wsdm 2021
THE 14TH ACM INTERNATIONAL CONFERENCE ON WEB SEARCH AND DATA MINING

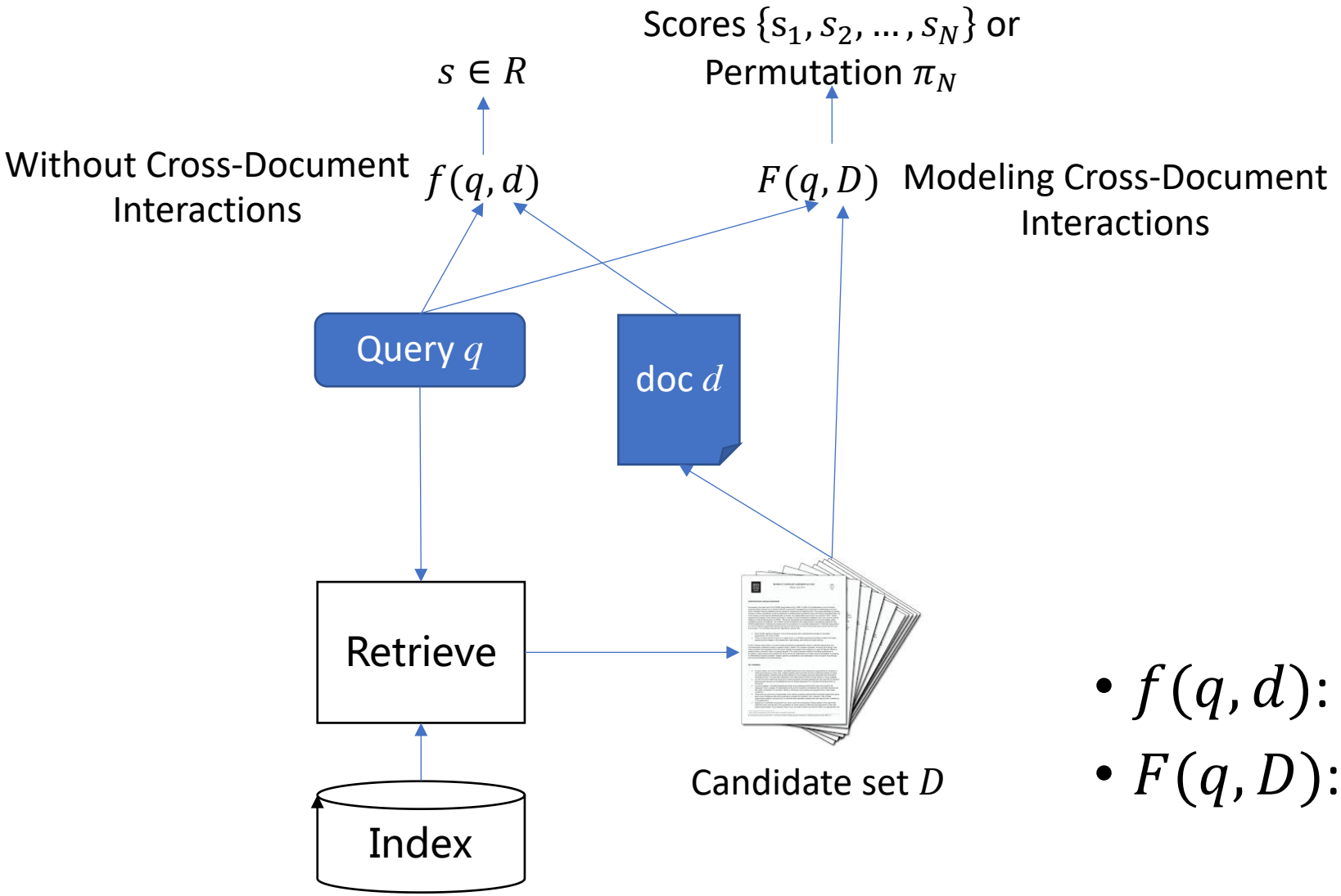
Tutorial
Morning (March, 8)
9:00-12:00, GMT+2

Beyond Probability Ranking Principle: Modeling the Dependencies among Documents

Liang Pang[1], Qingyao Ai[2], Jun Xu[3]

1. CAS Key Lab of Network Data Science and Technology
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
2. The University of Utah, USA
3. Renmin University of China, Beijing, China

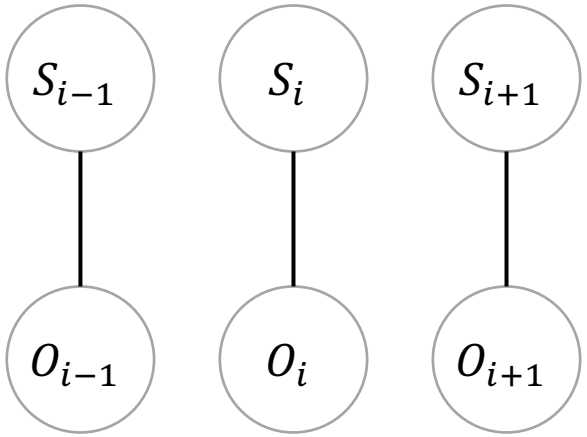

Modeling the Dependency in IR



- $f(q,d)$: independent model
- $F(q,D)$: dependent model

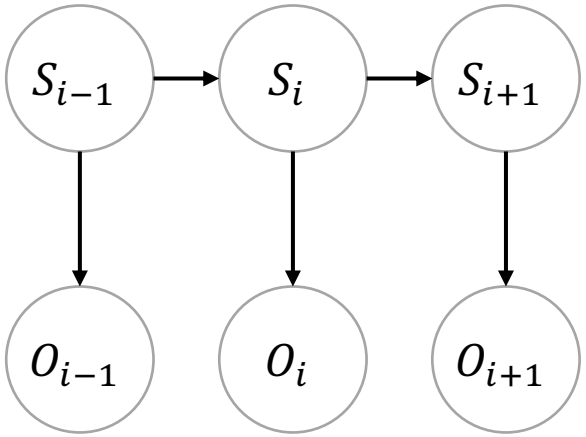
Modeling the Sequential Dependency: Experiences from Sequence Prediction

No Dependency



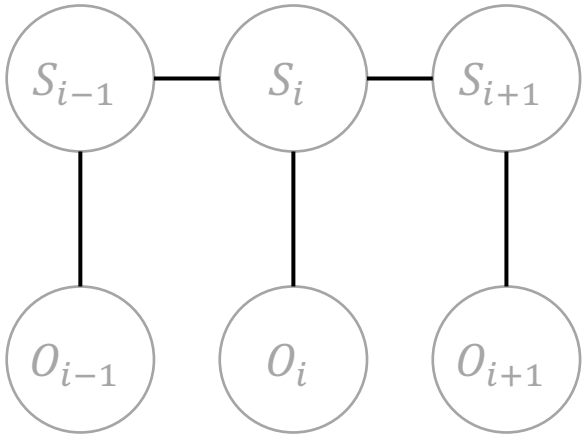
Logistic Regression

Sequential Dependency



HMM

Global Dependency



CRF

4. Ranking with Sequential Dependency

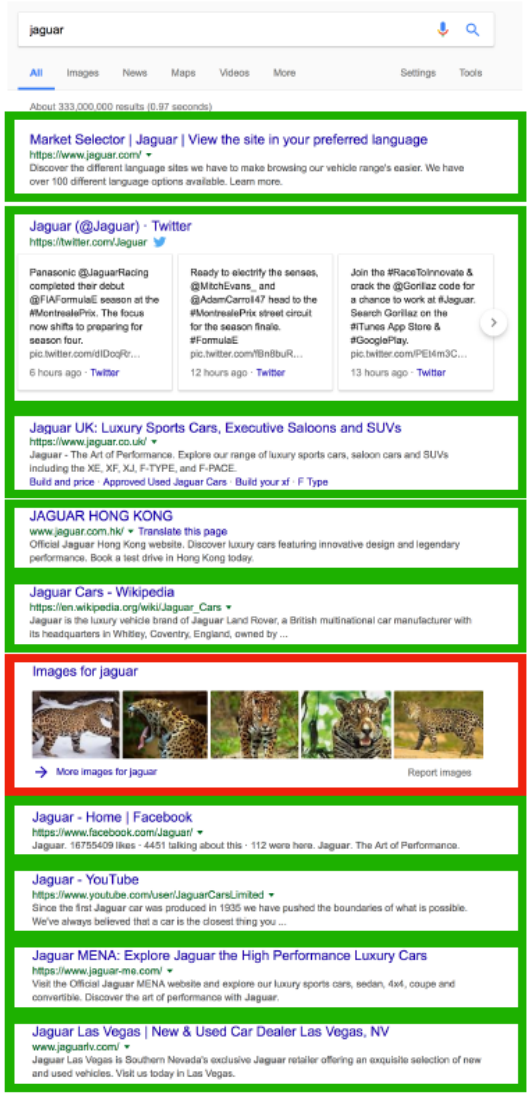
4.1 Heuristic Sequential Ranking Models

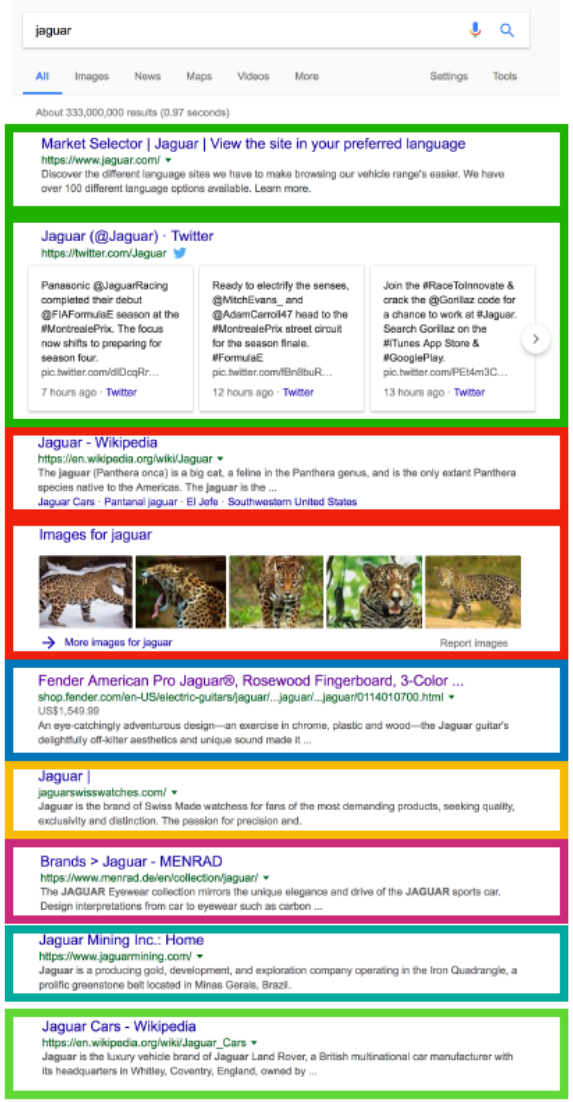
4.2 Learning Sequential Ranking Models

4.3 Challenges

Search Result Diversification

Query: jaguar





Luxury car

Animal

Electric

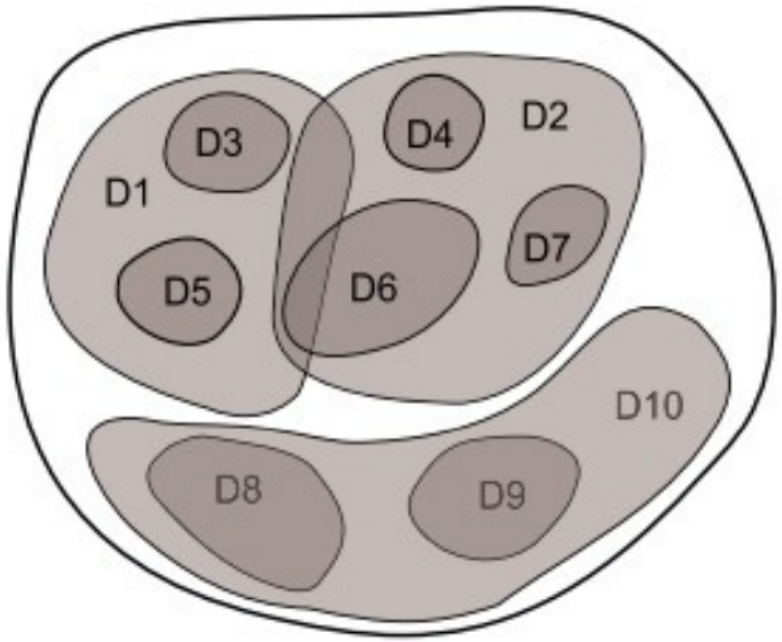
Swiss

Eyewear

Mining Inc.

- Different user needs
  - Ambiguous queries , e.g., Apple, Jaguar, Band
  - Multi-faceted needs, e.g., Britney spears (news, videos, photos…)
- Information redundancy
  - Many duplicate or similar results

Greedy approach to diverse ranking



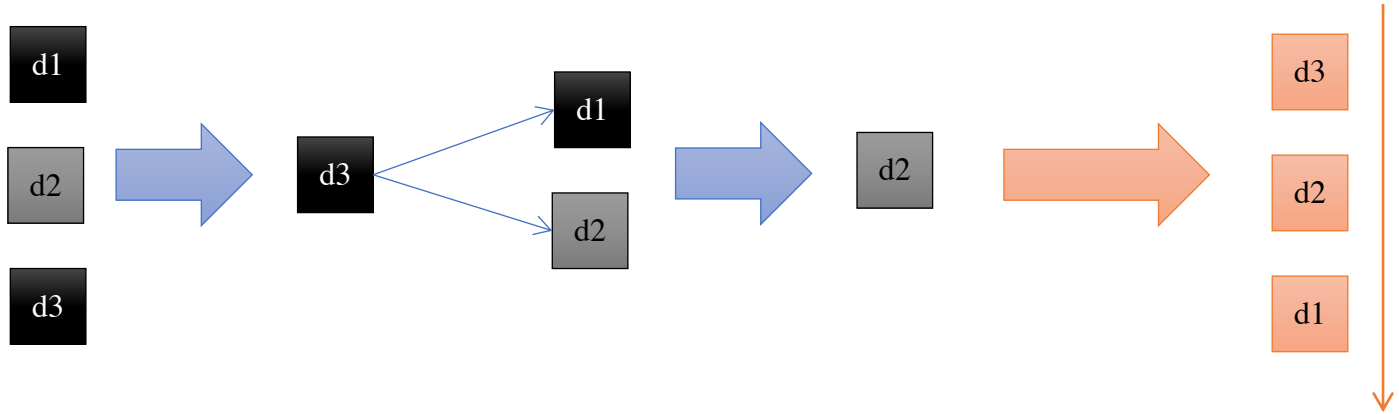
Select k documents to cover more subtopics

subset selection

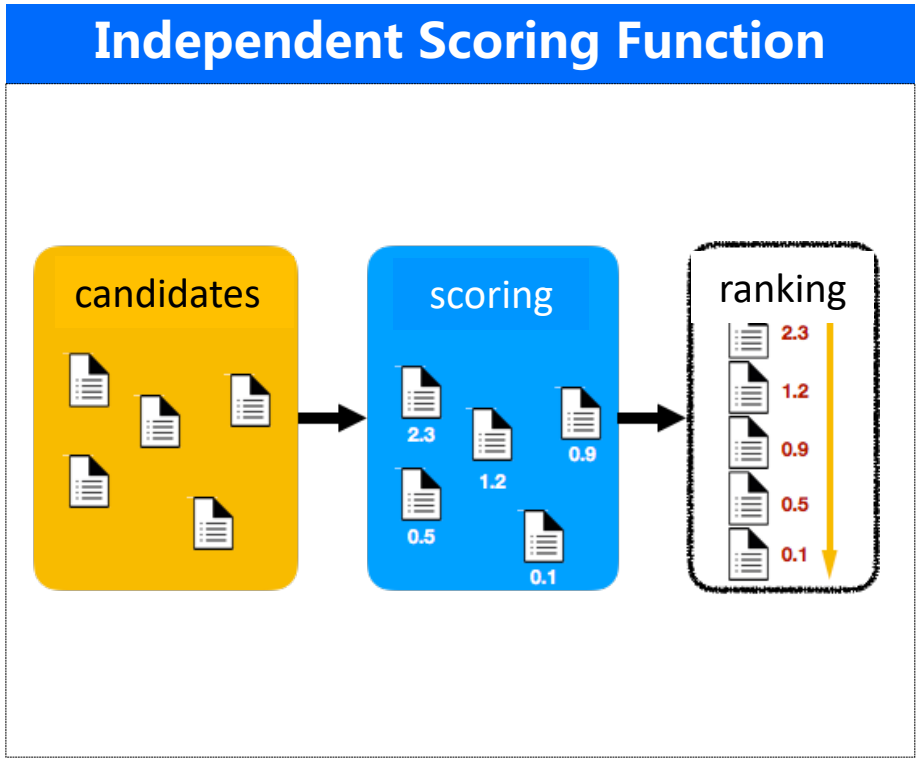
Diversify$(q, \mathcal{R}_q, \tau)$

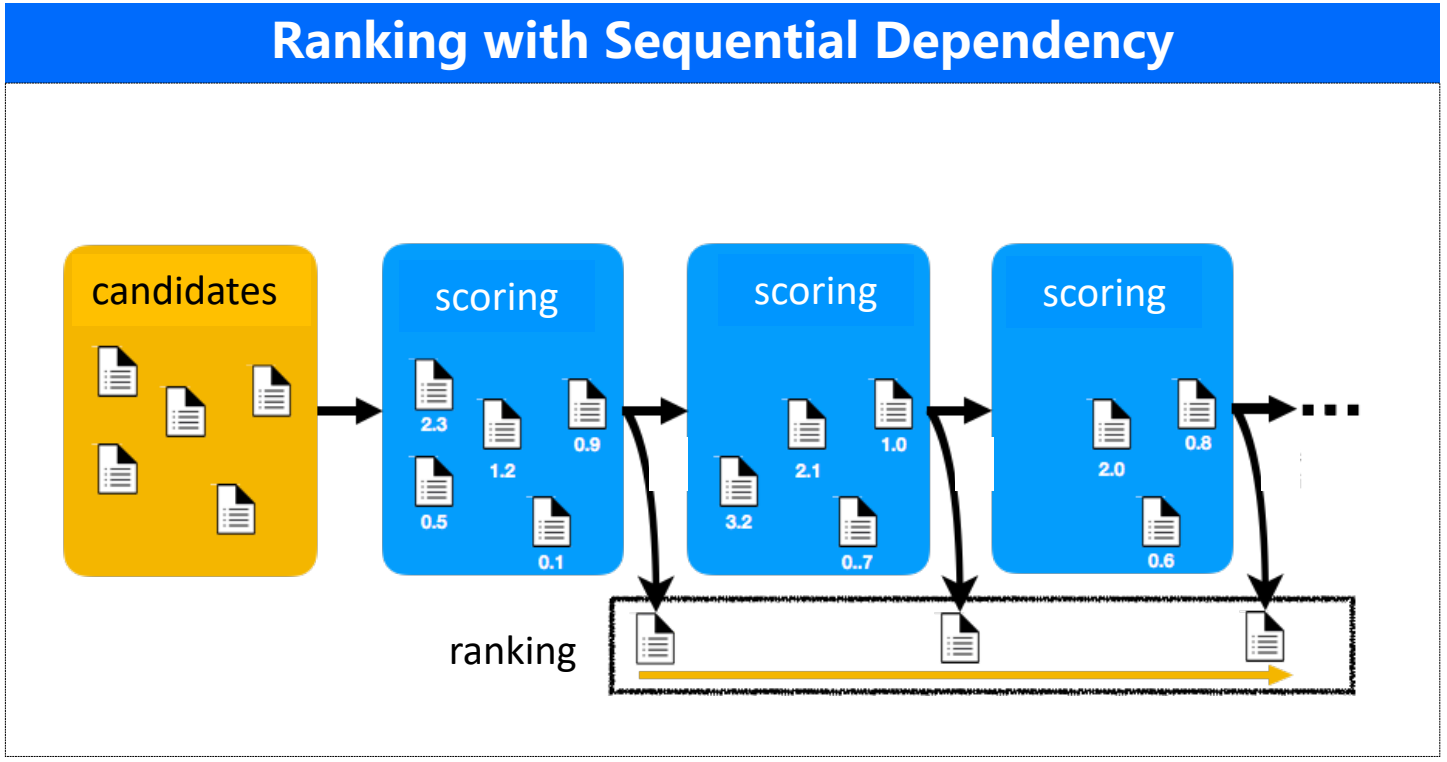
1 $\mathcal{D}_q \leftarrow \emptyset$
2 **while** $|\mathcal{D}_q| < \tau$ **do**
3 $\quad d^* \leftarrow \arg\max_{d \in \mathcal{R}_q \setminus \mathcal{D}_q} f(q, d, \mathcal{D}_q)$
4 $\quad \mathcal{R}_q \leftarrow \mathcal{R}_q \setminus \{d^*\}$
5 $\quad \mathcal{D}_q \leftarrow \mathcal{D}_q \cup \{d^*\}$
6 **end while**
7 **return** $\mathcal{D}_q$

Greedy Approach to Ranking



Greedy Approach Models the Dependency Sequentially





Obeys PRP

Beyond PRP

Pre-defined $f(q, d, D_q)$: MMR (Carbonell and Goldstein, 1998)

- Maximal marginal relevance
  - An implicit diversification model
- Key: defining $f(q, d, D_q)$

Marginal relevance of d given the query q and selected documents D_q

Prefer dissimilar documents

$$f_{\text{MMR}}(q, d, \mathcal{D}_q) = \lambda f_1(q, d) - (1 - \lambda) \max_{d_j \in \mathcal{D}_q} f_2(d, d_j)$$

Relevance between q and d

aggregation

Similarity between d and a selected document d_j
(dependency between target d and d_j)

Diversify$(q, \mathcal{R}_q, \tau)$

1 $\mathcal{D}_q \leftarrow \emptyset$
2 **while** $|\mathcal{D}_q| < \tau$ **do**
3 $\quad d^* \leftarrow \arg\max_{d \in \mathcal{R}_q \setminus \mathcal{D}_q} f(q, d, \mathcal{D}_q)$
4 $\quad \mathcal{R}_q \leftarrow \mathcal{R}_q \setminus \{d^*\}$
5 $\quad \mathcal{D}_q \leftarrow \mathcal{D}_q \cup \{d^*\}$
6 **end while**
7 **return** $\mathcal{D}_q$

Pre-defined $f(q, d, D_q)$

- Implicit Diversification
  - Risk minimization RM (Zhai et al., 2008)

$$f_{\text{RM}}(q, d, \mathcal{D}_q) = f_1(\theta_q, \theta_d)(1 - \lambda - f_2(\theta_d, \theta_{\mathcal{D}_q}))$$

- Explicit Diversification
  - Intent-aware selection (IA-Select) (Agrawal et al., 2009)

$$f_{\text{IA-Select}}(q, d, \mathcal{D}_q) = \sum_{c \in \mathcal{T}} f(c|q, \mathcal{D}_q)\, f(d|q, c)$$

  - Proportionality model (PM-2) (Dang and Croft, 2012)

$$f_{\text{PM-2}}(q, d, \mathcal{D}_q) = \sum_{s \in \mathcal{S}_q} b_s\, \zeta(s|q)\, p(d|s)$$

  - Explicit query aspect diversification (xQuAD) (Santos et al., 2010)

$$f_{\text{xQuAD}}(q, d, \mathcal{D}_q) = (1 - \lambda)\, \mathrm{P}(d|q) + \lambda \sum_{q_i \in Q} \left[\mathrm{P}(q_i|q)\, \mathrm{P}(d|q_i) \prod_{d_j \in S} (1 - \mathrm{P}(d_j|q_i)) \right]$$

Diversify$(q, \mathcal{R}_q, \tau)$

1 $\mathcal{D}_q \leftarrow \emptyset$
2 **while** $|\mathcal{D}_q| < \tau$ **do**
3 $\quad d^* \leftarrow \arg\max_{d \in \mathcal{R}_q \setminus \mathcal{D}_q} f(q, d, \mathcal{D}_q)$
4 $\quad \mathcal{R}_q \leftarrow \mathcal{R}_q \setminus \{d^*\}$
5 $\quad \mathcal{D}_q \leftarrow \mathcal{D}_q \cup \{d^*\}$
6 **end while**
7 **return** $\mathcal{D}_q$

4. Ranking with Sequential Dependency

4.1 Heuristic Sequential Ranking Models

4.2 Learning Sequential Ranking Models

4.3 Challenges

Learning $f(q, d, D_q)$: R-LTR (Zhu et al., 2014)

- Learning implicit diversification model $f(q, d, D_q)$ from labeled data
- A learnable extension of MMR

$$f_{\text{MMR}}(q, d, \mathcal{D}_q) = \lambda f_1(q, d) - (1 - \lambda) \max_{d_j \in \mathcal{D}_q} f_2(d, d_j)$$

$$f_{\text{R-LTR}}(q, d, D_q) = \langle \mathbf{w}_r, \boldsymbol{\phi}(q, d) \rangle + \left\langle \mathbf{w}_d, \mathbf{h}_{D_q}(R_d) \right\rangle$$

Features for describing the relevance (traditional learning to rank features)

Aggregated features for describing novelty (dependency) of d given the selected documents in D_q

Diversify$(q, \mathcal{R}_q, \tau)$

1 $\mathcal{D}_q \leftarrow \emptyset$
2 **while** $|\mathcal{D}_q| < \tau$ **do**
3 $\quad d^* \leftarrow \arg\max_{d \in \mathcal{R}_q \setminus \mathcal{D}_q} f(q, d, \mathcal{D}_q)$
4 $\quad \mathcal{R}_q \leftarrow \mathcal{R}_q \setminus \{d^*\}$
5 $\quad \mathcal{D}_q \leftarrow \mathcal{D}_q \cup \{d^*\}$
6 **end while**
7 **return** $\mathcal{D}_q$

$$\mathbf{h}_{D_q}(R_d) = \left\{ \max_{d_j \in D_q} R_{dj1}, \ldots, \max_{x_j \in D_q} R_{djK} \right\}$$

- Estimating the parameters $\mathbf{w}_r$ and $\mathbf{w}_d$ with list-wise MLE (*Plackett-Luce based Probability*) or Perceptron (PAMM) (Xia et al., 2015)

Dependency features of d and d_j $[R_{dj1}, R_{dj2}, \cdots, R_{djK}]$

- PLSA topic difference: $R_{dj1} = \sqrt{\sum_{k=1}^{m} \left(p(z_k|d) - p(z_k|d_j) \right)}$
- Text diversity: $R_{dj2} = 1 - \frac{\mathbf{d} \cdot \mathbf{d}_j}{\|\mathbf{d}\| \|\mathbf{d}_j\|}$

Learning $f(q, d, D_q)$

- SVM-DIV (Yue et al., 2008)
  - Focus on diversity only (subtopic coverage)
  - Defining f as a linear function on features

    $f_{\text{SVMDIV}}(q, d, D_q) = \langle \mathbf{w}^{\mathrm{T}}, \Psi(q, D_q \cup \{d\}) \rangle$

  - Estimating $\mathbf{w}$ based on structural SVMs

Algorithm 1 Greedy subset selection by maximizing weighted word coverage

Input: $\mathbf{w}$, $\mathbf{x}$
Initialize solution $\hat{\mathbf{y}} \leftarrow \emptyset$
for $k = 1, \ldots, K$ **do**
  $\hat{x} \leftarrow \text{argmax}_{x: x \notin \hat{\mathbf{y}}} \mathbf{w}^T \Psi(\mathbf{x}, \hat{\mathbf{y}} \cup \{d\})$
  $\hat{\mathbf{y}} \leftarrow \hat{\mathbf{y}} \cup \{\hat{x}\}$
end for
return $\hat{\mathbf{y}}$

- Determinantal Point Process (DPP) for diverse ranking (Chen et al., 2018)
  - Selecting item via greedy submodular maximization
  - f is defined based on the item kernel matrix

    $j = \arg\max_{i \in Z \setminus Y_g} f(Y_g \cup \{i\}) - f(Y_g)$

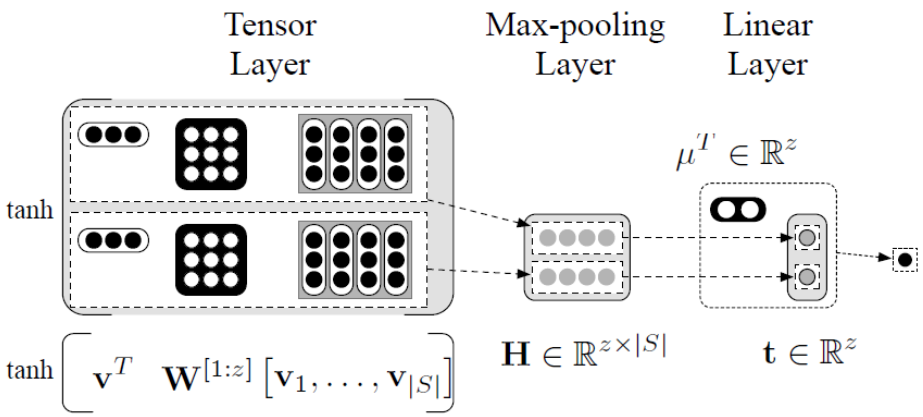
  See also (Wilhelm et al., 2018)

Algorithm 1 Fast Greedy MAP Inference

Input: Kernel $\mathbf{L}$, *stopping criteria*
Initialize: $\mathbf{c}_i = []$, $d_i^2 = \mathbf{L}_{ii}$, $j = \arg\max_{i \in Z} \log(d_i^2)$, $Y_g = \{j\}$
while *stopping criteria* not satisfied **do**
  for $i \in Z \setminus Y_g$ **do**
    $e_i = (\mathbf{L}_{ji} - \langle \mathbf{c}_j, \mathbf{c}_i \rangle)/d_j$
    $\mathbf{c}_i = [\mathbf{c}_i \quad e_i]$, $d_i^2 = d_i^2 - e_i^2$
  end for
  $j = \arg\max_{i \in Z \setminus Y_g} \log(d_i^2)$, $Y_g = Y_g \cup \{j\}$
end while
Return: Y_g

Learning deep $f(q, d, D_q)$

- Neural Tensor Network for representing diversity (Xia et al., 2016)
  - Implicit diversification based on R-LTR and PAMM
  - Modeling the novelty part of f with a neural tensor network (without linear and bias terms)



$$f(d, S) = g_r(\mathbf{x}) + g_n(\mathbf{v}, S)$$
$$= \omega^T \mathbf{x} + \mu^T \max\left\{\tanh\left(\mathbf{v}^T \mathbf{W}^{[1:z]} \left[\mathbf{v}_1, \ldots, \mathbf{v}_{|S|}\right]\right)\right\}$$

relevance

diversity

- Document Sequence with Subtopic Attention, DSSA (Jiang et al., 2017)
  - Explicit Diversification
  - Use RNNs with attention to model novelty part of f
  - Training by minimizing list pairwise loss

$$S_{\text{DSSA}}(q, d_t, C_{t-1}, \mathcal{I}_q) = s_{d_t} =$$
$$(1-\lambda)\mathcal{S}^{\text{rel}}(\boldsymbol{v}_{d_t}, \boldsymbol{v}_q) + \quad \Rightarrow \text{relevance}$$
$$\lambda \mathcal{S}^{\text{div}}\left(\boldsymbol{v}_{d_t}, \boldsymbol{v}_{i_{(\cdot)}}, \underbrace{\mathcal{A}\left(\mathcal{H}([\boldsymbol{v}_{d_1}, \ldots, \boldsymbol{v}_{d_{t-1}}]), \boldsymbol{v}_{i_{(\cdot)}}\right)}_{\text{subtopic attention}}\right), \quad \Rightarrow \text{diversity}$$

Learning deep $f(q, d, D_q)$ (cont')

- DVGAN (Liu et al., 2020)
  - Explicit & implicit diversification
  - Use RNNs with attention to model novelty part of f (same as in DSSA)
  - Training the parameters with GAN
    - Generator: DSSA
    - Discriminator: R-LTR

$$f_\theta(d_t|q,S) = (1-\lambda)S^{\text{rel}}(d_t,q) + \lambda \sum_{i \in I_q} A(i|S) * S^{\text{sub}}(d_t,i),$$

$$S^{\text{rel}}(d_t,q) = \mathcal{S}(e_{d_t}, e_q) + w_r^T(g) * x_{d_t,q},$$

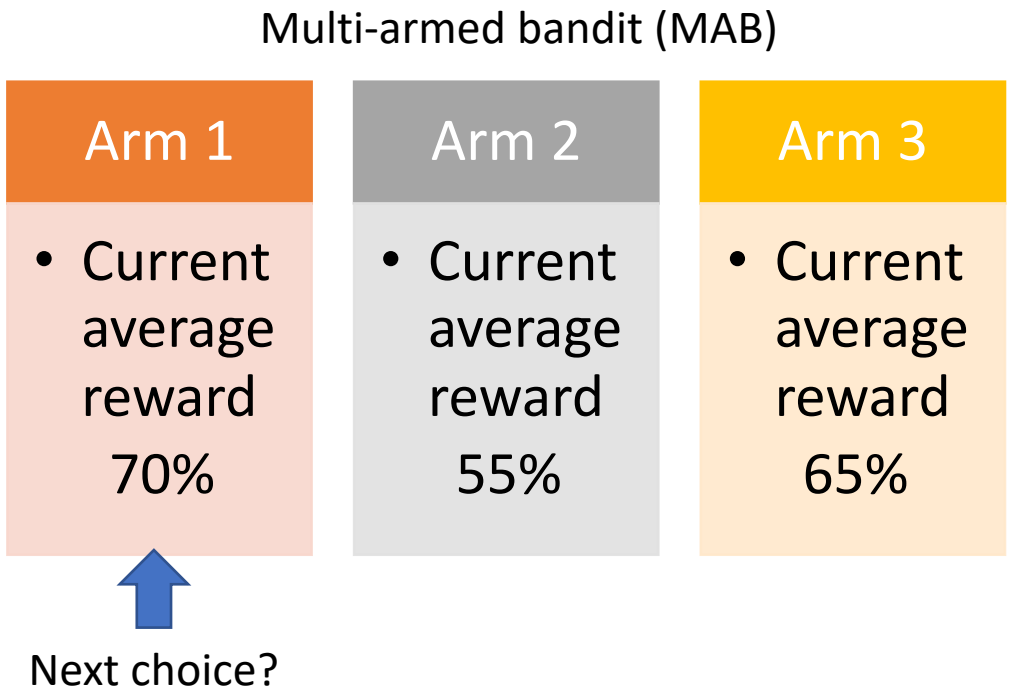
$$S^{\text{sub}}(d_t,i_k) = \mathcal{S}(e_{d_t}, e_{i_k}) + w_r^T(g) * x_{d_t,i_k},$$

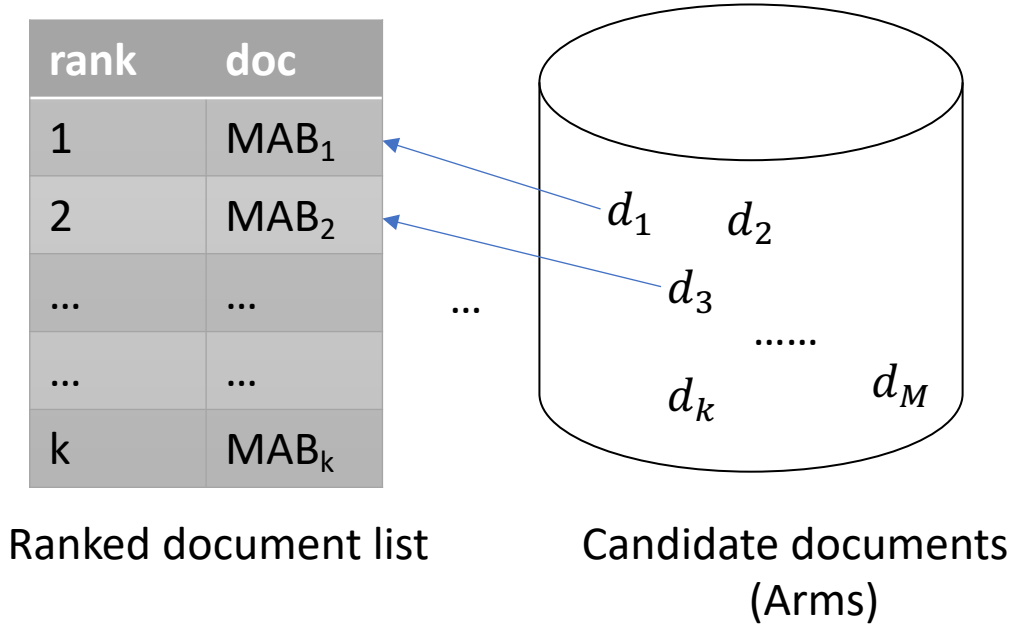
Diversify$(q, \mathcal{R}_q, \tau)$

$\mathcal{D}_q \leftarrow \emptyset$
while $|\mathcal{D}_q| < \tau$ **do**
  $d^* \leftarrow \arg\max_{d \in \mathcal{R}_q \setminus \mathcal{D}_q} f(q, d, \mathcal{D}_q)$
  $\mathcal{R}_q \leftarrow \mathcal{R}_q \setminus \{d^*\}$
  $\mathcal{D}_q \leftarrow \mathcal{D}_q \cup \{d^*\}$
end while
return $\mathcal{D}_q$

Inference by the learned generator

Multi-Slot Bandits for Ranking





- K arms, unknown sequence of *stochastic* rewards
- For each round $t = 1, 2, \dots,$
  - Choose arm A_t
  - Obtain reward

- Multi-slot MAB for ranking
  - MAB_i: model for the i-th rank
  - Arm j: the j-th document
  - Reward: online collecting user clicks
  - Goal: maximize the expected clicks

Ranked Bandit Algorithm [Radlinski et al., ICML '08]

- Runs an MAB instance at each rank
  - MAB_1 is responsible for choosing document shown at rank 1
  - MAB_2 is responsible for choosing document shown at rank 2
  - … until top K documents are selected
- Show top K to users and receive response
  - If click slot i
    - MAB_i gets reward 1
    - MAB_j $(j < i)$ get rewards 0
- Update MABs according to the received feedback

Document selection for k positions

Update bandits

Algorithm 2 Ranked Bandits Algorithm

```
initialize MAB_1(n), ..., MAB_k(n)                    Initialize MABs
for t = 1 ... T do
    for i = 1 ... k do                                 Sequentially select documents
        b̂_i(t) ← select-arm (MAB_i)
        if b̂_i(t) ∈ {b_1(t), .., b_{i-1}(t)} then      Replace repeats
            b_i(t) ← arbitrary unselected document
        else
            b_i(t) ← b̂_i(t)
        end if
    end for
    display {b_1(t), ..., b_k(t)} to user; record clicks
    for i = 1 ... k do                                 Determine feedback for MAB_i
        if user clicked b_i(t) and b̂_i(t) = b_i(t) then
            f_it = 1
        else
            f_it = 0
        end if
        update (MAB_i, arm = b̂_i(t), reward = f_it)
    end for
end for
```

Problem Abstraction: Learning of Assignments [Streeter et al., NIPS ' 09]

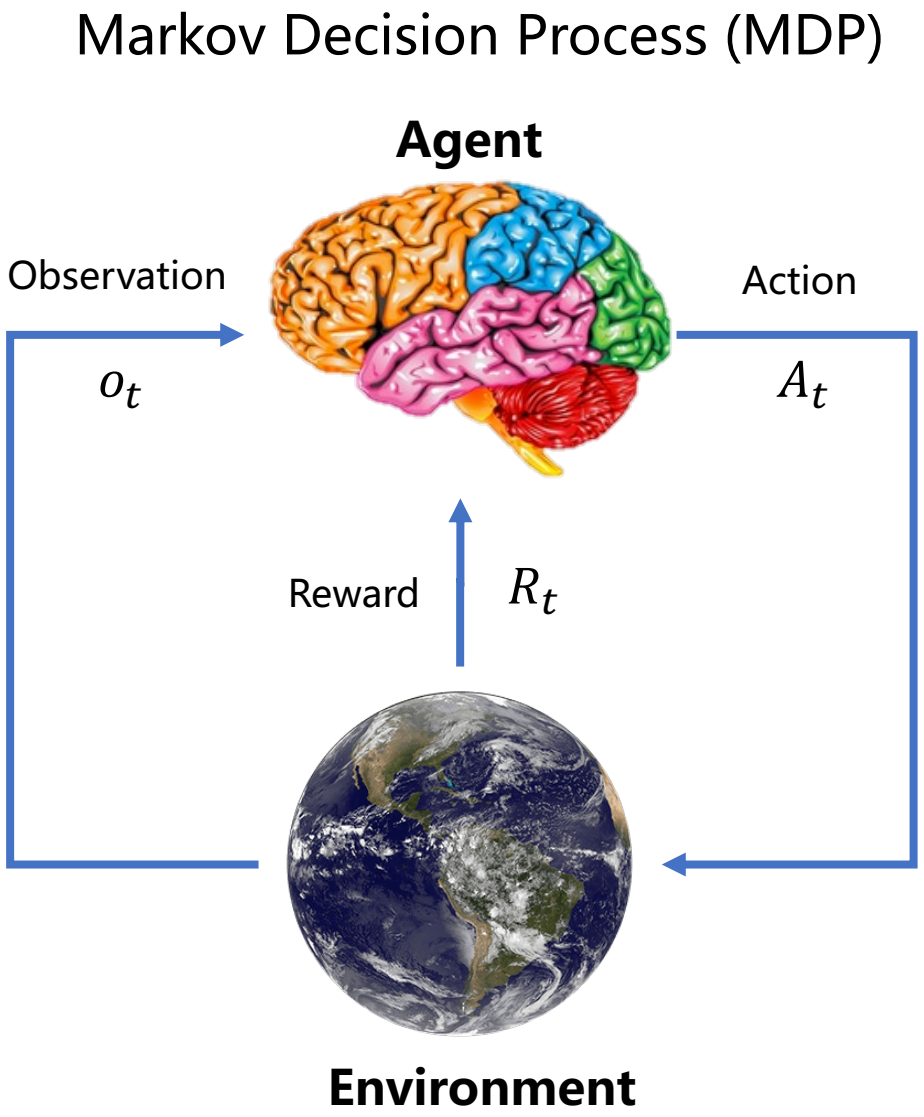
- Ranking: selecting an assignment of documents to positions
  - K positions: K sets of candidate documents
  - Dependences described by *submodular functions*
  - Repeatedly executing the greedy selection step

Algorithm: TABULARGREEDY

Input: integer C, sets $P_1, P_2, \ldots, P_K$, function $f : 2^{\mathcal{V}} \to \mathbb{R}_{\geq 0}$ (where $\mathcal{V} = \bigcup_{k=1}^{K} P_k$)

set $G := \emptyset$.
for c **from** 1 **to** C **do** /* For each color */
  for k **from** 1 **to** K **do** /* For each partition */
    set $g_{k,c} = \arg\max_{x \in P_k \times \{c\}} \{F(G + x)\}$ /* Greedily pick $g_{k,c}$ */
    set $G := G + g_{k,c}$;
for each $k \in [K]$, choose c_k uniformly at random from $[C]$.
return sample$_{\vec{c}}(G)$, where $\vec{c} := (c_1, \ldots, c_K)$.

Markov Decision Process (MDP)



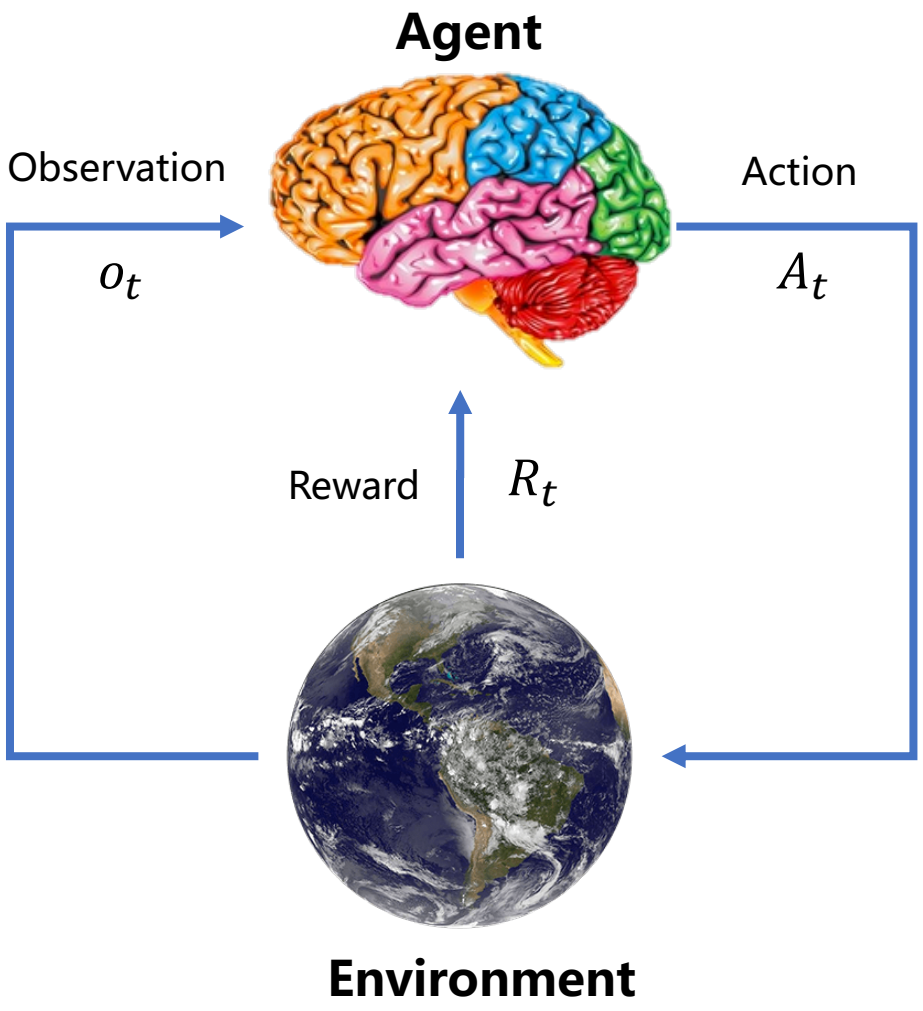
- An **agent** with the capacity to **act**
- Each **action** influences the **environment state**
- Success is measured by a scalar **reward** signal

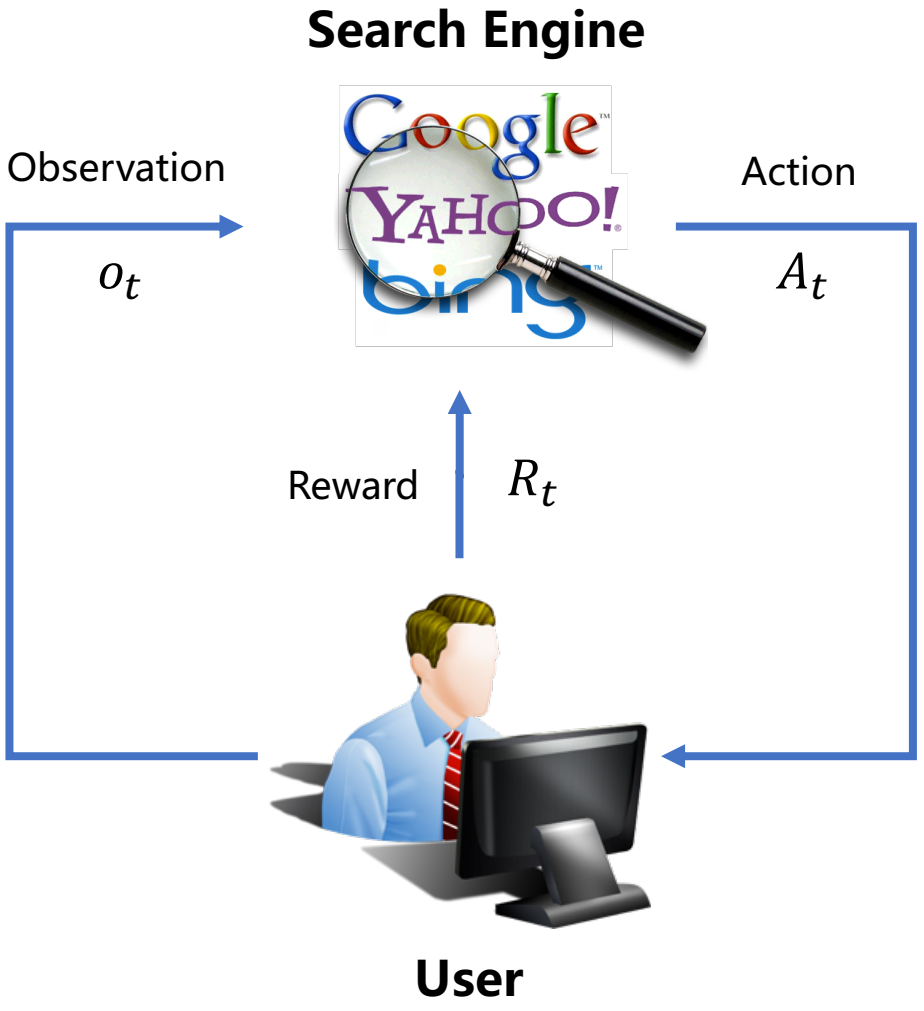
Goal: select actions to maximize future reward

- At each step t the agent:
  - Executes action A_t
  - Receives observation o_t
  - Receives scalar reward r_t
- The environment:
  - Receives action A_t
  - Emits observation o_{t+1}
  - Emits scalar reward r_{t+1}

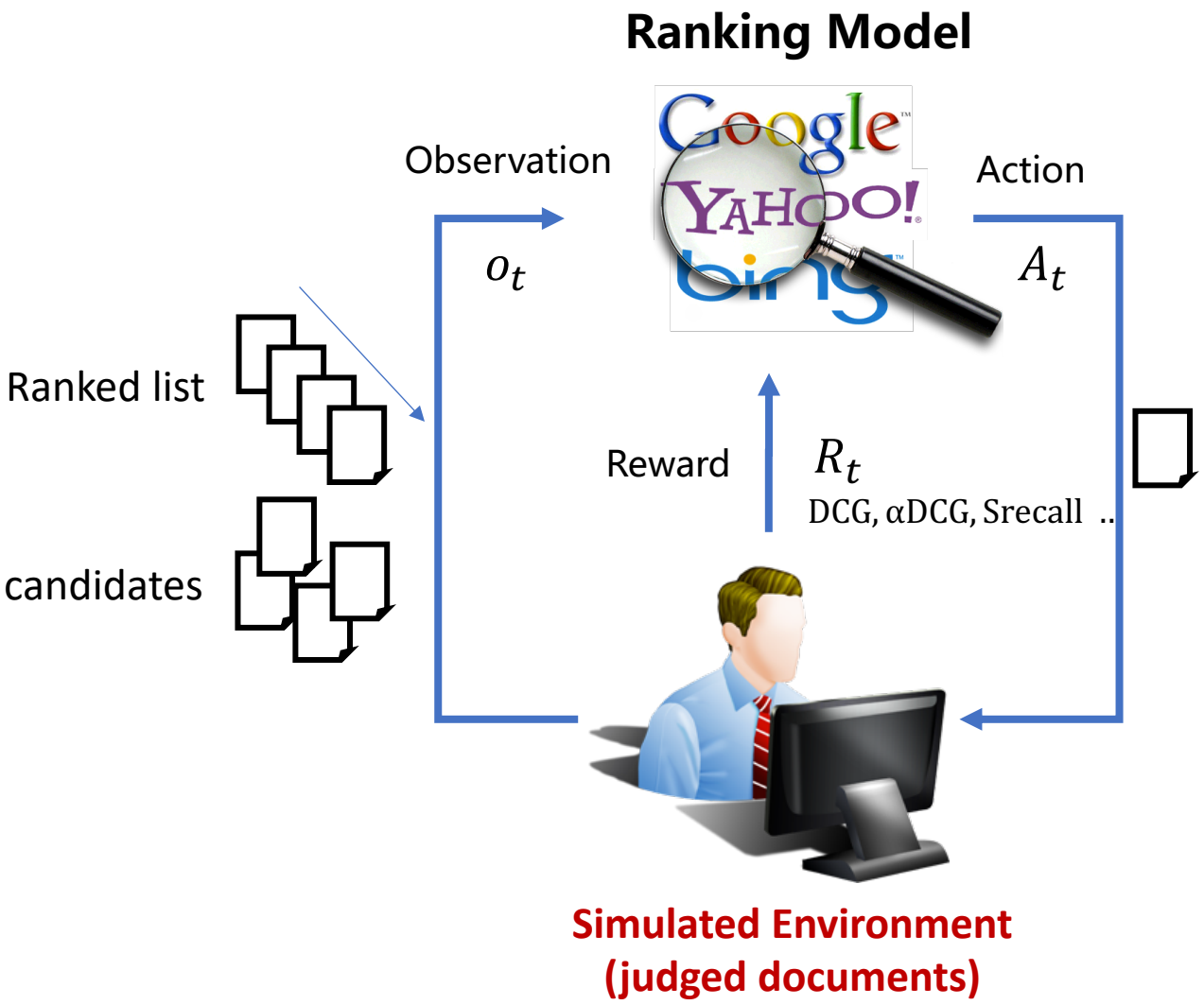
Modeling Sequential Dependency with MDP

Markov Decision Process (MDP)





Modeling Sequential Dependency with MDP



- **Action** A_t :
  - Selects a document and puts ranking list
- **Observation** o_t :
  - query, top t ranked list, candidate set
- **Reward** R_t :
  - designed based on rank evaluation measures

MDP configurations for diverse ranking [Xia et al., 2017]

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

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

- 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
    - initialized with query: $\mathbf{h}_0 = \sigma(\mathbf{V}_q \mathbf{q})$

Learning with Policy Gradient

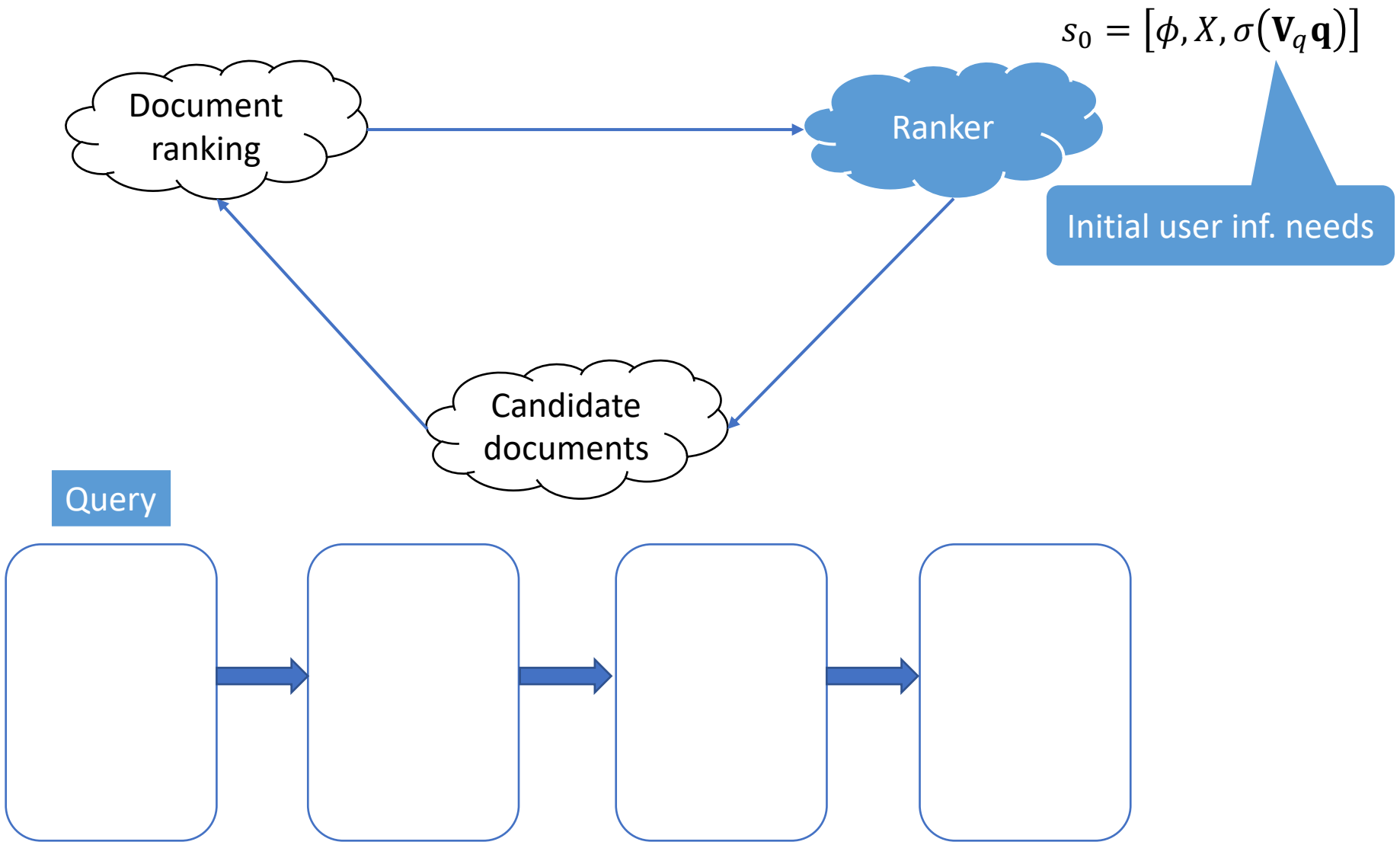
- Model parameters $\boldsymbol{\Theta} = \{\mathbf{V}_q, \mathbf{U}, \mathbf{V}, \mathbf{W}\}$
- Learning objective: maximizing expected return (discounted sum of rewards) of each training query

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

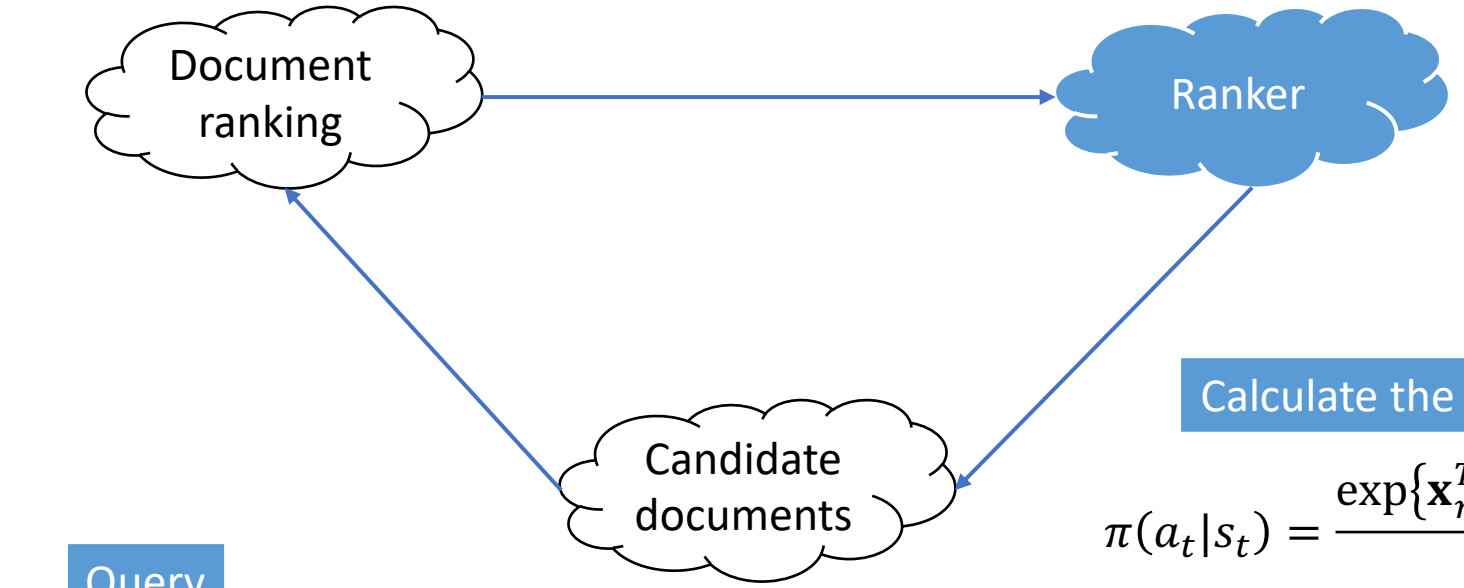
  - Directly optimizes evaluation measure as $G_0 = \alpha \mathrm{DCG@}M$
- Monte-Carlo stochastic gradient ascent is used to conduct the optimization (REINFORCE algorithm)

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

Greedy Sequential Decision: Initialize State



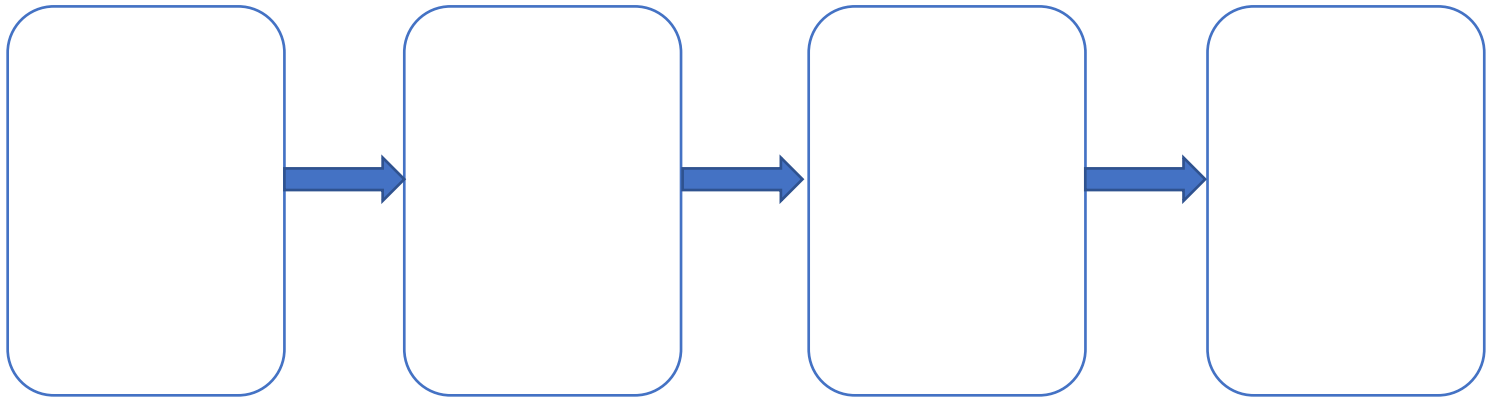
Greedy Sequential Decision: Policy



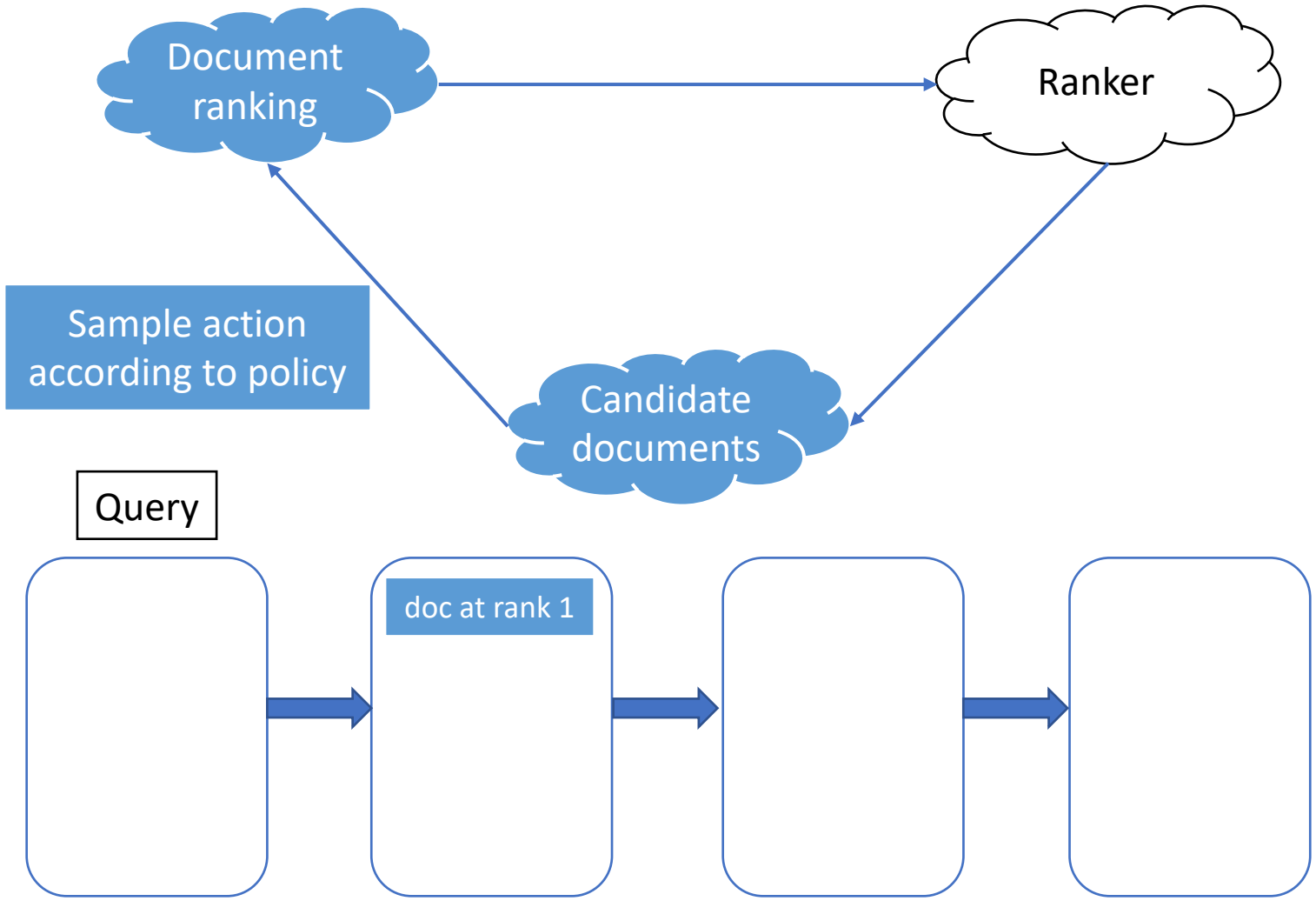
Calculate the policy

$$\pi(a_t|s_t) = \frac{\exp\{\mathbf{x}_{m(a_t)}^T \mathbf{U}\mathbf{h}_t\}}{Z}$$

Query

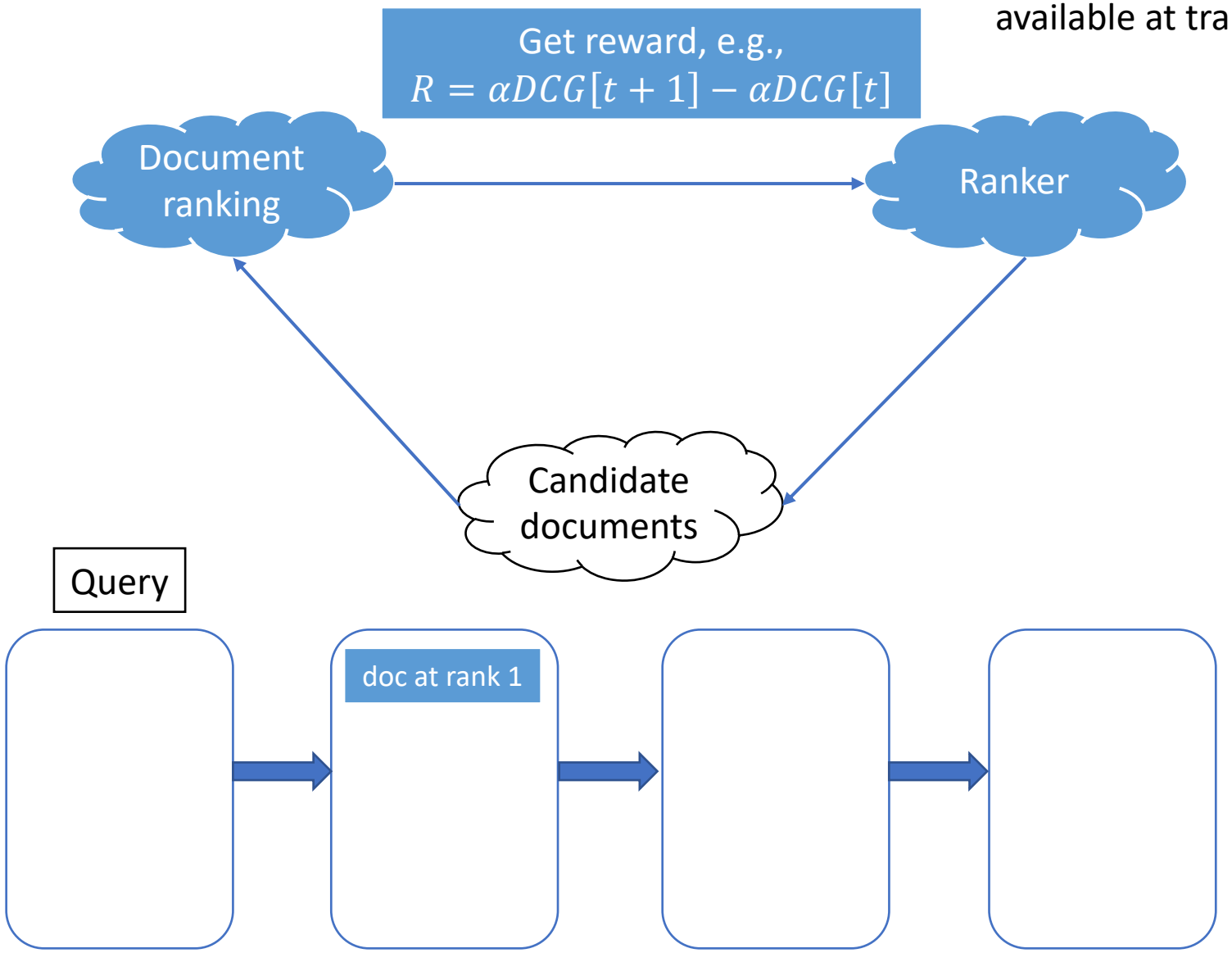


Greedy Sequential Decision: Action



Greedy Sequential Decision: Reward

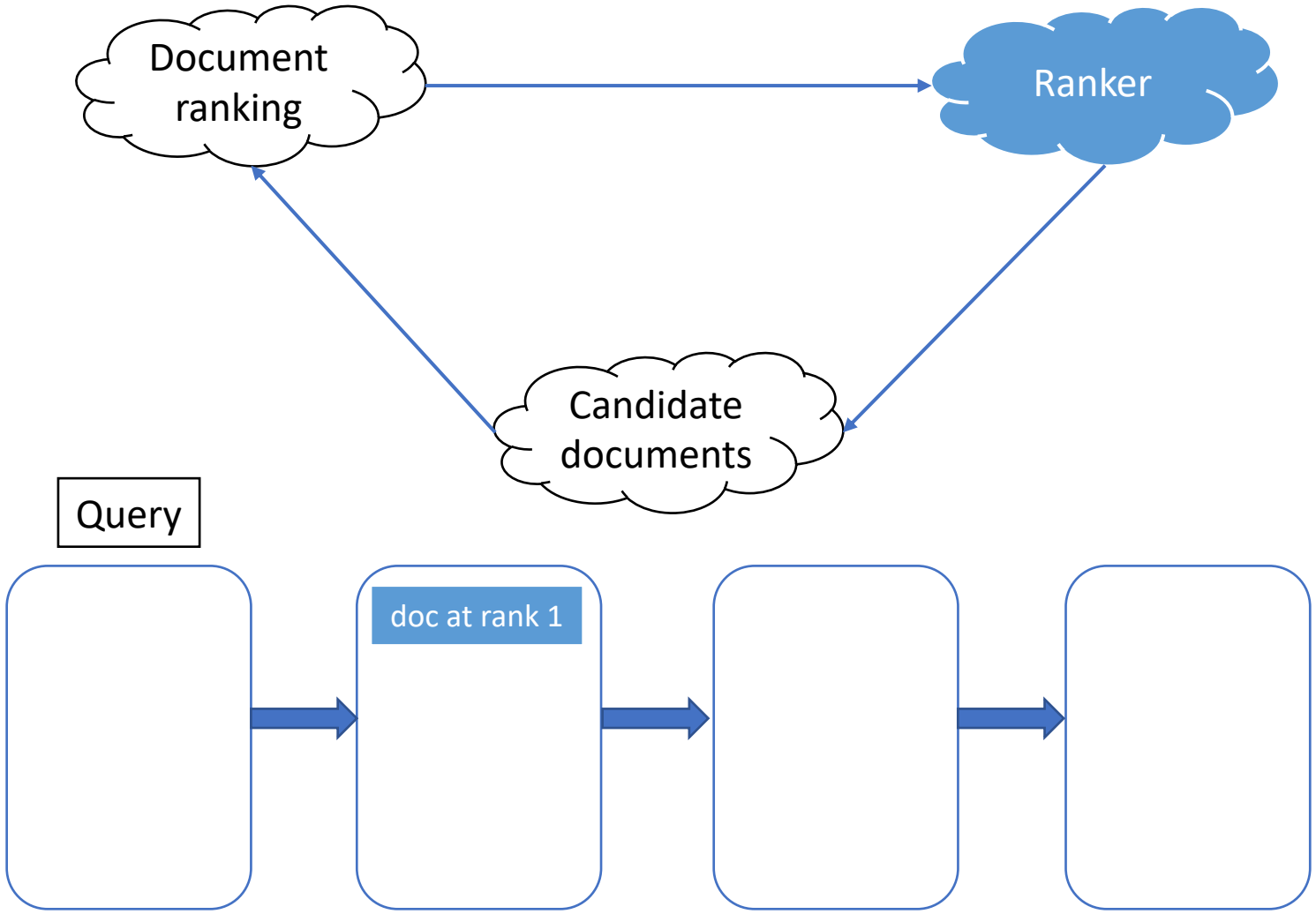
Note: rewards are only available at training time



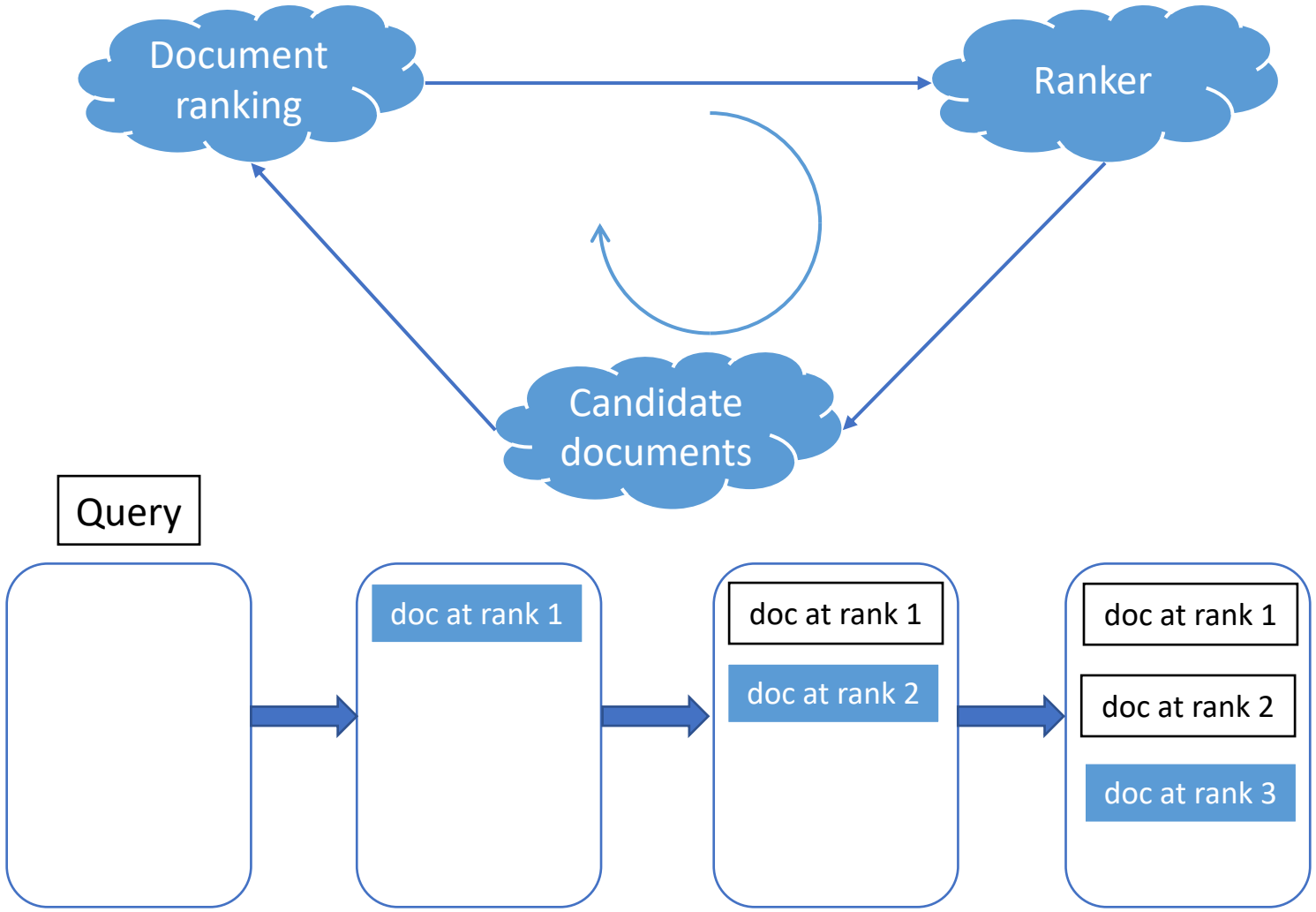
Greedy Sequential Decision: State Transition

Update ranked list, candidate set, and latent vector

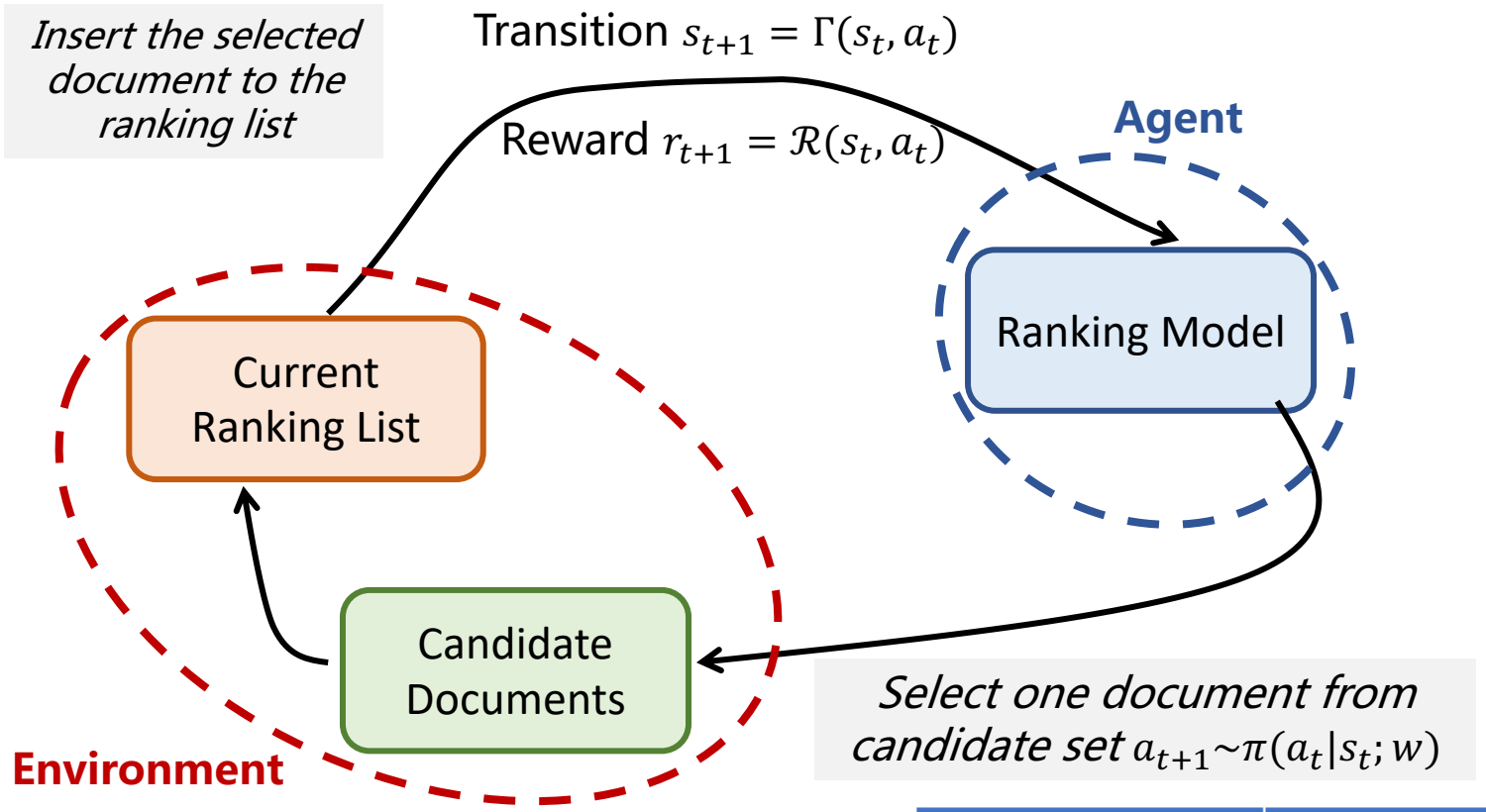
$$s_{t+1} = \left[Z_t \oplus \{\mathbf{x}_{m(a_t)}\}, X_t \backslash \{\mathbf{x}_{m(a_t)}\}, \sigma\left(\mathbf{V}\mathbf{x}_{m(a_t)} + \mathbf{W}\mathbf{h}_t\right)\right]$$



Greedy Sequential Decision: Iterate



MDPRank – General Learning to Rank (Zeng et al., 2017)



MDP Elements	Definitions in Relevance Ranking
State	$S_t = [t, X_t]$
Transition	$\Gamma([t, X_t], a_t) = [t+1, X_t/\{d_{m(a_t)}\}]$
Reward	$\mathcal{R}(s_t, a_t) = [\mathrm{DCG}[t+1] - \mathrm{DCG}[t]]$
Policy	$\pi(a_t|s_t; w) = \frac{\exp\{w^T x_{m(a_t)}\}}{\sum_{a \in A_t} \exp\{w^T x_{m(a)}\}}$

Empirical Evaluations

Method	α-NDCG@5	α-NDCG@10	ERR-IA@5	ERR-IA@10
MMR	0.2753	0.2979	0.2005	0.2309
xQuAD	0.3165	0.3941	0.2314	0.2890
PM-2	0.3047	0.3730	0.2298	0.2814
SVM-DIV	0.3030	0.3699	0.2268	0.2726
R-LTR	0.3498	0.4132	0.2521	0.3011
PAMM(α-NDCG)	0.3712	0.4327	0.2619	0.3029
NTN-DIV(α-NDCG)	0.3962	0.4577	0.2773	0.3285
MDP-DIV(α-DCG)	0.4189	0.4762	0.2988	0.3494

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	**0.5925**	**0.4992**	**0.4909**	0.4587
MDPRank(ReturnOnly)	0.5363	0.4885	0.46949	**0.4591**

MDP-based ranking for search result diversification and relevance ranking

4. Ranking with Sequential Dependency

4.1 Heuristic Sequential Ranking Models

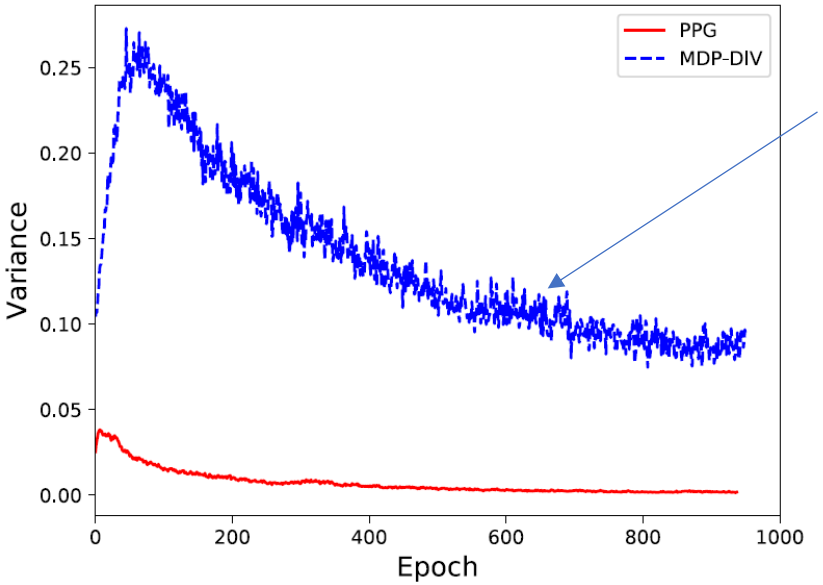
4.2 Learning Sequential Ranking Models

4.3 Challenges

Challenges in Greedy Sequential Decision

1 Low sampling efficiency

REINFORCE (Gradient Policy) is **unbiased**, but is known to have **high variance**.
Because of huge difference in episodic rewards.



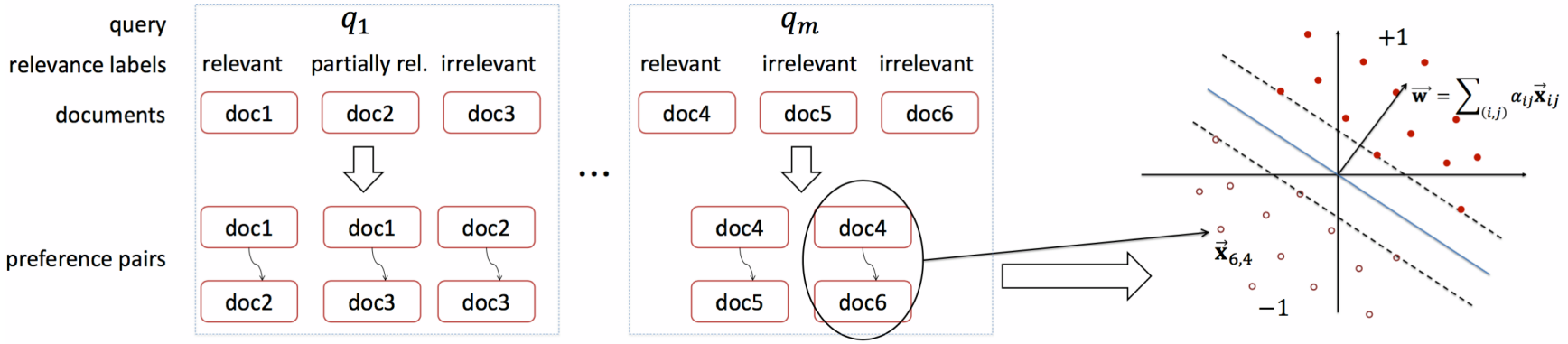
Variance curve of MDP-DIV

- Possible ways to improve the sampling efficiency
  - Sampling guided by experts
    - E.g., imitation learning
    - Efficient while biased estimation
  - **Control variates method**
    - Reducing variance via comparisons
    - E.g., SVRG, policy gradient with baseline
    - Unbiased estimation

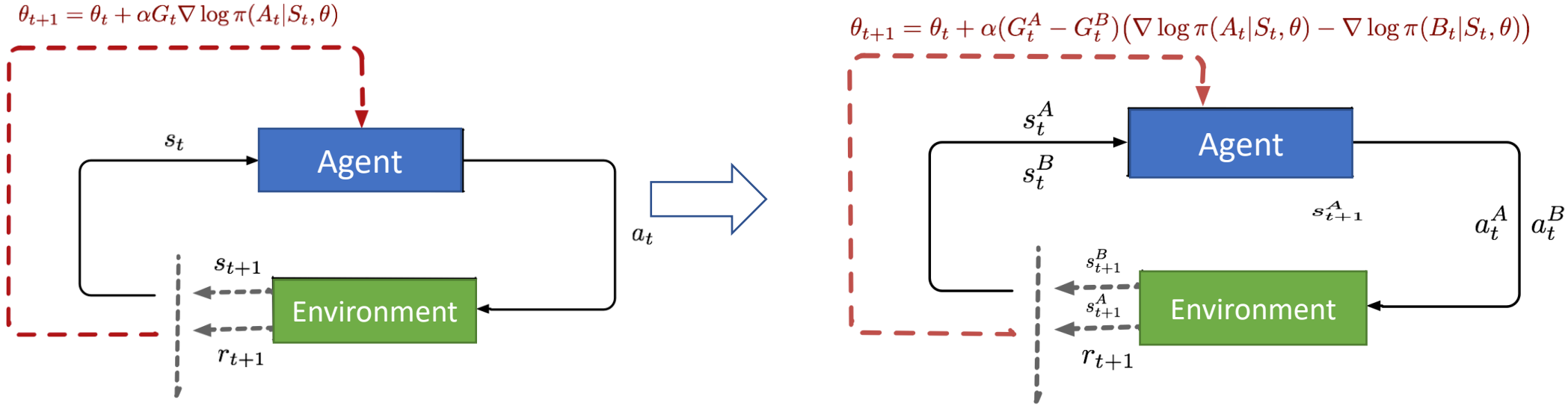
PPG – Pairwise Policy Gradient (Xu et al., ’20)

1 Improving sampling efficiency

Pairwise learning to rank



Pairwise policy gradient



Jun Xu, Wei Zeng, Yanyan Lan, Long Xia, Dawei Yin, Jirong Wen and Xueqi Cheng. Reinforcement Learning to Rank with Pairwise Policy Gradient. SIGIR’20

Theoretical and Empirical Analysis of PPG

- The gradients estimated by PPG are unbiased and have **low variance**

THEOREM 4.1. *The gradient of $J(\theta)$ in Equation (2) can be represented as*

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a \sum_b (q_\pi(s,a) - q_\pi(s,b)) \cdot (\pi(b|s;\theta)\nabla\pi(a|s;\theta) - \pi(a|s;\theta)\nabla\pi(b|s;\theta)),$$

Unbiased gradient

THEOREM 4.2. *Given state $s \sim \mu_\pi$ where μ_π is is on-policy distribution under π, and given two actions $a \sim \pi(\cdot|s;\theta)$ and $b \sim \pi(\cdot|s;\theta)$. Considering the following two representations of the gradient:*

$$\mathbf{g}_1 = (q_\pi(s,a) - q_\pi(s,b)) \cdot (\nabla \log \pi(a|s;\theta) - \nabla \log \pi(b|s;\theta)),$$
$$\mathbf{g}_2 = q_\pi(s,a) \cdot \nabla \log \pi(a|s;\theta) + q_\pi(s,b) \cdot \nabla \log \pi(b|s;\theta).$$

The variances of $\mathbf{g}_1$ and $\mathbf{g}_2$ satisfy

$$\mathrm{Var}(\mathbf{g}_1) \le \mathrm{Var}(\mathbf{g}_2),$$

Low variance gradient

- PPG improves the sorting **accuracy** and **convergence speed**

Table 4: Performance comparison on LETOR OHSUMED.

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	0.4790	0.3730	0.4673	0.4496
MDPRank	0.5743	0.5045	0.4784	0.4558
PPG	**0.5771**	**0.5218**	**0.4911**	**0.4664**

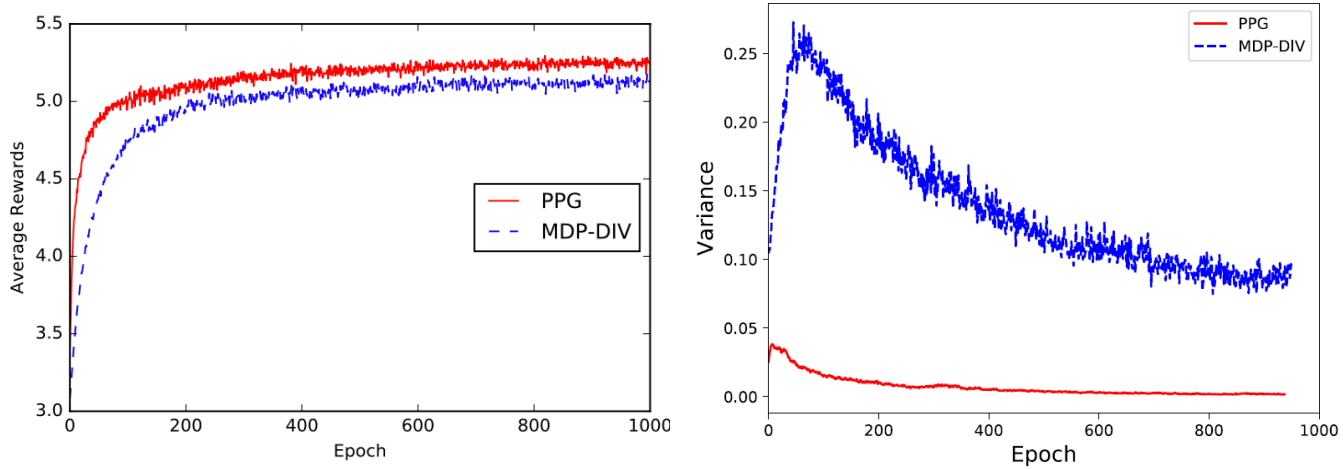
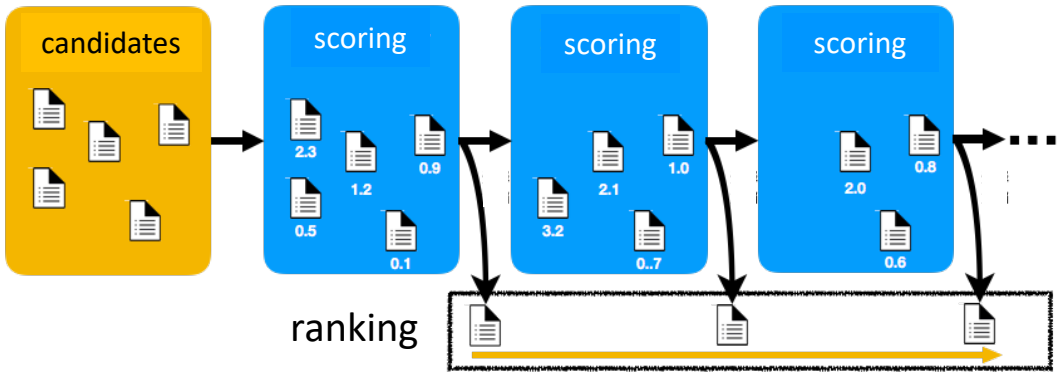


Figure 3: Learning curves of PPG and MDP-DIV.

Challenges in Greedy Sequential Decision

2 Local Optimal

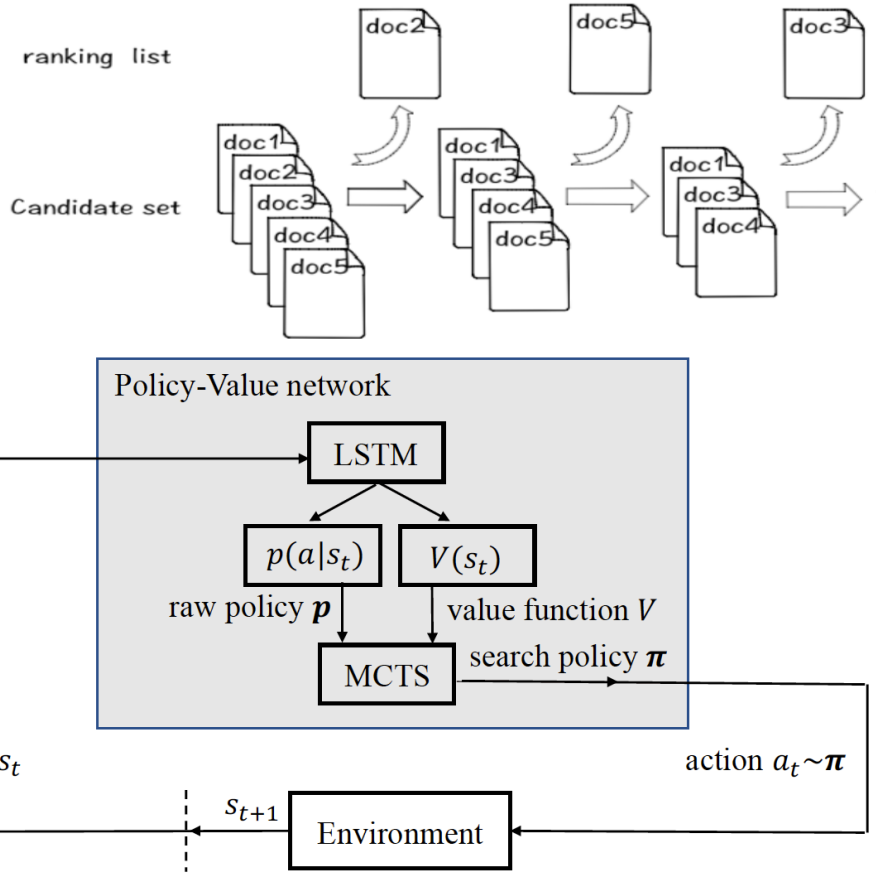
Sequential decision making is **greedy**.
The choice of the previous steps will affect the choice of the next document, but **NOT** vice versa.



- Possible ways to avoid local optimal
  - Exhaustive search (Brute-force search)
    - Enumerating all possible candidate rankings
    - Checking their performances at each position
    - Exact optimal solution but extremely costly
  - **Monte Carlo tree search (MCTS)**
    - Search tree based on random sampling
    - *Near-optimal* solution but much faster

MCTS Enhanced MDP Ranking (Feng et al., 2018)

2 Avoiding Local Optimal with MCTS



Value Function + MCTS

To see the future

. . . . . .

$$
\begin{aligned}
\mathbf{f}_k &= \sigma(\mathbf{W}_f \mathbf{x}_k + \mathbf{U}_f \mathbf{h}_{k-1} + \mathbf{b}_f), \\
\mathbf{i}_k &= \sigma(\mathbf{W}_i \mathbf{x}_k + \mathbf{U}_i \mathbf{h}_{k-1} + \mathbf{b}_i), \\
\mathbf{o}_k &= \sigma(\mathbf{W}_o \mathbf{x}_k + \mathbf{U}_o \mathbf{h}_{k-1} + \mathbf{b}_o), \\
\mathbf{c}_k &= \mathbf{f}_k \circ \mathbf{c}_{k-1} + \mathbf{i}_k \circ \tanh(\mathbf{W}_c \mathbf{x}_k + \mathbf{U}_c \mathbf{h}_{k-1} + \mathbf{b}_c), \\
\mathbf{h}_k &= \mathbf{o}_k \circ \tanh(\mathbf{c}_k),
\end{aligned}
$$

$$
\mathrm{LSTM}(s) = \left[\mathbf{h}_t{}^T, \mathbf{c}_t{}^T \right]^T
$$

$$
V(s) = \sigma\left(\langle \mathbf{w}, \mathrm{LSTM}(s) \rangle + b_v \right)
$$

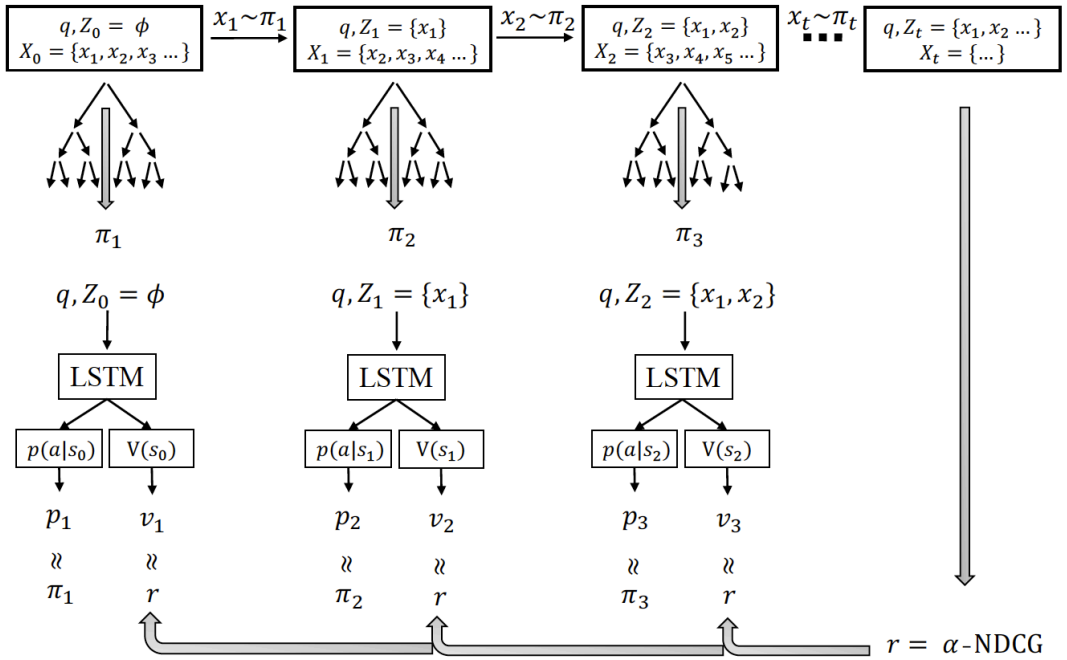
$$
p(a|s) = \frac{\exp\left\{ \mathbf{x}_{m(a)}^T \mathbf{U}_p \ \mathrm{LSTM}(s) \right\}}{\sum_{a' \in \mathcal{A}(s)} \exp\left\{ \mathbf{x}_{m(a')}^T \mathbf{U}_p \ \mathrm{LSTM}(s) \right\}}
$$

- Ranking as an MDP
- MCTS guided by the predicted policies and values

Yue Feng, Jun Xu, Yanyan Lan, Jiafeng Guo, Wei Zeng, Xueqi Cheng. From Greedy Selection to Exploratory Decision-Making: Diverse Ranking with Policy-Value Networks. SIGIR' 18

Learning the Parameters

$$\ell(E, r) = \sum_{t=1}^{|E|} \left((V(s_t) - r)^2 + \sum_{a \in \mathcal{A}(s_t)} \pi_t(a|s_t) \log \frac{1}{p(a|s_t)} \right)$$



- Predicted value is as close to the real α-NDCG as possible
- Raw policy is as close to the search policy as possible

Evaluation of existing models

Method	α-NDCG@5	α-NDCG@10	ERR-IA@5	ERR-IA@10
MMR	0.2753	0.2979	0.2005	0.2309
xQuAD	0.3165	0.3941	0.2314	0.2890
PM-2	0.3047	0.3730	0.2298	0.2814
SVM-DIV	0.3030	0.3699	0.2268	0.2726
R-LTR	0.3498	0.4132	0.2521	0.3011
PAMM(α-NDCG)	0.3712	0.4327	0.2619	0.3029
NTN-DIV(α-NDCG)	0.3962	0.4577	0.2773	0.3285
MDP-DIV(α-DCG)	0.4189	0.4762	0.2988	0.3494
M^2Div(without MCTS)	0.4386*	0.4835	0.3435*	0.3668*
M^2Div(with MCTS)	**0.4424***	**0.4852**	**0.3459***	**0.3686***

Heuristic Sequential Models

Supervised Learning of Sequential Ranking Models

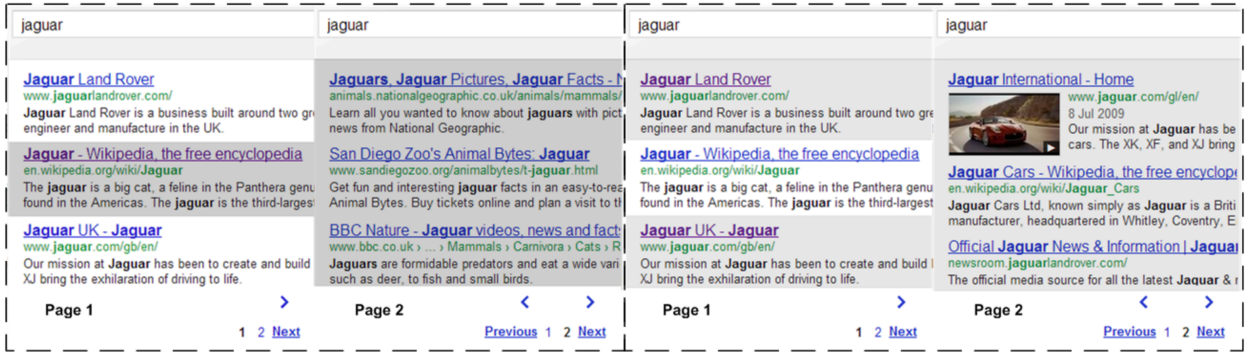
MDP-based Sequential Ranking Models

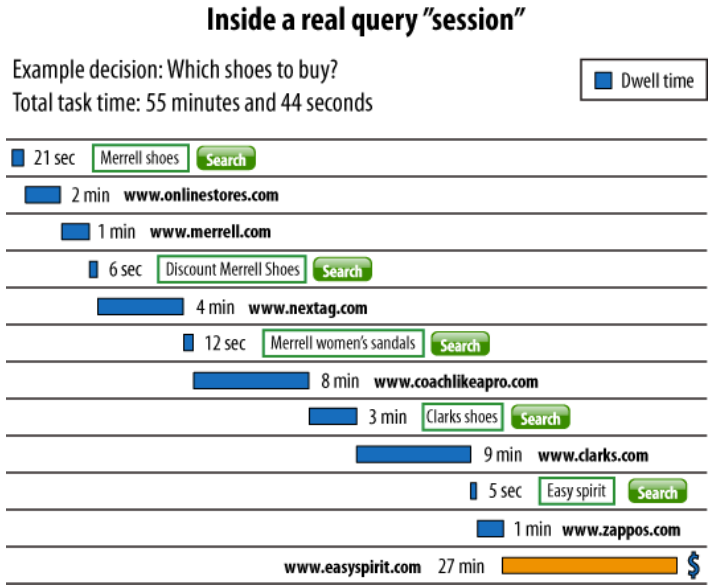
MCTS enhanced MDP

Sequential Ranking Models for Search Result Diversification

More Sequential Dependent Ranking Models

- For multi-page search (depends on results in previous result page)
  - MDP-MPS (Zeng et al., ICTIR ’18)
  - E-commerce Search as MDP: DPG-FBE (Hu et al., KDD ’18)
- For session Search (depends on results in previous search session)
  - Query Change Model (QCM) (Yan et al, ’15)
  - Win-Win (Luo et al, ’14)
  - DPL (Luo et al, ’15)
- Interactively optimizing IR systems
  - Dueling Bandit Gradient Descent (DBGD) (Yue and Joachims, ICML ’09)
  - Balancing Exploration and Exploitation (Hofmann et al., IRJ ‘13)

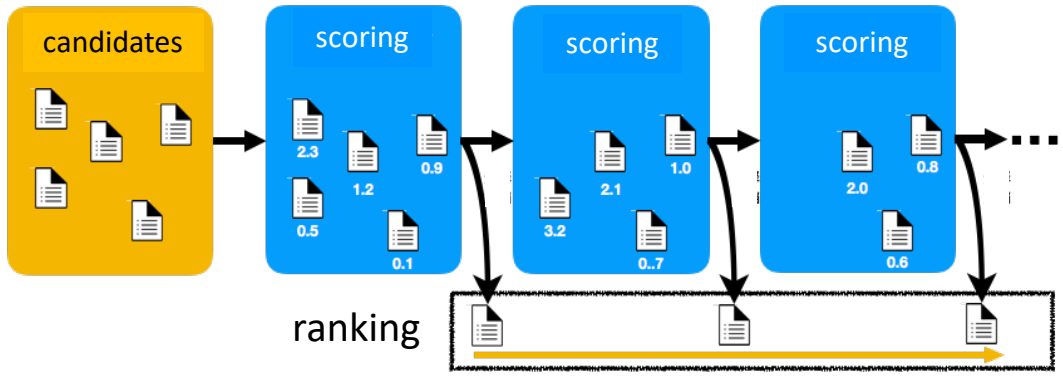




Brief Summary: Sequential Dependent Ranking Models

- Ranking as **greedy sequential decision making**
- Key: the scoring function f
  - Heuristically predefined f
  - Learned f
    - Linear / kernel / deep function
    - Multi-armed bandits
    - Markov decision process
- Challenges
  - Low efficiency
  - Local optimal

➔ Next: globally dependent ranking models



$\mathcal{D}_q \leftarrow \emptyset$
while $|\mathcal{D}_q| < \tau$ **do**
  $d^* \leftarrow \arg\max_{d \in \mathcal{R}_q \setminus \mathcal{D}_q} f(q, d, \mathcal{D}_q)$
  $\mathcal{R}_q \leftarrow \mathcal{R}_q \setminus \{d^*\}$
  $\mathcal{D}_q \leftarrow \mathcal{D}_q \cup \{d^*\}$
end while
return $\mathcal{D}_q$

Reference

- R. L. T. Santos, C. Macdonald, and I. Ounis. Search Result Diversification. FnTIR Vol. 9, No. 1 (2015) 1–90.
- J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. SIGIR 1998, 335–336.
- C. Zhai, W. W. Cohen, and J. Lafferty. Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. SIGIR 2003, 10–17.
- V. Dang and W. B. Croft. Diversity by proportionality: an election-based approach to search result diversification. SIGIR 2012, 65–74.
- R. L. T. Santos, C. Macdonald, and I. Ounis. Exploiting query reformulations for web search result diversification. WWW 2010, 881–890.
- Y. Yue and T. Joachims. Predicting Diverse Subsets Using Structural SVMs, ICML 2008.
- R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. WSDM 2009, 5–14.
- Y. Zhu , Y. Lan, J.Guo, X. Cheng and S. Niu. Learning for Search Result Diversification. SIGIR 2014, 293-302.
- L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng. Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures. SIGIR 2015, 113-122.
- L. Xia, J. Xu, Y. Lan, J. Guo, and X. Cheng. Modeling Document Novelty with Neural Tensor Network for Search Result Diversi_cation. SIGIR 2016, 395-404.
- Z. Jiang, J. R. Wen, Z. Dou, W. X. Zhao, J.-Y. Nie, M. Yue. Learning to Diversify Search Results via Subtopic Attention. SIGIR 2017, 545-554
- F. Radlinski, R. Kleinberg, and T. Joachims. Learning diverse rankings with multi-armed bandits. ICML 2008, 784–791.
- L. Xia, J. Xu, Y. Lan, J. Guo, W. Zeng, and X. Cheng. Adapting Markov Decision Process for Search Result Diversification. SIGIR 2017, 535-544.

Reference

- W. Zeng, J. Xu, Y. Lan, J. Guo, and X. Cheng. Reinforcement Learning to Rank with Markov Decision Process. SIGIR 2017, 945-948.
- J. Xu, W. Zeng, L. Xia, Y. Lan, D. Yin, X. Cheng, and J.-R. Wen. Reinforcement Learning to Rank with Pairwise Policy Gradient. SIGIR 2020, 509–518
- Y. Feng, J. Xu, Y. Lan, J. Guo, W. Zeng, X. Cheng. From Greedy Selection to Exploratory Decision-Making: Diverse Ranking with Policy-Value Networks. SIGIR 2018, 125-134.
- M. Streeter, D. Golovin, and A. Krause. Online learning of assignments. In NIPS 2009.
- Y. Hu, Q. Da, A. Zeng , Y. Yu, Y. Xu. Reinforcement Learning to Rank in E-Commerce Search Engine: Formalization, Analysis, and Application. KDD 2018, 368–377.
- W. Zeng, J. Xu, Y. Lan, J. Guo, X. Cheng. Multi Page Search with Reinforcement Learning to Rank. ICTIR 2018, 175-178.
- H. Yang, D. Guan, S. Zhang. The Query Change Model: Modeling Session Search as a Markov Decision Process. TOIS, No. 20, 2015.
- J. Luo, S. Zhang, H. Yang. Win-Win Search: Dual-Agent Stochastic Game in Session Search. SIGIR 2014,
- J. Luo, S. Zhang, X. Dong, H. Yang. Designing states, actions, and rewards for using POMDP in session search. ECIR 2015, 526-537.
- Y. Yue, T. Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. ICML 2009.
- J. Liu, Z. Dou, X. Wang, S. Lu, and J.-R. Wen. DVGAN: A Minimax Game for Search Result Diversification Combining Explicit and Implicit Features. SIGIR 2020. 479-488.
- M. Streeter, D. Golovin, A. Krause. Online learning of assignments. NIPS 2009. 1794-1802.
- L. Chen, G. Zhang, H. Zhou. Fast Greedy MAP Inference for Determinantal Point Process to Improve Recommendation Diversity. NIPS 2018. 1-12.
- M. Wilhelm, A. Ramanathan, A. Bonomo, S. Jain, E. H. Chi, J. Gillenwater. Practical Diversified Recommendation on Youtube with Determinantal Point Process. CIKM 2018.

Thank you!