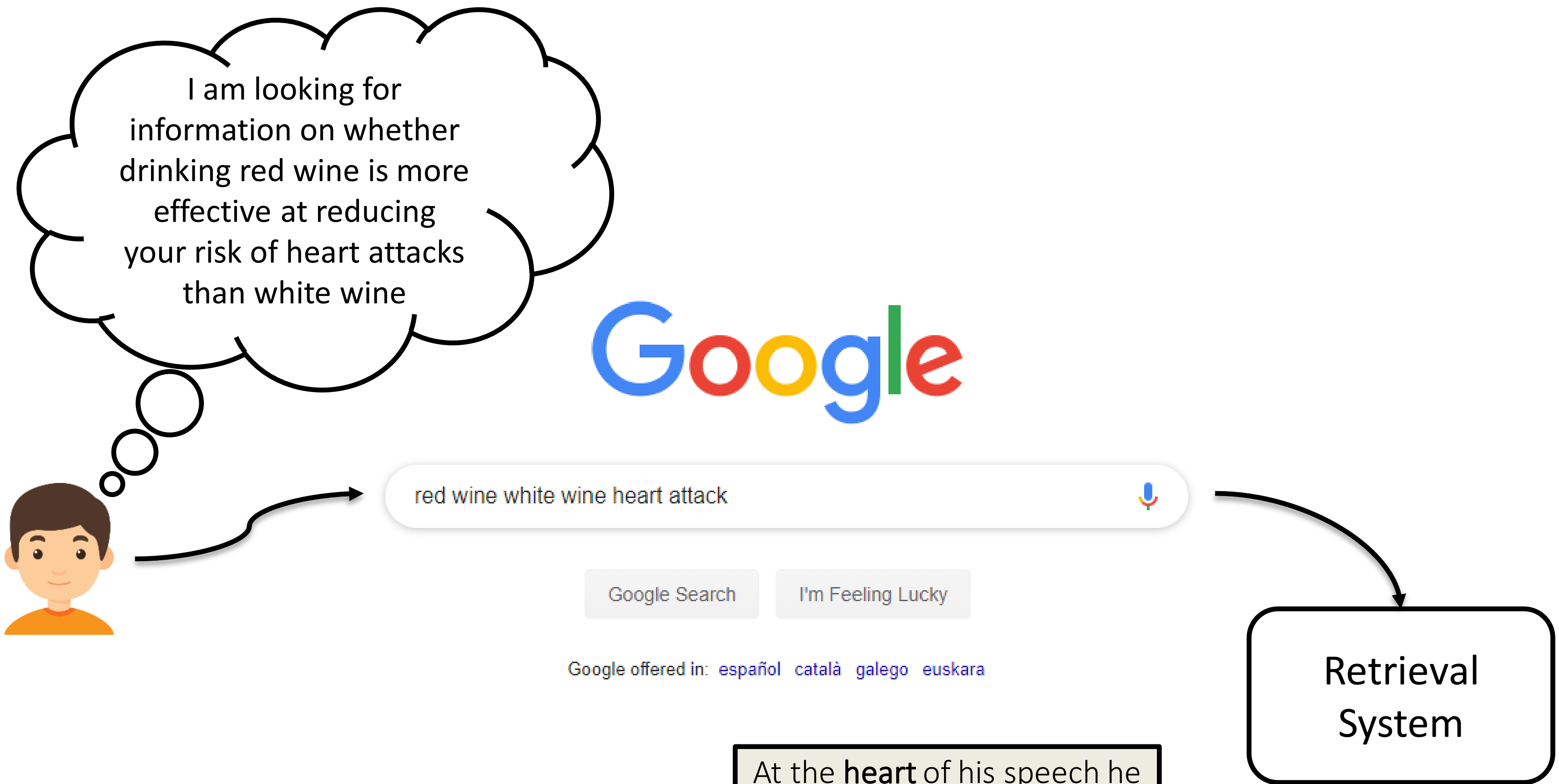
How much information does it index
(e.g., number of Web pages)

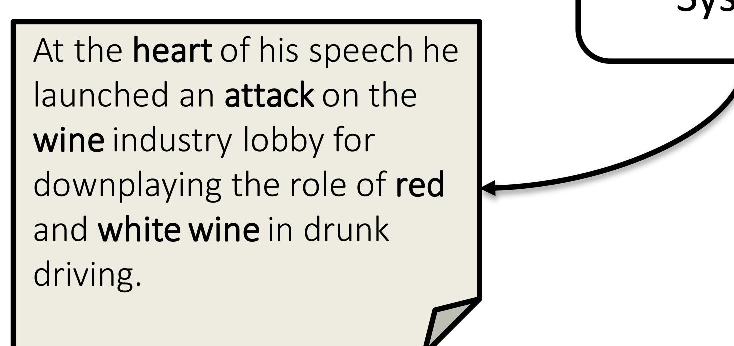
How fast does it search
(e.g., latency as a function of queries per second)

What is the cost per query?
(in dollars)

**Relevance of the results**
**(*how do you measure this?*)**

# Relevance in respect to what?



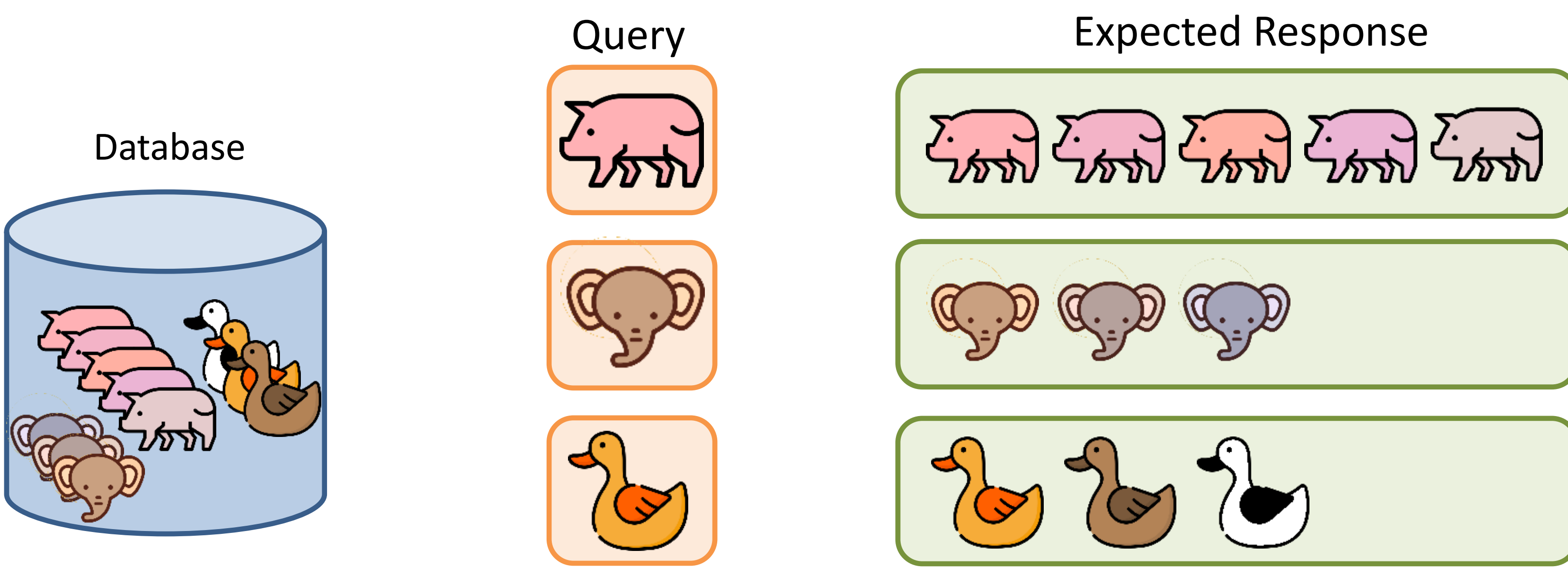




Relevance should be measured in terms of an **information need**, not in terms of "similarity" to the query.

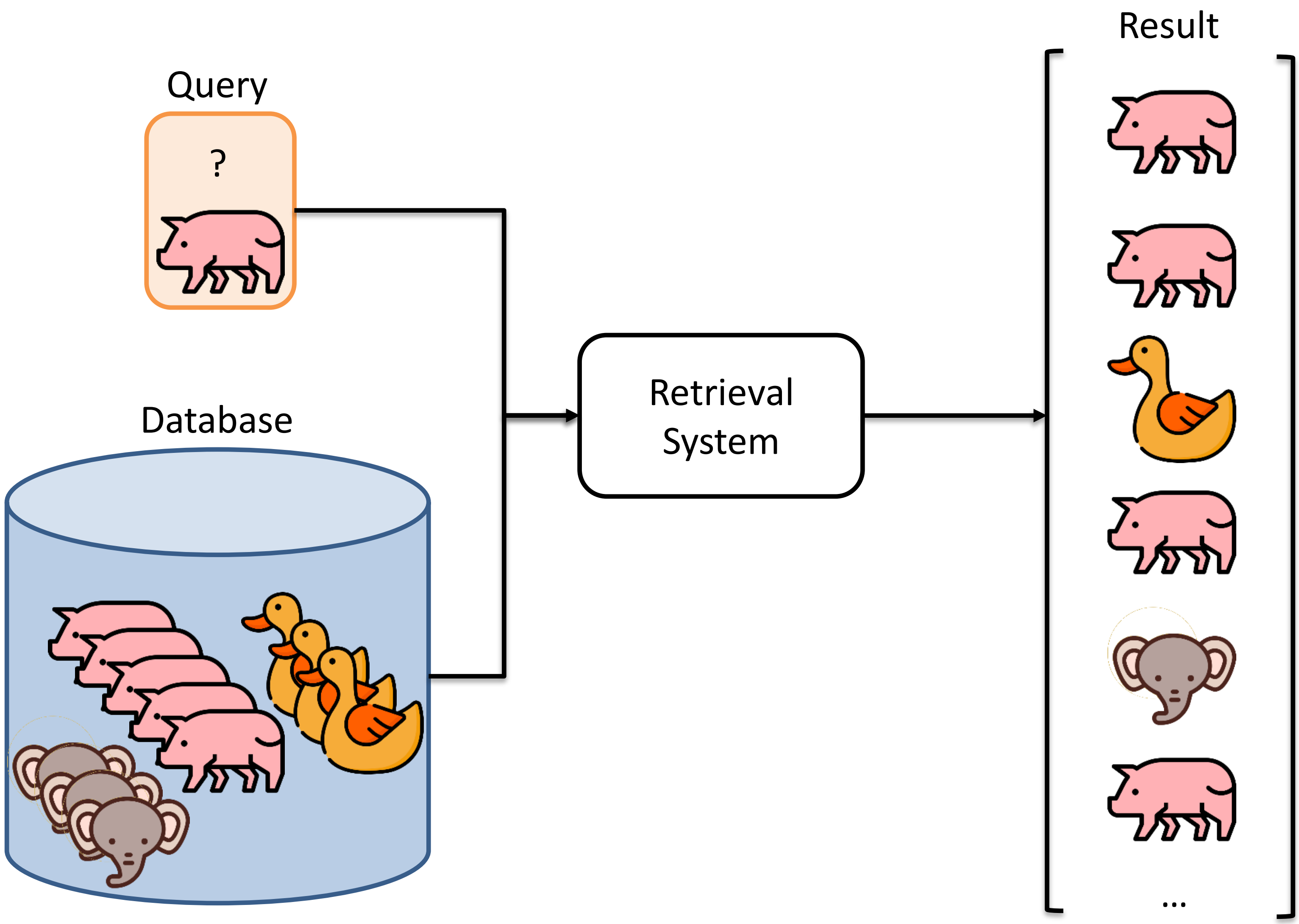
# Measuring Relevance

Standard methodology in information retrieval performance evaluation consists of three elements:

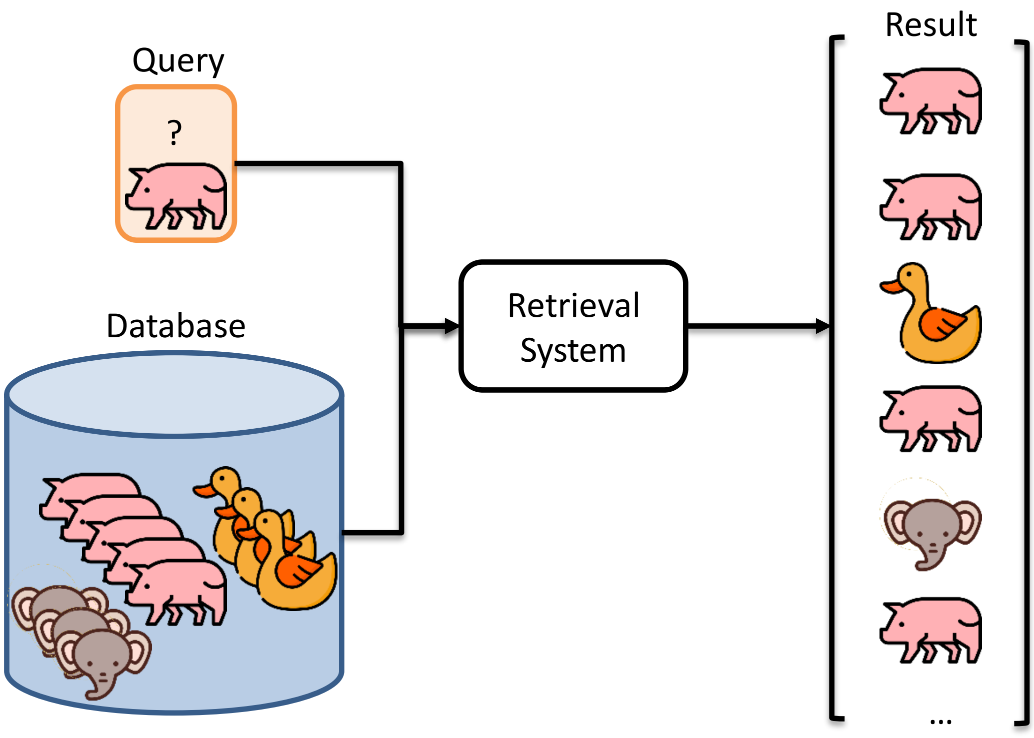
- A benchmark collection of items
- A benchmark suite of queries
- An assessment of the relevance of each query-item pair (each item is relevant or not to the query)



# Retrieving Information



# Retrieving Information

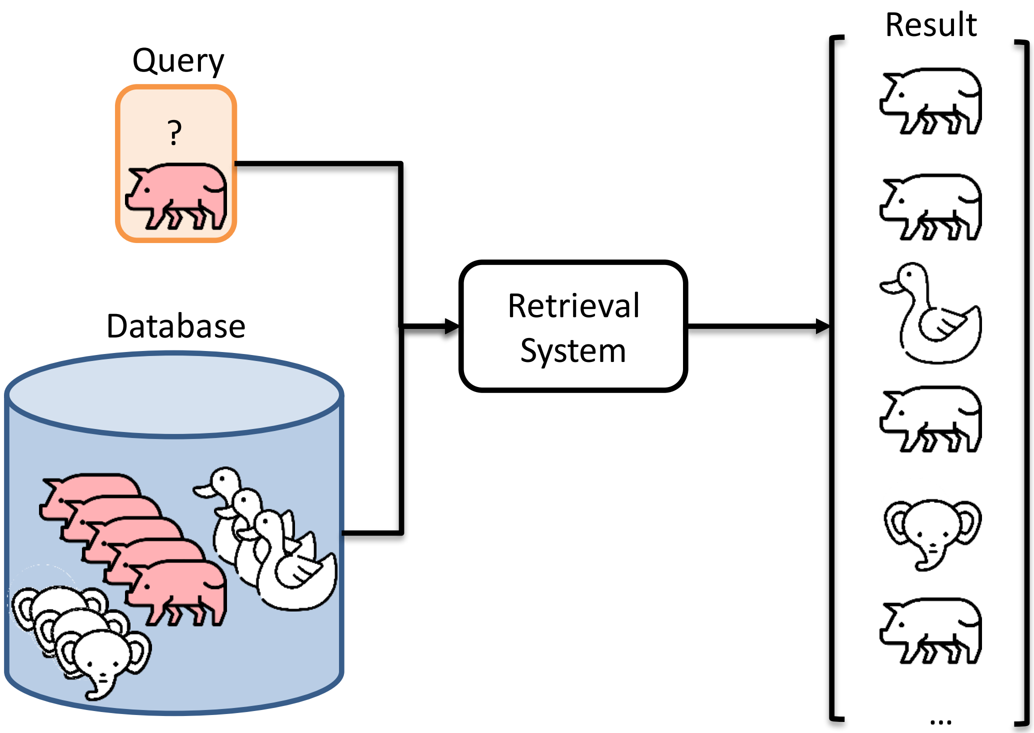


# Retrieving Information

**Relevant**

$|rel|$

Relevant:

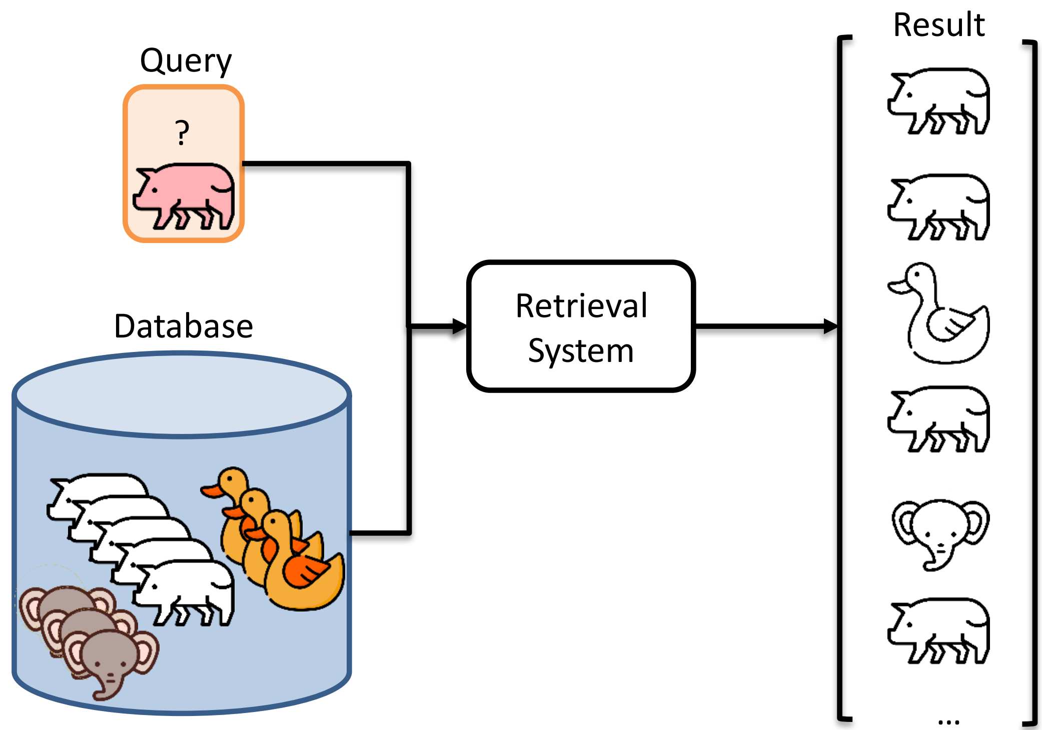


# Retrieving Information

**Relevant** **Non-Relevant**

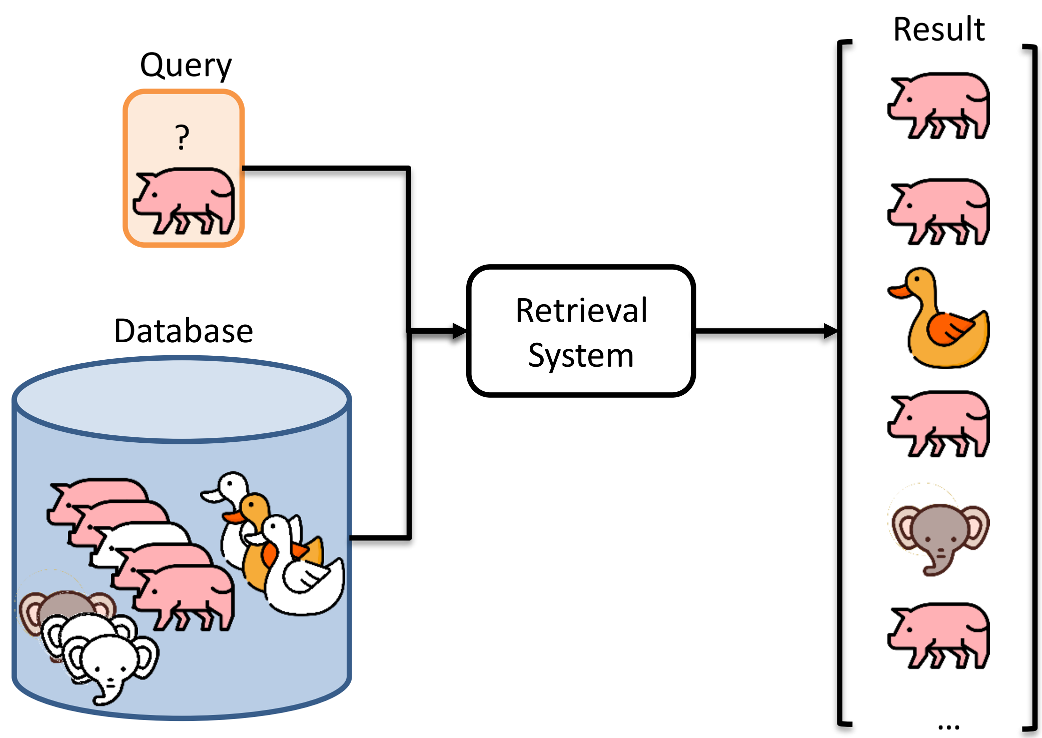
$|rel|$ $|\overline{rel}|$

Non-Relevant:



# Retrieving Information

| | Relevant | Non-Relevant | |
|---|---|---|---|
| **Retrieved** | | | $|ret|$ |
| | $|rel|$ | $|\overline{rel}|$ | |

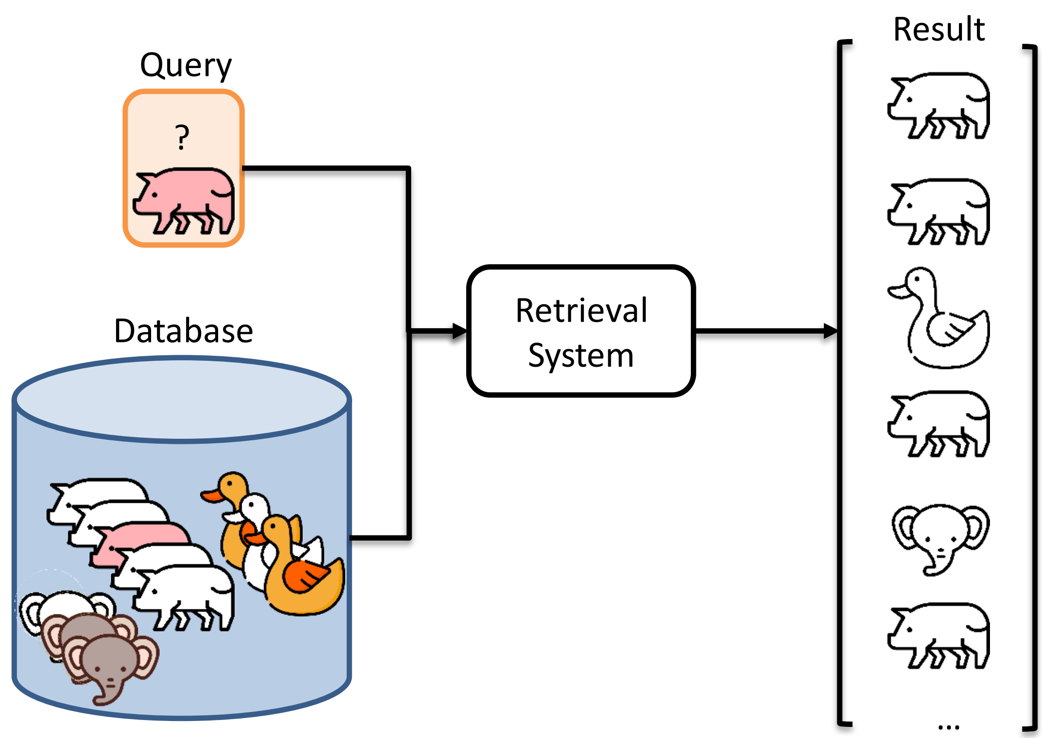



Retrieved:

# Retrieving Information

| | **Relevant** | **Non-Relevant** | |
|---|---|---|---|
| **Retrieved** | | | $\lvert ret \rvert$ |
| **Not Retrieved** | | | $\lvert \overline{ret} \rvert$ |
| | $\lvert rel \rvert$ | $\lvert \overline{rel} \rvert$ | |

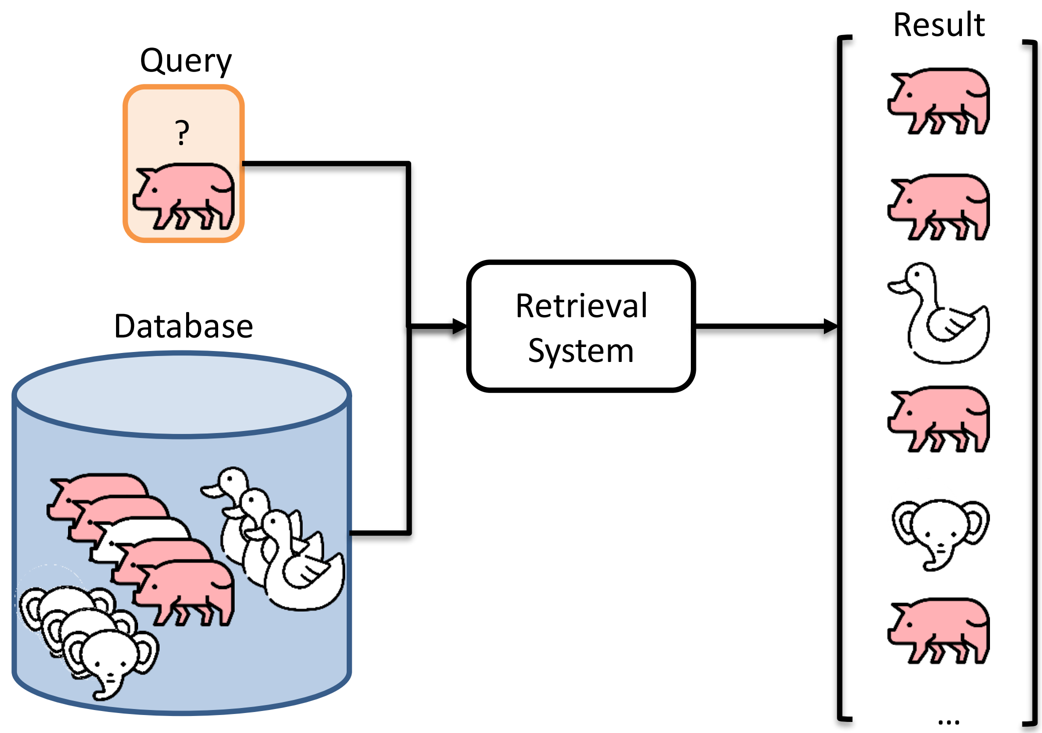
Not Retrieved:



# Retrieving Information

| | **Relevant** | **Non-Relevant** | |
|---|---|---|---|
| **Retrieved** | $\lvert ret \cap rel \rvert$ | | $\lvert ret \rvert$ |
| **Not Retrieved** | | | $\lvert \overline{ret} \rvert$ |
| | $\lvert rel \rvert$ | $\lvert \overline{rel} \rvert$ | |

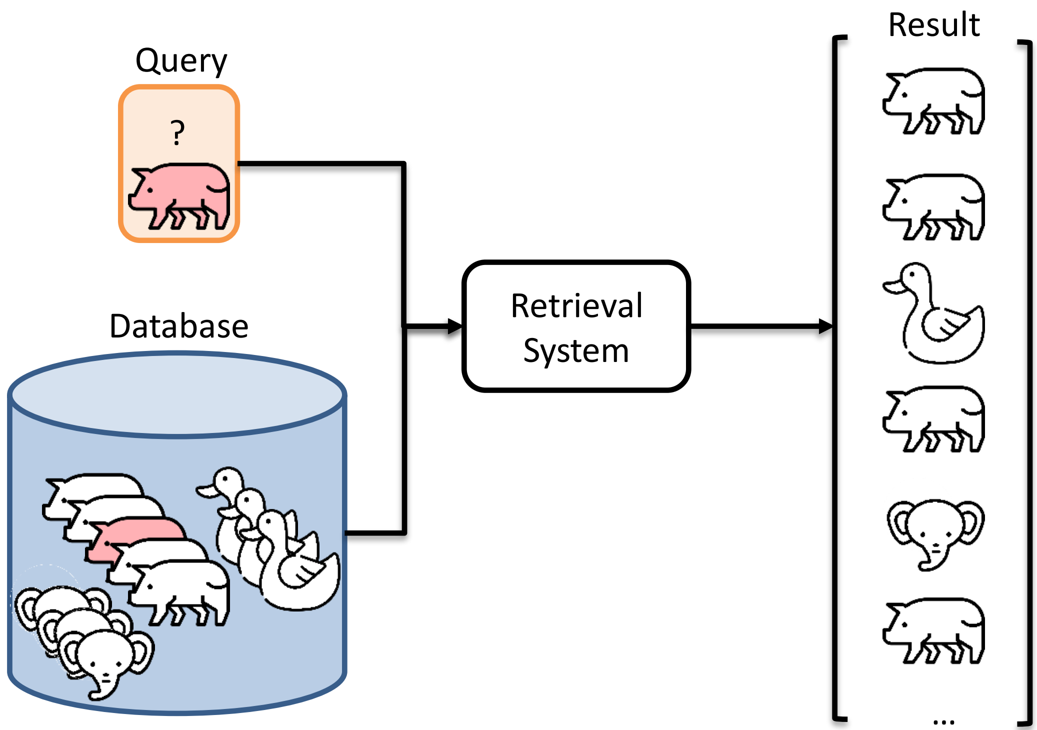
Retrieved and Relevant:



# Retrieving Information

| | Relevant | Non-Relevant | |
|---|---|---|---|
| **Retrieved** | $\lvert ret \cap rel \rvert$ | | $\lvert ret \rvert$ |
| **Not Retrieved** | $\lvert \overline{ret} \cap rel \rvert$ | | $\lvert \overline{ret} \rvert$ |
| | $\lvert rel \rvert$ | $\lvert \overline{rel} \rvert$ | |

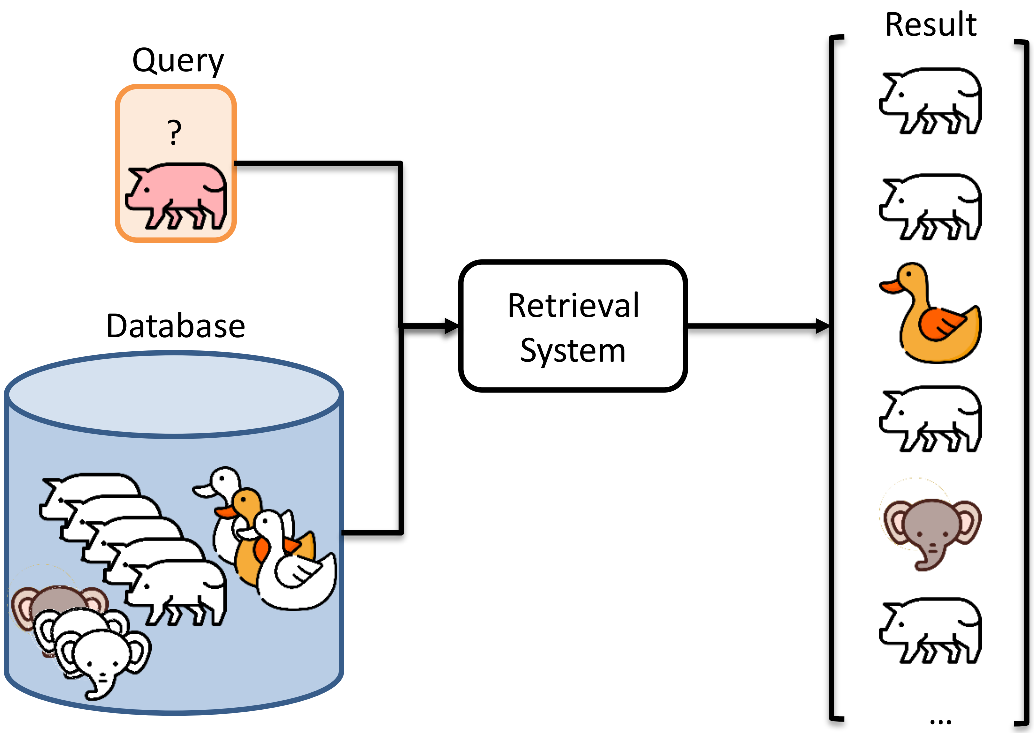
Not Retrieved and Relevant:



# Retrieving Information

| | Relevant | Non-Relevant | |
|---|---|---|---|
| **Retrieved** | $\lvert ret \cap rel \rvert$ | $\lvert ret \cap \overline{rel} \rvert$ | $\lvert ret \rvert$ |
| **Not Retrieved** | $\lvert \overline{ret} \cap rel \rvert$ | | $\lvert \overline{ret} \rvert$ |
| | $\lvert rel \rvert$ | $\lvert \overline{rel} \rvert$ | |

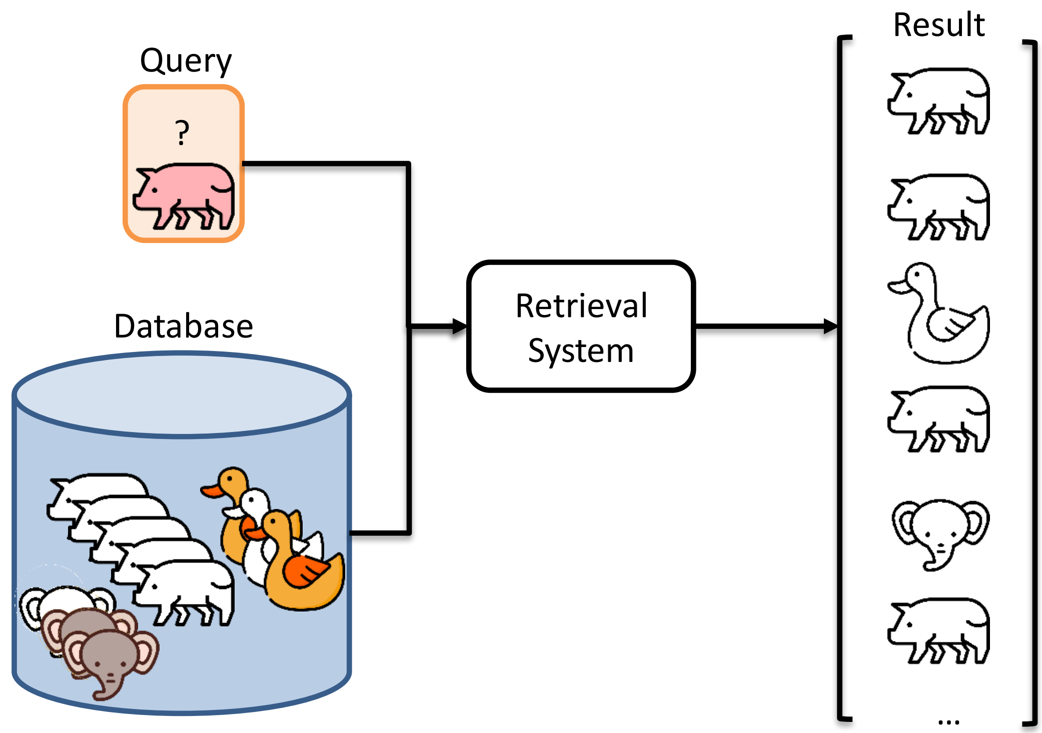
Retrieved and Non-Relevant:



# Retrieving Information

| | **Relevant** | **Non-Relevant** | |
|---|---|---|---|
| **Retrieved** | $\lvert ret \cap rel \rvert$ | $\lvert ret \cap \overline{rel} \rvert$ | $\lvert ret \rvert$ |
| **Not Retrieved** | $\lvert \overline{ret} \cap rel \rvert$ | $\lvert \overline{ret} \cap \overline{rel} \rvert$ | $\lvert \overline{ret} \rvert$ |
| | $\lvert rel \rvert$ | $\lvert \overline{rel} \rvert$ | |

Not Retrieved and Non-Relevant:



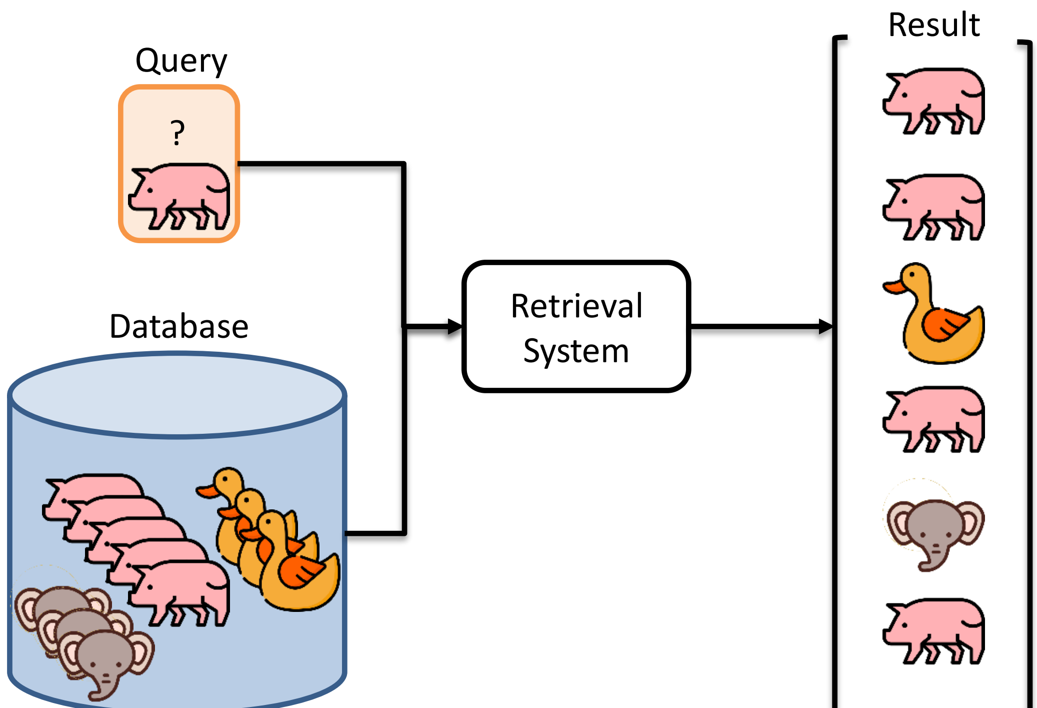
# Precision and Recall

| | Relevant | Non-Relevant | TOTAL |
|---|---|---|---|
| **Retrieved** | $\lvert ret \cap rel \rvert$ | $\lvert ret \cap \overline{rel} \rvert$ | $\lvert ret \rvert$ |
| **Not Retrieved** | $\lvert \overline{ret} \cap rel \rvert$ | $\lvert \overline{ret} \cap \overline{rel} \rvert$ | $\lvert \overline{ret} \rvert$ |
| **TOTAL** | $\lvert rel \rvert$ | $\lvert \overline{rel} \rvert$ | |

**Precision** is the fraction of retrieved documents which are relevant.

$$\boldsymbol{P} = \frac{\lvert \boldsymbol{ret} \cap \boldsymbol{rel} \rvert}{\lvert \boldsymbol{ret} \rvert}$$

It measures the quality of the retrieval system in terms of its ability to only include relevant items in the result



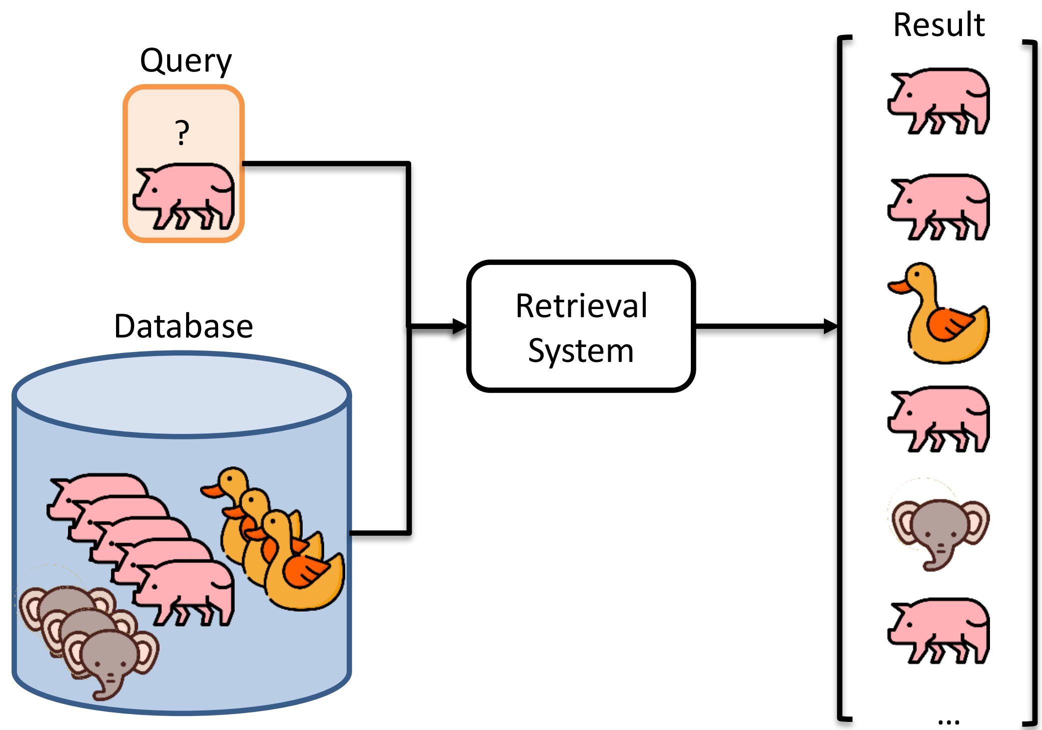
# Precision and Recall

| | Relevant | Non-Relevant | TOTAL |
|---|---|---|---|
| **Retrieved** | $\lvert ret \cap rel \rvert$ | $\lvert ret \cap \overline{rel} \rvert$ | $\lvert ret \rvert$ |
| **Not Retrieved** | $\lvert \overline{ret} \cap rel \rvert$ | $\lvert \overline{ret} \cap \overline{rel} \rvert$ | $\lvert \overline{ret} \rvert$ |
| **TOTAL** | $\lvert rel \rvert$ | $\lvert \overline{rel} \rvert$ | |

**Recall** is the fraction of relevant documents which has been retrieved

$$\boldsymbol{R} = \frac{\lvert \boldsymbol{ret} \cap \boldsymbol{rel} \rvert}{\lvert \boldsymbol{rel} \rvert}$$

It measures the effectiveness of the system in retrieving all relevant items that exists



# Precision and Recall

**Precision** is the fraction of retrieved documents which are relevant.

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

**Recall** is the fraction of relevant documents which has been retrieved

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

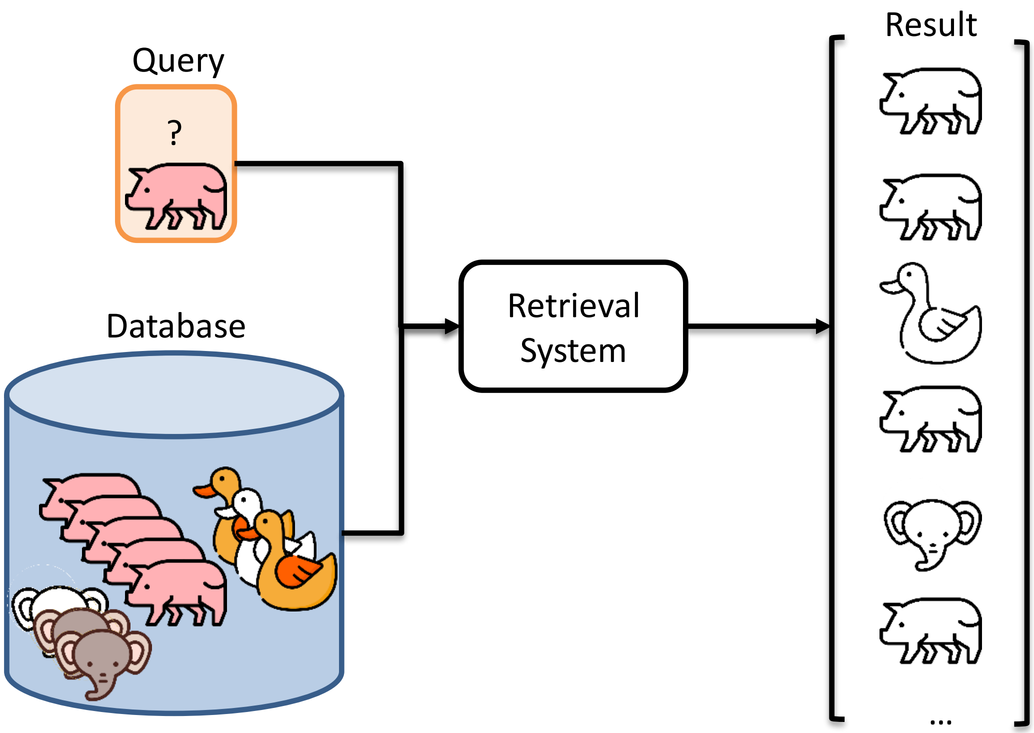

20

# Precision and Recall

| | **Relevant** | **Non-Relevant** |
|---|---|---|
| **Retrieved** | True positives (*TP*) | False Positives (*FP*) |
| **Not Retrieved** | False Negatives (*FN*) | True Negatives (*TN*) |

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$



# Precision / Recall Trade-off

You can (*usually*) increase **Precision** by being more strict in what you return. Assuming that you have ranked items well.

**Recall** is a non-decreasing function of the number of items retrieved.

You can increase recall by returning more items.

A system that returns all items in the dataset by definition yields 100% recall.

Return less stuff

Return more stuff

# A combined measure: F

The F-measure allows us to combine Precision and Recall in a single value.

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}}$$

To balance the importance between Precision and Recall, we can set $\alpha = 0.5$.

This particular value is called the $F_1$ or H-mean metric, as it is the harmonic mean of P and R:

$$\frac{1}{F_1} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)$$

Or equivalently:

$$F_1 = \frac{2PR}{P+R}$$

# Example

| | **Relevant** | **Non-Relevant** |
|---|---|---|
| **Retrieved** | 20 | 40 |
| **Not Retrieved** | 60 | 1,000,000 |

$$P = \frac{20}{20+40} = \frac{1}{3} \qquad R = \frac{20}{20+60} = \frac{1}{4}$$

$$F_1 = \frac{2\frac{1}{3}\frac{1}{4}}{\frac{1}{3}+\frac{1}{4}} = \frac{2}{7}$$

# Why not accuracy?

| | Relevant | Non-Relevant |
|---|---|---|
| **Retrieved** | True positives (*TP*) | False Positives (*FP*) |
| **Not Retrieved** | False Negatives (*FN*) | True Negatives (*TN*) |

**Accuracy** is the fraction of decisions (relevant or non relevant) that are correct (retrieved or not retrieved correspondingly)

$$\boldsymbol{Accuracy = \frac{TP + TN}{TP + FP + FN + TN}}$$

Why is accuracy not a useful measure for information retrieval?


25

# Try this out

| | Relevant | Non-Relevant |
|---|---|---|
| **Retrieved** | 2 | 18 |
| **Not Retrieved** | 98 | 1,000,000,000 |

Compute Precision, Recall and Accuracy for the above result

$$Precision = 10.0\%$$

$$Recall = 2.0\%$$

$$Accuracy = 99.9999\%$$

A search engine that always returns 0 results, regardless of the query would yield a very high Accuracy in the above scenario.


26

# Taking ranking into account

Precision, Recall and F are measures for unranked sets.

| | P | R |
|---|---|---|
| | 5/8 | 5/9 |

$$\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Total Relevant in the dataset: 9

# Taking ranking into account

Precision, Recall and F are measures for unranked sets.

To convert such set measures into measures of ranked lists, we can compute the set measure for each of the top-N sets

For each value of N, we would then obtain the "Precision at N" and "Recall at N"

$$
\begin{array}{cccc}
 & P & R & \\
\begin{bmatrix} \boxed{1} \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} &
\begin{bmatrix} 1/1 \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} &
\begin{bmatrix} 1/9 \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} & \begin{matrix} @1 \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \\
\dots & & &
\end{array}
$$

Total Relevant in the dataset: 9

# Taking ranking into account

Precision, Recall and F are measures for unranked sets.

To convert such set measures into measures of ranked lists, we can compute the set measure for each of the top-N sets

For each value of N, we would then obtain the "Precision at N" and "Recall at N"

$$\begin{array}{cccc}
 & P & R & \\
\begin{bmatrix}1\\1\\0\\1\\0\\0\\1\\1\end{bmatrix} & \begin{bmatrix}1/1\\2/2\\ \\ \\ \\ \\ \\ \\ \end{bmatrix} & \begin{bmatrix}1/9\\2/9\\ \\ \\ \\ \\ \\ \\ \end{bmatrix} & \begin{matrix}@1\\@2\\ \\ \\ \\ \\ \\ \\ \end{matrix} \\
\dots & & &
\end{array}$$

Total Relevant in the dataset: 9

# Taking ranking into account

Precision, Recall and F are measures for unranked sets.

To convert such set measures into measures of ranked lists, we can compute the set measure for each of the top-N sets

For each value of N, we would then obtain the "Precision at N" and "Recall at N"

| | P | R | |
|---|---|---|---|
| 1 | 1/1 | 1/9 | @1 |
| 1 | 2/2 | 2/9 | @2 |
| 0 | 2/3 | 2/9 | @3 |
| 1 | | | |
| 0 | | | |
| 0 | | | |
| 1 | | | |
| 1 | | | |
| … | | | |

Total Relevant in the dataset: 9

# Taking ranking into account

Precision, Recall and F are measures for unranked sets.

To convert such set measures into measures of ranked lists, we can compute the set measure for each of the top-N sets

For each value of N, we would then obtain the “Precision at N” and “Recall at N”

$$
\begin{array}{cccc}
 & P & R & \\
\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} &
\begin{bmatrix} 1/1 \\ 2/2 \\ 2/3 \\ 3/4 \\ \\ \\ \\ \\ \end{bmatrix} &
\begin{bmatrix} 1/9 \\ 2/9 \\ 2/9 \\ 3/9 \\ \\ \\ \\ \\ \end{bmatrix} &
\begin{matrix} @1 \\ @2 \\ @3 \\ @4 \\ \\ \\ \\ \\ \end{matrix} \\
\dots & & &
\end{array}
$$

Total Relevant in the dataset: 9

# Taking ranking into account

Precision, Recall and F are measures for unranked sets.

To convert such set measures into measures of ranked lists, we can compute the set measure for each of the top-N sets

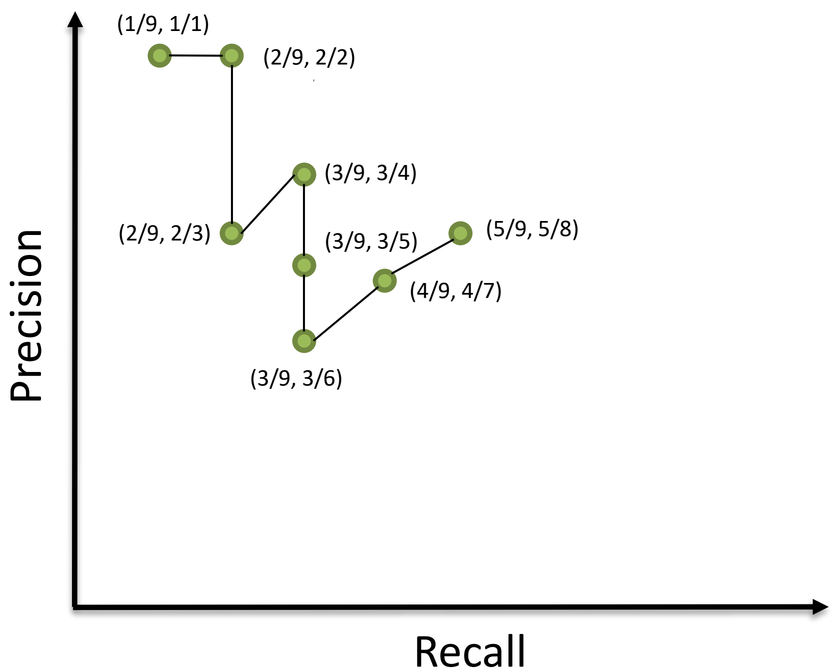
For each value of N, we would then obtain the "Precision at N" and "Recall at N"

All this can be summarised in the **precision-recall curve**.

| | P | R | |
|---|---|---|---|
| 1 | 1/1 | 1/9 | @1 |
| 1 | 2/2 | 2/9 | @2 |
| 0 | 2/3 | 2/9 | @3 |
| 1 | 3/4 | 3/9 | @4 |
| 0 | 3/5 | 3/9 | @5 |
| 0 | 3/6 | 3/9 | @6 |
| 1 | 4/7 | 4/9 | @7 |
| 1 | 5/8 | 5/9 | @8 |
| … | … | … | … |

Total Relevant in the dataset: 9

# Precision – Recall Curve




[1, 1, 0, 1, 0, 0, 1, 1, …, 1, …., 0, 1, …, 1, 0, …, 1, …]

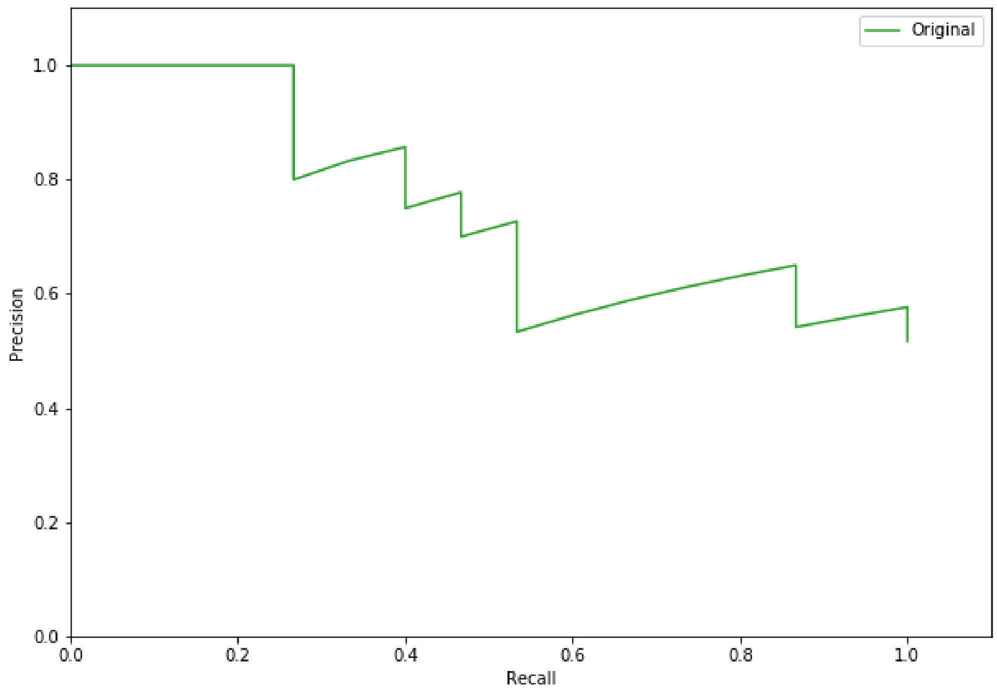
⇧ ⇧ ⇧ ⇧ ⇧ ⇧ ⇧ ⇧

@1 @2 @3 @4 @5 @6 @7 @8

Total Relevant in the dataset: 9

# Summarising the P-R plot

Rather than comparing curves, its sometimes useful to have a single number that characterizes the performance of a retrieval system for a given query (or classifier). A common metric is the **average precision**.

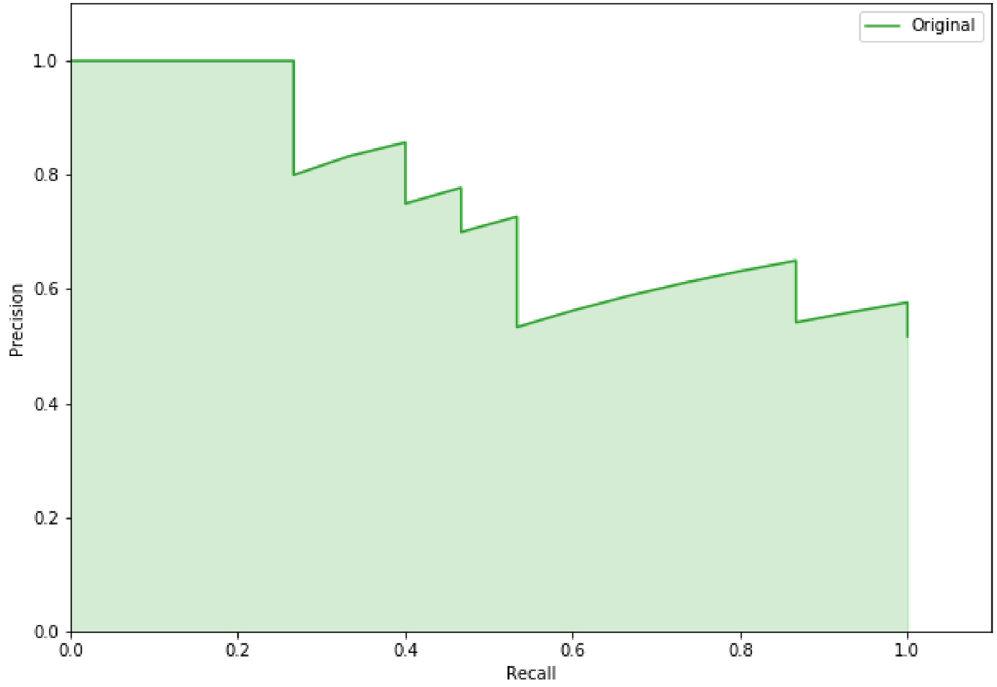



Average Precision rewards the earliest return of relevant items. Ranking is important.

Retrieving all relevant items in the collection and ranking them perfectly will lead to an average precision of 1.

# Average Precision

The average precision the precision averaged across all values of recall between 0 and 1. This is equal to the area under the P-R curve.

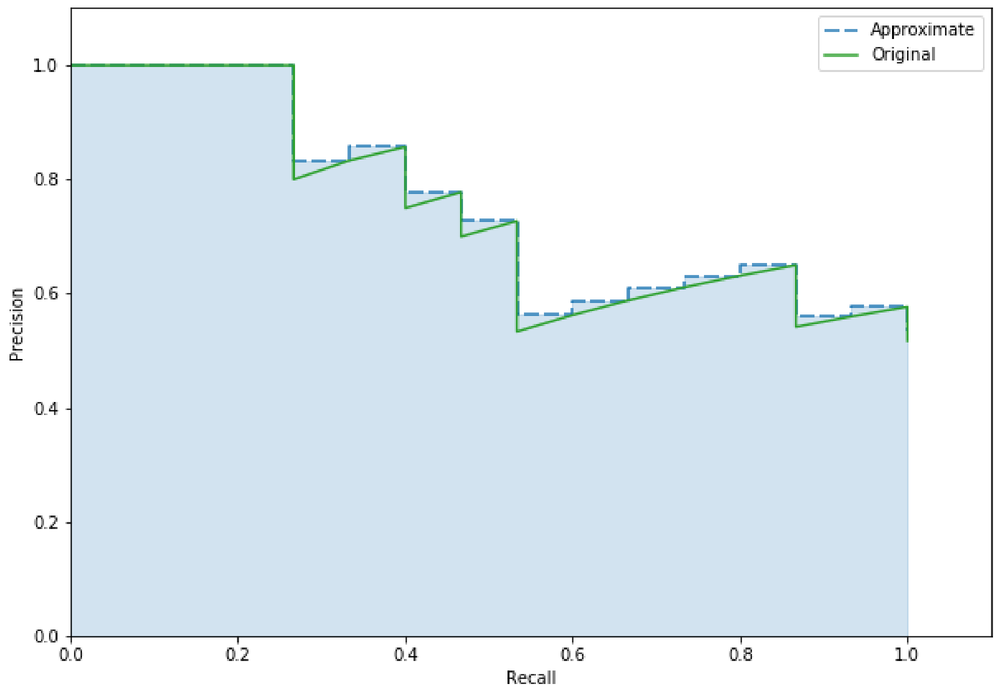



$$AP = \int_0^1 p(r)\,\mathrm{d}r$$

Result = [1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0]

# Average Precision (approximated)

In practice, the integral is closely approximated by a sum over the precisions at every possible threshold value, multiplied by the change in recall:




$$AP = \sum_{k=1}^{N} P(k)\,\Delta r(k)$$

Or alternatively:

$$AP = \frac{\sum_{k=1}^{N} P(k)\, rel(k)}{|rel|}$$

Notice that the points at which the recall doesn't change (the non-relevant ones) don't contribute to this sum:

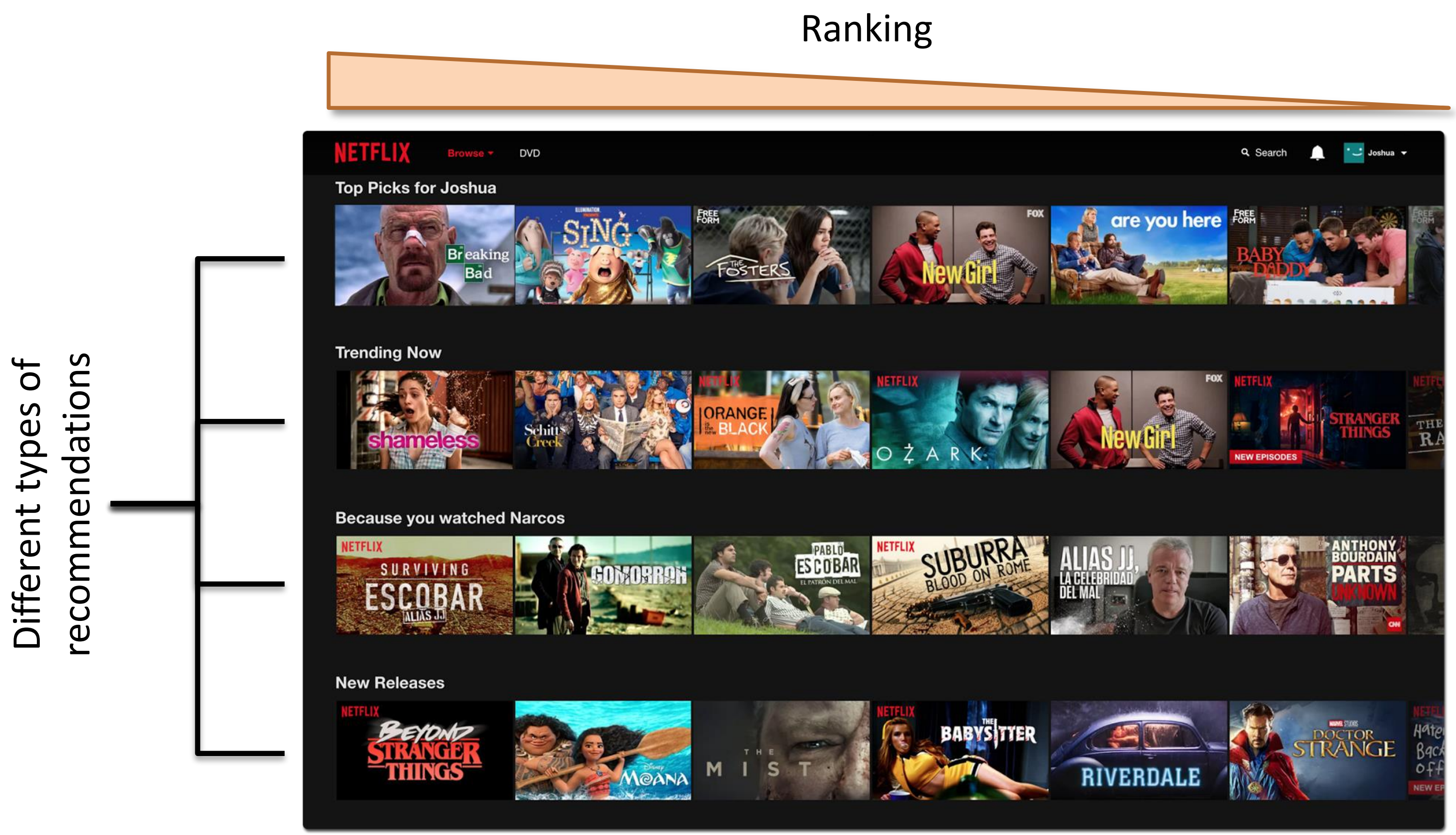
$$AP = \left(\frac{1}{1}\frac{1}{15}\right) + \left(\frac{2}{2}\frac{1}{15}\right) + \left(\frac{3}{3}\frac{1}{15}\right) + \left(\frac{4}{4}\frac{1}{15}\right) + \left(\frac{4}{5}0\right) + \left(\frac{5}{6}\frac{1}{15}\right) + \cdots$$

Result = [ 1, 1, 1, 1, 0, 1, … ]

# RECOMMENDER SYSTEMS

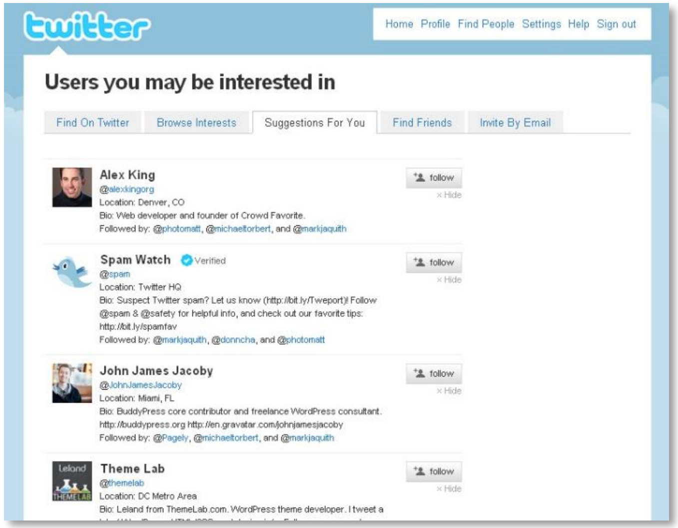
# Recommender systems

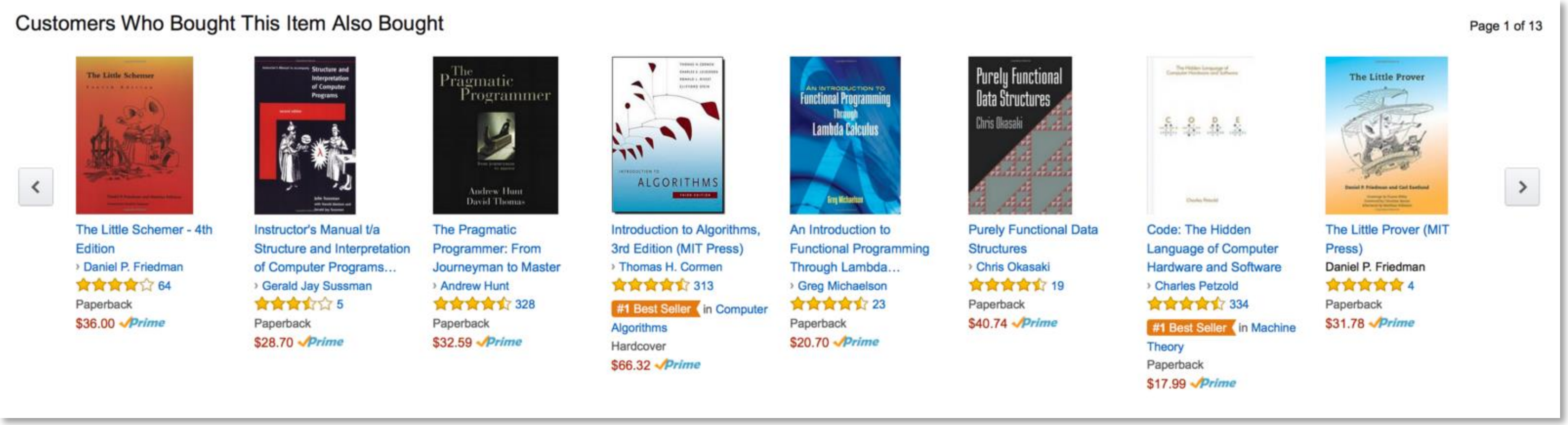
A special case of (implicit) information retrieval (or filtering) scenario is the recommender (or recommendation) system.



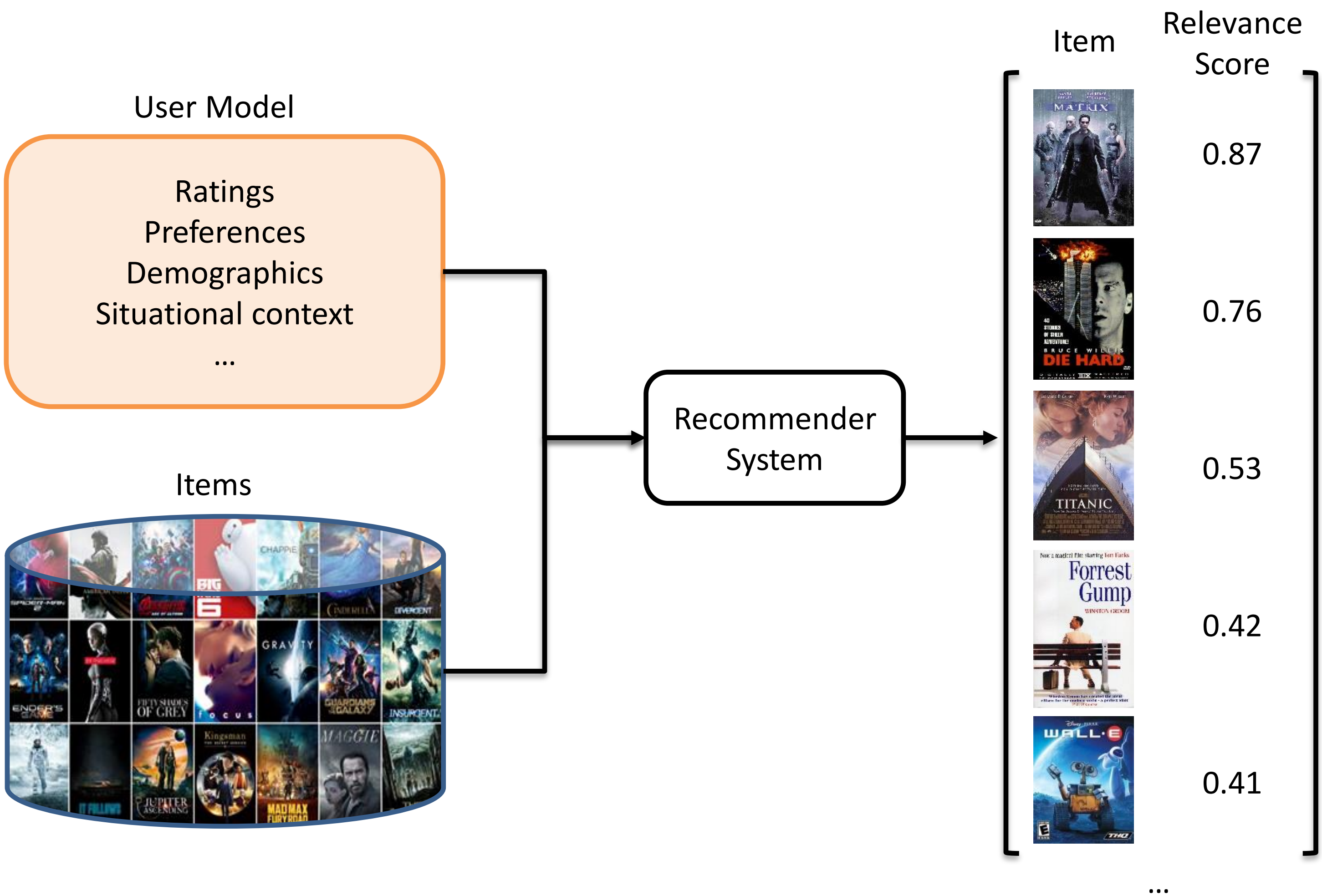
# Recommender systems

In this case, the user is not explicitly seeking for information (there is no query), instead the system is implicitly predicting the "rating" or "preference" the user would give to an item.



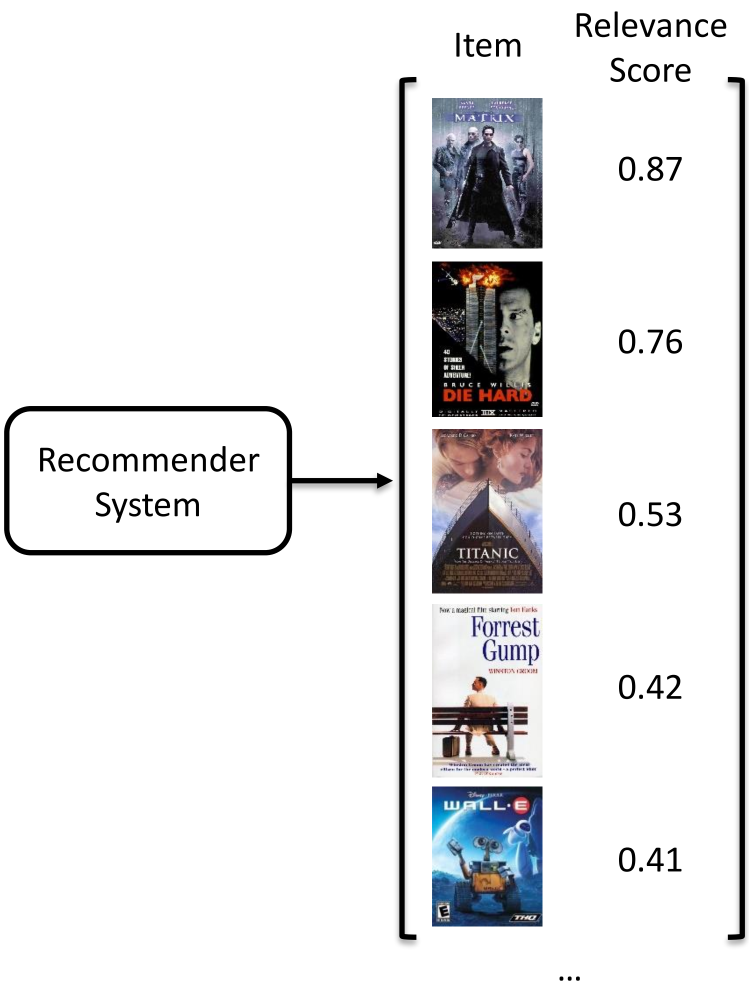



# Recommender systems



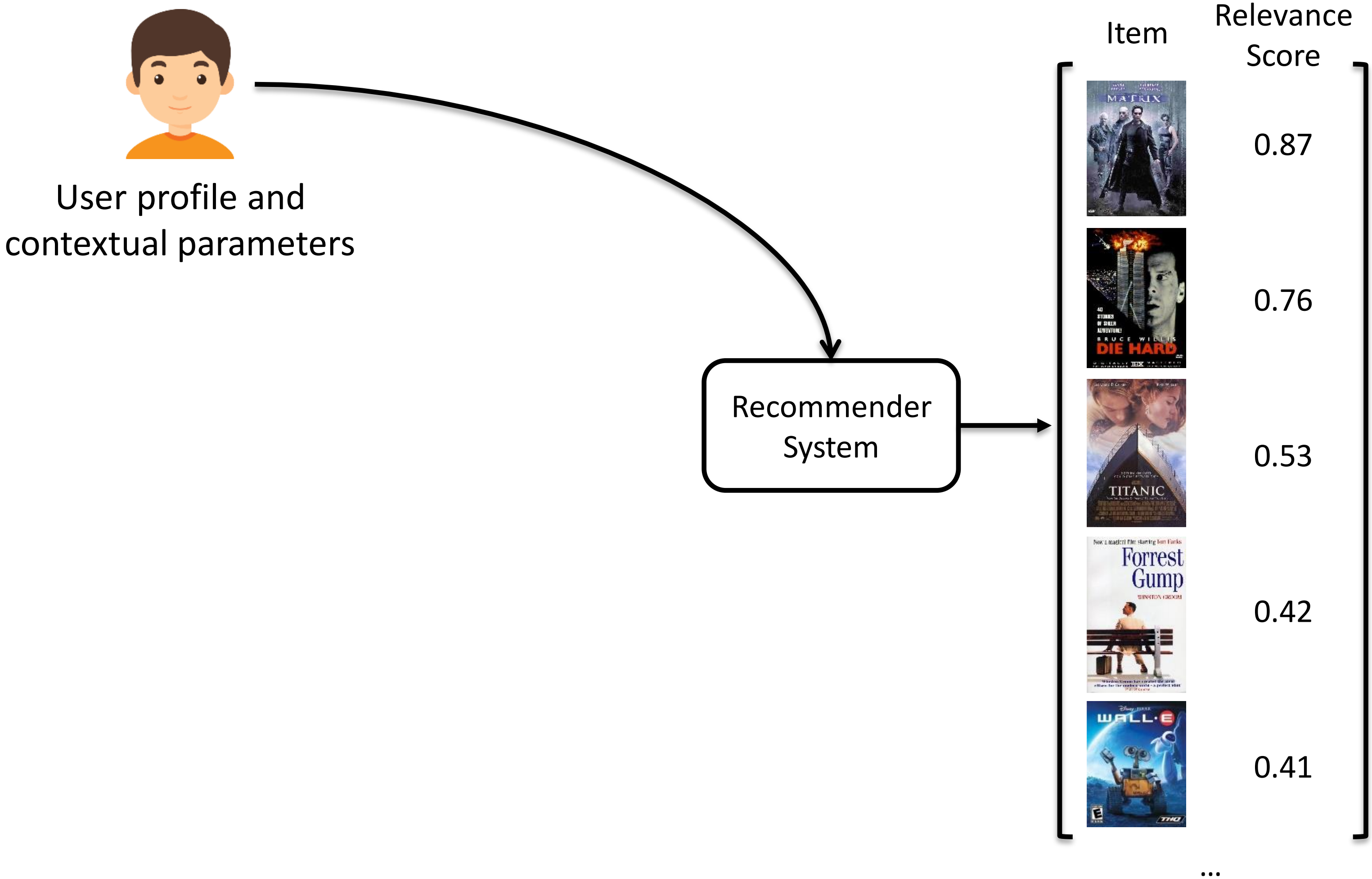

Recommender systems reduce information overload by estimating relevance

# Paradigms of recommender systems



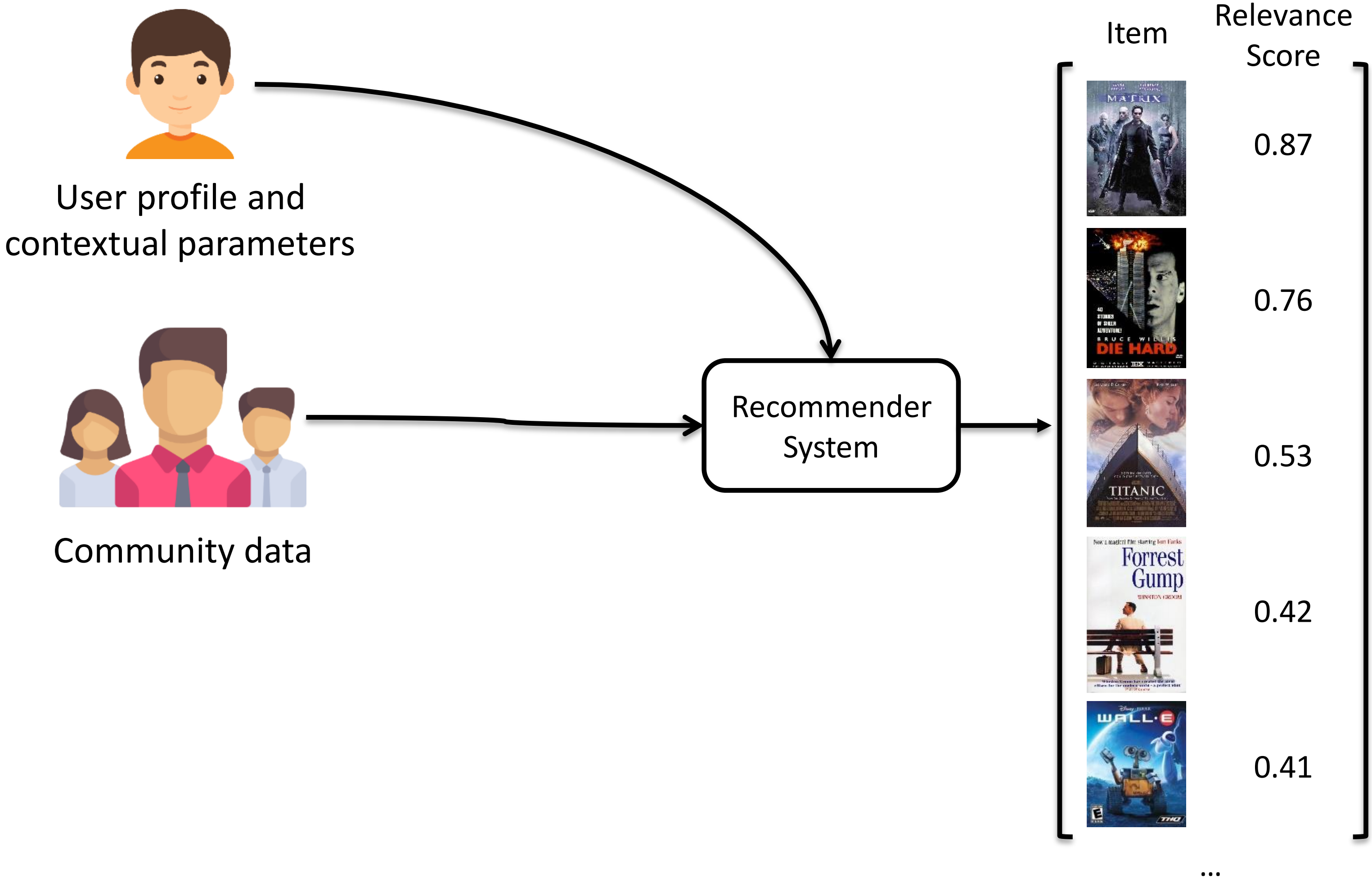

Recommender systems reduce information overload by estimating relevance

# Paradigms of recommender systems



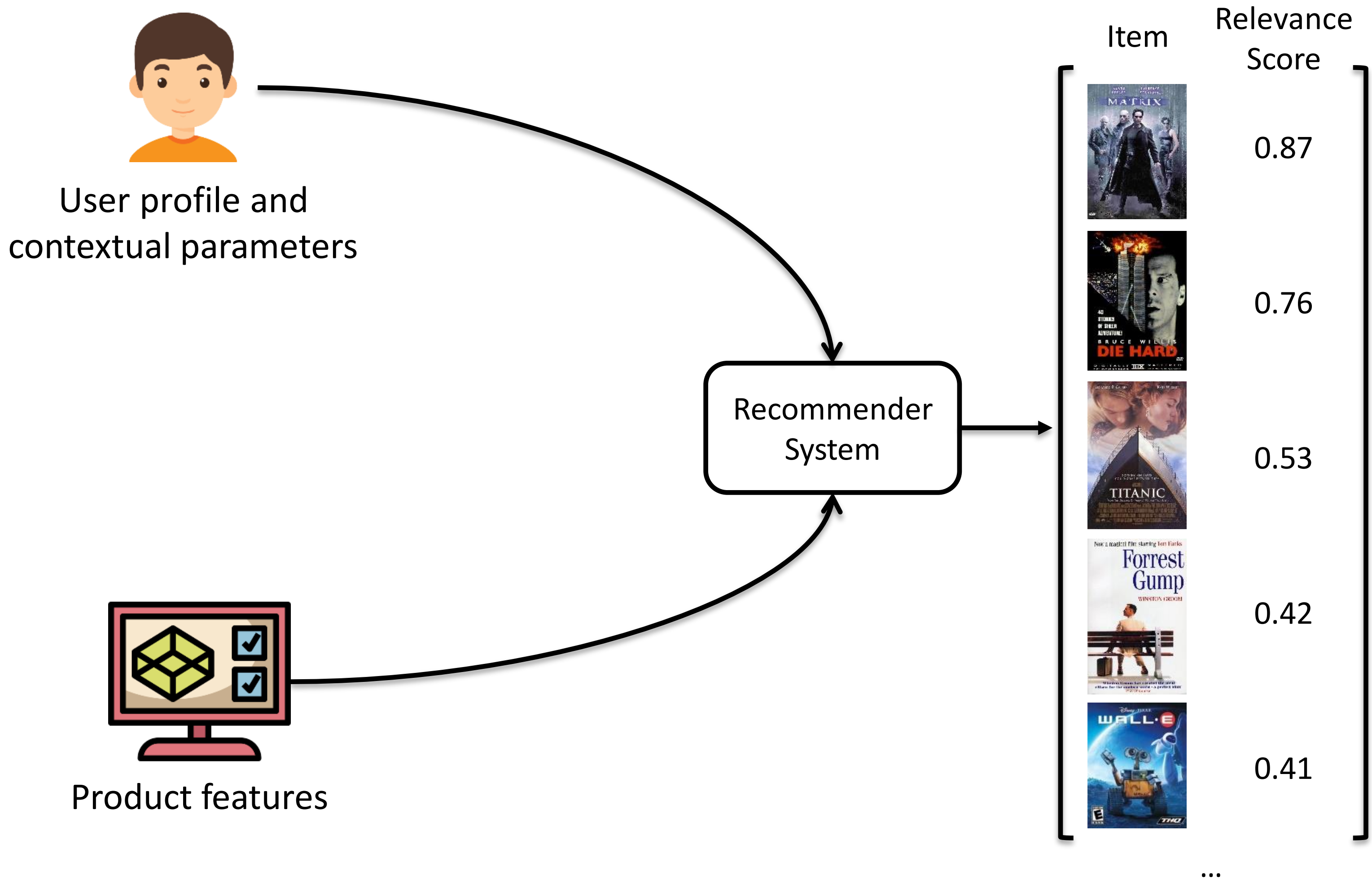

Personalized recommendations

# Paradigms of recommender systems



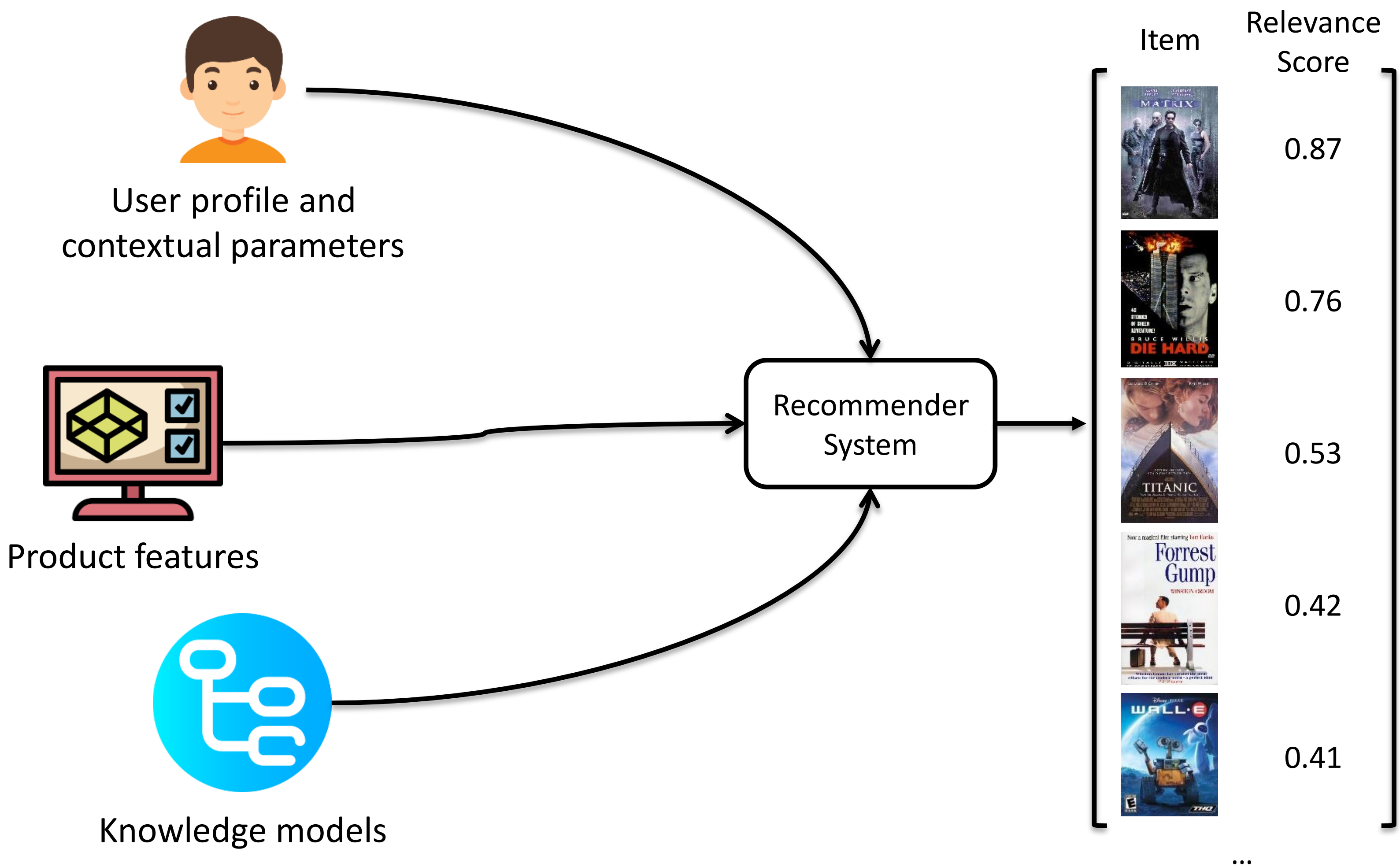

Collaborative: "Tell me what's popular among my peers”

# Paradigms of recommender systems



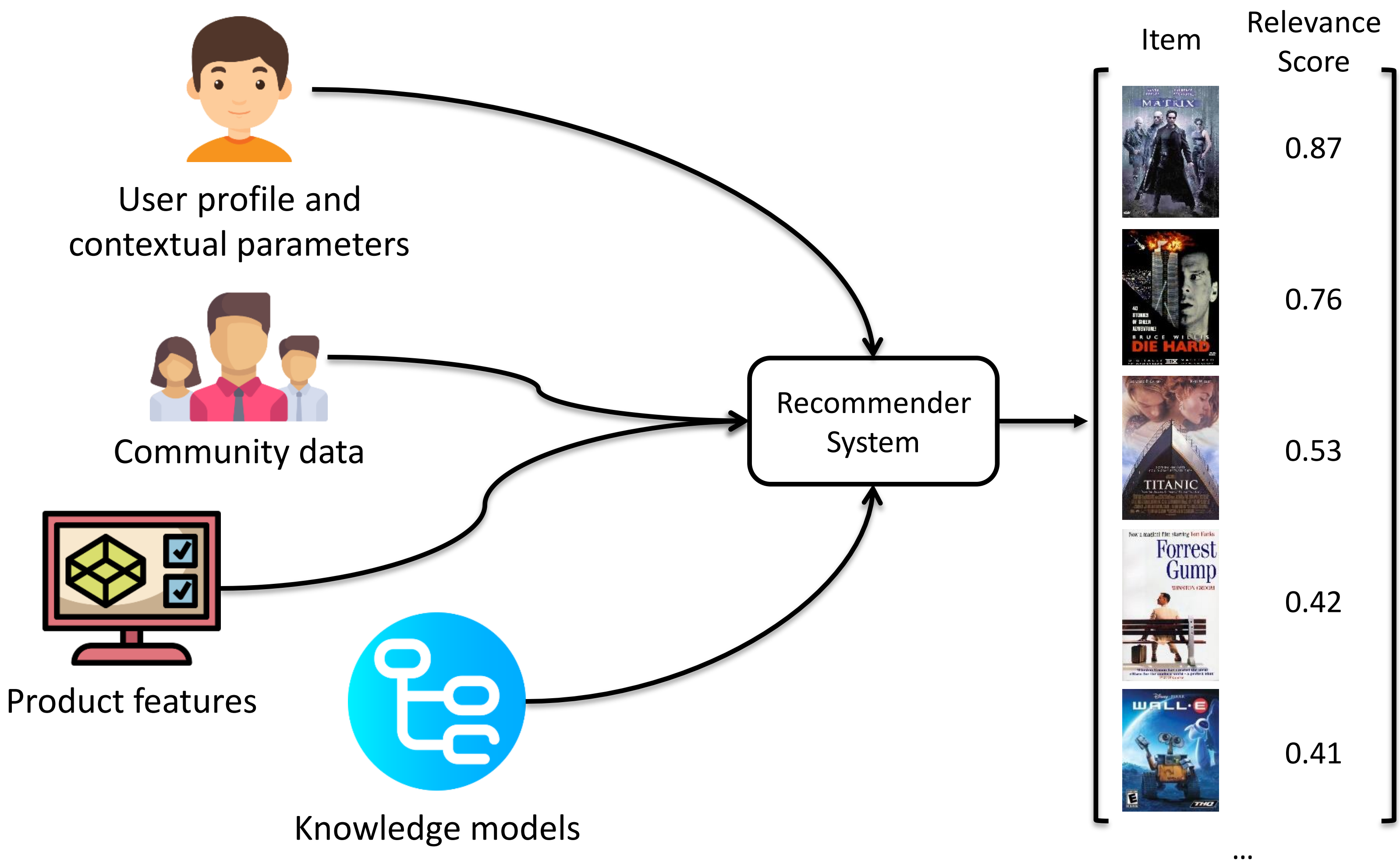

Content-based: "Show me more items like the ones I've liked"

# Paradigms of recommender systems




Knowledge-based: "Tell me what is best based on my needs"

# Paradigms of recommender systems




Hybrid: combinations of various inputs
and/or composition of different mechanism

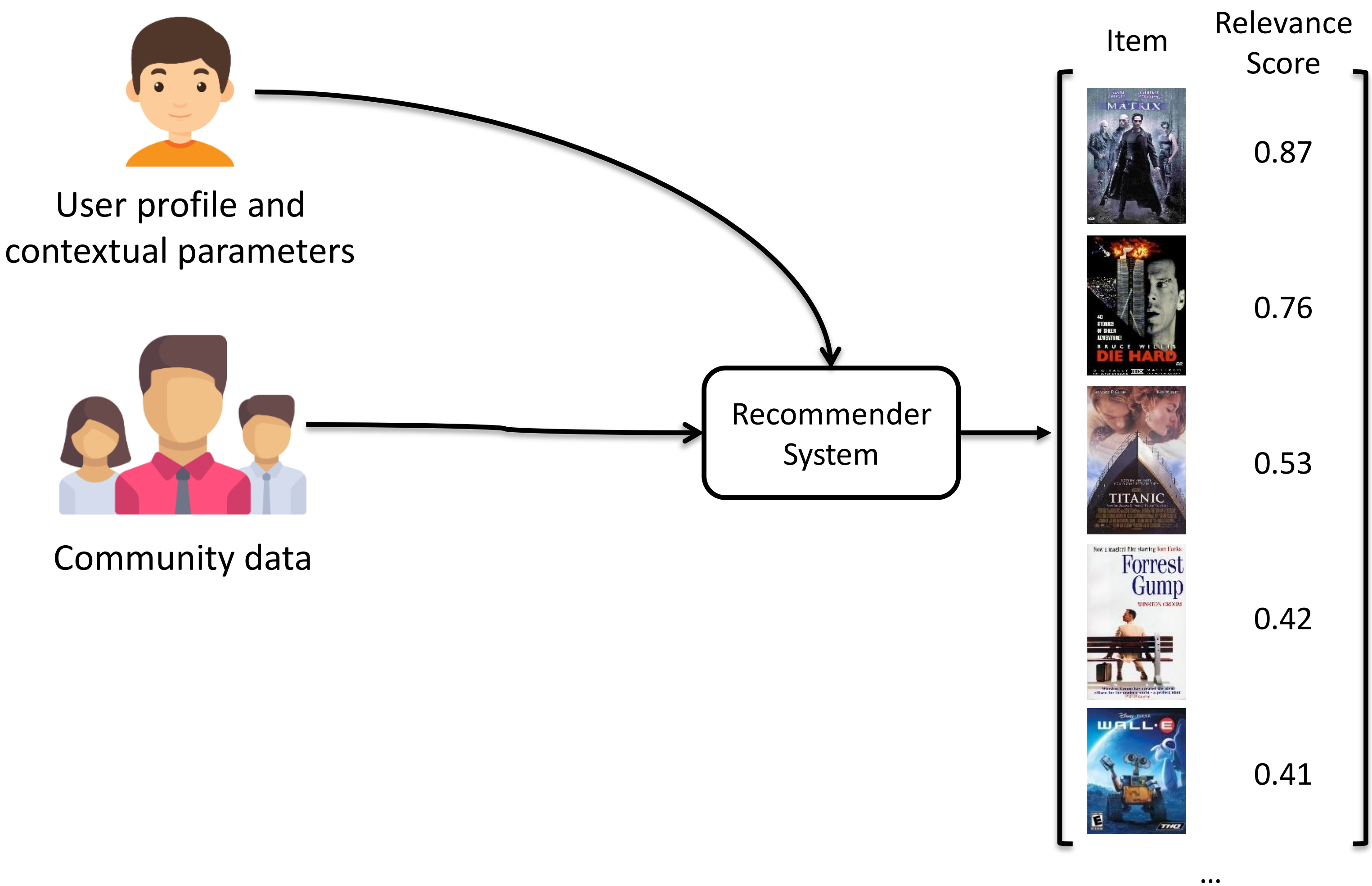
# Collaborative Filtering

**Collaborative Filtering** uses the "wisdom of the crowd" to recommend items

- Widely used by large, commercial e-commerce sites, social media, etc
- Well-understood, various algorithms and variations exist
- Applicable in many domains (book, movies, DVDs, ..)

**Basic assumption**: User preferences remain stable and consistent over time (i.e. customers who had similar tastes in the past, will have similar tastes in the future)

# Collaborative Filtering




The input can be summarised into a matrix of given user–item ratings

# Collaborative Filtering

Items

Users

Ratings

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 4 | | | | 2 | | | | | | | 3 | | | 2 |
| **2** | | | 1 | | | 4 | | | 3 | | | | 1 | | 2 |
| **3** | | | | | 3 | | 4 | | | | 2 | | | 1 | 1 |
| **4** | | 1 | | 5 | | 3 | | | | | 5 | | 3 | | |
| **5** | 3 | | | 5 | | | 1 | | | | | | 3 | | |
| **6** | | | | | 5 | | | 2 | | 1 | | | | | 5 |
| **7** | | | 1 | | | 3 | | 2 | | | | | | 4 | |
| **8** | | 4 | | | | | | | | | 5 | | 5 | | 1 |
| **9** | | | | 4 | | | | | 1 | | | | | 4 | |
| **10** | | | 3 | | | | 3 | | | 3 | | | 5 | | |
| **11** | | 3 | | | | | 3 | | | 1 | | | | | 5 |
| **12** | 5 | | 4 | 5 | 1 | | | | | | | 3 | | | |
| **13** | | | | | | | | | 5 | | 4 | | | | |
| **14** | | | 1 | 1 | | | | | | | | | | 2 | 3 |
| **15** | | 5 | | | | | | | | | 4 | | | | |

Recommender System

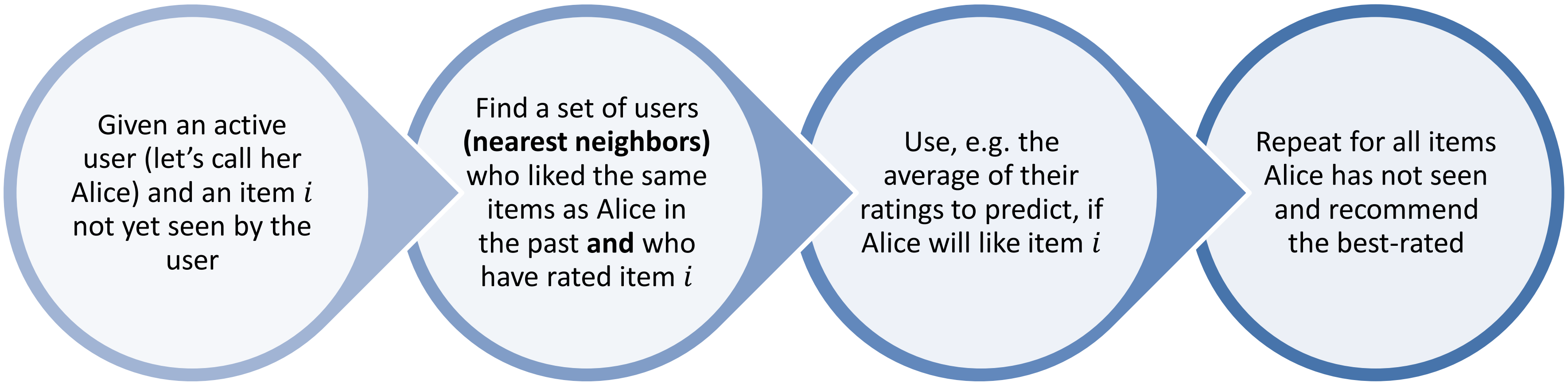
| Item | Relevance Score |
|---|---|
|  | 0.87 |
| | 0.76 |
| | 0.53 |
| | 0.42 |
| | 0.41 |



...

The input can be summarised into a matrix of given user–item ratings

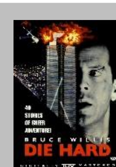
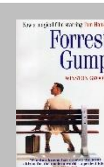
# User-based (user-to-user) collaborative filtering




**Basic assumption**: User preferences remain stable and consistent over time (i.e. customers who had similar tastes in the past, will have similar tastes in the future)

# User-based (user-to-user) collaborative filtering

Example: Determine whether the Active User will like or dislike “Titanic”, which she has not yet rated or seen

| | <br>**The Matrix** | <br>**Die Hard** | <br>**Forest Gump** | <br>**Wall-E** | <br>**Titanic** |
|---|---|---|---|---|---|
| **Active User** | 2 | 3 | 5 | 4 | ? |
| **User1** | 5 | | 2 | 2 | 1 |
| **User2** | 1 | 2 | 5 | 5 | 5 |
| **User3** | 4 | 5 | 3 | | 3 |
| **User4** | 1 | 4 | 1 | 4 | 1 |
| **User5** | 1 | 2 | 4 | 3 | 4 |
| **User6** | 4 | 3 | 1 | 2 | 1 |
| **User7** | 1 | 1.5 | 2.5 | 2 | 3 |
| **User8** | 2 | 3 | 4 | 1 | |

- How do we measure similarity?
- How many neighbours should we consider?
- How do we generate a prediction from the neighbours' ratings?

# Measuring user similarity

**Pearson correlation** is a popular similarity measure in user-based collaborative filtering. It returns similarity values in the range of $[-1, 1]$

Values centred to average rating given by each user

$$\boldsymbol{sim(a,b)} = \frac{\sum_{p \in P}(r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \bar{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \bar{r}_b)^2}}$$

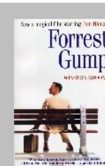
Values normalised by the variance of the ratings given by each user

$a$, $b$ : users
$r_{a,p}$ : rating of user $a$ for item $p$
$P$ : set of items, rated both by $a$ and $b$

# Measuring user similarity

| |  The Matrix |  Die Hard |  Forest Gump |  Wall-E |  Titanic | Pearson |
|---|---|---|---|---|---|---|
| Active User | 2 | 3 | 5 | 4 | ? | 1.00 |
| User1 | 5 | | 2 | 2 | 1 | -0.94 |
| User2 | 1 | 2 | 5 | 5 | 5 | 0.94 |
| User3 | 4 | 5 | 3 | | 3 | -0.65 |
| User4 | 1 | 4 | 1 | 4 | 1 | 0.00 |
| User5 | 1 | 2 | 4 | 3 | 4 | 1.00 |
| User6 | 4 | 3 | 1 | 2 | 1 | -1.00 |
| User7 | 1 | 1.5 | 2.5 | 2 | 3 | 1.00 |
| User8 | 2 | 3 | 4 | 1 | | 0.40 |

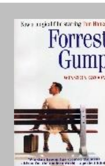
# Measuring user similarity

| |  The Matrix |  Die Hard |  Forest Gump |  Wall-E |  Titanic | Pearson |
|---|---|---|---|---|---|---|
| **Active User** | 2 | 3 | 5 | 4 | ? | 1.00 |
| **User1** | 5 | | 2 | 2 | 1 | -0.94 |
| **User2** | 1 | 2 | 5 | 5 | 5 | 0.94 |
| **User3** | 4 | 5 | 3 | | 3 | -0.65 |
| **User4** | 1 | 4 | 1 | 4 | 1 | 0.00 |
| **User5** | 1 | 2 | 4 | 3 | 4 | 1.00 |
| **User6** | 4 | 3 | 1 | 2 | 1 | -1.00 |
| **User7** | 1 | 1.5 | 2.5 | 2 | 3 | 1.00 |
| **User8** | 2 | 3 | 4 | 1 | | 0.40 |

Similarity is calculated taking into account ONLY the items that BOTH users have ranked

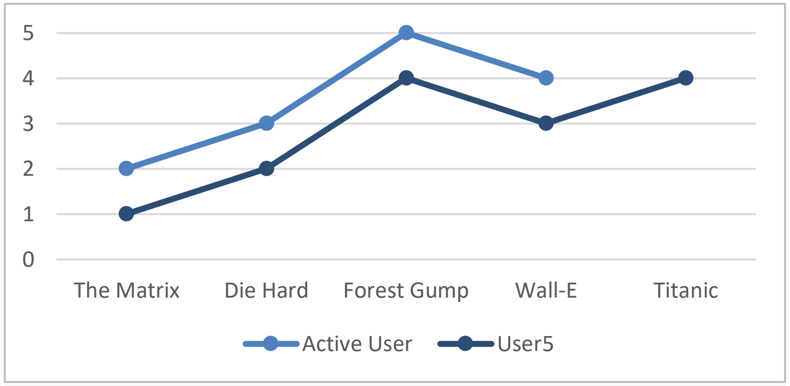
# Measuring user similarity

An alternative distance used frequently is cosine similarity (the angle between the vectors). The ranking remains the same (if number of items is consistent), although Pearson is more intuitive.

| |  The Matrix |  Die Hard |  Forest Gump |  Wall-E |  Titanic | Pearson | Cosine |
|---|---|---|---|---|---|---|---|
| **Active User** | 2 | 3 | 5 | 4 | ? | 1.00 | 1.00 |
| **User1** | 5 | | 2 | 2 | 1 | -0.94 | 0.73 |
| **User2** | 1 | 2 | 5 | 5 | 5 | 0.94 | 0.97 |
| **User3** | 4 | 5 | 3 | | 3 | -0.65 | 0.87 |
| **User4** | 1 | 4 | 1 | 4 | 1 | 0.00 | 0.82 |
| **User5** | 1 | 2 | 4 | 3 | 4 | 1.00 | 0.99 |
| **User6** | 4 | 3 | 1 | 2 | 1 | -1.00 | 0.75 |
| **User7** | 1 | 1.5 | 2.5 | 2 | 3 | 1.00 | 1.00 |
| **User8** | 2 | 3 | 4 | 1 | | 0.40 | 0.92 |

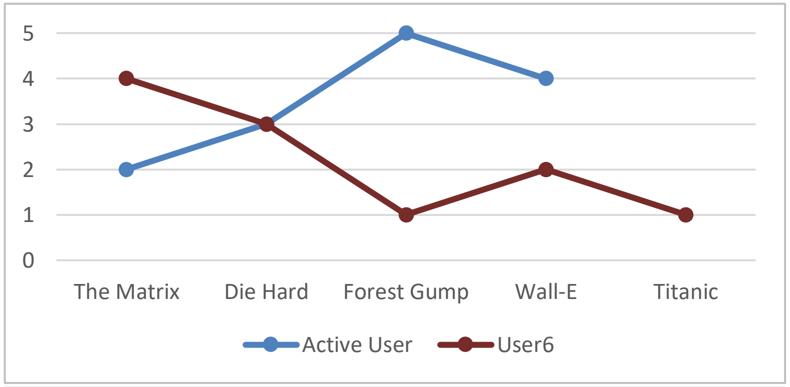
$$\boldsymbol{sim}(\vec{a},\vec{b}) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}| * |\vec{b}|}$$

# Pearson vs Cosine



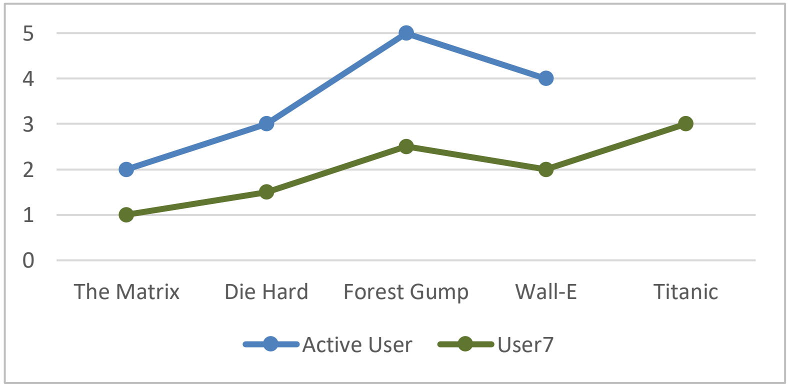

User 5 ratings: *Active user -1*

**Pearson Similarity** = 1.00
**Cosine Similarity** = 0.99




User 6 ratings: *Inverse to active user*

**Pearson Similarity** = -1.00
**Cosine Similarity** = 0.75




User 7 ratings: *0.5 * active user*

**Pearson Similarity** = 1.00
**Cosine Similarity** = 1.00

# Making predictions

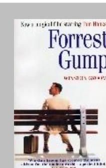
A common prediction function is using the similarities to calculate a weighted average of (centred) rankings

$$\boldsymbol{pred(a,i)} = \overline{\boldsymbol{r_a}} + \frac{\sum_{\boldsymbol{b} \in \boldsymbol{N}} \boldsymbol{sim(a,b)} * (\boldsymbol{r_{b,i}} - \overline{\boldsymbol{r_b}})}{\sum_{\boldsymbol{b} \in \boldsymbol{N}} \boldsymbol{sim(a,b)}}$$

For a user $\alpha$ and an item $i$:

- Calculate whether the other users' ($b \in N$) ratings for the unseen item $i$ are higher or lower than their respective average
- Combine the rating differences using the similarity of each user with user $a$ as a weight
- Add the result to the active user's average and use this as a prediction

# Making predictions

| |  The Matrix |  Die Hard |  Forest Gump |  Wall-E |  Titanic | Pearson |
|---|---|---|---|---|---|---|
| **Active User** | 2 | 3 | 5 | 4 | ? | 1.00 |
| **User1** | 5 | | 2 | 2 | 1 | -0.94 |
| **User2** | 1 | 2 | 5 | 5 | 5 | 0.94 |
| **User3** | 4 | 5 | 3 | | 3 | -0.65 |
| **User4** | 1 | 4 | 1 | 4 | 1 | 0.00 |
| **User5** | 1 | 2 | 4 | 3 | 4 | 1.00 |
| **User6** | 4 | 3 | 1 | 2 | 1 | -1.00 |
| **User7** | 1 | 1.5 | 2.5 | 2 | 3 | 1.00 |
| **User8** | 2 | 3 | 4 | 1 | | 0.40 |

User 8 is irrelevant for the prediction, as she has not ranked the item we are interested in. User 8 is therefore excluded from the process.

# Improving the prediction function

- Not all ratings might be equally “valuable” for determining similarity between users
  - I.e. Agreement on commonly liked items is not so informative as agreement on controversial items
  - **Possible solution**:  Give more weight to items that have a higher variance in rankings
- The number of co-rated items (how many items both users have rated) should tell you something about how confident we are that two users are similar or not
  - **Possible solution**: Use some kind of “significance weighting”, by e.g., linearly reducing the weight when the number of co-rated items is low
- Why use all the users
  - **Possible solution**: use only the nearest neighbours. Use similarity threshold or fixed number of neighbors.

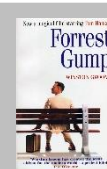
# Item based collaborative filtering

Rather than matching user-to-user similarity, **item-based (item-to-item) collaborative filtering** matches items purchased or rated by a target user to **similar items** and combines those similar items in a recommendation list.


Similarity can be computed in a number of ways:

- Using product descriptions / characteristics
- Using co-occurrence of the items in the user bags of past purchases
- **Using the user ratings**

# Measuring item similarity

| |  The Matrix |  Die Hard |  Forest Gump |  Wall-E |  Titanic |
|---|---|---|---|---|---|
| **Active User** | 2 | 3 | 5 | 4 | ? |
| **User1** | 5 | | 2 | 2 | 1 |
| **User2** | 1 | 2 | 5 | 5 | 5 |
| **User3** | 4 | 5 | 3 | | 3 |
| **User4** | 1 | 4 | 1 | 4 | 1 |
| **User5** | 1 | 2 | 4 | 3 | 4 |
| **User6** | 4 | 3 | 1 | 2 | 1 |
| **User7** | 1 | 1.5 | 2.5 | 2 | 3 |
| **User8** | 2 | 3 | 4 | 1 | |
| **Cosine** | 0.55 | 0.73 | 0.99 | 0.82 | 1.00 |

# Similarity and prediction

The “adjusted cosine distance” is typically used for item-based collaborative filtering

Compare to cosine distance

$$sim_{cosine}(\vec{a},\vec{b}) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}| * |\vec{b}|}$$

$$sim_{AdjustedCosine}(\vec{a},\vec{b}) = \frac{\sum_{u\in U}(r_{u,a}-\overline{r_u})(r_{u,b}-\overline{r_u})}{\sqrt{\sum_{u\in U}(r_{u,a}-\overline{r_u})^2}\sqrt{\sum_{u\in U}(r_{u,b}-\overline{r_u})^2}}$$

*<u>Note:</u>*

The adjusted cosine similarity takes **mean-centered user ratings** into account, and the formula is exactly the same as Pearson…

The difference is that in this case it is **applied to ALL users**, but considering a rating equal to the mean whenever user $u$ has not rated item $i$. Therefore in these cases: $(r_{u,i} - \overline{r_u}) = 0$

This has the effect of dropping such items that one user has not rated from the nominator, but counting them in the denominator. This produces a **self-damping effect** – the more users that have rated the two items the better – without a need to explicitly introduce this as a “*significance weighting*”

# Scalability

- Item-based filtering itself does not solve the scalability problem
- Pre-processing possible: calculate all pair-wise item similarities in advance
  - Memory requirements: Up to $N^2$ pair-wise similarities to be memorized (N = number of items) in theory
  - In practice, this is significantly lower (items with no co-ratings)
- Incremental (similarities calculated for every new item / every time we have n more reviews)
- Typically small neighborhood is used at run-time
- Only items which the user has rated / seen / purchased are taken into account

# Scalability - example

Items

Users

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 4 | | | | 2 | | | | | | | 3 | | | 2 | | | 3 | | | 5 | |
| **2** | | | 1 | | | 4 | | | 3 | | | | 1 | | 2 | | 5 | | | 1 | | 5 |
| **3** | | | | | 3 | | 4 | | | | 2 | | | 1 | 1 | | | 5 | | | | |
| **4** | | 1 | | 5 | | 3 | | | | | 5 | | 3 | | | | | | | | | |
| **5** | 3 | | | 5 | | | 1 | | | | | | 3 | | | | 1 | | | 5 | 5 | |
| **6** | | | | | 5 | | | 2 | | 1 | | | | | 5 | | | | | | | |
| **7** | | | 1 | | | 3 | | 2 | | | | | | 4 | | 4 | | | | | | |
| **8** | | 4 | | | | | | | | | 5 | | 5 | | 1 | | | | 1 | | | |
| **9** | | | | 4 | | | | | 1 | | | | | 4 | | | | | | | | 5 |
| **10** | | | 3 | | | | 3 | | | 3 | | | 5 | | | | | 1 | | 1 | 2 | |
| **11** | | 3 | | | | | 3 | | | 1 | | | | | 5 | | | | | | | |
| **12** | 5 | | 4 | 5 | 1 | | | | | | | 3 | | | | | | | | | 5 | |
| **13** | | | | | | | | | 5 | | 4 | | | | | 4 | | | | 2 | | 4 |
| **14** | | | 1 | 1 | | | | | | | | | | 2 | 3 | | 3 | | | | 4 | |
| **15** | | 5 | | | | | | | | | 4 | | | | | | | 1 | | | | 3 |

To compute the similarity of item $i_1$ to the others, first notice that only users $u_1$, $u_5$ and $u_{12}$ have ranked / seen / purchased item $i_1$.

Therefore we can only calculate the similarity of item $i_1$ to the union of the items ranked by these users: items[3, 4, 5, 7, 12, 13, 15, 17, 18, 20, 21]

# User-based vs Item-based collaborative filtering

- Item similarities are supposed to be **more stable** than user similarities
- Item-based CF provides better predictions than user-based **when there are more users than items** (which is the case in the most popular scenarios)
- Item **neighbourhood is fairly static**, hence enables pre-computation (which in turn improves online performance)

# Planning

| JUEVES | Practicas (9:30 - 11:30) | Teoria (11:30 - 13:30) | Problemas (13:30 - 14:30) |
|---|---|---|---|
| 18 / 02 | | Introducción, Datos y casos de uso, Conceptos basicos de estadistica | Introducción |
| 25 / 02 | | Conceptos basicos de estadistica, Algebra lineal | Manipulación de datos con Python |
| 04 / 03 | [612] Introducción (1) | Intro Pattern Recognition and Regresión lineal | Regresión lineal |
| 11 / 03 | [611] Introducción (1) | Regresión multiple, regresion polinomial, normalización | Normalizacion, regresión multiple, regresion polynomial |
| 18 / 03 | [612] Introducción (2) | Regresión logística | Regresión logística |
| 25 / 03 | [611] Introducción (2) | Regularización, descomposición "bias-variance" | Regularización |
| 01 / 04 | *Semana Santa* | *Semana Santa* | *Semana Santa* |
| 08 / 04 | [611/612] Proyecto 1 | Reducción de dimensionalidad (PCA) | PCA |
| 15 / 04 | | EXAMEN PARCIAL | |
| 22 / 04 | | Probabilidades and Bayesian inference | Probabilities |
| 29 / 04 | *MEM Enginy* | *MEM Enginy* | *MEM Enginy* |
| 06 / 05 | [612] Presentaciónes Proyecto 1 | Algoritmo de "Nearest Neighbours" | Nearest Neighbours |
| 13 / 05 | [611] Presentaciónes Proyecto 1 | Busqueda de datos, precisión / recall, Sistemas de recomendación | Busqueda de datos, Sistemas de recomendación |
| 20 / 05 | [611/612] Proyecto 2 | Agrupación (clustering), algoritmo K-means | K-means |
| 27 / 05 | | Revisión | |
| 03 / 06 | [611] Presentaciones Proyecto 2 | [612] Presentaciones Proyecto 2 | |

| Fecha | Hora | |
|---|---|---|
| 18 / 06 | 12:00 - 14:30 | Segundo Parcial (Q3/1003) |
| 02 / 07 | 12:00 - 14:30 | Examen Recuperacion (Q2/1005) |

Examenes presenciales