Learning Context-dependent Label Permutations for Multi-label Classification

Jinseok Nam

Amazon Alexa AI

Joint work with
Young-Bum Kim, Eneldo Loza Mencía, Sunghyun Park, Ruhi Sarikaya
and Johannes Fürnkranz

TECHNISCHE
UNIVERSITÄT
DARMSTADT

amazon alexa

Multi-label Classification (MLC)

- **Goal**: learn a function f that maps instances to a subset of labels

$\xrightarrow{f}$

Sea
~~Desert~~
Building
Sky
~~Cloud~~
Mountain

- It is important to take into account ***label dependencies***.
- Joint probability of labels

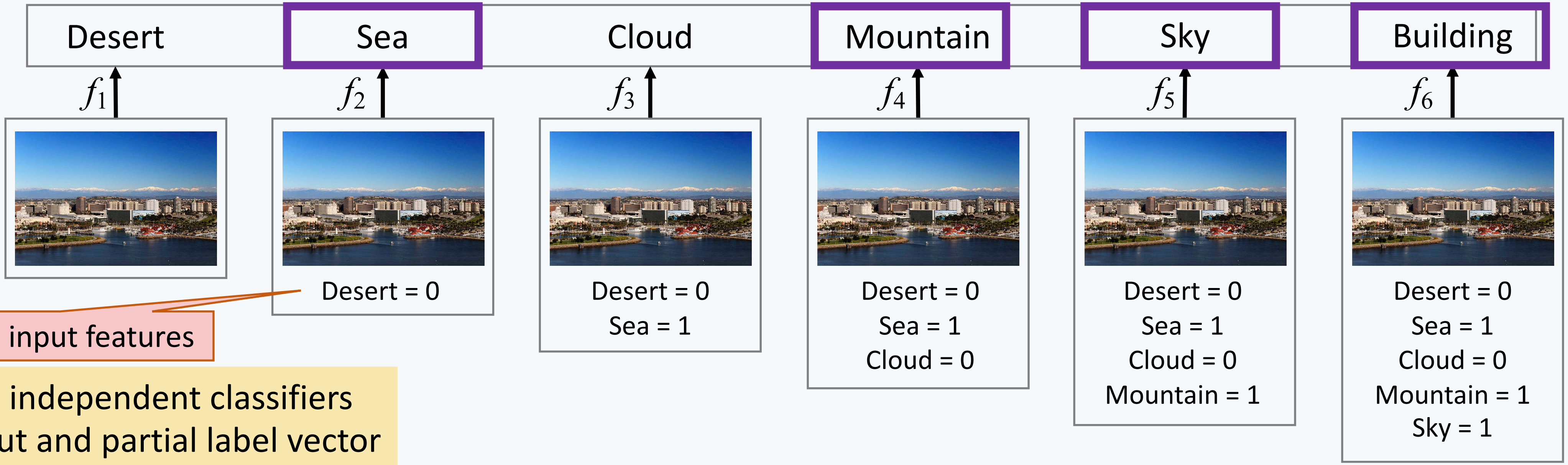
$$P(y_1, y_2, \cdots, y_L|\boldsymbol{x}) = \prod_{i=1}^{L} P(y_i|\boldsymbol{y}_{<i}, \boldsymbol{x})$$

Maximization of the joint probability

- Traditional approaches for minimizing **subset 0/1 loss**:
  - (Probabilistic) classifier chain (Dembczyński et al., ICML 2010; Read et al., MLJ 2011)

Y = {Sea, Desert, Building, Sky, Cloud, Mountain}

1. Creates a chain of L labels



Additional input features

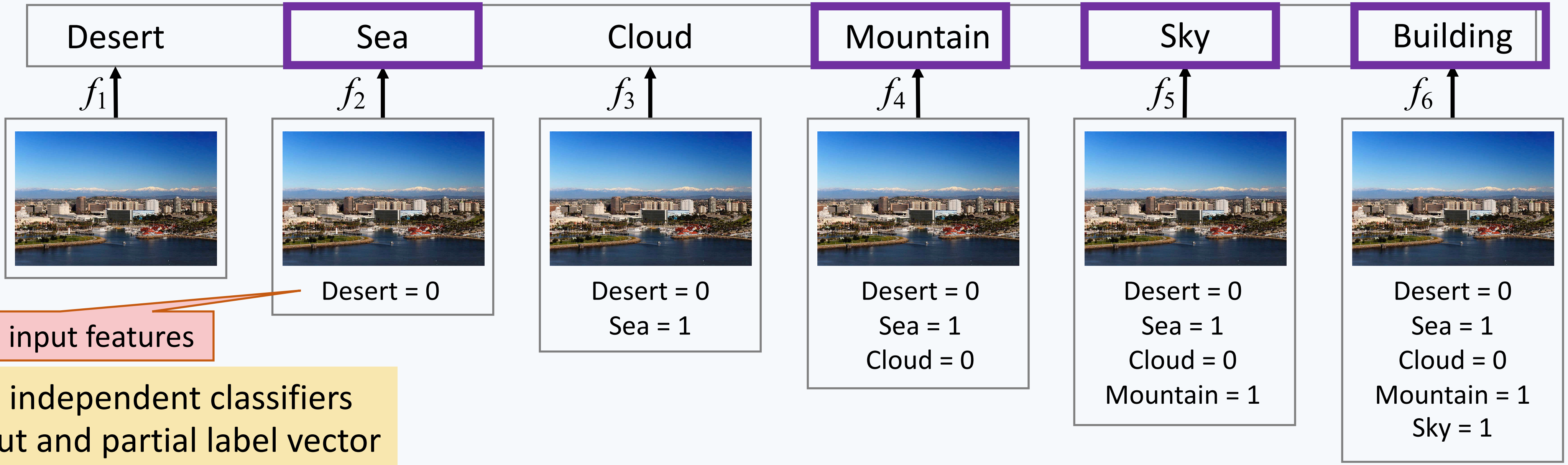
2. Train L independent classifiers given input and partial label vector

Maximization of the joint probability

- Traditional approaches for minimizing **subset 0/1 loss**:
  - (Probabilistic) classifier chain (Dembczyński et al., ICML 2010; Read et al., MLJ 2011)

Y = {Sea, Desert, Building, Sky, Cloud, Mountain}

1. Creates a chain of L labels



Additional input features

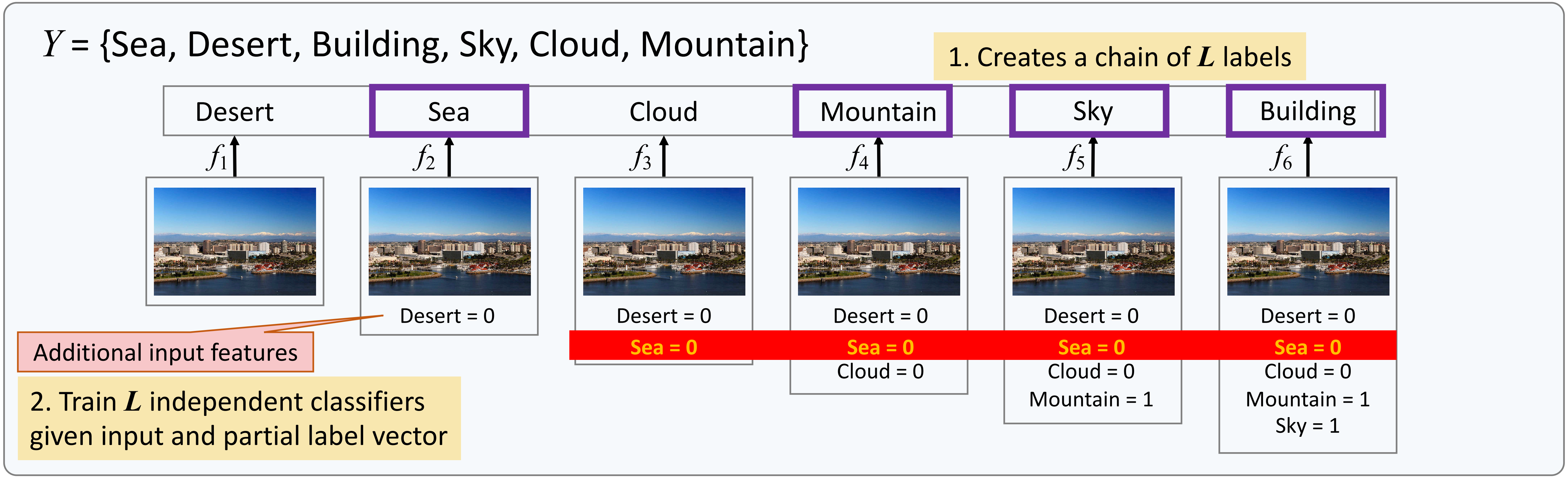
2. Train L independent classifiers given input and partial label vector

Limitations

- **Error-propagation at test time**
- **Effect of label orders in the chain**

Maximization of the joint probability

- Traditional approaches for minimizing **subset 0/1 loss**:
  - (Probabilistic) classifier chain (Dembczyński et al., ICML 2010; Read et al., MLJ 2011)

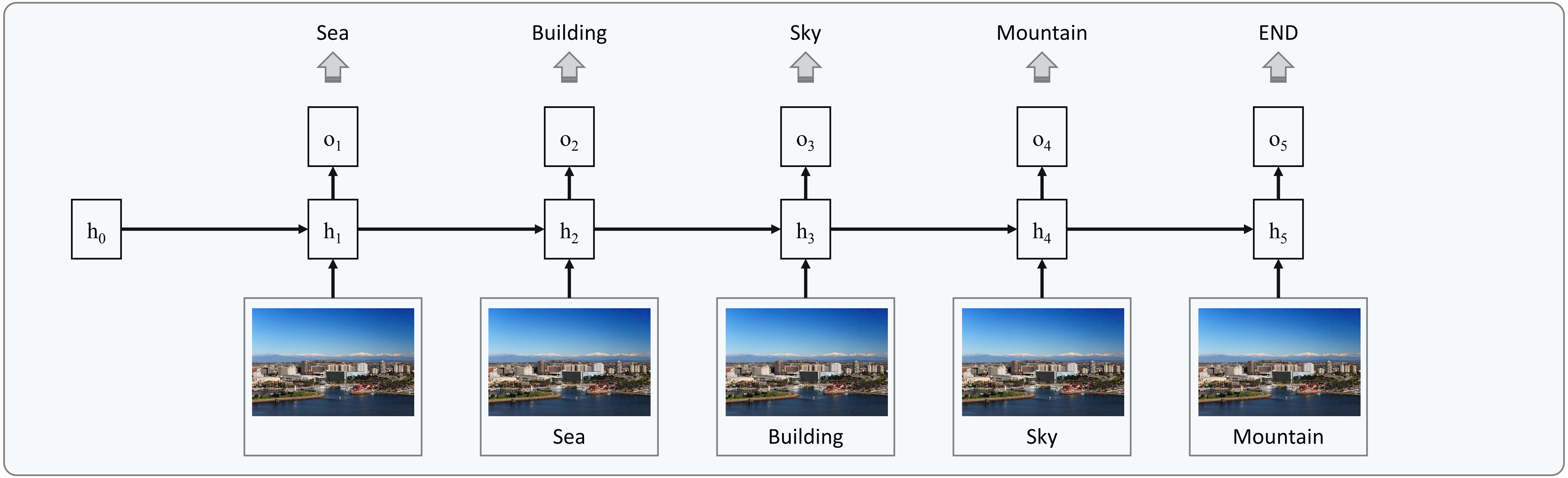


Limitations

- **Error-propagation at test time**
- **Effect of label orders in the chain**

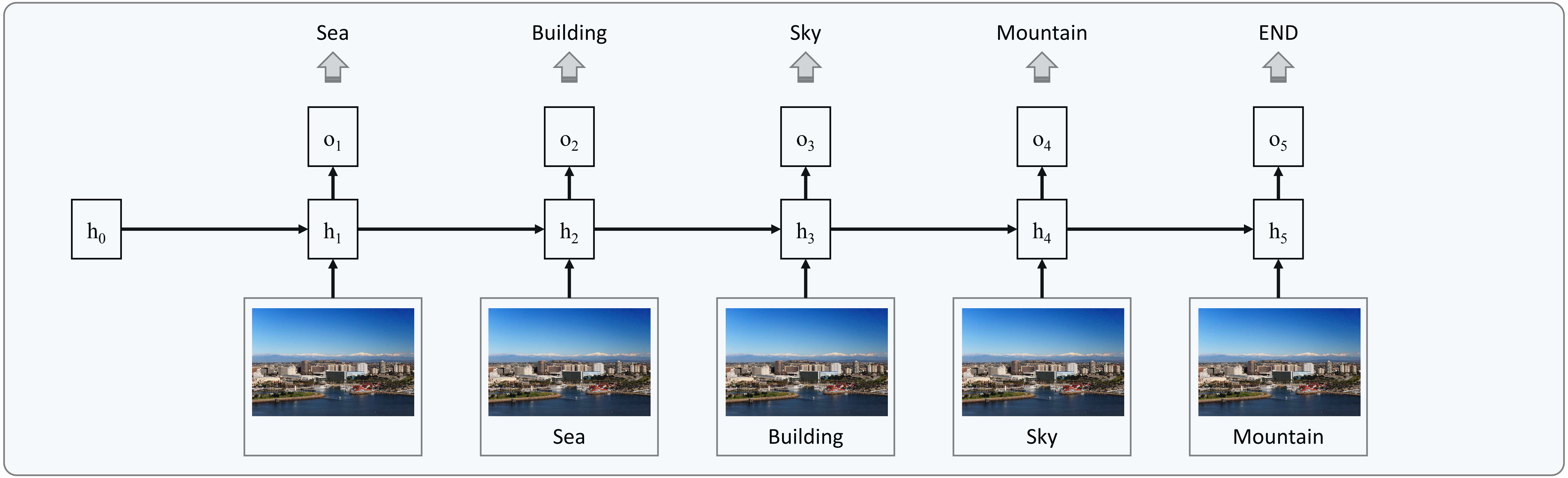
Recurrent Neural Networks for MLC

- Learning from **a set of relevant labels** in a **sequential** manner (Nam et al., NIPS 2017)
    - Number of relevant labels is much smaller than the total number of labels



Recurrent Neural Networks for MLC

- Learning from **a set of relevant labels** in a **sequential** manner (Nam et al., NIPS 2017)
  - Number of relevant labels is much smaller than the total number of labels



- **Question**: The effect of label permutation remain!
  How to determine the target label permutation?

Target label permutations for RNN training

- Static label permutation for ***all*** instances
  - Arbitrary label sequence randomly chosen at the beginning
  - Label frequency distribution: *freq2rare*, *rare2freq*
  - Label structures (e.g., pairwise label dependencies)

→ *Suboptimal* choice; learn from only one permutation

Target label permutations for RNN training

- Static label permutation for ***all*** instances
  - Arbitrary label sequence randomly chosen at the beginning
  - Label frequency distribution: *freq2rare*, *rare2freq*
  - Label structures (e.g., pairwise label dependencies)

→ *Suboptimal* choice; learn from only one permutation

- Different label permutations for ***individual*** instances
  - Choosing randomly every time
  - Learning from all possible label permutations

→ More robust to the effect of label permutation; *Computational complexity*

Target label permutations for RNN training

- Static label permutation for ***all*** instances
  - Arbitrary label sequence randomly chosen at the beginning
  - Label frequency distribution: *freq2rare*, *rare2freq*
  - Label structures (e.g., pairwise label dependencies)

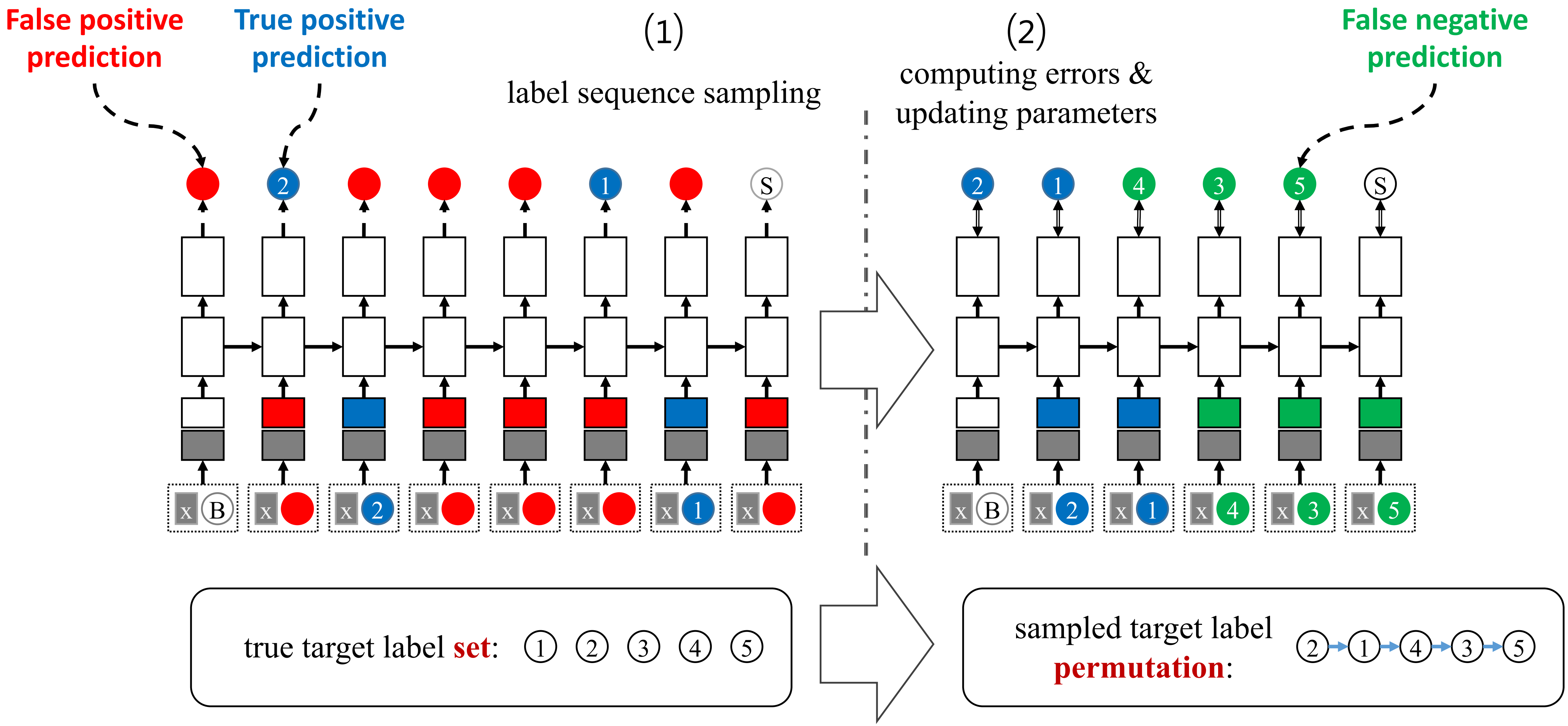
→ *Suboptimal* choice; learn from only one permutation

- Different label permutations for ***individual*** instances
  - Choosing randomly every time
  - Learning from all possible label permutations

→ More robust to the effect of label permutation; *Computational complexity*

We need MLC algorithms that learn context-dependent label permutations ***efficiently***!

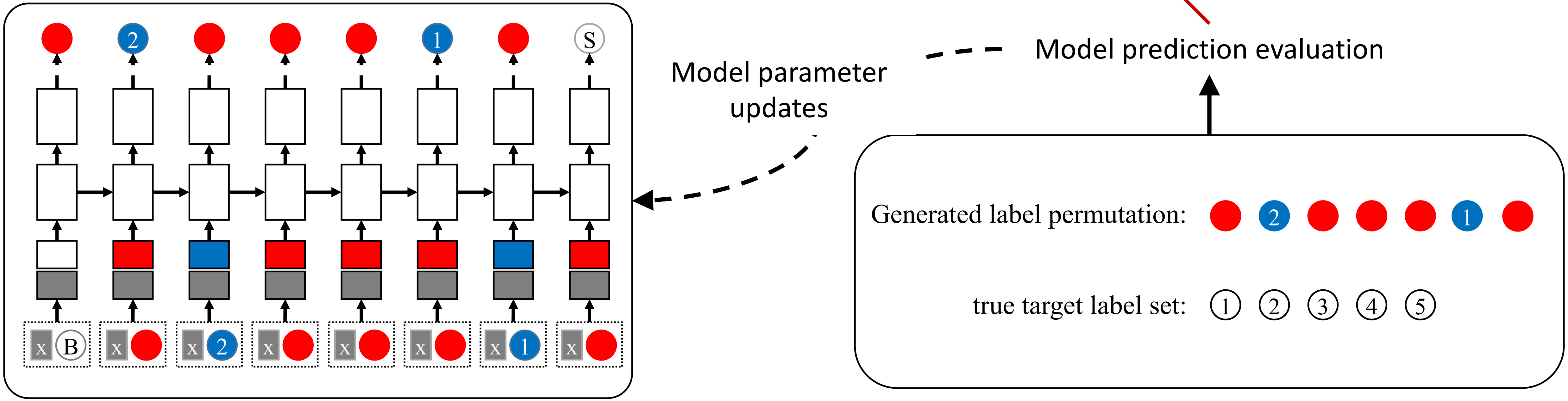
Model based label permutation



Policy gradient

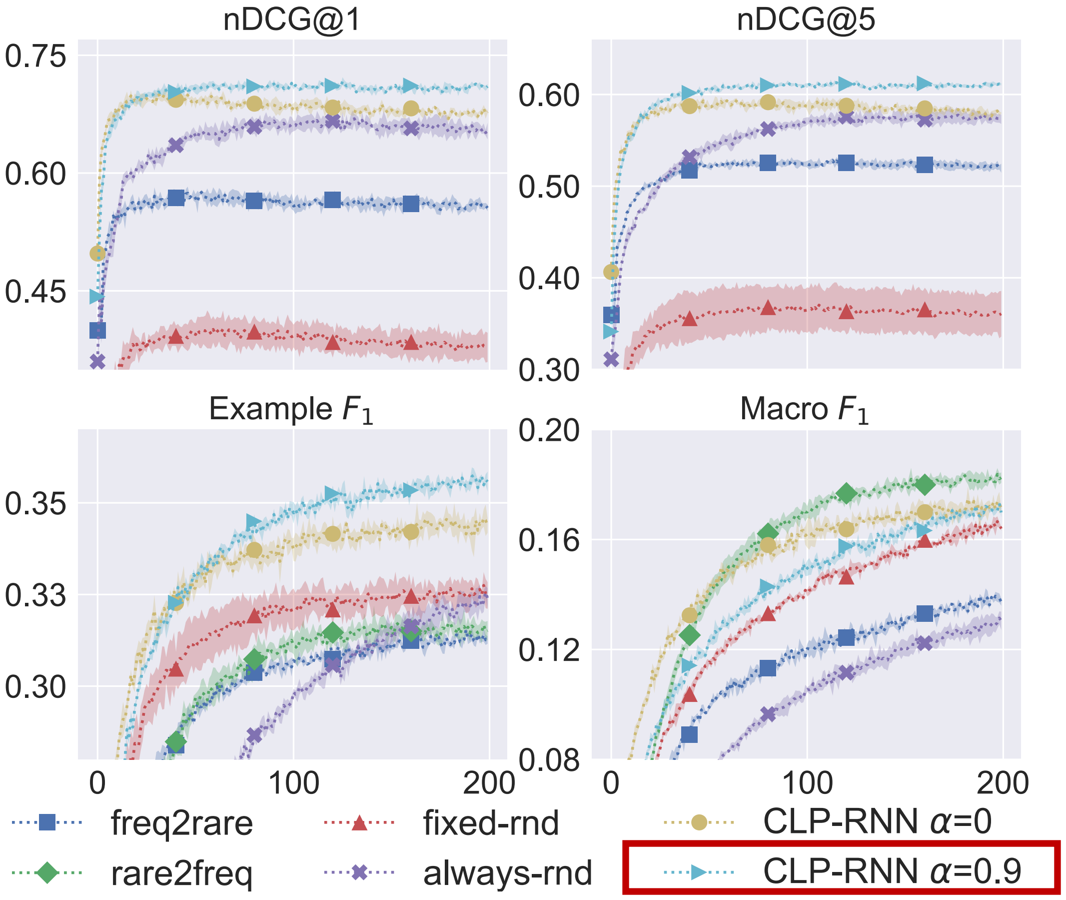
$$\nabla_\theta J(\theta) = \mathbb{E}_{P_\theta^\tau}\left[\sum_{i=0}^{T-1} \nabla_\theta \log P_\theta(a_i|s_i)(R_i - b(s_i))\right]$$

Label policy distribution



Experiments

- We combined two approaches! Context-dependent label permutation learning clearly outperforms static label permutation approaches



	Methods	Example F_1	Macro F_1	Prec@1	Prec@3	Prec@5
Mediamill	SLEEC	-	-	87.82	73.45	59.17
	FastXML	-	-	84.22	67.33	53.04
	Parabel	-	-	83.91	67.12	52.99
	freq2rare	66.63±0.33	39.68±0.69	90.05±0.31	74.20±0.18	58.39±0.29
	rare2freq	66.95±0.26	**43.33**±0.62	53.67±1.31	59.57±0.78	52.49±0.37
	fixed-rnd	67.21±0.25	41.85±0.90	73.95±5.20	65.58±2.31	55.55±0.83
	always-rnd	66.25±0.25	34.03±0.58	89.08±0.18	73.90±0.24	59.45±0.31
	CLP-RNN (α=0)	67.22±0.15	38.75±0.88	89.40±0.42	73.84±0.30	59.29±0.17
	CLP-RNN (α=0.6)	**67.27**±0.30	36.49±0.74	**91.27**±0.28	**75.25**±0.32	**59.75**±0.30
Delicious	SLEEC	-	-	67.59	61.38	56.56
	FastXML	-	-	69.61	64.12	59.27
	Parabel	-	-	67.44	61.83	56.75
	freq2rare	31.36±0.17	13.94±0.29	57.21±0.38	54.28±0.31	51.16±0.36
	rare2freq	31.60±0.15	**18.00**±0.31	17.46±0.38	18.49±0.51	20.31±0.72
	fixed-rnd	32.74±0.27	16.48±0.31	40.59±1.31	37.21±3.06	35.74±2.60
	always-rnd	32.45±0.05	13.00±0.25	66.58±0.90	60.46±0.54	54.95±0.55
	CLP-RNN (α=0)	34.43±0.54	17.33±0.17	69.57±0.43	61.57±0.69	55.73±0.56
	CLP-RNN (α=0.9)	**35.80**±0.35	**18.00**±0.51	**70.54**±0.77	**63.39**±0.65	**57.72**±0.58

Poster #233