

# Reinforcement Learning to Rank with Markov Decision Process

Jun Xu  
ICT, CAS

# Outline

- Background: learning to rank for IR
- Reinforcement learning to rank
- Summary

# Ranking is Important for Web Search

The screenshot shows a search engine interface with the query 'Data Mining' in the search bar. Below the search bar are navigation tabs for 'Web', 'Images', 'Videos', 'Maps', 'News', and 'My saves'. The search results are displayed as a list of links with titles and snippets. The first result is from Wikipedia, followed by a blog post from frandweb.net, a page from thearling.com, a page from investopedia.com, a page from sas.com, a page from whatis.com, and a page from microsoft.com.

Data Mining

Web Images Videos Maps News My saves

1,050,000 RESULTS Any time ▾

**Data mining - Wikipedia**  
[https://en.wikipedia.org/wiki/Data\\_mining](https://en.wikipedia.org/wiki/Data_mining) ▾  
Data mining is the computing process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and ...

**Data Mining: What is Data Mining? - frandweb.net**  
[www.frandweb.net/jason](http://www.frandweb.net/jason) ▾  
Welcome to Jason Frand's Homepage. September 1, 2006 was the start of an entirely new career for me.

**An Introduction to Data Mining - Analytics and Data ...**  
[www.thearling.com/text/dmwhite/dmwhite.htm](http://www.thearling.com/text/dmwhite/dmwhite.htm) ▾  
An Introduction to Data Mining. Discovering hidden value in your data warehouse. Overview. Data mining, the extraction of hidden predictive information from large ...

**Data Mining - Investopedia**  
[www.investopedia.com/terms/d/datamining.asp](http://www.investopedia.com/terms/d/datamining.asp) ▾  
Data mining is a process used by companies to turn raw data into useful information. By using software to look for patterns in large batches of data, businesses can ...

**What is data mining? | SAS**  
[https://www.sas.com/en\\_us/insights/analytics/data-mining.html](https://www.sas.com/en_us/insights/analytics/data-mining.html) ▾  
Data Mining History and Current Advances. The process of digging through data to discover hidden connections and predict future trends has a long history.

**What is data mining? - Definition from WhatIs.com**  
[searchsqlserver.techtarget.com/definition/data-mining](http://searchsqlserver.techtarget.com/definition/data-mining) ▾  
Data mining is the process of sorting through large data sets to identify patterns and establish relationships to solve problems through data analysis.

**Data Mining - Microsoft Research**  
[www.microsoft.com/en-us/research/project/data-mining](http://www.microsoft.com/en-us/research/project/data-mining) ▾  
The Knowledge Discovery and Data Mining (KDD) process consists of data selection, data cleaning, data transformation and reduction, mining, interpretation and ...

- Criteria

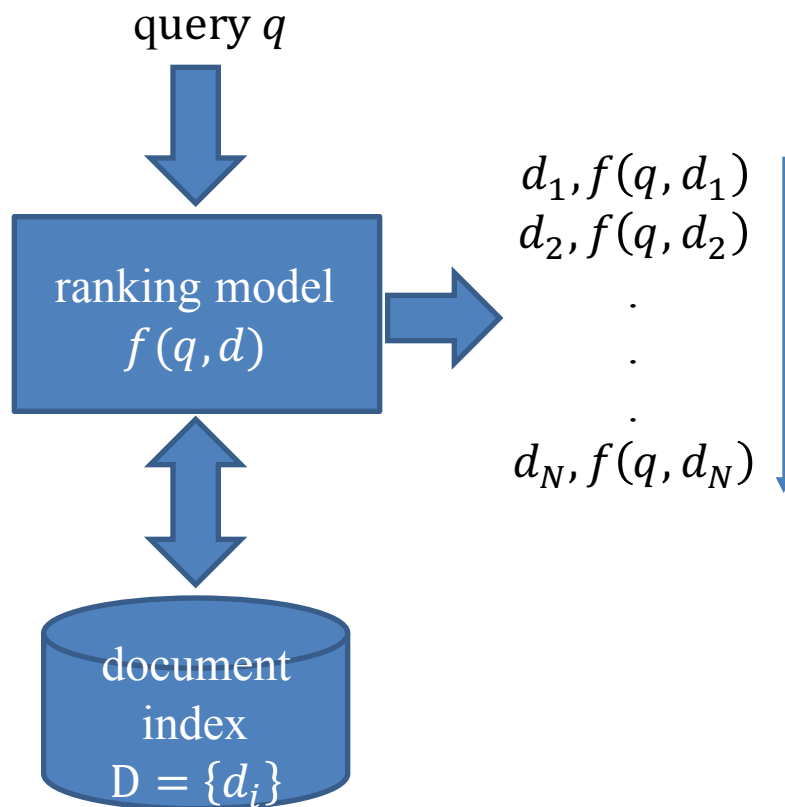
- Relevance
- Diversity
- Freshness

.....

- Ranking model

- Heuristic
  - Relevance: BM25, LMIR
  - Diversity: MMR, xQuAD
- Learning to rank

# Ranking in Information Retrieval



Learning to Rank

Web 1-10 of 8,430,000 results - [Advanced](#)  
See also: [Images](#), [Video](#), [News](#), [Maps](#)<sup>beta</sup>, [More](#) ▾

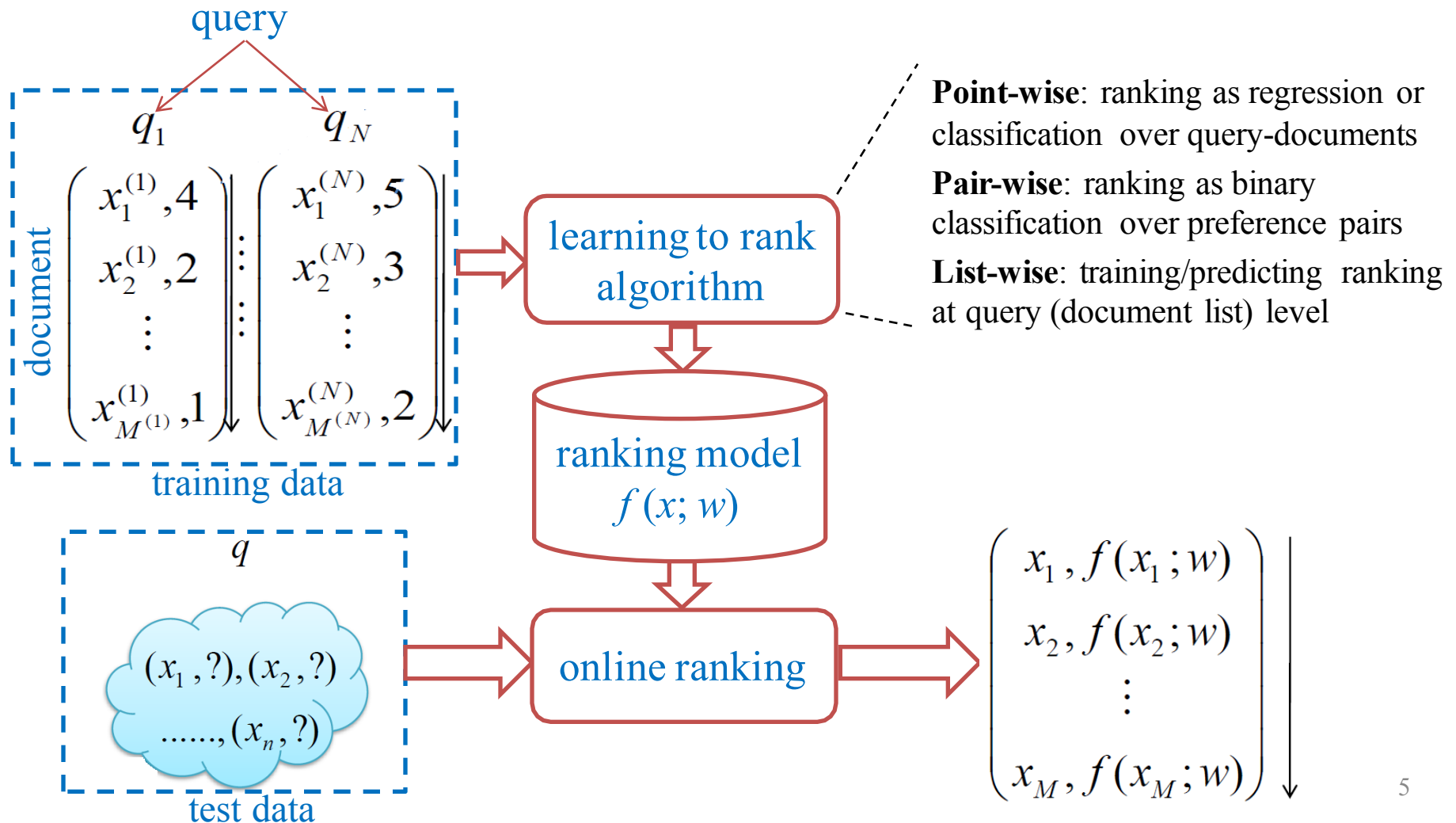
[Libra: Learning to rank with non - smooth cost functions](#)  
[Learning to rank with non - smooth cost functions\(2006\)](#) (Citation:4) C. Burges R. Ragno Q. Le View or Download: <http://research.microsoft.com/~cburges/papers/LambdaRank.pdf> Live Search [libra.msra.cn/paperdetail.aspx?id=4114251](#) · [Cached page](#)

[Query-Level Stability and Generalization in Learning to Rank](#)  
Query-Level Stability and Generalization in [Learning to Rank](#) We propose anew probabilistic formulation of [learning to rank](#) for IR. The formulation can naturally represent the pointwise, pairwiseandlistwise approaches in a unified framework. Within the framework, we introduce the concepts of query-level loss, query-level risk, and particularly query ... [www.amt.ac.cn/member/mazhiming/papers/ma081004-2.pdf](http://www.amt.ac.cn/member/mazhiming/papers/ma081004-2.pdf) · [Cached page](#) · PDF file

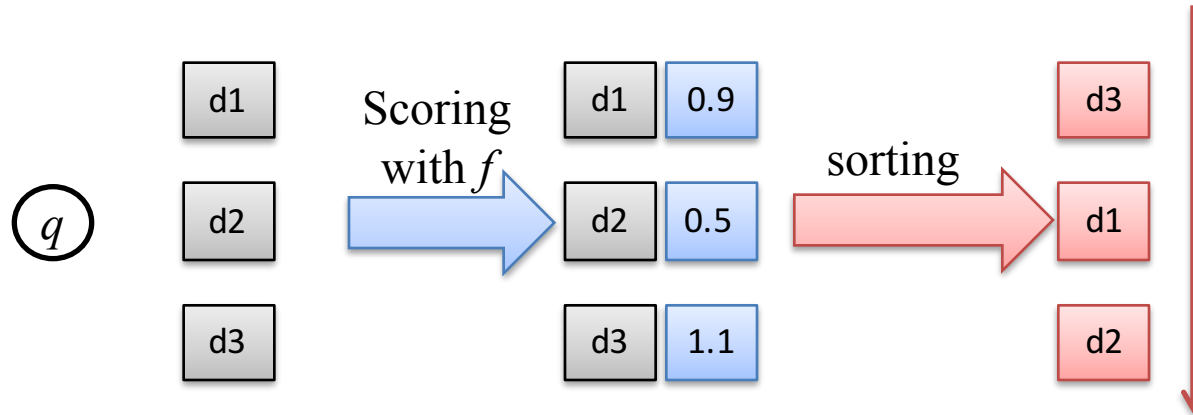
[Libra: Learning to rank using classification and gradient boosting](#)  
On Using Simultaneous Perturbation Stochastic Approximation for [Learning to Rank](#), and the Empirical Optimality of LambdaRank Yisong Yue One shortfall of existing machine [learning](#) (ML) methods when ap-plied to information retrieval (IR) is the inability to directly optimize for typical IR performance measures. [libra.msra.cn/paperdetailed.aspx?id=4114249](http://libra.msra.cn/paperdetailed.aspx?id=4114249) · [Cached page](#)

# Learning to Rank for Information Retrieval

- Machine learning algorithms for relevance ranking



# Independent Relevance Assumption



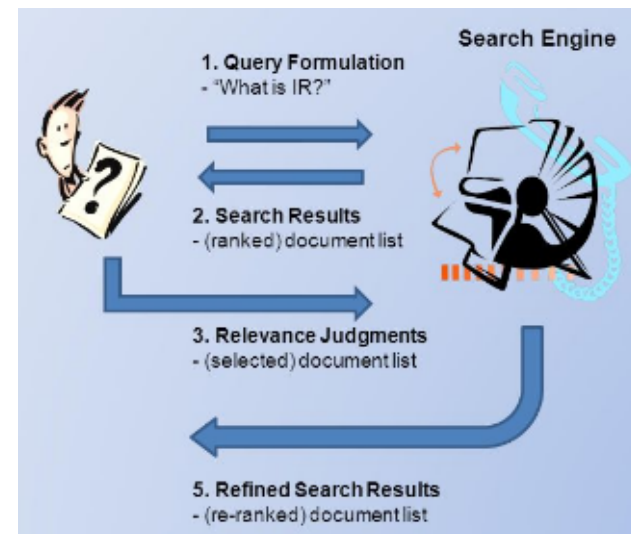
- Utility of a doc is independent of other docs
- Ranking as scoring & sorting
  - Each documents can be scored independently
  - Scores are independent of the rank

# Beyond Independent Relevance

- More ranking criteria, e.g., search result diversification
  - Covering as much subtopics as possible with a few documents
  - Need consider the **novelty** of a document given preceding documents
- Complex application environment, e.g., Interactive IR
  - Human interacts with the system during the ranking process
  - User feedback is helpful for improving the remaining results

Query: Programming language

Good	Bad
Java	Java
C++	Java
Python	Java



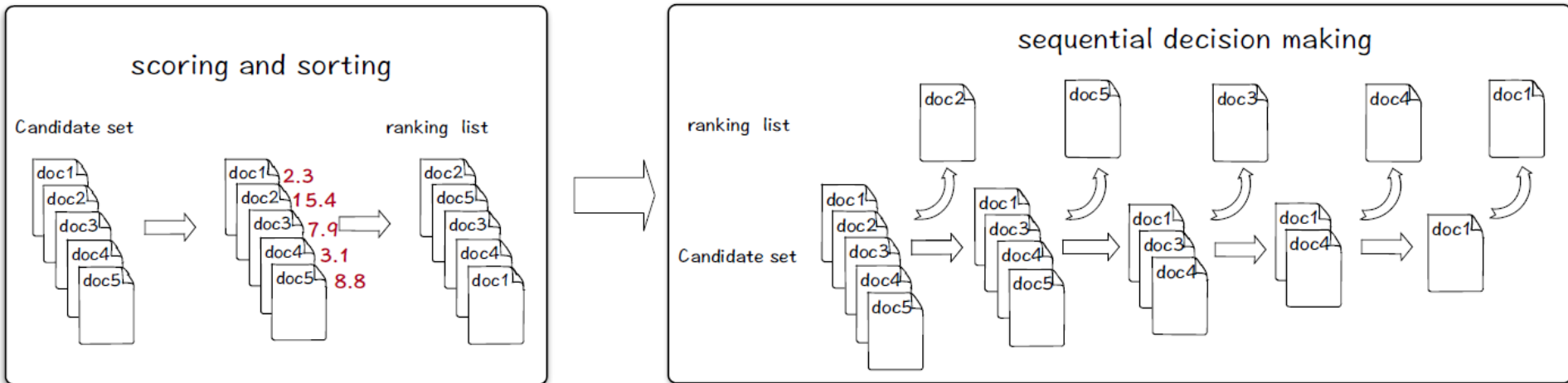
Need more powerful ranking mechanism!

# Outline

- Background: learning to rank for IR
- Reinforcement learning to rank
  - Ranking as Markov decision process
  - Adapting MDP for relevance and diverse ranking
- Summary



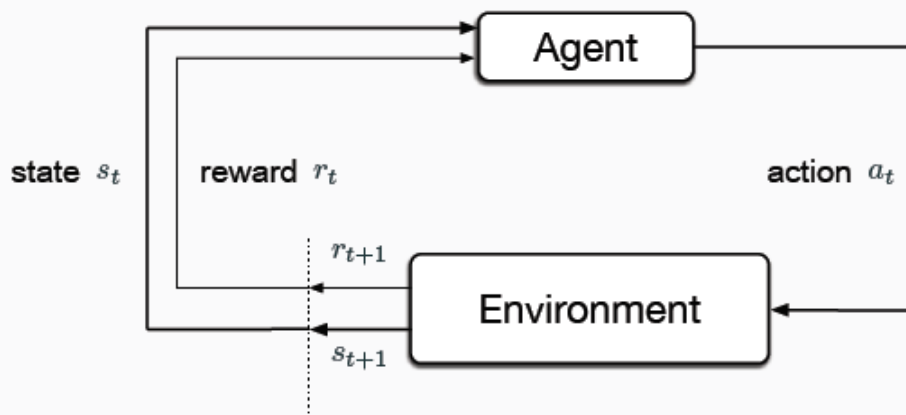
# From Scoring & Sorting to Sequential Decision Making



- Advantages: beyond independent relevance
  - Modeling the dependencies between documents
  - Taking the ranking positions into consideration

# Markov Decision Process (MDP)

- An MDP is composed by states, actions, rewards, policy, and transitions, and represented by a tuple  $\langle S, A, T, R, \pi \rangle$



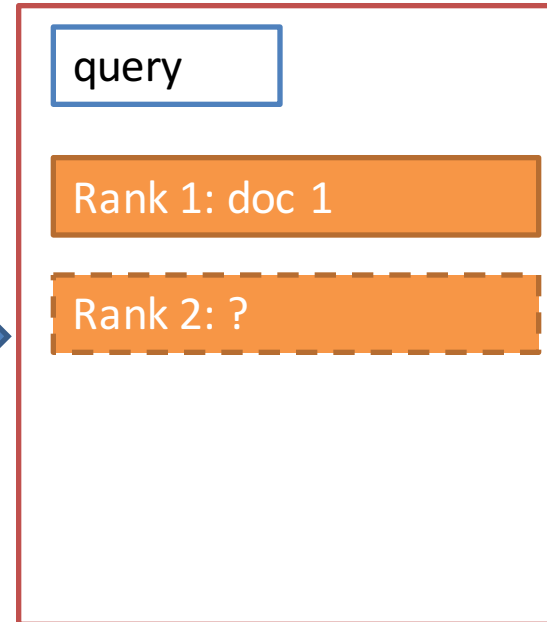
- **States  $S$** : a set of states.
- **Actions  $A$** : a discrete set of actions that an agent can take.
- **Transition  $T$** : the state transition function  $s_{t+1} = T(s_t, a_t)$
- **Reward  $r = R(s, a)$** : the immediate reward, also known as reinforcement
- **Policy  $\pi(a|s)$** : a probability distribution over the possible actions.

# Ranking as Markov Decision Process

Candidate document set



Decide which doc  
should be selected for  
the 2<sup>nd</sup> position



- Time steps: ranks
- State: query, preceding docs, candidates, .....
- Policy: distribution over remaining candidate documents
- Action (Decision): selecting a doc and placing it to current pos
- Reward
  - Additional utility (e.g., the increase of DCG) from the selected doc
  - Calculated based on widely used evaluation measures (e.g., DCG, ERR-IA)

# Learning and Online Ranking

- Learning the parameters
  - Model parameters: policy function, state initialization and transition etc.
  - Reinforcement learning: policy gradient
  - Rewards based on relevance labels as supervision
- Online ranking
  - Without rewards (rewards are based on relevance labels)
  - Fully trust the learned policy

# Example 1: Learning for Search Result Diversification

Long Xia, Jun Xu, Yanyan Lan, et al., Adapting Markov Decision Process for Search Result Diversification. Proceedings of SIGIR 2017, pp. 535-544.

# Search Result Diversification

Query: jaguar


Market Selector | Jaguar | View the site in your preferred language  
<https://www.jaguar.com/> +  
 Discover the different language sites we have to make browsing our vehicle range's easier. We have over 100 different language options available. Learn more.

Jaguar (@Jaguar) · Twitter  
<https://twitter.com/Jaguar> +  
 Panasonic @JaguarRacing completed their debut @FIAFormulaE season at the #MontrealEPro. The focus now shifts to preparing for season four.  
 pic.twitter.com/dDQqFr...  
 6 hours ago · Twitter

Jaguar UK: Luxury Sports Cars, Executive Saloons and SUVs  
<https://www.jaguar.co.uk/> +  
 Jaguar - The Art of Performance. Explore our range of luxury sports cars, saloon cars and SUVs including the XE, XF, XJ, F-TYPE, and F-PACE.  
 Build and price. Approved Used Jaguar Cars. Build your sf. F Type

JAGUAR HONG KONG  
[www.jaguar.com.hk/](http://www.jaguar.com.hk/) + Translate this page  
 Official Jaguar Hong Kong website. Discover luxury cars featuring innovative design and legendary performance. Book a test drive in Hong Kong today.

Jaguar Cars - Wikipedia  
[https://en.wikipedia.org/wiki/Jaguar\\_Cars](https://en.wikipedia.org/wiki/Jaguar_Cars) +  
 Jaguar is the luxury vehicle brand of Jaguar Land Rover, a British multinational car manufacturer with its headquarters in Whitley, Coventry, England, owned by ...

Images for jaguar  
  
 → More images for jaguar Report Images

Jaguar - Home | Facebook  
<https://www.facebook.com/Jaguar/> +  
 Jaguar · 18755409 likes · 4451 talking about this · 112 were here. Jaguar. The Art of Performance.

Jaguar - YouTube  
<https://www.youtube.com/user/JaguarCarsLimited> +  
 Since the first Jaguar car was produced in 1935 we have pushed the boundaries of what is possible. We've always believed that a car is the closest thing you ...


Jaguar MENA: Explore Jaguar the High Performance Luxury Cars  
<https://www.jaguar-mena.com/> +  
 Visit the Official Jaguar MENA website and explore our luxury sports cars, sedan, 4x4, coupe and convertible. Discover the art of performance with Jaguar.

Jaguar Las Vegas | New & Used Car Dealer Las Vegas, NV  
[www.jaguarlv.com/](http://www.jaguarlv.com/) +  
 Jaguar Las Vegas is Southern Nevada's exclusive Jaguar retailer offering an exquisite selection of new and used vehicles. Visit us today in Las Vegas.

Market Selector | Jaguar | View the site in your preferred language  
<https://www.jaguar.com/> +  
 Discover the different language sites we have to make browsing our vehicle range's easier. We have over 100 different language options available. Learn more.

Jaguar (@Jaguar) · Twitter  
<https://twitter.com/Jaguar> +  
 Panasonic @JaguarRacing completed their debut @MitchEvans\_and @AdamCarroll47 head to the #MontrealEPro street circuit for the season finale. #FormulaE  
 pic.twitter.com/9BnBbuR...  
 12 hours ago · Twitter

Jaguar - Wikipedia  
<https://en.wikipedia.org/wiki/Jaguar> +  
 The jaguar (*Panthera onca*) is a big cat, a feline in the Panthera genus, and is the only extant Panthera species native to the Americas. The jaguar is the ...  
 Jaguar Cars · Pantanal jaguar · El Jefe · Southwestern United States

Images for jaguar  
  
 → More images for jaguar Report Images

Fender American Pro Jaguar® Rosewood Fingerboard, 3-Color ...  
[shop.fender.com/en-US/electro-guitars/jaguar/jaguar-jaguar0114010700.html](http://shop.fender.com/en-US/electro-guitars/jaguar/jaguar-jaguar0114010700.html) +  
 US\$1,549.99  
 An eye-catchingly adventurous design—an exercise in chrome, plastic and wood—the Jaguar guitar's delightfully off-kilter aesthetics and unique sound made it ...

Jaguar | jaguar-swisswatches.com/ +  
 Jaguar is the brand of Swiss Made watches for fans of the most demanding products, seeking quality, exclusivity and distinction. The passion for precision and ...

Brands > Jaguar - MENRAD  
<https://www.menrad.de/en/collection/jaguar/> +  
 The JAGUAR Eyewear collection mirrors the unique elegance and drive of the JAGUAR sports car. Design interpretations from car to eyewear such as carbon ...

Jaguar Mining Inc.: Home  
<https://www.jaguarmining.com/> +  
 Jaguar is a producing, grid, development, and exploration company operating in the Iron Quadrangle, a prolific greenstone belt located in Minas Gerais, Brazil.

Jaguar Cars - Wikipedia  
[https://en.wikipedia.org/wiki/Jaguar\\_Cars](https://en.wikipedia.org/wiki/Jaguar_Cars) +  
 Jaguar is the luxury vehicle brand of Jaguar Land Rover, a British multinational car manufacturer with its headquarters in Whitley, Coventry, England, owned by ...

- Luxury car
- Animal
- Electric
- Swiss
- Eyewear
- Mining Inc.

- Query: information needs are ambiguous and multi-faceted
- Search results: may contain redundant information
- Goal: covering as much subtopics as possible with a few documents

# Modeling Diverse Ranking with MDP

- Key points
  - Mimic user top-down browsing behaviors
  - Model dynamic information needs with MDP state
- States  $s_t = [Z_t, X_t, \mathbf{h}_t]$ 
  - $Z_t$ : sequence of  $t$  preceding documents,  $Z_0 = \phi$
  - $X_t$ : set of candidate documents,  $X_0 = X$
  - $\mathbf{h}_t \in R^K$ : latent vector, encodes user **perceived utility from preceding documents**, initialized with the information needs from the query:

$$\mathbf{h}_0 = \sigma(\mathbf{V}_q \mathbf{q})$$

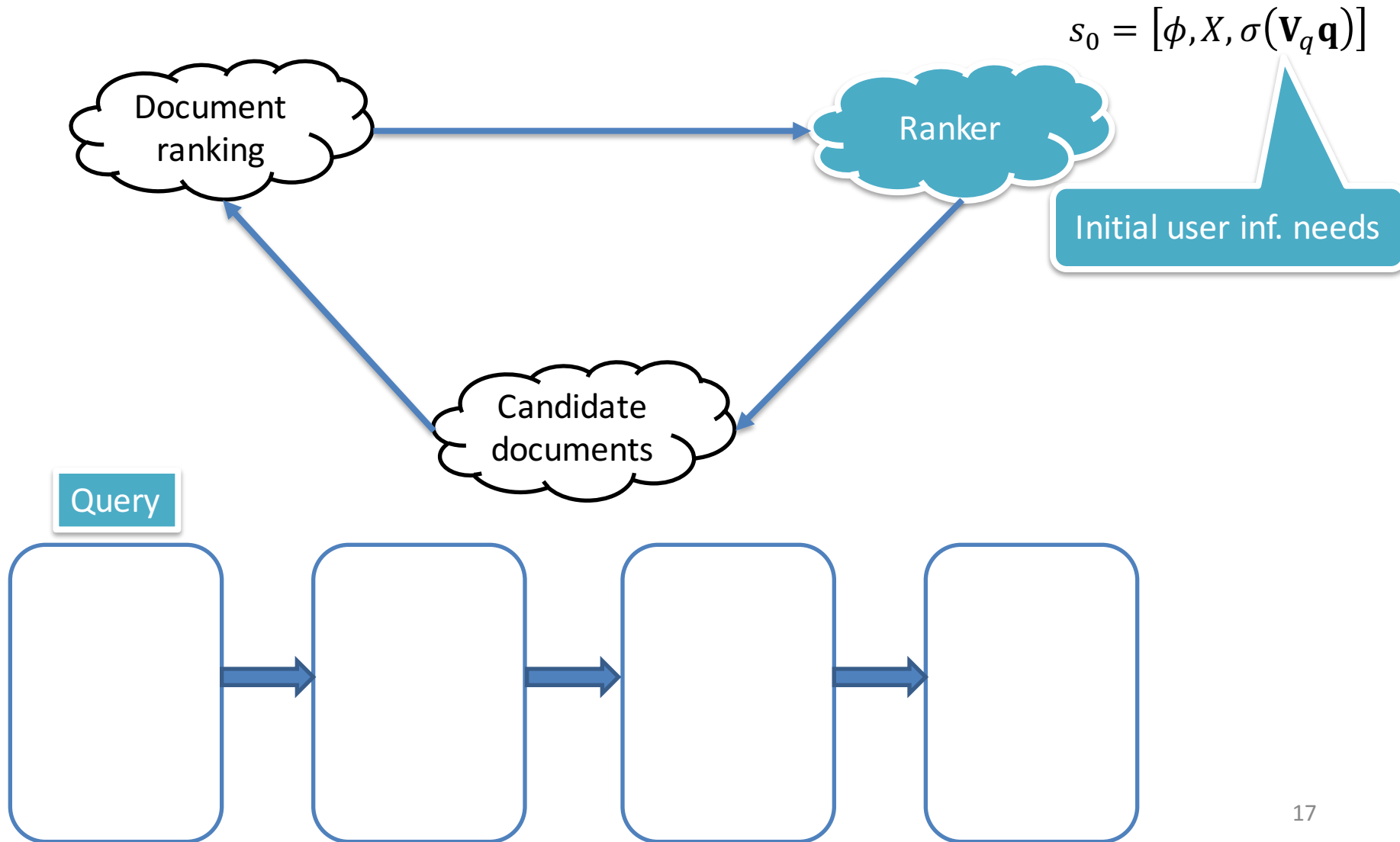
# Modeling Diverse Ranking with MDP

$\mathbf{x}_{m(a_t)}$ : document embedding

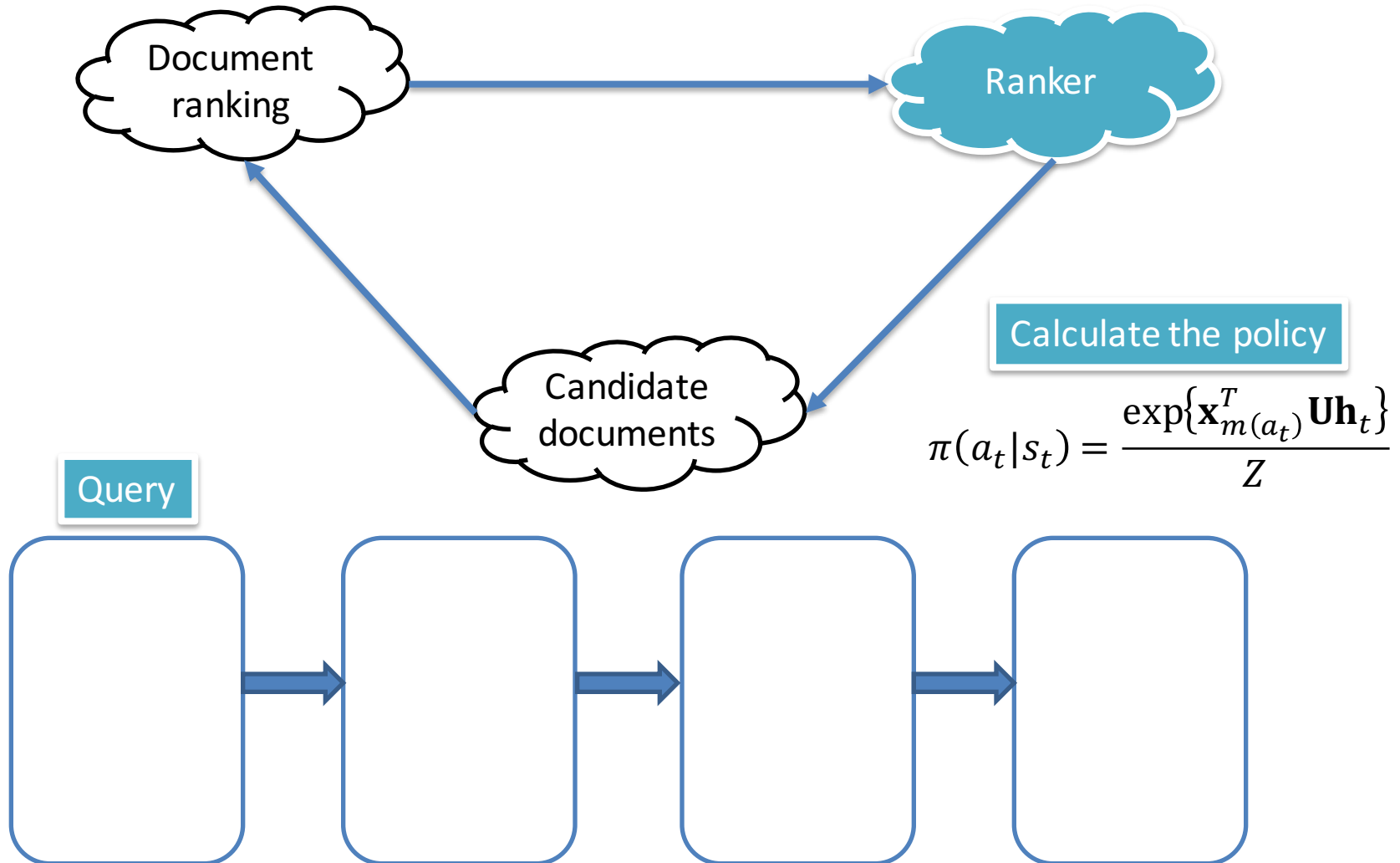
MDP factors	Corresponding diverse ranking factors
Time steps	The ranking positions
State	$s_t = [Z_t, X_t, \mathbf{h}_t]$
Policy	$\pi(a_t   s_t = [Z_t, X_t, \mathbf{h}_t]) = \frac{\exp\{\mathbf{x}_{m(a_t)}^T \mathbf{U} \mathbf{h}_t\}}{Z}$
Action	Selecting a doc and placing it to rank $t + 1$
Reward	Based on evaluation measure $\alpha$ DCG, SRecall etc. For example: $R = \alpha \text{DCG}[t + 1] - \alpha \text{DCG}[t];$ $R = \text{SRecall}[t + 1] - \text{SRecall}[t]$
State Transition	$s_{t+1} = T(s_t = [Z_t, X_t, \mathbf{h}_t], a_t)$ $= [Z_t \oplus \{\mathbf{x}_{m(a_t)}\}, X_t \setminus \{\mathbf{x}_{m(a_t)}\}, \sigma(\mathbf{V} \mathbf{x}_{m(a_t)} + \mathbf{W} \mathbf{h}_t)]$



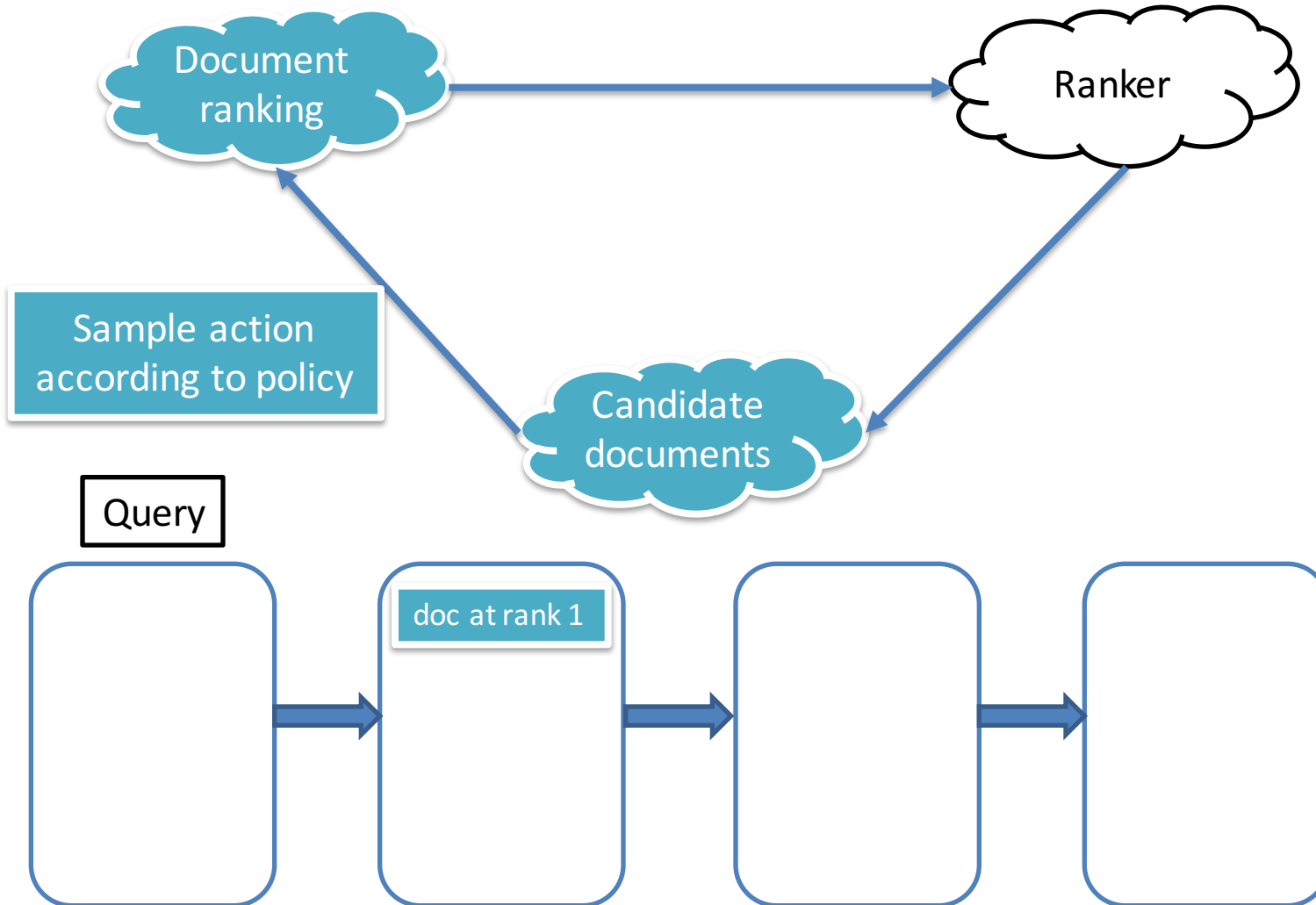
# Ranking Process: Initialize State



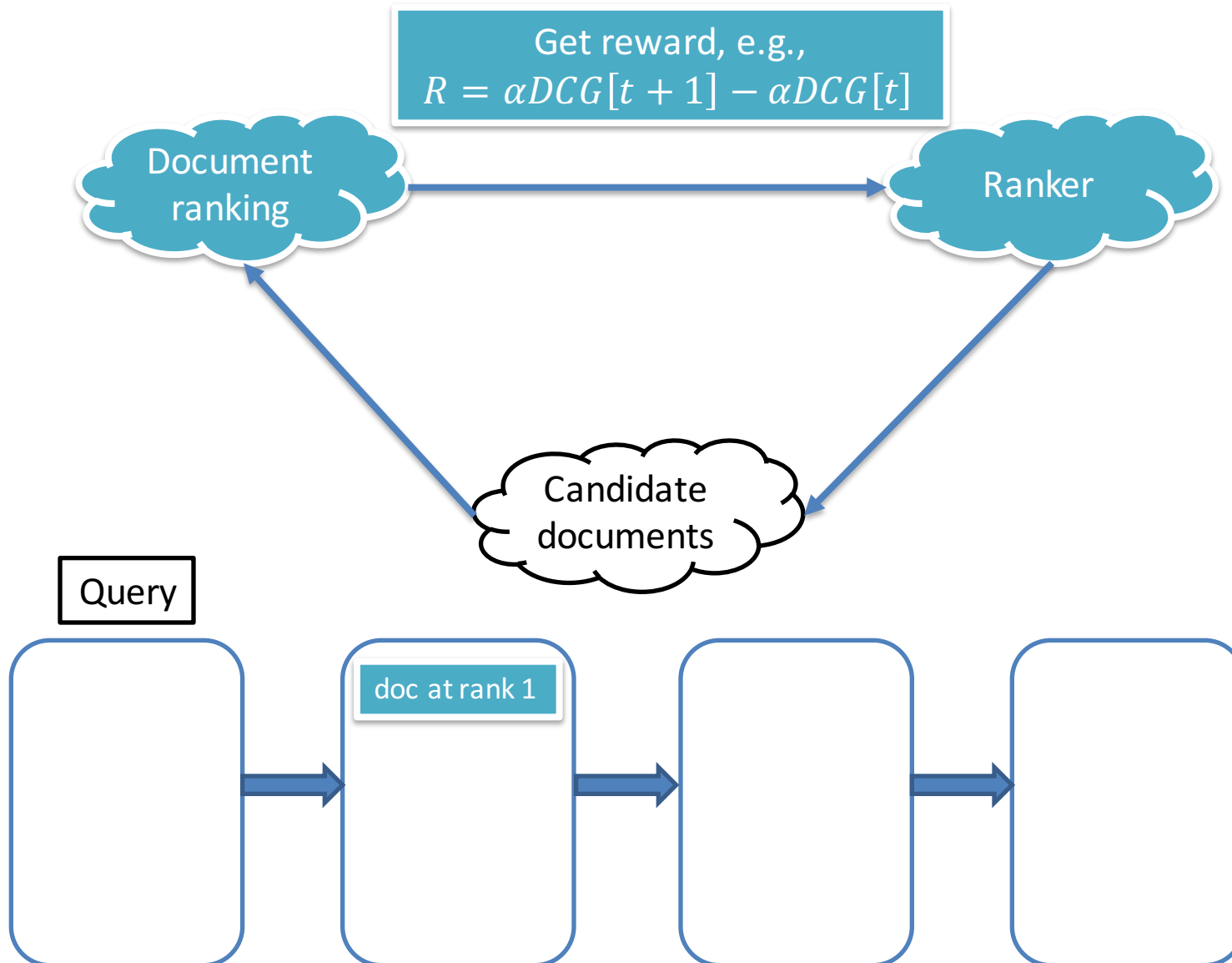
# Ranking Process: Policy



# Ranking Process: Action



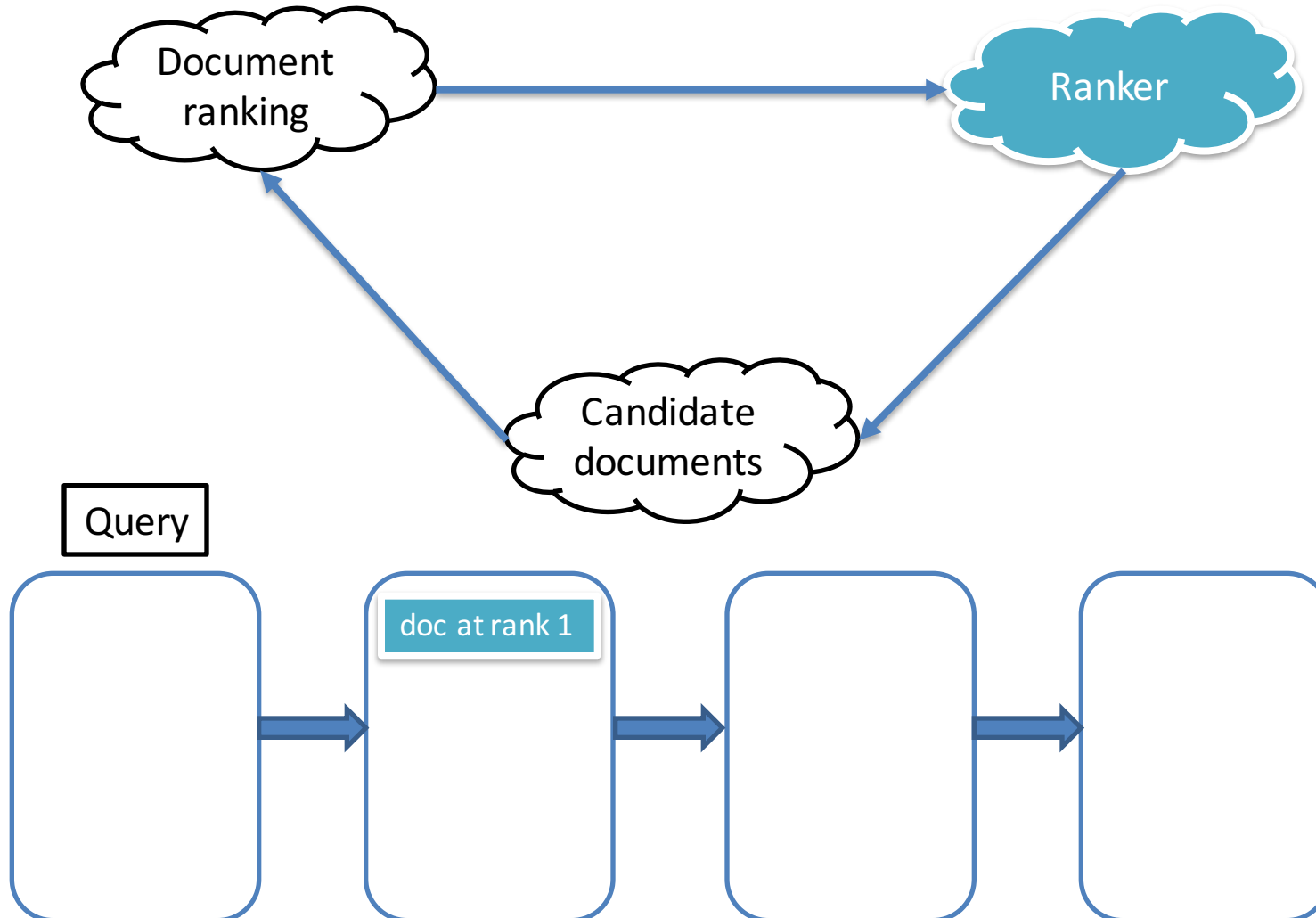
# Ranking Process: Reward



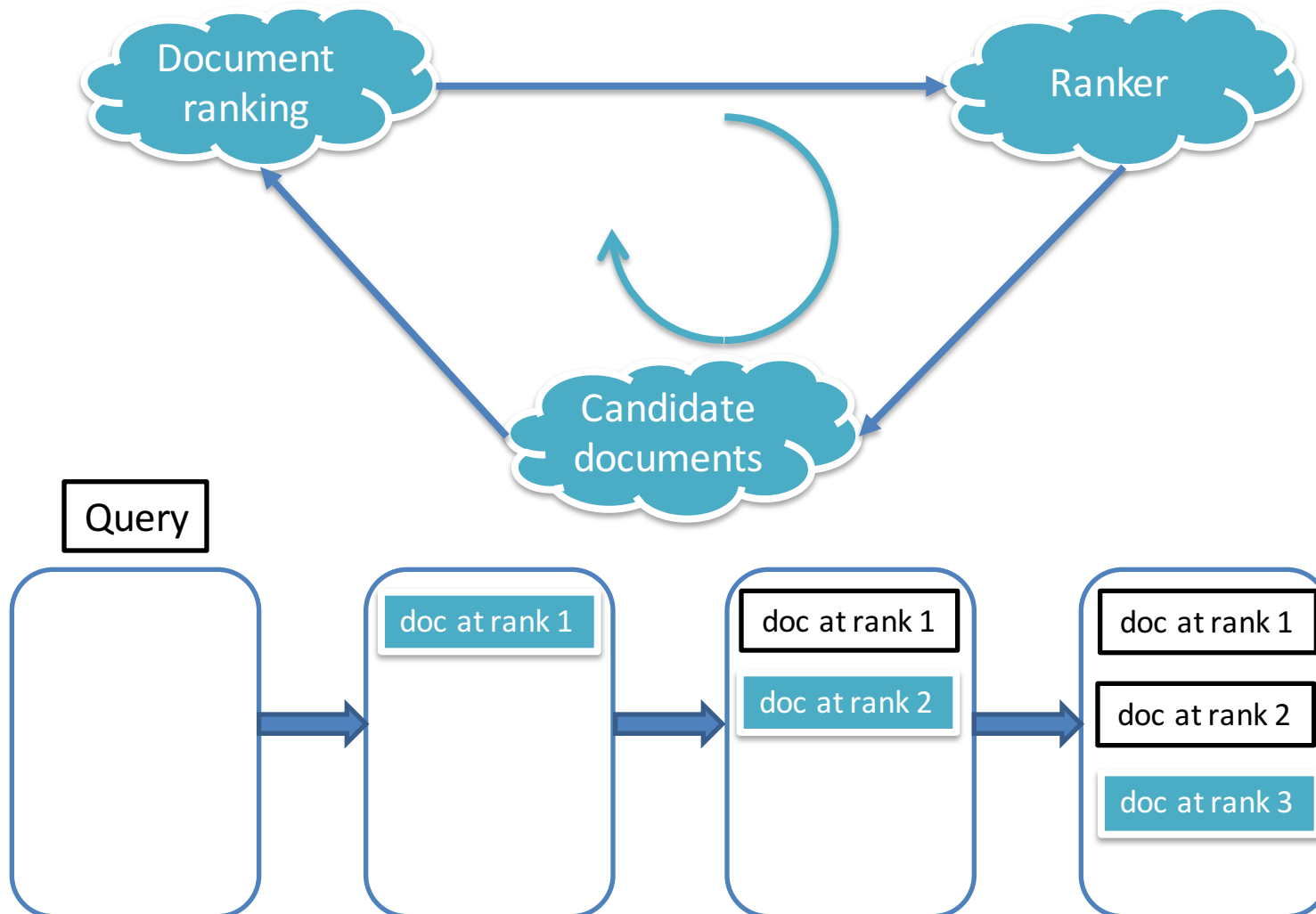
# Ranking Process: State Transition

Update ranked list, candidate set, and latent vector

$$s_{t+1} = [Z_t \oplus \{\mathbf{x}_{m(a_t)}\}, X_t \setminus \{\mathbf{x}_{m(a_t)}\}, \sigma(\mathbf{V}\mathbf{x}_{m(a_t)} + \mathbf{W}\mathbf{h}_t)]$$



# Ranking Process: Iterate



# Learning with Policy Gradient

- Model parameters  $\Theta = \{\mathbf{V}_q, \mathbf{U}, \mathbf{V}, \mathbf{W}\}$
- Learning objective: maximizing expected return (discounted sum of rewards) of each training query

$$\max_{\Theta} v(\mathbf{q}) = E_{\pi} G_0 = E_{\pi} \left[ \sum_{k=0}^{M-1} \gamma^k r_{k+1} \right]$$

– Directly optimizes evaluation measure as  $G_0 = \alpha \text{DCG}@M$

- Monte-Carlo stochastic gradient ascent is used to conduct the optimization (REINFORCE algorithm)

$$\widehat{\nabla_{\Theta} v(\mathbf{q})} = \gamma^t G_t \nabla_{\Theta} \log \pi(a_t | s_t; \Theta)$$

# Analysis

- Optimize general diversity evaluation measures (e.g.,  $\alpha$ -DCG, S-recall)

discounted sum of the rewards, starting from position 0 (return)

$$\max_{\Theta} V(\mathbf{q}) = \mathbb{E}_{\pi} G_0$$

- Given an episode and time step  $t$

$$\begin{aligned} \widehat{\nabla_{\Theta} V(\Theta)} &\stackrel{\text{sample}}{=} \gamma^t \sum_{a \in A(s_t)} \nabla_{\Theta} \pi(a|s_t) Q^{\pi}(s_t, a) \\ &= \gamma^t \sum_{a \in A(s_t)} \pi(a|s_t) \cdot \left( Q^{\pi}(s_t, a) \frac{\nabla_{\Theta} \pi(a|s_t)}{\pi(a|s_t)} \right) \end{aligned}$$

$$\stackrel{\text{sample}}{=} \gamma^t Q^{\pi}(s_t, a_t) \frac{\nabla_{\Theta} \pi(a_t|s_t)}{\pi(a_t|s_t)}$$

$$\stackrel{\text{sample}}{=} \gamma^t G_t \nabla_{\Theta} \log \pi(a_t|s_t).$$

Maximizing the return starting from position  $t$



# The Learning Algorithm

## Algorithm 1 MDP-DIV learning

**Input:** Labeled training set  $D = \{(q^{(n)}, X^{(n)}, I^{(n)})\}_{n=1}^N$ , learning rate  $\eta$ , discount factor  $\gamma$ , and reward function  $R$

**Output:**  $\Theta = \{V_q, U, V, W\}$

- 1: Initialize  $\Theta = \{V_q, U, V, W\} \leftarrow$  random
- 2: **repeat**
- 3:   **for all**  $(q, X, J) \in D$  **do**
- 4:      $(s_0, a_0, r_1, \dots, s_{M-1}, a_{M-1}, r_M) \leftarrow$  Sample from  $S$  {Algorithm (2), and  $M = |X|$ }
- 5:     **for**  $t = 0$  **to**  $M - 1$  **do**
- 6:        $G_t \leftarrow \sum_{k=0}^{M-1-t} \gamma^k r_{t+k+1}$  {Equation (1)}
- 7:        $\Theta \leftarrow \Theta + \eta \gamma^t G_t \nabla_{\Theta} \log \pi(a_t | s_t; \Theta)$
- 8:     **end for**
- 9:   **end for**
- 10: **until** converge
- 11: **return**  $\Theta$

## Algorithm 2 SampleEpisode

**Input:** Parameters  $\Theta = \{V_q, U, V, W\}$ ,  $q, X, J$ , and  $R$

**Output:** An episode

- 1: Initialize  $s \leftarrow [\emptyset, X, \sigma(V_q q)]$  {Equation (1)}
- 2:  $M \leftarrow |X|$
- 3:  $E = ()$  {empty episode}
- 4: **for**  $t = 0$  **to**  $M - 1$  **do**
- 5:    $A \leftarrow A(s)$  {Possible actions according to  $X$  in state  $s$ }
- 6:   **for all**  $a \in A$  **do**
- 7:      $P(a) \leftarrow \pi(a | s; \Theta)$
- 8:   **end for**
- 9:   Sample an action  $\hat{a} \in A$ , according to  $P$
- 10:    $r \leftarrow R(s, \hat{a})$  {Calculation on the basis of  $J$ }
- 11:   Append  $(s, \hat{a}, r)$  to the tail of  $E$
- 12:    $[\mathcal{Z}, X, \mathbf{h}] \leftarrow s$
- 13:    $s \leftarrow [\mathcal{Z} \oplus \{\mathbf{x}_{m(\hat{a})}\}, X \setminus \{\mathbf{x}_{m(\hat{a})}\}, \sigma(V \mathbf{x}_{m(\hat{a})} + W \mathbf{h})]$
- 14: **end for**
- 15: **return**  $E = (s_0, a_0, r_1, \dots, s_{M-1}, a_{M-1}, r_M)$

# Online Ranking Algorithm

- Fully trust the policy

---

**Algorithm 3** MDP-DIV online ranking

---

**Input:** Parameters  $\Theta = \{\mathbf{V}_q, \mathbf{U}, \mathbf{V}, \mathbf{W}\}$ , query  $\mathbf{q}$ , documents  $X$

**Output:** Permutation of documents  $\tau$

- 1: Initialize  $s \leftarrow [\emptyset, X, \sigma(\mathbf{V}_q \mathbf{q})]$  {Equation (1)}
  - 2:  $M \leftarrow |X|$
  - 3: **for**  $t = 0$  **to**  $M - 1$  **do**
  - 4:    $A \leftarrow A(s)$  {Possible actions according to  $X$  in state  $s$ }
  - 5:    $\hat{a} \leftarrow \arg \max_{a \in A} \pi(a|s; \Theta)$  {Choosing most possible action}
  - 6:    $\tau[t + 1] \leftarrow m(\hat{a})$  {Document  $\mathbf{x}_{m(\hat{a})}$  is ranked at  $t + 1$ }
  - 7:    $[\mathcal{Z}, X, \mathbf{h}] \leftarrow s$
  - 8:    $s \leftarrow [\mathcal{Z} \oplus \{\mathbf{x}_{m(\hat{a})}\}, X \setminus \{\mathbf{x}_{m(\hat{a})}\}, \sigma(\mathbf{V}\mathbf{x}_{m(\hat{a})} + \mathbf{W}\mathbf{h})]$
  - 9: **end for**
  - 10: **return**  $\tau$
- 

using max instead  
of sampling

# Experimental Results

Method	$\alpha$ -NDCG@5	$\alpha$ -NDCG@10	S-recall@5	S-recall@10
MMR	0.2753	0.2979	0.4388	0.5151
xQuAD	0.3165	0.3941	0.4933	0.6043
PM-2	0.3047	0.3730	0.4910	0.6012
SVM-DIV	0.3030	0.3699	0.5122	0.6230
R-LTR	0.3498	0.4132	0.5397	0.6511
PAMM( $\alpha$ -NDCG)	0.3712	0.4327	0.5561	0.6612
NTN-DIV( $\alpha$ -NDCG)	0.3962	0.4577	0.5817	0.6872
MDP-DIV(S-recall)	0.4156	0.4734	<b>0.6123</b>	<b>0.7155</b>
MDP-DIV( $\alpha$ -DCG)	<b>0.4189</b>	<b>0.4762</b>	0.6102	0.7117

- Based on combination of TREC 2009 ~ 2012 Web Track
- Directly optimize a predefined measure via defining the rewards based on the measure

# How it works?

## Using Query 93 as Example

q

raffles

- |                                    |   |
|------------------------------------|---|
| [1] : “Raffles Hotel in Singapore” | $d_1$ : “Stamford Raffles – Wikipedia, the free encyclopedia” [2] |
| [2] : “Sir Stamford Raffles”       | $d_2$ : “Fundraiser Raffle Ideas” [3, 5]                          |
| [3] : “organizing a raffle”        | $d_3$ : “Luxury Hotel Guide   Raffles Hotels” [1, 4]              |
| [4] : “the Raffles hotel in Dubai” | $d_4$ : “National Corvette Museum – Corvette Raffles” [5]         |
| [5] : “car raffles”                | $d_5$ : “Raffles Hotels and Resorts” [1, 4]                       |

# How it works?

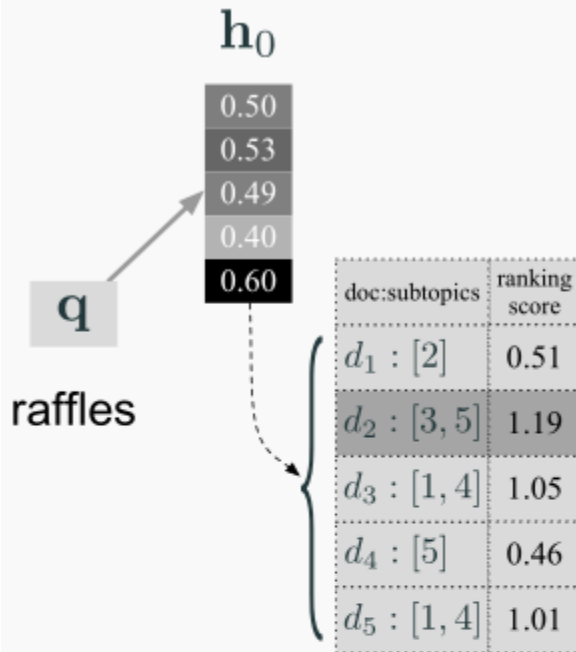
## Using Query 93 as Example



- |                                    |   |
|------------------------------------|---|
| [1] : "Raffles Hotel in Singapore" | $d_1$ : "Stamford Raffles – Wikipedia, the free encyclopedia" [2] |
| [2] : "Sir Stamford Raffles"       | $d_2$ : "Fundraiser Raffle Ideas" [3, 5]                          |
| [3] : "organizing a raffle"        | $d_3$ : "Luxury Hotel Guide   Raffles Hotels" [1, 4]              |
| [4] : "the Raffles hotel in Dubai" | $d_4$ : "National Corvette Museum – Corvette Raffles" [5]         |
| [5] : "car raffles"                | $d_5$ : "Raffles Hotels and Resorts" [1, 4]                       |

# How it works?

## Using Query 93 as Example

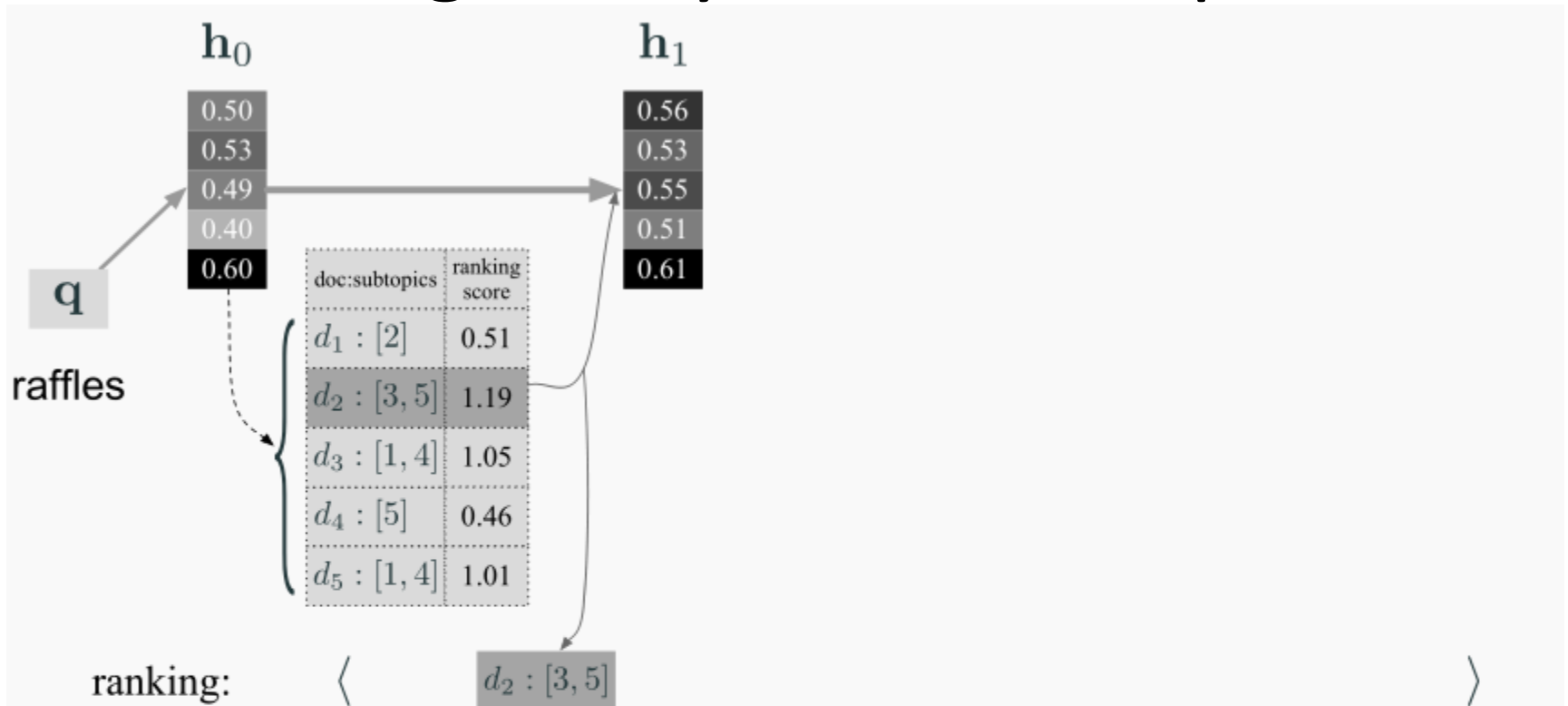


[1] : “Raffles Hotel in Singapore”  
[2] : “Sir Stamford Raffles”  
[3] : “organizing a raffle”  
[4] : “the Raffles hotel in Dubai”  
[5] : “car raffles”

*d*<sub>1</sub> : “Stamford Raffles – Wikipedia, the free encyclopedia” [2]  
*d*<sub>2</sub> : “Fundraiser Raffle Ideas” [3, 5]  
*d*<sub>3</sub> : “Luxury Hotel Guide | Raffles Hotels” [1, 4]  
*d*<sub>4</sub> : “National Corvette Museum – Corvette Raffles” [5]  
*d*<sub>5</sub> : “Raffles Hotels and Resorts” [1, 4]

# How it works?

## Using Query 93 as Example



[1] : “Raffles Hotel in Singapore”

[2] : “Sir Stamford Raffles”

[3] : “organizing a raffle”

[4] : “the Raffles hotel in Dubai”

[5] : “car raffles”

$d_1$  : “Stamford Raffles – Wikipedia, the free encyclopedia” [2]

$d_2$  : “Fundraiser Raffle Ideas” [3, 5]

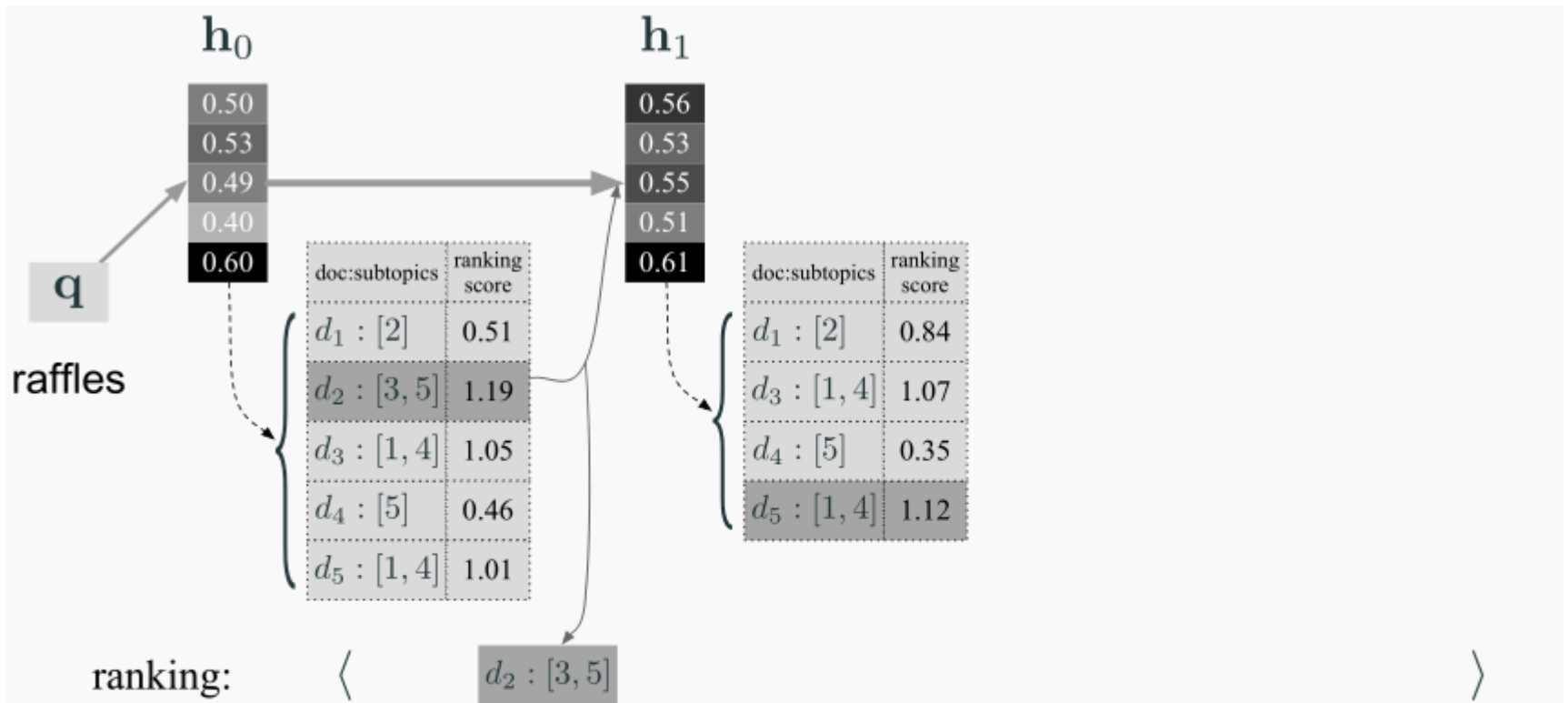
$d_3$  : “Luxury Hotel Guide | Raffles Hotels” [1, 4]

$d_4$  : “National Corvette Museum – Corvette Raffles” [5]

$d_5$  : “Raffles Hotels and Resorts” [1, 4]

# How it works?

## Using Query 93 as Example



[1] : "Raffles Hotel in Singapore"

[2] : "Sir Stamford Raffles"

[3] : "organizing a raffle"

[4] : "the Raffles hotel in Dubai"

[5] : "car raffles"

$d_1$  : "Stamford Raffles – Wikipedia, the free encyclopedia" [2]

$d_2$  : "Fundraiser Raffle Ideas" [3, 5]

$d_3$  : "Luxury Hotel Guide | Raffles Hotels" [1, 4]

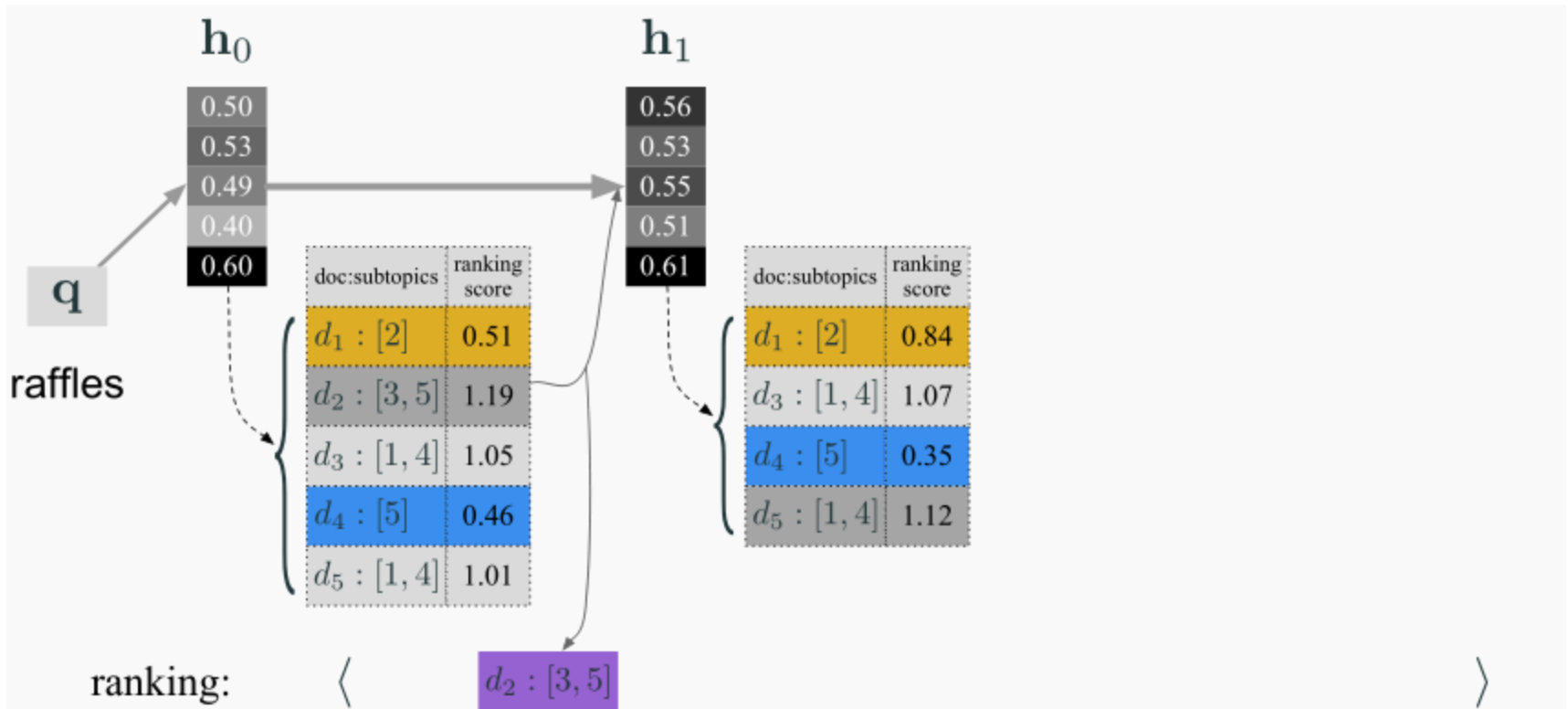
$d_4$  : "National Corvette Museum – Corvette Raffles" [5]

$d_5$  : "Raffles Hotels and Resorts" [1, 4]



# How it works?

## Using Query 93 as Example



[1] : "Raffles Hotel in Singapore"

[2] : "Sir Stamford Raffles"

[3] : "organizing a raffle"

[4] : "the Raffles hotel in Dubai"

[5] : "car raffles"

$d_1$  : "Stamford Raffles – Wikipedia, the free encyclopedia" [2]

$d_2$  : "Fundraiser Raffle Ideas" [3, 5]

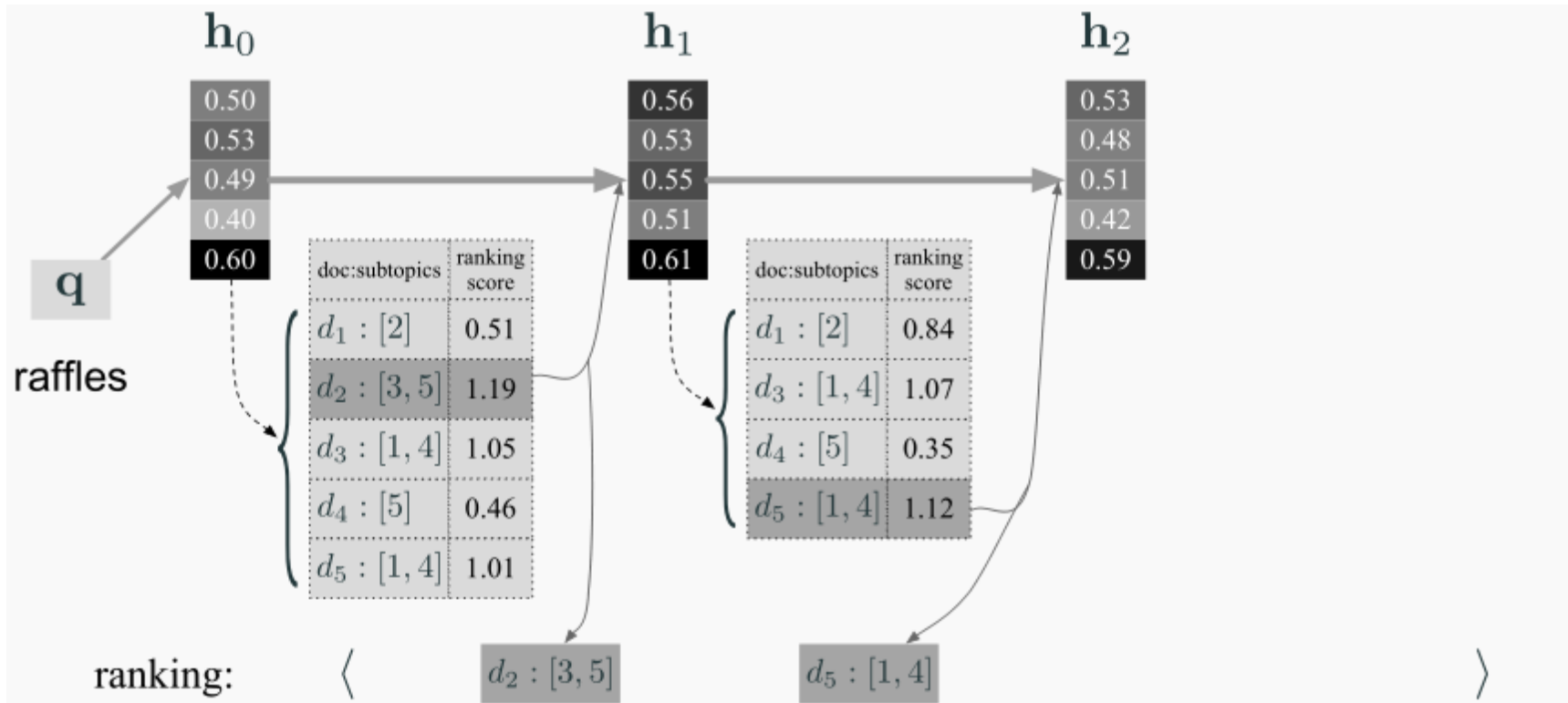
$d_3$  : "Luxury Hotel Guide | Raffles Hotels" [1, 4]

$d_4$  : "National Corvette Museum – Corvette Raffles" [5]

$d_5$  : "Raffles Hotels and Resorts" [1, 4]

# How it works?

## Using Query 93 as Example

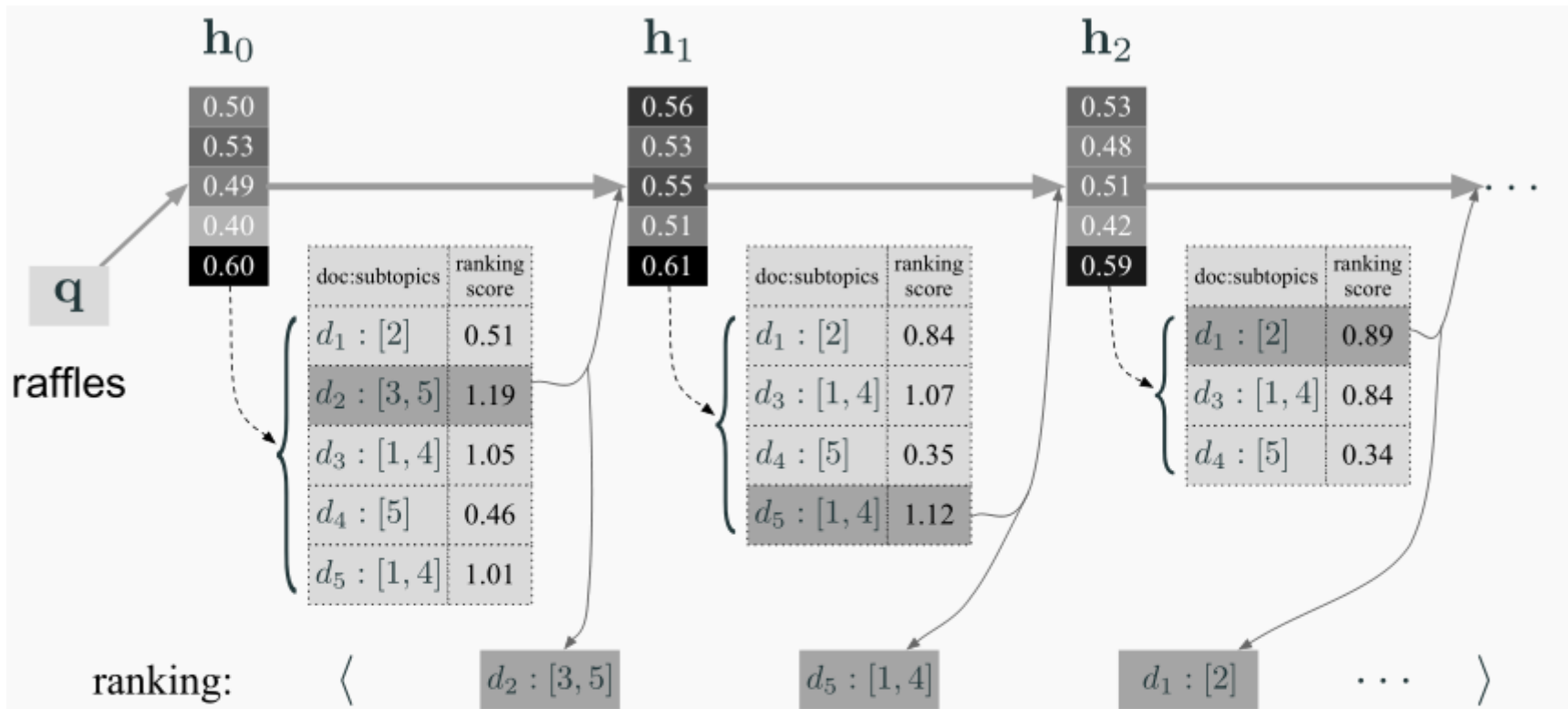


- [1] : "Raffles Hotel in Singapore"
- [2] : "Sir Stamford Raffles"
- [3] : "organizing a raffle"
- [4] : "the Raffles hotel in Dubai"
- [5] : "car raffles"

- $d_1$  : "Stamford Raffles – Wikipedia, the free encyclopedia" [2]
- $d_2$  : "Fundraiser Raffle Ideas" [3, 5]
- $d_3$  : "Luxury Hotel Guide | Raffles Hotels" [1, 4]
- $d_4$  : "National Corvette Museum – Corvette Raffles" [5]
- $d_5$  : "Raffles Hotels and Resorts" [1, 4]

# How it works?

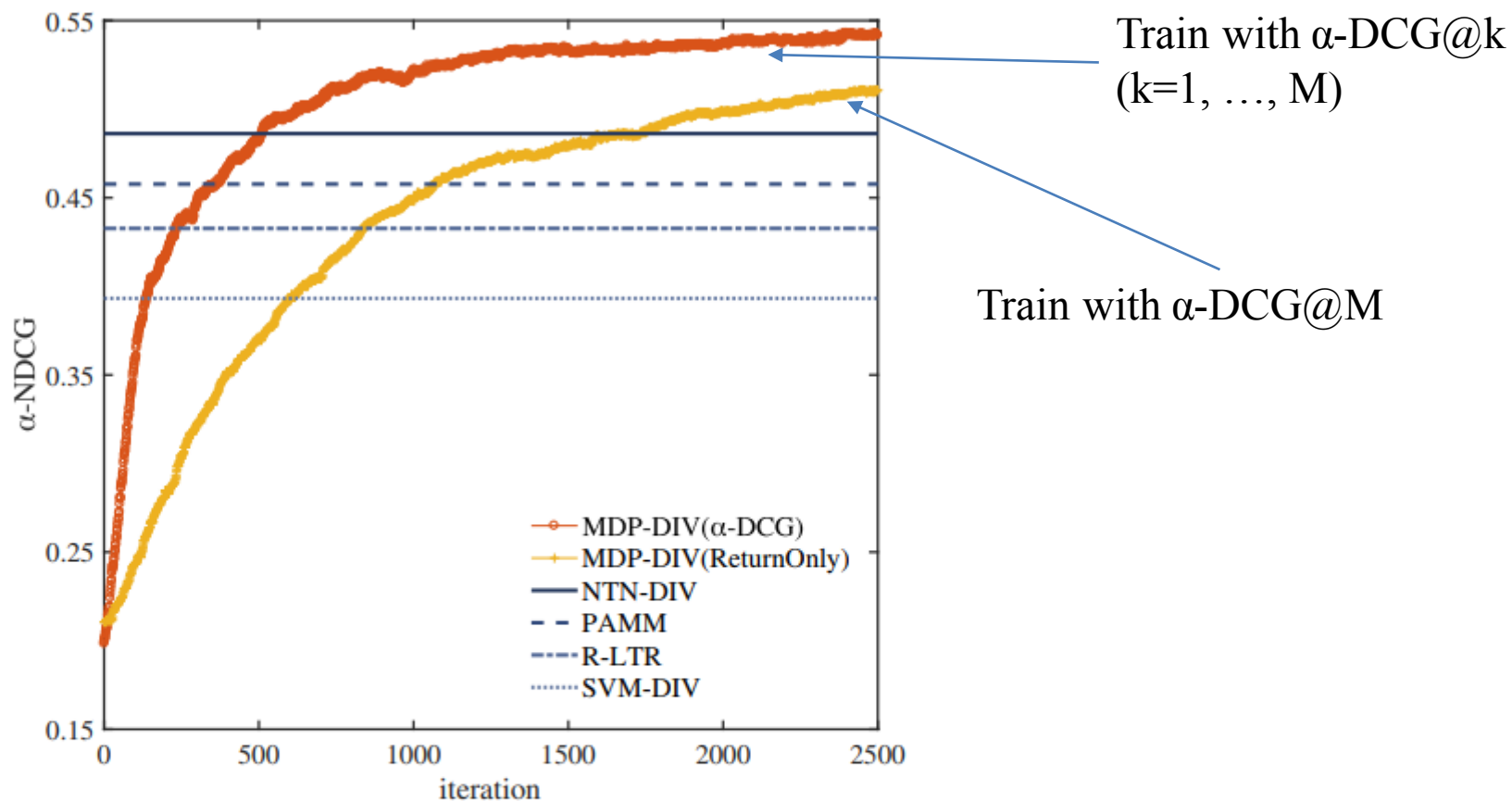
## Using Query 93 as Example



- [1] : "Raffles Hotel in Singapore"
- [2] : "Sir Stamford Raffles"
- [3] : "organizing a raffle"
- [4] : "the Raffles hotel in Dubai"
- [5] : "car raffles"

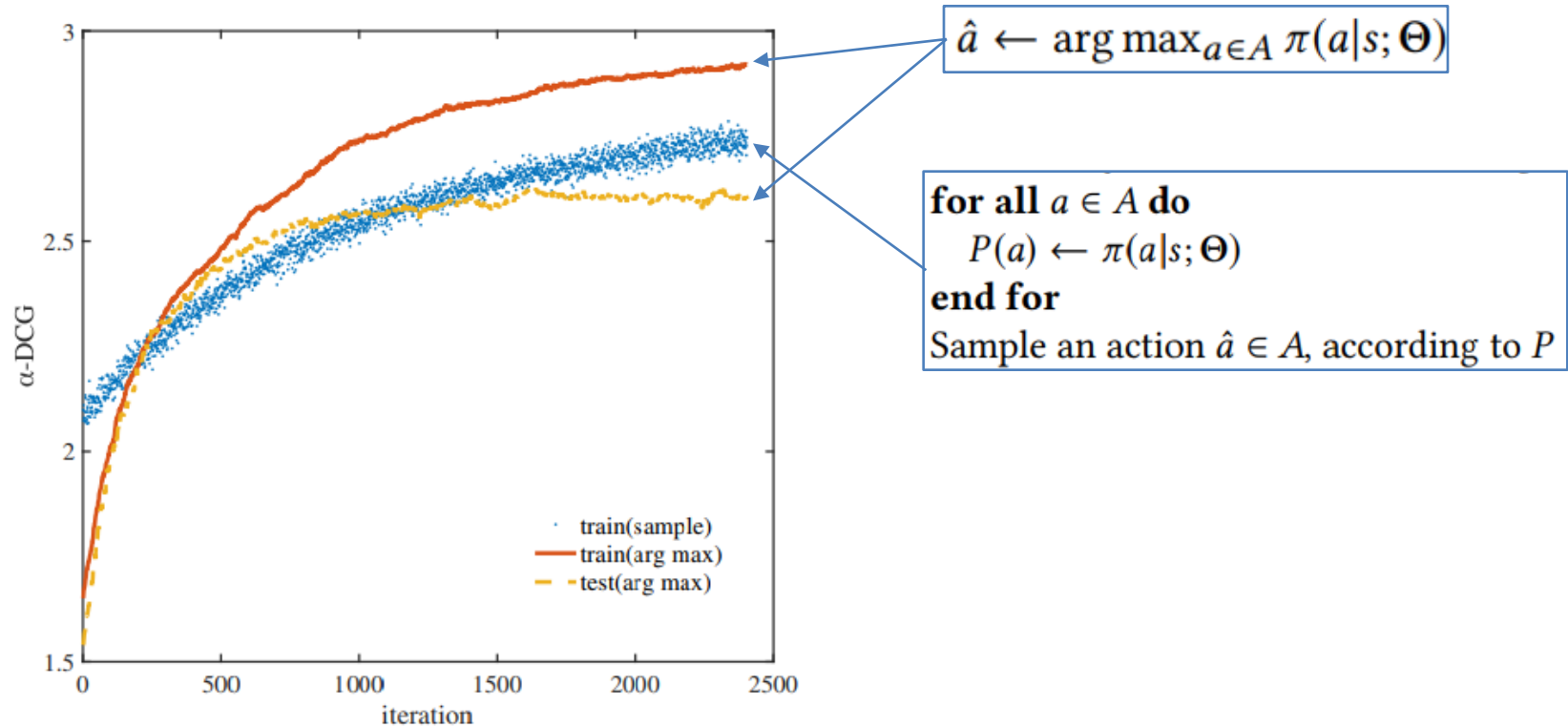
- $d_1$  : "Stamford Raffles – Wikipedia, the free encyclopedia" [2]
- $d_2$  : "Fundraiser Raffle Ideas" [3, 5]
- $d_3$  : "Luxury Hotel Guide | Raffles Hotels" [1, 4]
- $d_4$  : "National Corvette Museum – Corvette Raffles" [5]
- $d_5$  : "Raffles Hotels and Resorts" [1, 4]

# Using Immediate Rewards in Training



**Figure 4: The performance curves on the test data for MDP-DIV( $\alpha$ -DCG), and the modified MDP-DIV( $\alpha$ -DCG) in which the training only involves the long-term returns. The performances of other baselines are shown as horizontal lines.**

# Convergence and Online Ranking Criterion



**Figure 5: The performance curves in terms of  $\alpha$ -DCG on the training data (“train(arg max)”) and the test data (“test(arg max)”). The average performances of the sampled rankings over all training queries are also shown (“train(sample”).**

# Advantages

- Unified criterion (additional utility user can perceive) for selecting documents at each iteration
- End-to-end learning of the diverse ranking model
  - No need of handcrafted features
- Utilizes both the immediate rewards and the long-term returns as the supervision information during training

# Example 2: Relevance Ranking as an MDP

Wei Zeng, Jun Xu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. Reinforcement Learning to Rank with Markov Decision Process. Proceedings of SIGIR 2017, pp. 945-948.

# Modeling Relevance Ranking with MDP

$\mathbf{x}_{m(a_t)}$ : query-doc relevance features

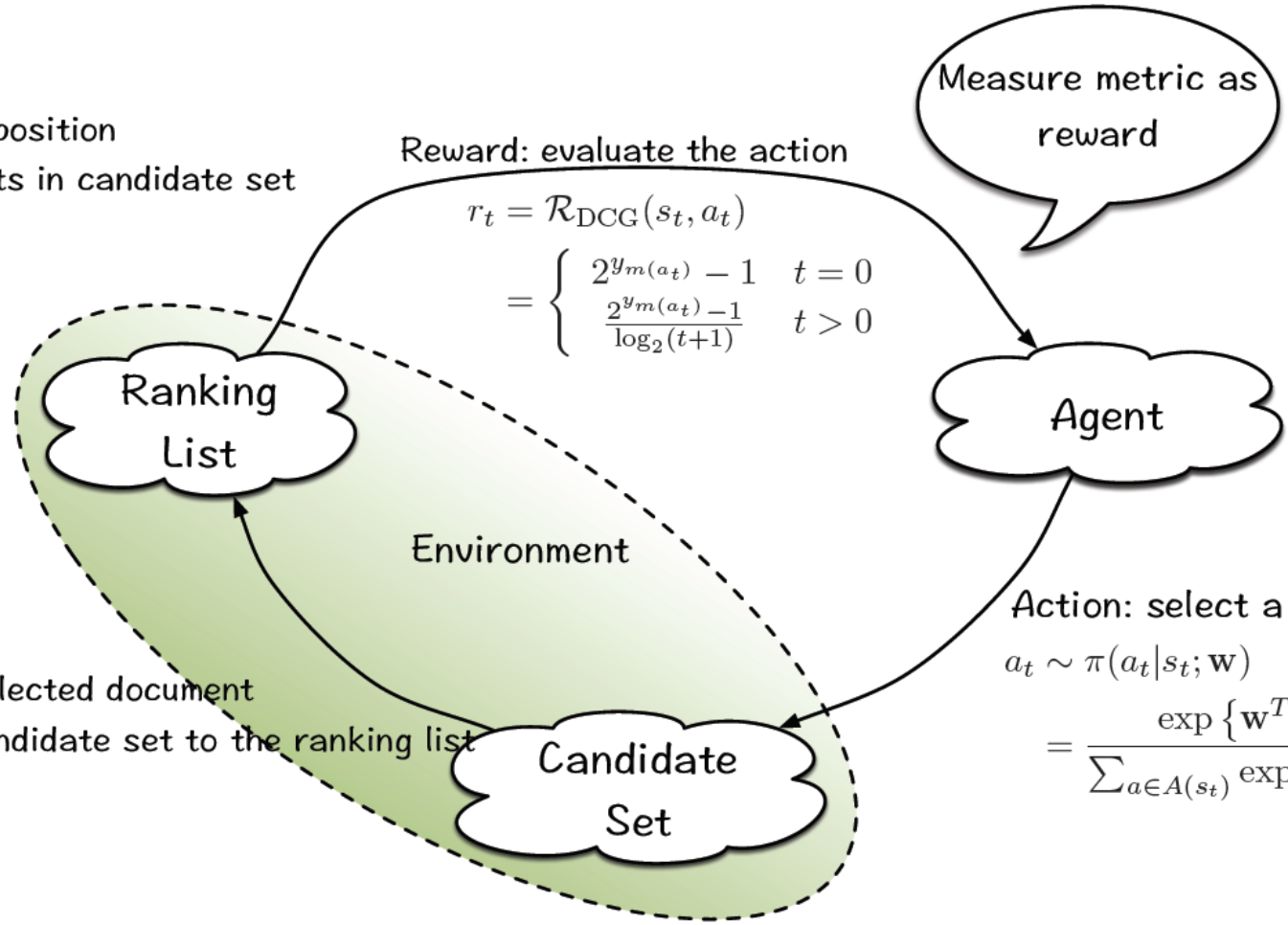
MDP factors	Corresponding relevance ranking factors
Time steps	The ranking positions
State	$s_t = [t, X_t]$
Policy	$\pi(a_t   s_t = [t, X_t]) = \frac{\exp\{\mathbf{w}^T \mathbf{x}_{m(a_t)}\}}{\sum_{a \in A(t)} \exp\{\mathbf{w}^T \mathbf{x}_{m(a)}\}}$
Action	Selecting a doc and placing it to current position
Reward	Based on evaluation measure DCG: $R = \begin{cases} 2^{y(a_t)} - 1 & t = 0 \\ \frac{2^{y(a_t)} - 1}{\log_2(t+1)} & t > 0 \end{cases}$
State Transition	$s_{t+1} = T(s_t = [t, X_t], a_t) = [t + 1, X_t \setminus \{\mathbf{x}_{m(a_t)}\}, ]$



# The Ranking Process

State:  $s_t = [t, X_t]$

- 1. the ranking position
- 2. the documents in candidate set



Reward: evaluate the action

$$r_t = \mathcal{R}_{\text{DCG}}(s_t, a_t) = \begin{cases} 2^{y_{m(a_t)}} - 1 & t = 0 \\ \frac{2^{y_{m(a_t)}} - 1}{\log_2(t+1)} & t > 0 \end{cases}$$

Measure metric as reward

Action: select a document

$$a_t \sim \pi(a_t | s_t; \mathbf{w}) = \frac{\exp\{\mathbf{w}^T \mathbf{x}_{m(a_t)}\}}{\sum_{a \in A(s_t)} \exp\{\mathbf{w}^T \mathbf{x}_{m(a)}\}}$$

Move the selected document from the candidate set to the ranking list

# Learning with Policy Gradient

---

## Algorithm 1 MDPRank learning

---

**Input:** Labeled training set  $D = \{(q^{(n)}, X^{(n)}, Y^{(n)})\}_{n=1}^N$ , learning rate  $\eta$ , discount factor  $\gamma$ , and reward function  $R$

**Output:**  $w$

- 1: Initialize  $w \leftarrow$  random values
- 2: **repeat**
- 3:    $\Delta w = 0$
- 4:   **for all**  $(q, X, Y) \in D$  **do**
- 5:      $(s_0, a_0, r_1, \dots, s_{M-1}, a_{M-1}, r_M)$   $\leftarrow$  {Algorithm (2), and  $M = \lfloor |X| \rfloor$ }
- 6:     **for**  $t = 0$  **to**  $M - 1$  **do**
- 7:        $G_t \leftarrow \sum_{k=1}^{M-t} \gamma^{k-1} r_{t+k}$
- 8:        $\Delta w \leftarrow \Delta w + \gamma^t G_t \nabla_w \pi(a_t | s_t; w)$
- 9:     **end for**
- 10:   **end for**
- 11:    $w \leftarrow w + \eta \Delta w$
- 12: **until** converge
- 13: **return**  $w$

---

## Algorithm 2 SampleAnEpisode

---

**Input:** Parameters  $w, q, X, Y$ , and  $\mathcal{R}$

**Output:** An episode

- 1: Initialize  $s_0 \leftarrow [0, X]$ ,  $M \leftarrow |X|$ , and episode  $E \leftarrow \emptyset$
  - 2: **for**  $t = 0$  **to**  $M - 1$  **do**
  - 3:   Sample an action  $a_t \in A(s_t) \sim \pi(a_t | s_t; w)$  {Equation (2)}
  - 4:    $r_{t+1} \leftarrow \mathcal{R}(s_t, a_t)$  {Equation (1), calculation on the basis of  $Y$ }
  - 5:   Append  $(s_t, a_t, r_{t+1})$  at the end of  $E$
  - 6:   State transition  $s_{t+1} \leftarrow [t + 1, X \setminus \{\mathbf{x}_{m(a_t)}\}]$
  - 7: **end for**
  - 8: **return**  $E = (s_0, a_0, r_1, \dots, s_{M-1}, a_{M-1}, r_M)$
-

# Experimental Results

Result on MQ2007 Dataset

Method	NDCG@1	NDCG@3	NDCG@5	NDCG@10
RankSVM	0.4045	0.4019	0.4072	0.4383
ListNet	0.4002	0.4091	0.4170	<b>0.4440</b>
AdaRank-MAP	0.3821	0.3984	0.4071	0.4335
AdaRank-NDCG	0.3876	0.4044	0.4102	0.4369
SVMMAP	0.3853	0.3899	0.3983	0.4187
MDPRank	<b>0.4061</b>	<b>0.4101</b>	<b>0.4171</b>	0.4416
MDPRank(return only)	0.4033	0.4059	0.4113	0.4350

Result on OHSUMED Dataset

Method	NDCG@1	NDCG@3	NDCG@5	NDCG@10
RankSVM	0.4958	0.4207	0.4164	0.4140
ListNet	0.5326	0.4732	0.4432	0.4410
AdaRank-MAP	0.5388	0.4682	0.4613	0.4429
AdaRank-NDCG	0.5330	0.4790	0.4673	0.4496
SVMMAP	0.5229	0.4663	0.4516	0.4319
MDPRank	<b>0.5925</b>	<b>0.4992</b>	<b>0.4909</b>	<b>0.4587</b>
MDPRank(return only)	0.5363	0.4885	0.4694	0.4591

- MDPRank is better because
  - Utilize the IR measures calculated at all the ranking positions as supervision information for training
  - Directly optimizes the IR measure on the training data without any approximation or upper bounding

# Outline

- Background: learning to rank for IR
- Reinforcement learning to rank
- **Summary**

# Summary

- Reinforcement learning to rank
  - Ranking as sequential decision making
  - Adapting MDP for the task
  - Learning with policy gradient
- Two examples
  - Diverse ranking
  - Relevance ranking

BDAS

Create Job Upload Program Upload Data Notebook

Program Data Job

examples

- ✓ 【实例】 Titanic Demo
- ✓ 【实例】 Twitter Demo
- ✓ 【实例】 分布式 移动垃圾短
- ✓ 【实例】 分布式 鲍鱼年龄
- ✓ 【实例】 单机 GBDT
- ✓ 【实例】 微博垃圾信息分类
- ✓ 【实例】 混合 移动垃圾短
- ✓ 【实例】 财新网新闻推荐
- ✓ 【实例】 CRF单机分词
- ✓ 【实例】 W1系统\_热词发现
- ✓ 【实例】 W1系统\_群发热点
- ✓ 【实例】 分布式 CART(分
- ✓ 【实例】 分布式 CART(回
- ✓ 【实例】 分布式 Feature\_Ir
- ✓ 【实例】 分布式 Feature\_Ir
- ✓ 【实例】 分布式 GBDT
- ✓ 【实例】 分布式 GBRT
- ✓ 【实例】 分布式 LDA
- ✓ 【实例】 分布式 LogisticRe
- ✓ 【实例】 分布式 MinMax\_S
- ✓ 【实例】 分布式 MinMax\_S
- ✓ 【实例】 分布式 NMF
- ✓ 【实例】 分布式 RandomFi
- ✓ 【实例】 分布式 RandomFi
- ✓ 【实例】 分布式 Row\_Norr
- ✓ 【实例】 分布式 SVDFeatu
- ✓ 【实例】 分布式 TF-IDF
- ✓ 【实例】 分布式 Word\_Indc
- ✓ 【实例】 分布式 格式转换
- ✓ 【实例】 单机 CART(分类)
- ✓ 【实例】 单机 CART(回归)
- ✓ 【实例】 单机 GBRT

垃圾短信分类

stopwords

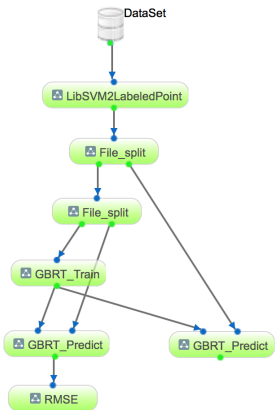
移动垃圾短信测试集

BinaryClassifier

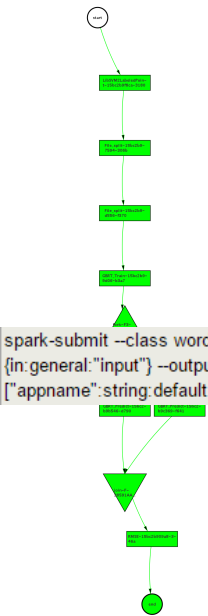
Run History Submit Clear Clone Stop Refresh

# Easy Machine Learning Project

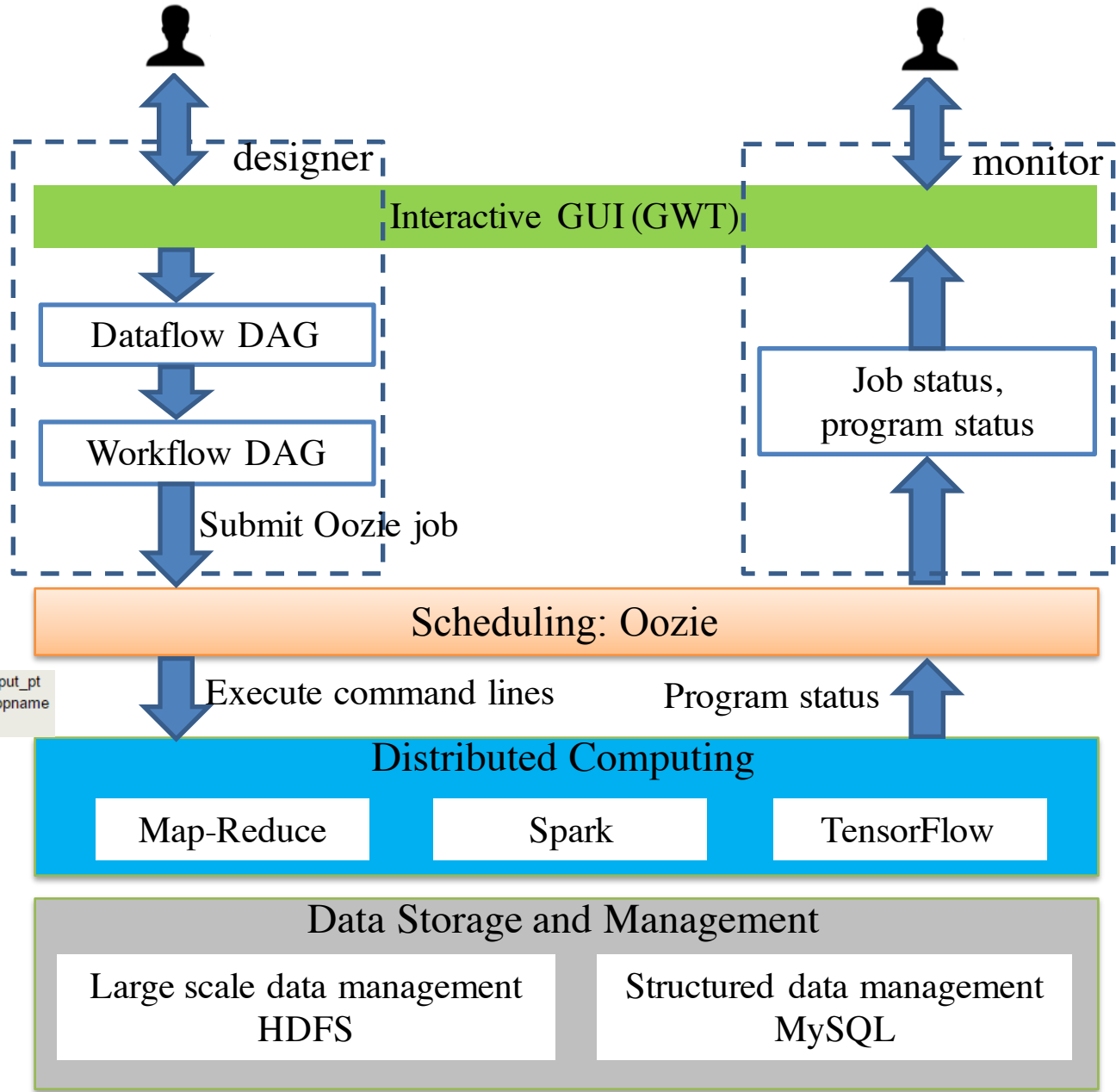
# Design of Easy Machine Learning



Node: program / data  
Edge: dataflow

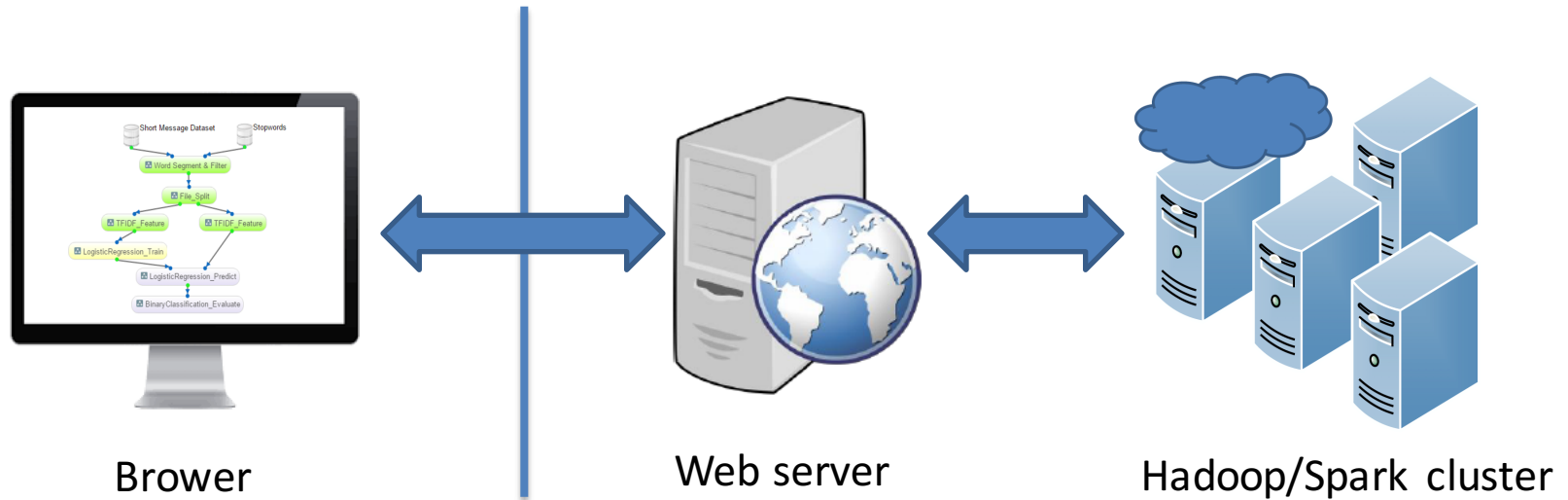


Node: program / start / end / fork / join  
Edge: dependency



# Deploy as Web Service

<http://159.226.40.104:18080/dev>



- Advantages

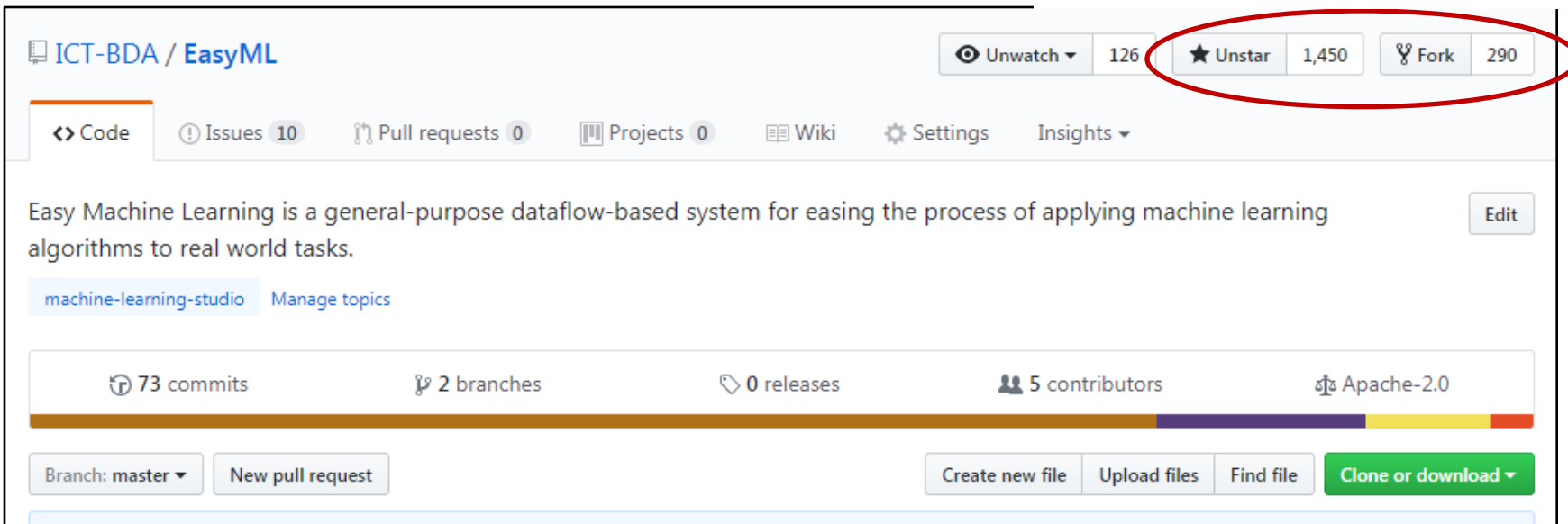
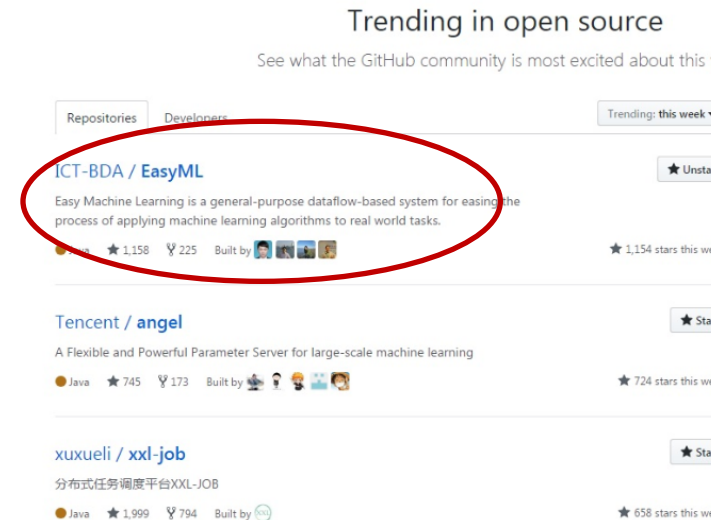
- **Sharing:** share data/programs/tasks among users
- **Collaborating:** working together for one task
- **Mobility:** accessing with web browsers anywhere
- **Open:** ETL for data import/export; can run third-party programs



# Source Shared at Github

<https://github.com/ICT-BDA/EasyML>

- Top 1 Java project at Github trending for one week
- 1400 + stars and ~300 forks
- CIKM 2016 best demo candidate [Guo et al., CIKM '16]



# Thanks!

[junxu@ict.ac.cn](mailto:junxu@ict.ac.cn)

<http://www.bigdatalab.ac.cn/~junxu>