Learning to Rank

Pradeep Ravikumar
pradeepr@cs.utexas.edu

CS 371R

Adapted from Hang Li, MSR Asia, 09

Ranking

Coke or Pepsi or Sprite)?

Ranking

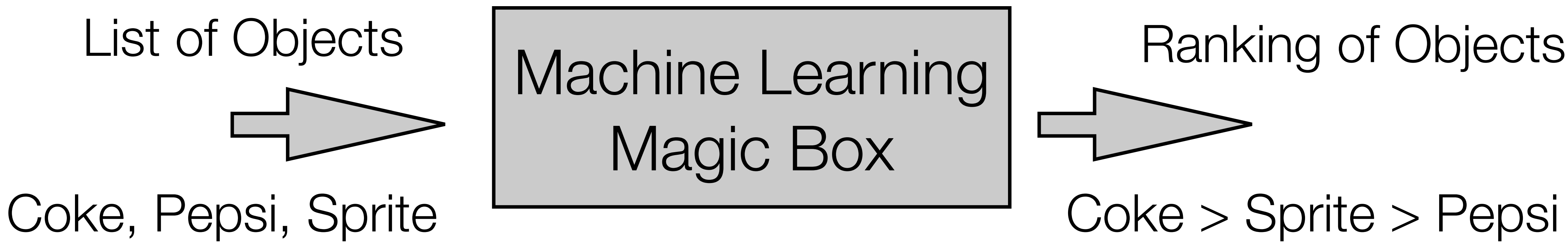
Ranking Chess Players

Ranking

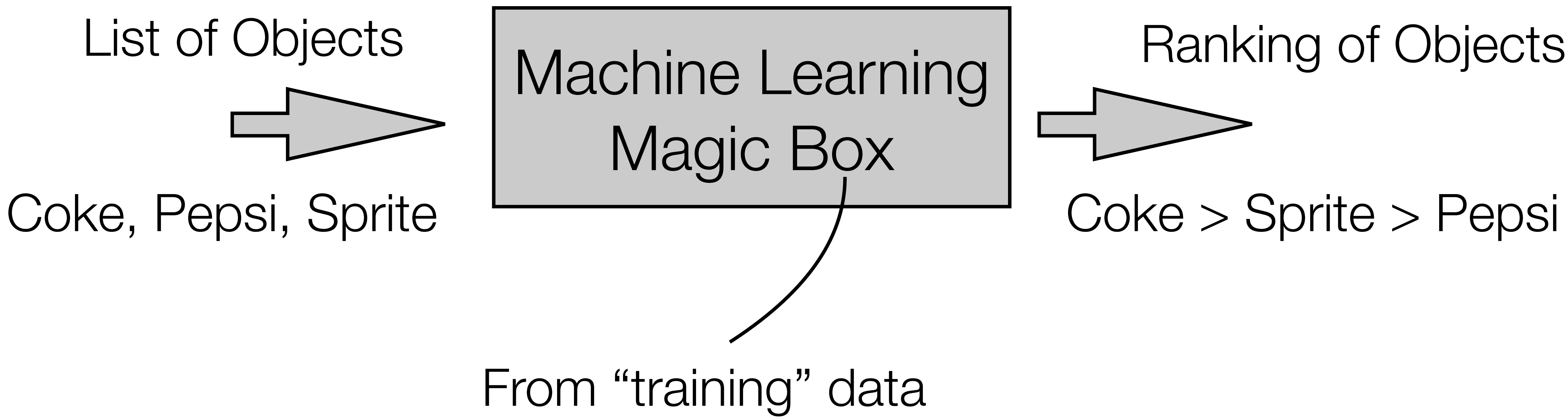
Ranking Chess Players

Golf Players, etc.

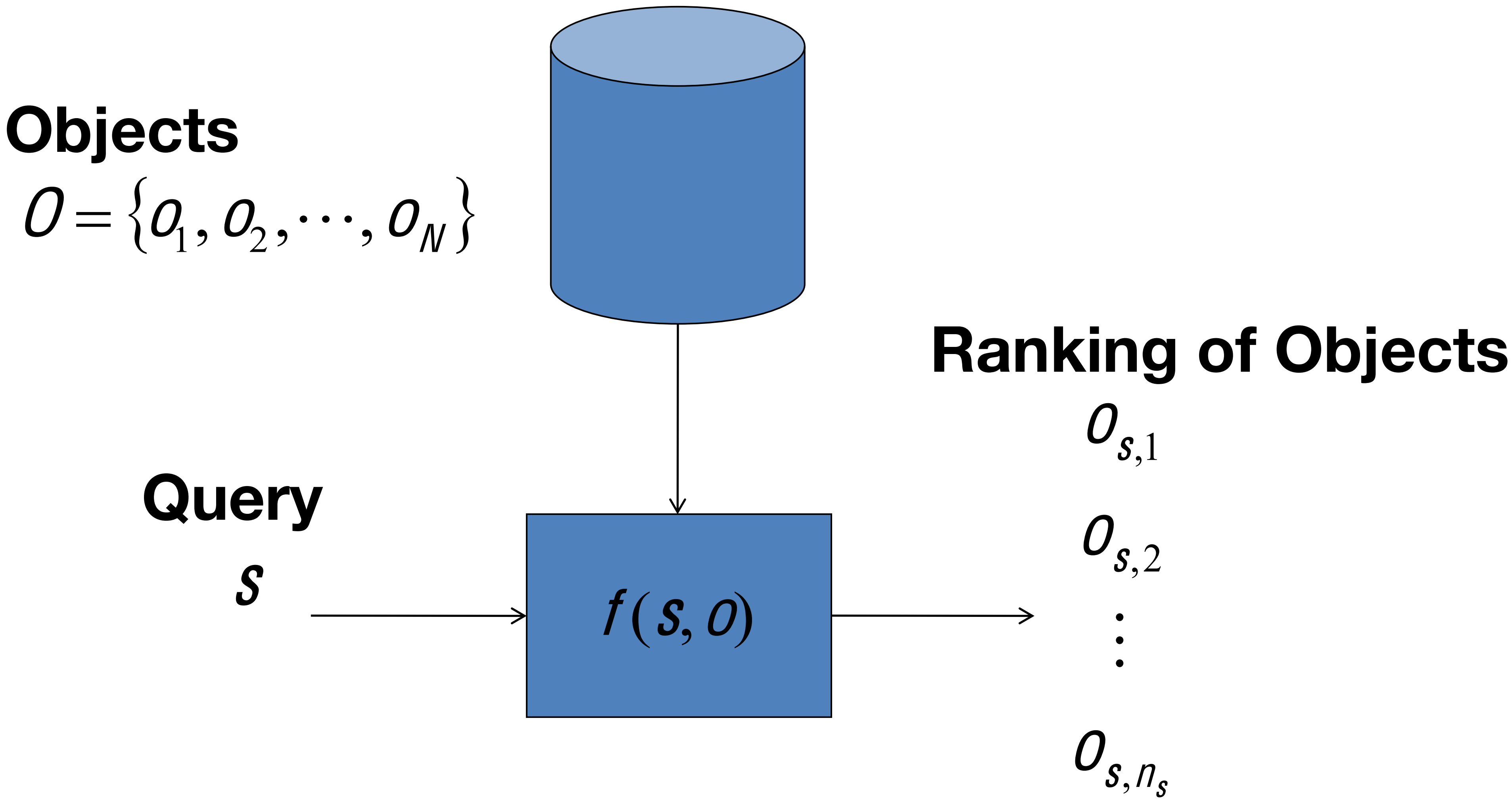
Whither Machine Learning?



Whither Machine Learning?

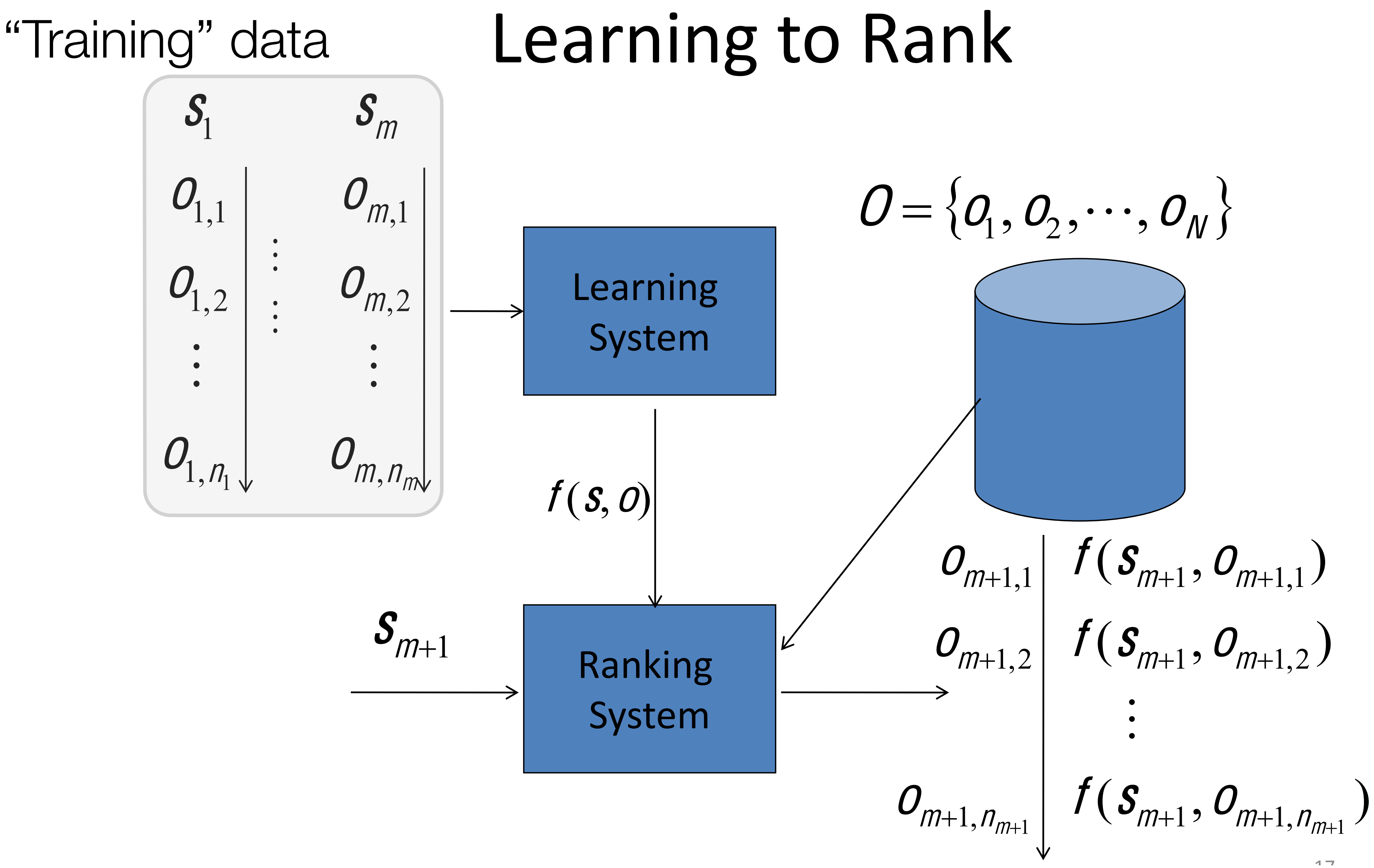


Ranking



Objects
$O = \{o_1, o_2, \cdots, o_N\}$
Ranking of Objects
$o_{s,1}$
$o_{s,2}$
$\vdots$
o_{s,n_s}
Query
s
$f(s, o)$

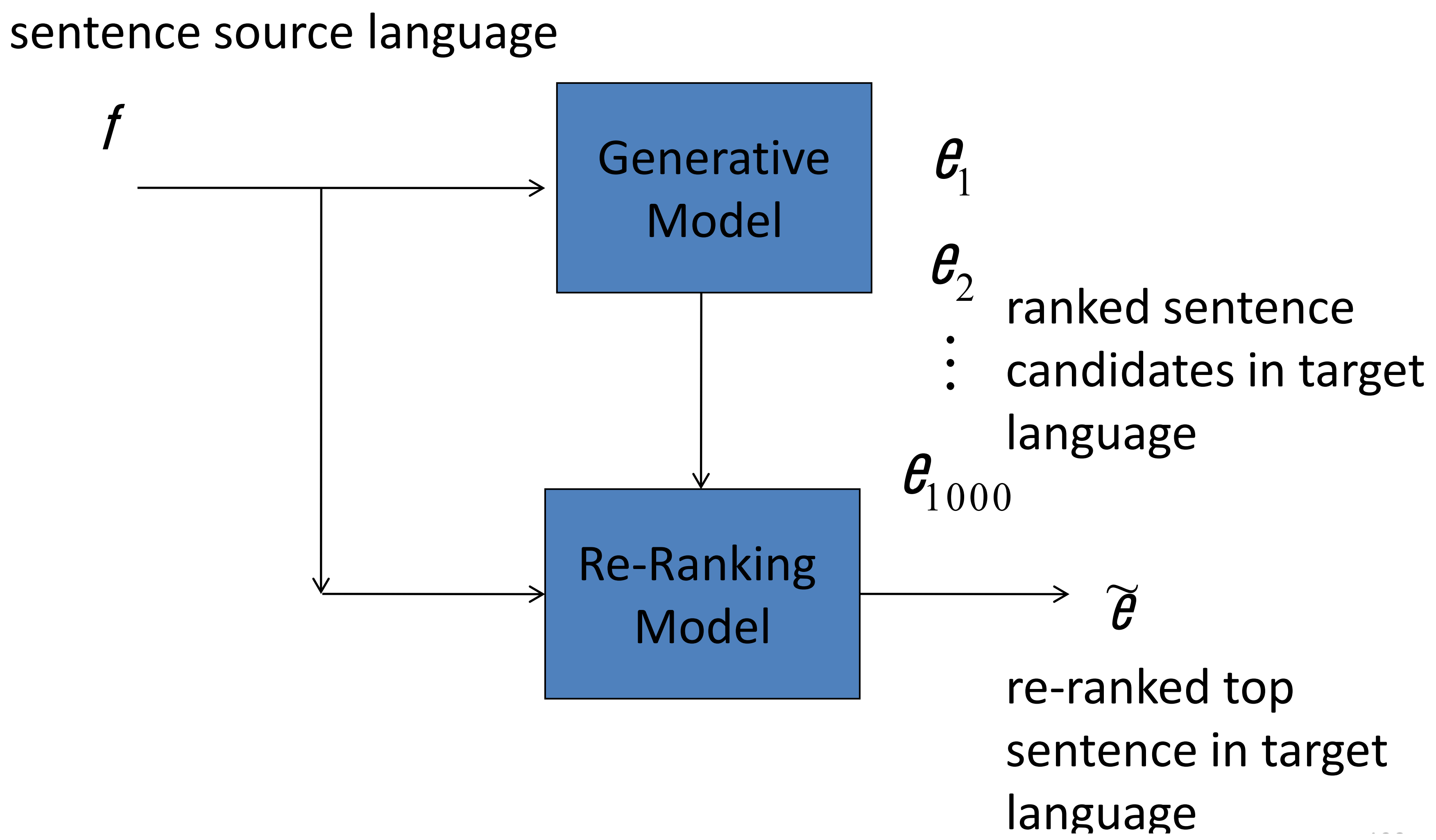
Ranking via Machine Learning



Is learning to rank necessary for IR?

- Has been successfully used in web-document search
- Has been the focus of increasing research at the intersection of machine learning + IR

Example: Machine Translation



Example: Collaborative Filtering

	Item1	Item2	Item3	...	
User1	5	4			
User2	1		2		2
...		?	?	?	
User M	4	3			

Example: Collaborative Filtering

	Item1	Item2	Item3	...	
User1	5	4			
User2	1		2		2
...		?	?	?	
UserM	4	3			

Input: User Ratings on *some* items

Output: User Ratings on *other* items

Example: Collaborative Filtering

	Item1	Item2	Item3	...	
User1	5	4			
User2	1		2		2
...		?	?	?	
UserM	4	3			

Any user is a query; based on this query, rank the set of items (based on user-item features)

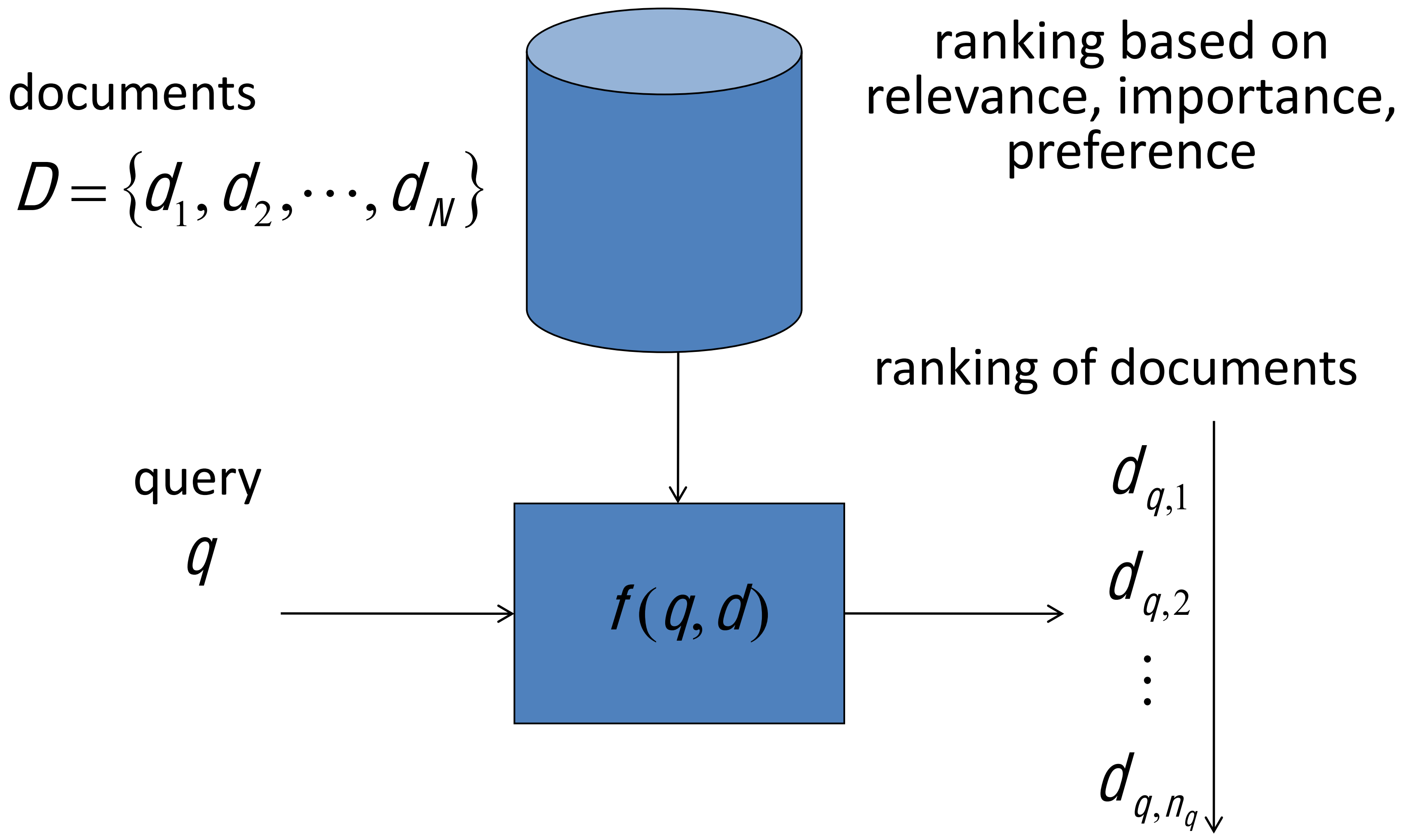
Example: Key Phrase Extraction

- Input: Document
- Output: Key phrases in document
- Pre-processing step: extraction of possible phrases in document
- Ranking approach: rank the extracted phrases

Other IR applications of learning to rank

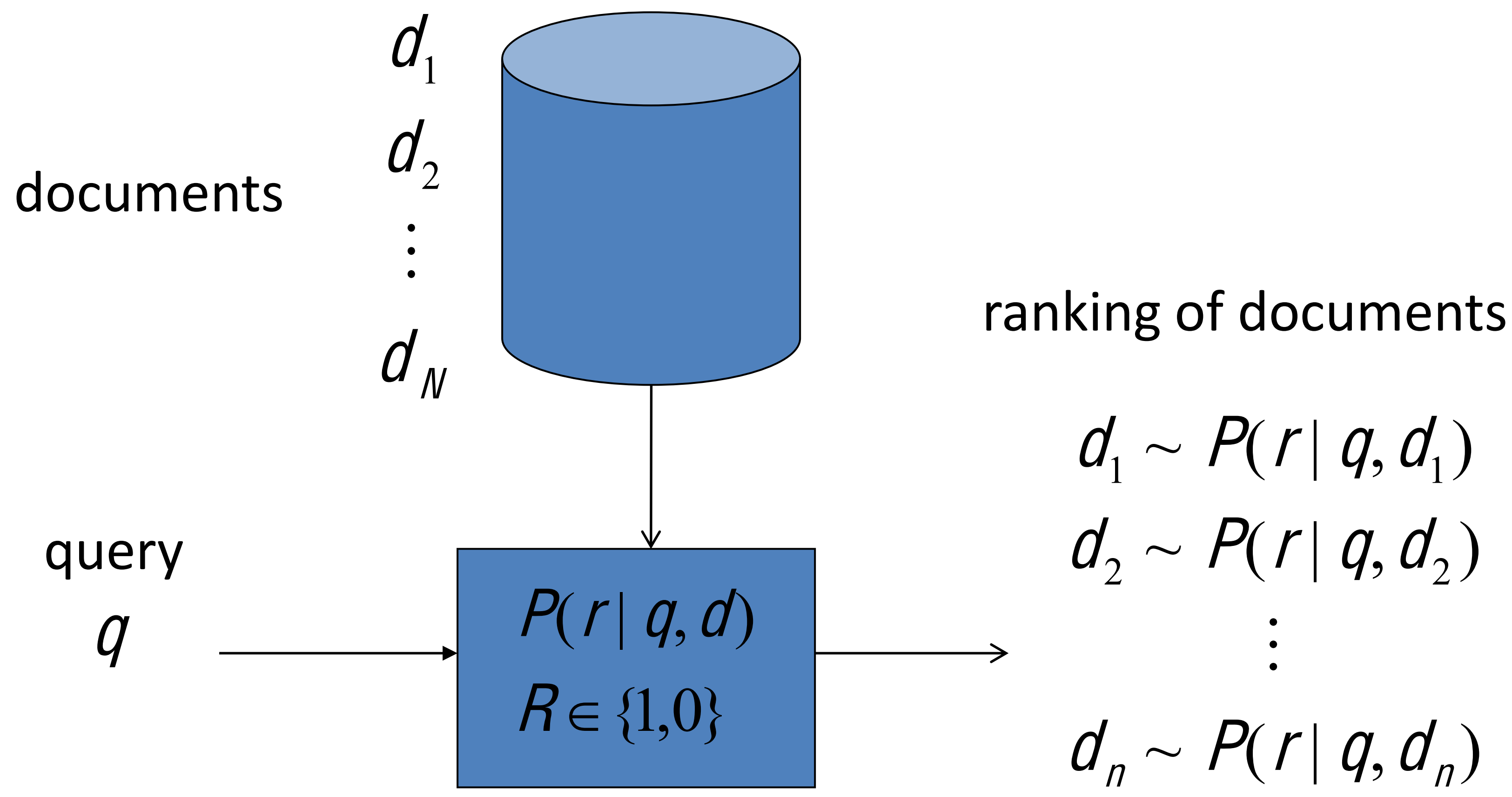
- Question Answering
- Document Summarization Opinion Mining
- Sentiment Analysis
- Machine Translation
- Sentence Parsing
- ….

Our focus: ranking documents

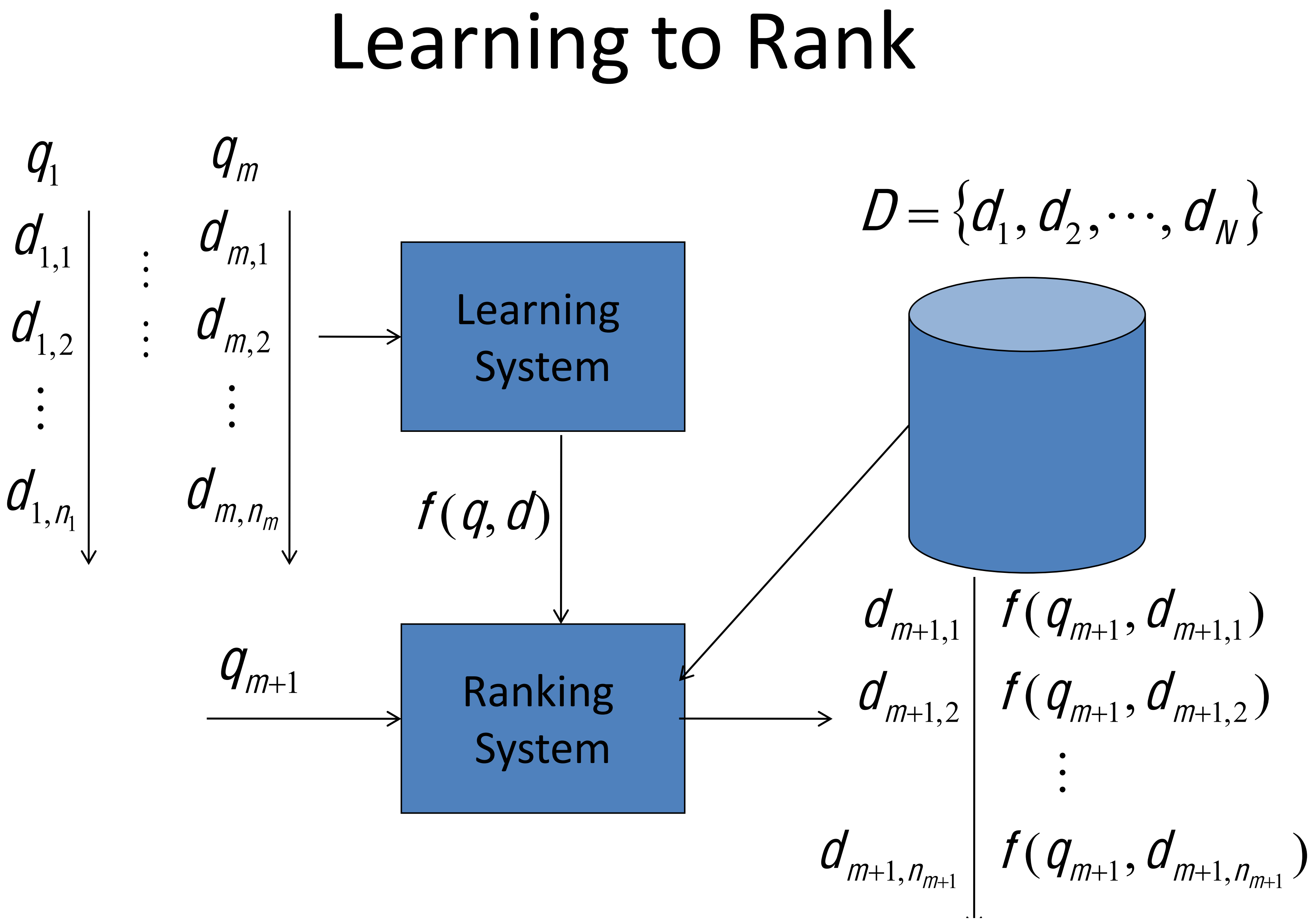


Traditional approach to “learning to rank”

Probabilistic Model



Modern approach to learning to rank



Learning to Rank: Training Process

$$q_1 \begin{cases} d_{1,1} \\ d_{1,2} \\ \vdots \\ d_{1,n_1} \end{cases}$$

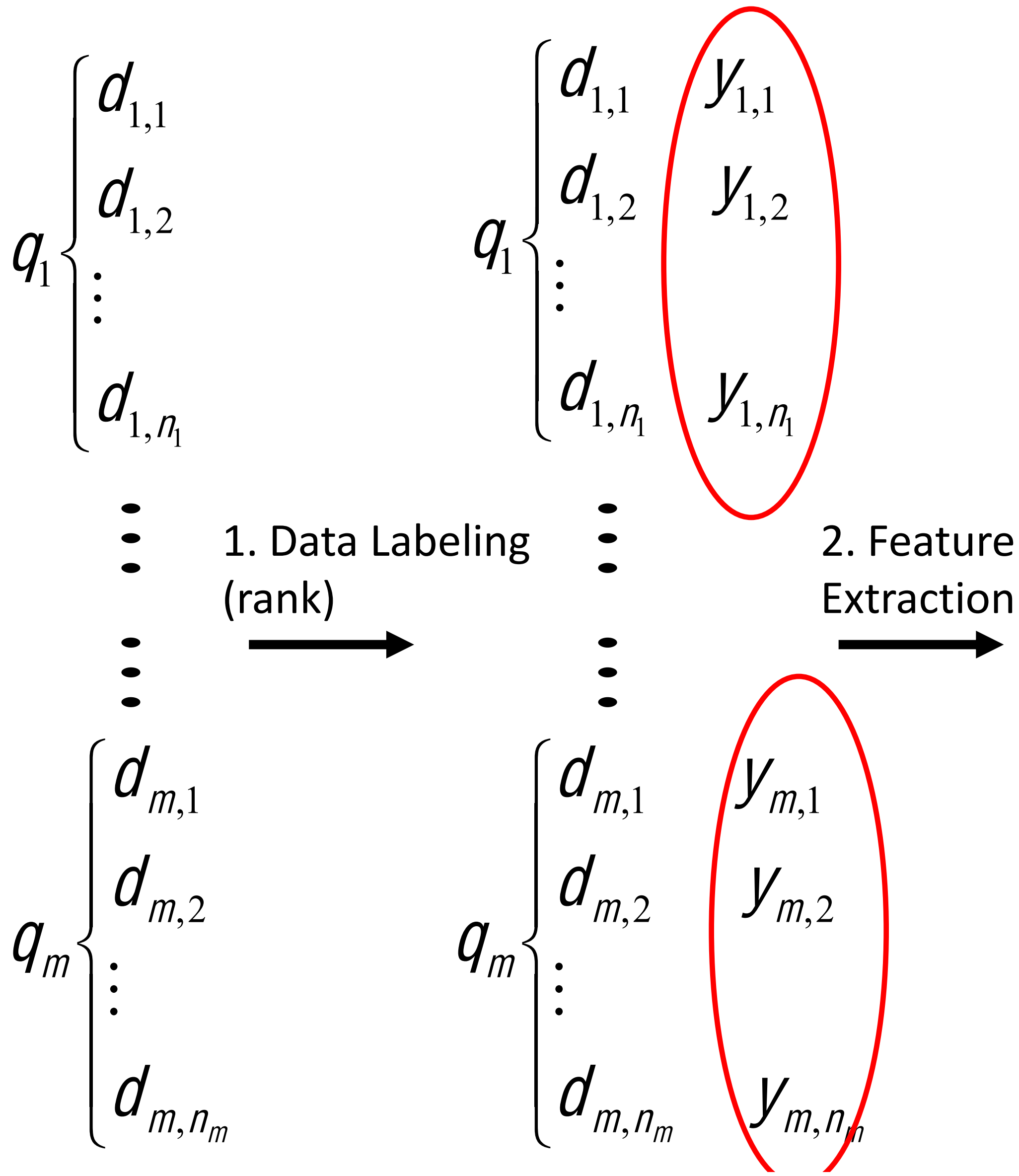
$$\vdots$$

$$\xrightarrow{\text{1. Data Labeling (rank)}}$$

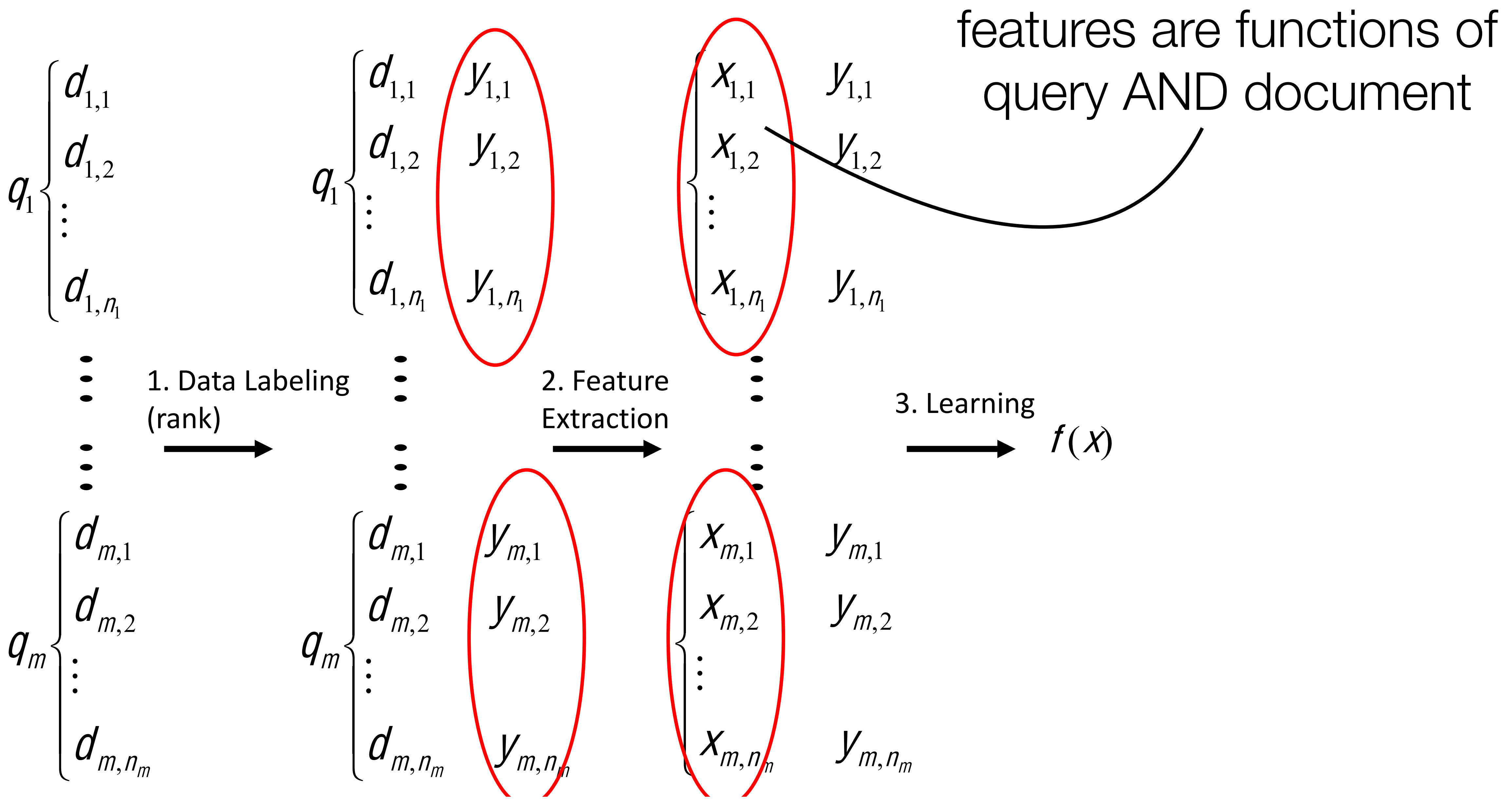
$$\vdots$$

$$q_m \begin{cases} d_{m,1} \\ d_{m,2} \\ \vdots \\ d_{m,n_m} \end{cases}$$

Learning to Rank: Training Process



Learning to Rank: Training Process



Learning to Rank: Testing Process

$$q_{m+1}\begin{cases} d_{m+1,1} \\ d_{m+1,2} \\ \vdots \\ d_{m+1,n_{m+1}} \end{cases}$$

1. Data Labeling
(rank)

$\longrightarrow$

Learning to Rank: Testing Process

Ground truth ranks

$$q_{m+1}\begin{cases} d_{m+1,1} \\ d_{m+1,2} \\ \vdots \\ d_{m+1,n_{m+1}} \end{cases}$$

1. Data Labeling (rank)

$\longrightarrow$

$$q_{m+1}\begin{cases} d_{m+1,1} & y_{m+1,1} \\ d_{m+1,2} & y_{m+1,2} \\ \vdots & \\ d_{m+1,n_{m+1}} & y_{m+1,n_{m+1}} \end{cases}$$

2. Feature Extraction

$\longrightarrow$

Learning to Rank: Testing Process

Ground truth ranks

$$q_{m+1}\begin{cases} d_{m+1,1} \\ d_{m+1,2} \\ \vdots \\ d_{m+1,n_{m+1}} \end{cases} \xrightarrow{\text{1. Data Labeling (rank)}} q_{m+1}\begin{cases} d_{m+1,1} & y_{m+1,1} \\ d_{m+1,2} & y_{m+1,2} \\ \vdots & \\ d_{m+1,n_{m+1}} & y_{m+1,n_{m+1}} \end{cases} \xrightarrow{\text{2. Feature Extraction}} \begin{cases} x_{m+1,1} & y_{m+1,1} \\ x_{m+1,2} & y_{m+1,2} \\ \vdots & \\ x_{m+1,n_{m+1}} & y_{m+1,n_{m+1}} \end{cases} \xrightarrow{\text{3. Ranking with } f(x)}$$

Learning to Rank: Testing Process

Ground truth ranks

$$q_{m+1}\begin{cases} d_{m+1,1} \\ d_{m+1,2} \\ \vdots \\ d_{m+1,n_{m+1}} \end{cases} \xrightarrow{\text{1. Data Labeling (rank)}} q_{m+1}\begin{cases} d_{m+1,1} & y_{m+1,1} \\ d_{m+1,2} & y_{m+1,2} \\ \vdots & \\ d_{m+1,n_{m+1}} & y_{m+1,n_{m+1}} \end{cases} \xrightarrow{\text{2. Feature Extraction}} \begin{cases} x_{m+1,1} & y_{m+1,1} \\ x_{m+1,2} & y_{m+1,2} \\ \vdots & \\ x_{m+1,n_{m+1}} & y_{m+1,n_{m+1}} \end{cases} \xrightarrow{\text{3. Ranking with } f(x)}$$

$$\longrightarrow \begin{cases} x_{m+1,1} & f(x_{m+1,1}) & y_{m+1,1} \\ x_{m+1,2} & f(x_{m+1,2}) & y_{m+1,2} \\ \vdots & & \\ x_{m+1,n_{m+1}} & f(x_{m+1,n_{m+1}}) & y_{m+1,n_{m+1}} \end{cases} \xrightarrow{\text{4. Evaluation}} \text{Evaluation Result}$$

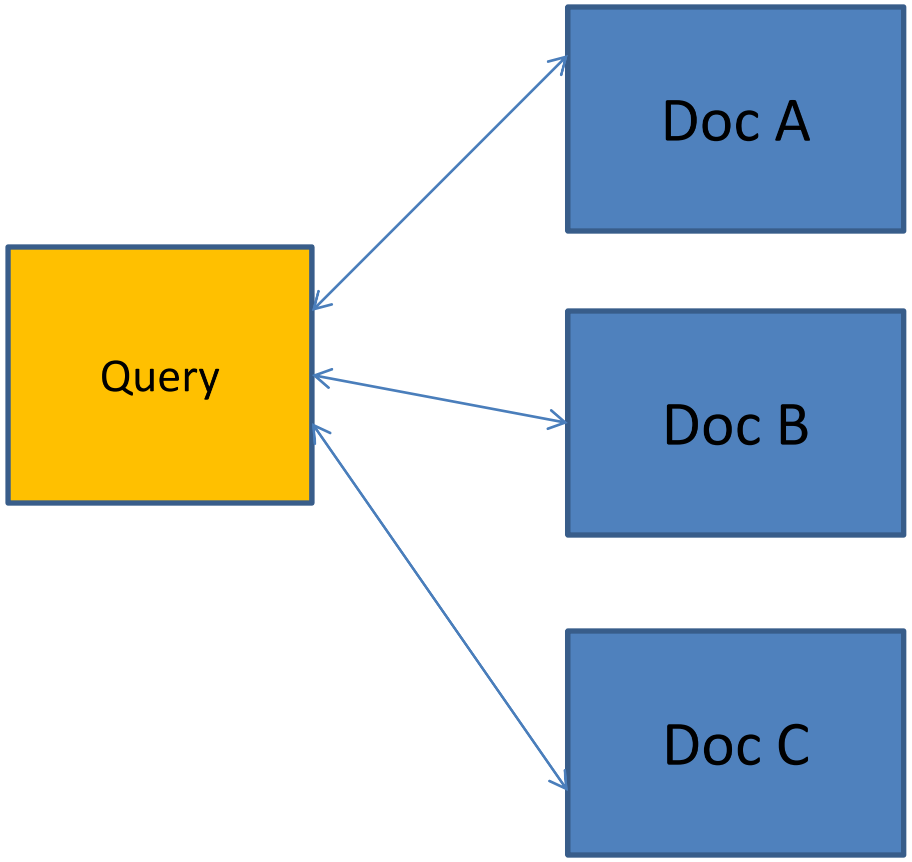
Issues in learning to rank

- Data Labeling
- Feature Extraction
- Evaluation Measure
- Learning Method (Model, Loss Function, Algorithm)

Data Labeling

- Ground truth/supervision: “ranks” of set of documents/objects

- E.g., relevance of documents w.r.t. query

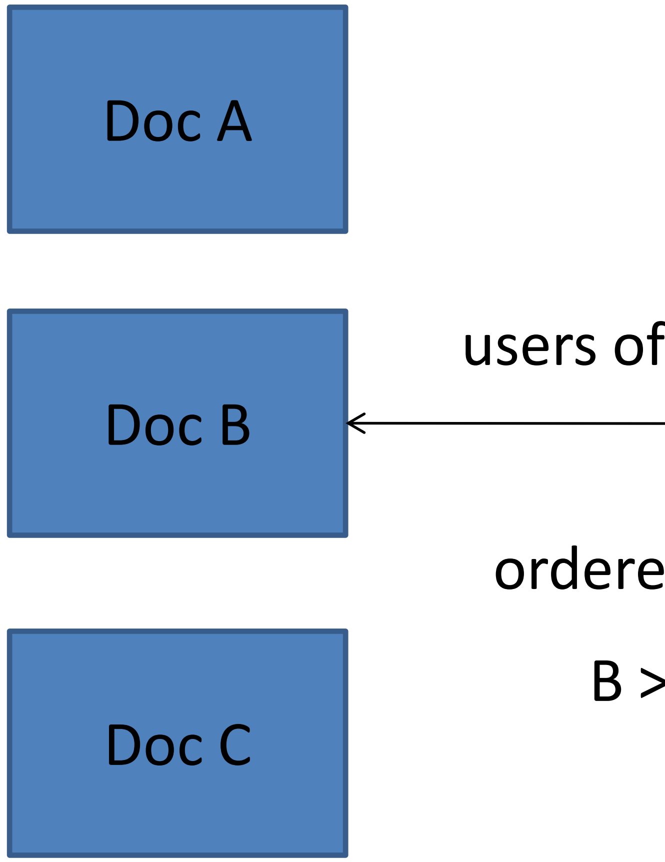


Data Labeling: Supervision takes different forms

- Multiple levels (e.g., relevant, partially relevant, irrelevant)
  - Widely used in IR
- Ordered Pairs
  - e.g. ordered pairs between documents (e.g. A>B, B>C)
  - e.g. implicit relevance judgments derived from click-through data
- Fully ordered list (i.e. a permutation) of documents is given
  - Ideal but difficult to implement

Data Labeling: Implicit Relevance Judgement

ranking of documents at search system

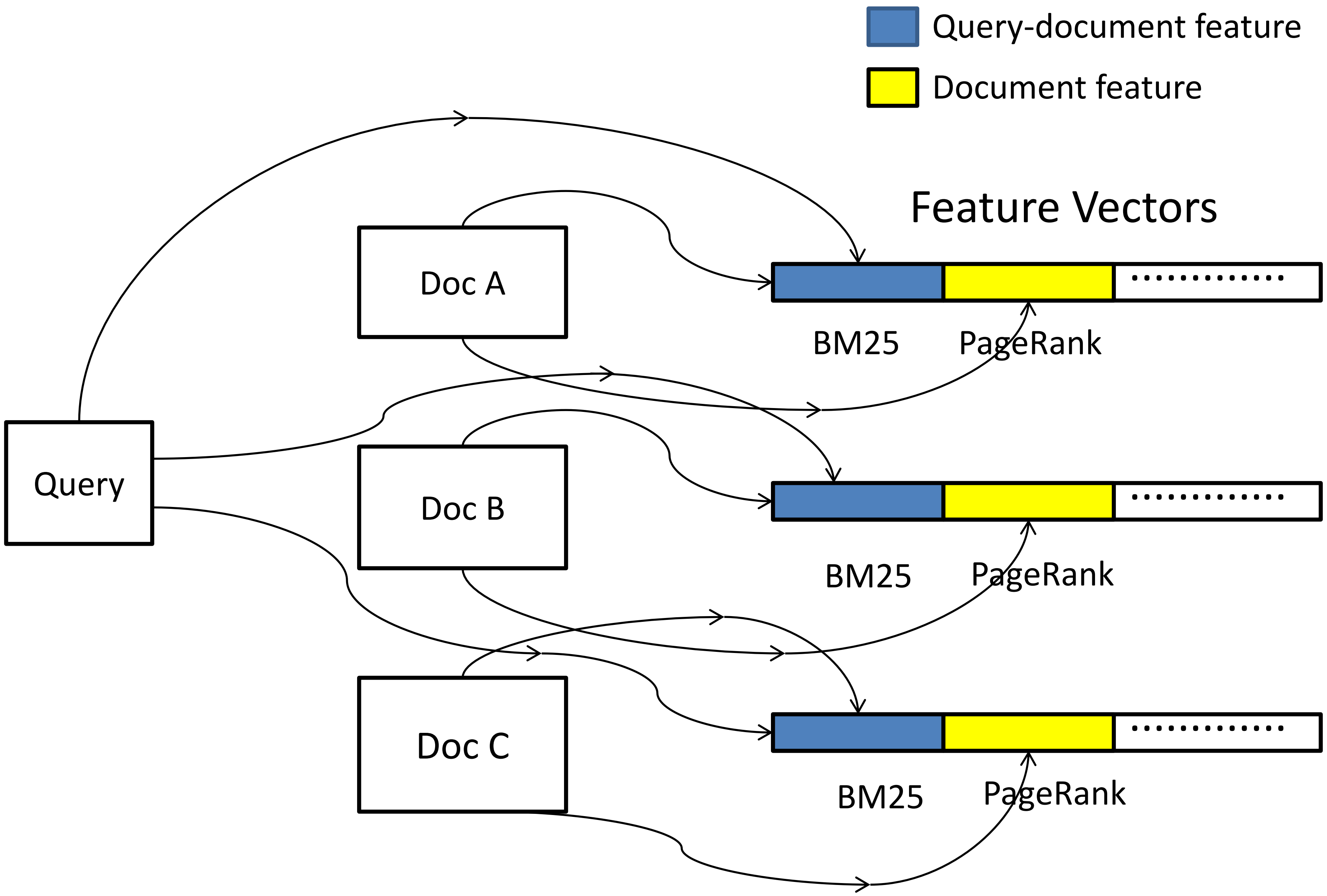


users often clicked on Doc B

ordered pair

B > A

Feature Extraction



Feature Extraction: Example Features

- BM25 (a classical ranking score function)
- Proximity/distance measures between query and document
- Whether query exactly occurs in document
- PageRank
- …

Evaluation Measures

- Allows us to quantify how good the predicted ranking is when compared with ground truth ranking
- Typical evaluation metrics care more about ranking **top results** correctly
- Popular Measures
  - NDCG (Normalized Discounted Cumulative Gain)
  - MAP (Mean Average Precision)
  - MRR (Mean Reciprocal Rank)
  - WTA (Winners Take All)
  - Kendall's Tau

Kendall's Tau

- Comparing agreement between two permutations

$$\tau = \frac{P - Q}{\frac{1}{2} n(n-1)}$$

- Number of pairs $\frac{1}{2} n(n-1)$
- Number of concordant pairs and disconcordant pairs P, Q
- Completely agree $\tau = +1$
- Completely disagree $\tau = -1$

Kendall's Tau

- Example

  A B C

  C A B

  $$\tau = \frac{1 - 2}{3} = -\frac{1}{3}$$

Learning to Rank Methods

- Three major approaches
  - Pointwise approach
  - Pairwise approach
  - Listwise approach

Pointwise Approach

- Transforming ranking to regression, classification, or ordinal regression
- Query-document group structure is ignored

Pointwise Approach

	Learning		
	Regression	**Classification**	**Ordinal Regression**
Input	Feature vector x		
Output	Real number $y = f(x)$	Category $y = \text{sign}(f(x))$	Ordered category $y = \text{thresold}(f(x))$
Model	Ranking function $f(x)$		
Loss Function	Regression loss	Classification loss	Ordinal regression loss

Pairwise Approach

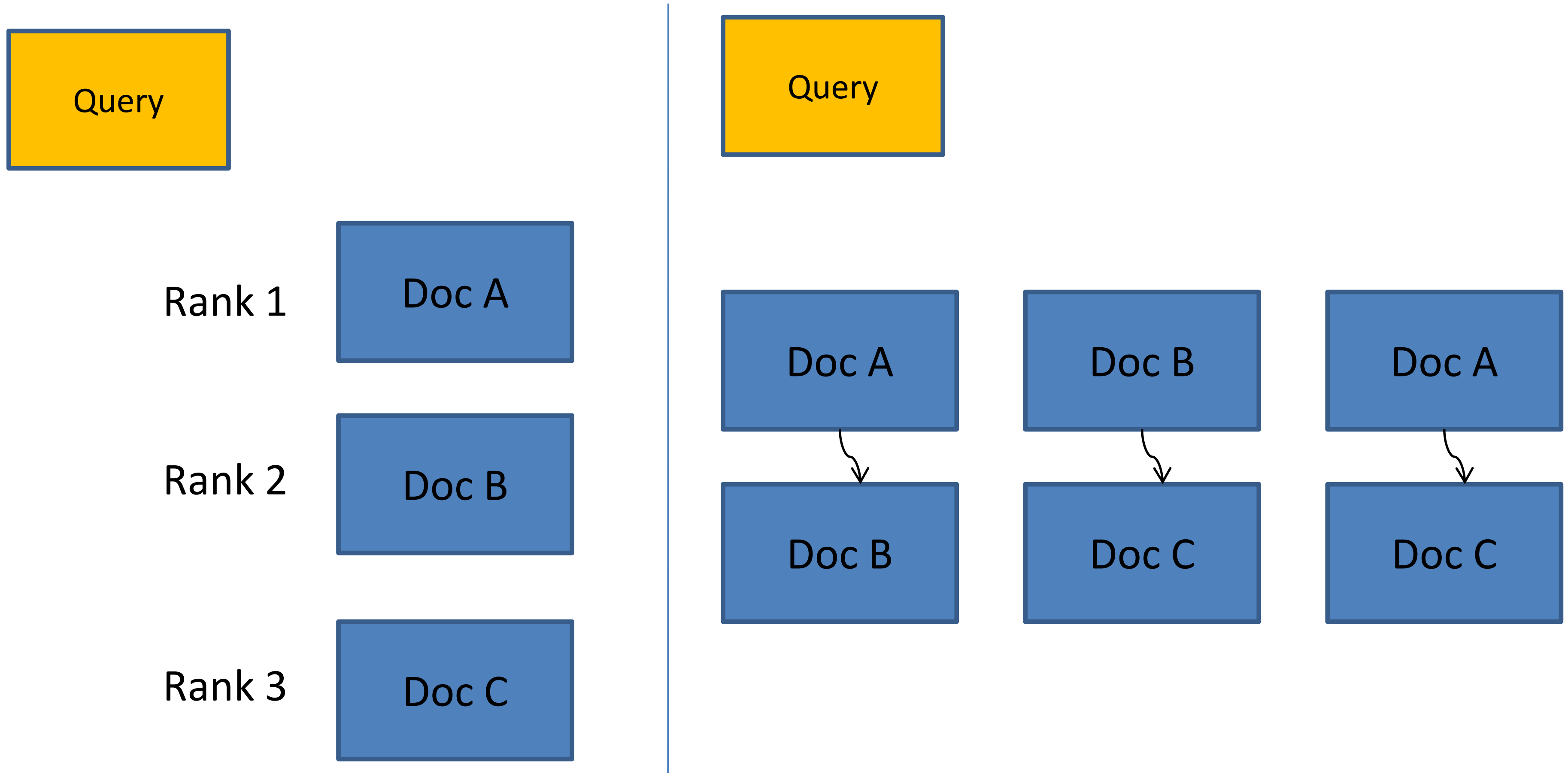
- Transforming ranking to pairwise classification
- Query-document group structure is ignored

Pairwise Approach

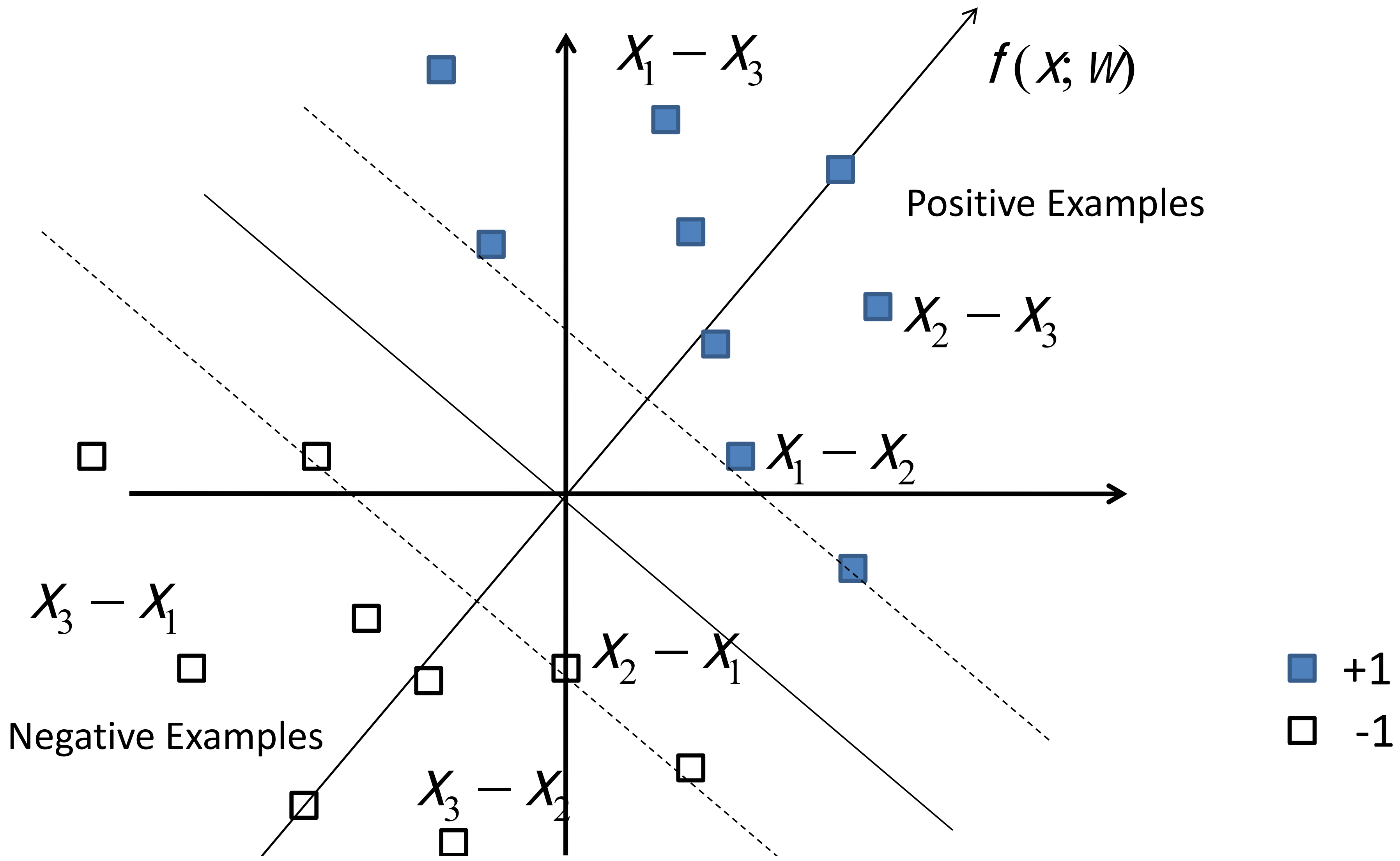
	Learning
Input	Ordered feature vector pair (x_i, x_j)
Output	Classification on order of vector pair $y_{i,j} = \mathrm{sign}(f(x_i - x_j))$
Model	Ranking function $f(x)$
Loss Function	Pairwise classification loss

Pairwise Ranking

- Converting document list to document pairs



Transformed Pairwise Classification Problem



Listwise Approach

- Entire list of documents as input-instance
- Query-document group structure is used
- Straightforwardly represents learning to rank problem
  - Many state of the art methods: ListNet, ListMLE, etc.

Listwise Approach

	Learning & Ranking
Input	Feature vectors $\mathrm{X} = \{x_i\}_{i=1}^{n}$
Output	Permutation on feature vectors $\boldsymbol{\pi} = \mathrm{sort}(\{f(x_i)\}_{i=1}^{n})$
Model	Ranking function $f(x)$
Loss Function	Listwise loss function (ranking evaluation measure)