ADAPTIVE HARD EXAMPLE MINING FOR IMAGE CAPTIONING

Yongzhuang Wang, Yangmei Shen, Hongkai Xiong, Weiyao Lin

Shanghai Jiao Tong University e-mail: wang_yongzhuang@outlook.com

INTRODUCTION

- In this paper, we propose an adaptive hard example mining method with additional supervised training for image captioning.
- Beam search algorithm is leveraged to estimate score expectation for each example. Examples whose caption scores are lower than expectation are selected automatically.
- For the selected hard examples, we propose an additional reward policy for high-scoring captions to force model learning from them. The proposed method is hyper-parameter free without tuning.

EXAMPLES

Easy Example	Captions	CIDEr
Beam search Results	A group of men standing around a table with pizza. A group of people standing in a kitchen.	1.578 1.476
Greedy Result	A group of people standing around a table with food.	1.638

Hard Example	Captions	CIDEr
Beam search Results	A man in a suit and tie standing in front of a door. A man smiling and walking through a doorway.	1.869 1.694
Greedy Result	A man in a suit and tie standing in a doorway.	1.081
Our Result	**A black and white photo of a man in a suit.**	**1.873**

Hard Example	Captions	CIDEr
Beam search Results	A little girl standing next to a red bike. A young girl standing next to a red bench.	1.685 1.176
Greedy Result	A woman standing next to a red bench on a sidewalk.	1.097
Our Result	**A little girl standing next to a red bike in a sidewalk.**	**1.852**

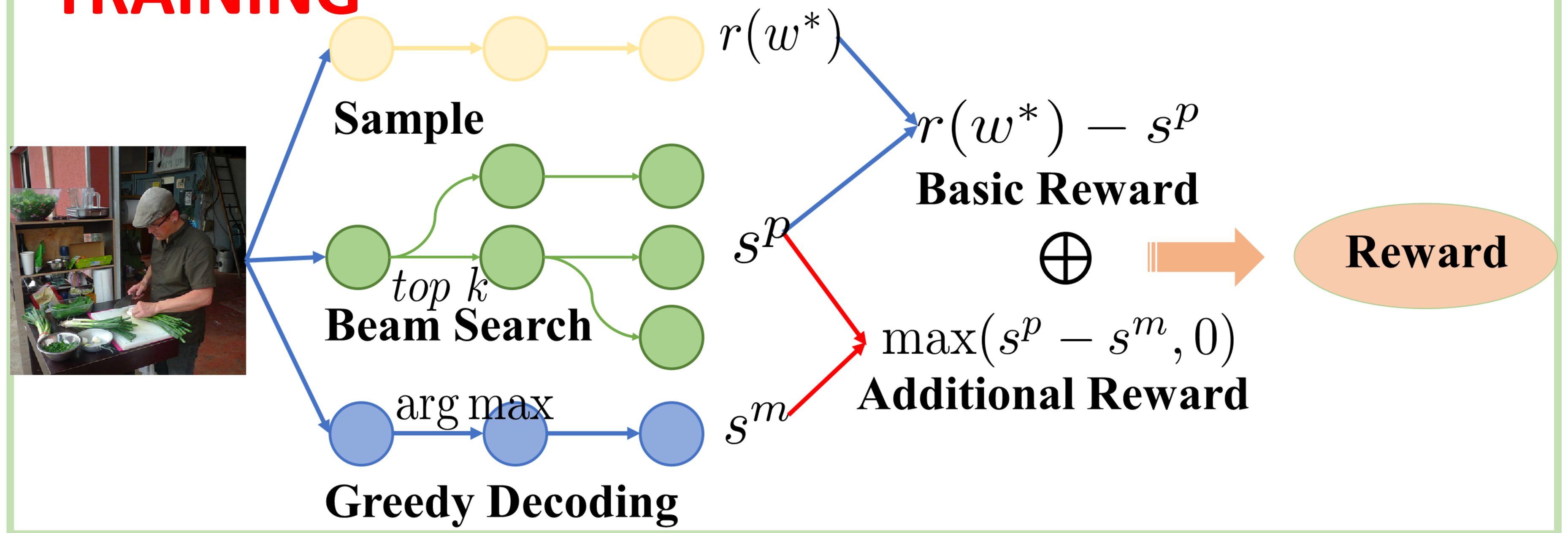
ALGORITHM

Algorithm 1 The proposed adaptive hard example mining and additional training for hard examples.

Input: The training set of image and captioning pairs;
The model parameter θ and loss function $L(\theta, r)$;
The beam size k in beam search algorithm;
Output: Model parameter θ.

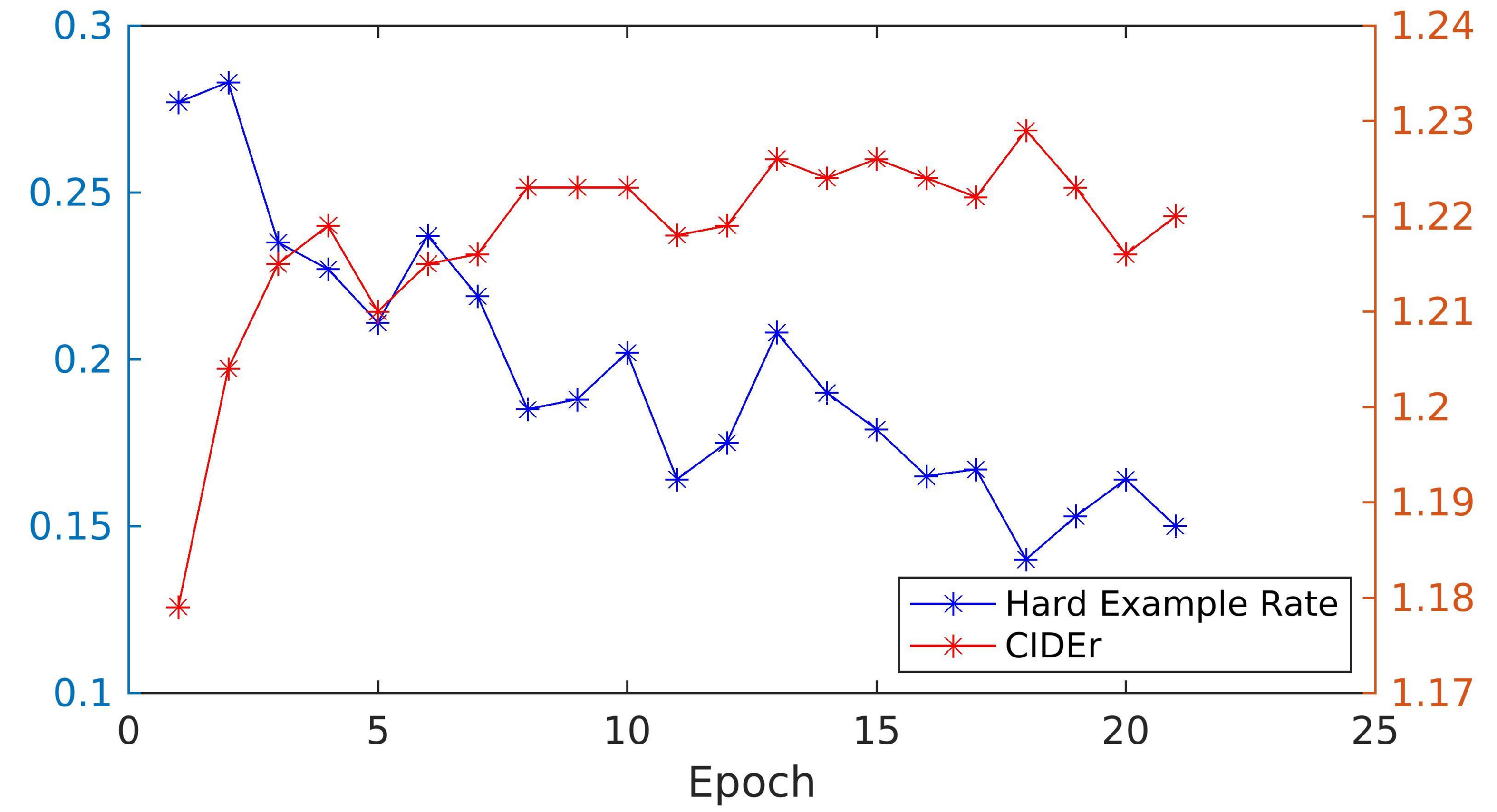
while Not converge **do**
  Given an image I, sample training caption w^* from ground-truth w^g;
  Get k-beam search results $\{w^{(1)}, \dots, w^{(k)}\}$;
  Get greedy decoding result w^m; **STEP 1. Pre-Evaluation.**
  Evaluate captions by CIDEr metric and get scores $\{r(w^*), s^m, s^{(1)}, \dots, s^{(k)}\}$;
  Get score expectation $s^p = E(\{s^{(1)}, \dots, s^{(k)}\})$ **STEP 2.**
  Get basic reward $r = r(w^*) - s^p$; **Adaptive threshold.**
  if $s^m < s^p$ **then** **STEP 3. Train hard examples.**
    Add additional $s^p - s^m$ to basic reward;
  end if
  Get loss $L(\theta, r)$;
  Update parameter θ by the policy gradient method.
end while
return Model parameter θ;

TRAINING



HARD EXAMPLE RATE

Since we select hard examples automatically in the training, the hard example rate reduces along with training and model performance improvement.



RESULTS

We train several models with the proposed method, results in the figure show that our training method can improve model performance significantly.

Model	Methods	B-4	M	R	C
Att2in [13]	XE	31.3	26.0	54.3	101.3
	MIXER∗ [9]	32.2	25.9	54.8	106.9
	SCST [13]	33.3	26.3	55.3	111.4
	Ours	**34.4**	**27.0**	**55.8**	**114.7**
Att2all [13]	XE	30.3	25.9	53.4	99.4
	MIXER [9]	32.8	25.2	26.1	110.5
	SCST [13]	34.2	26.7	55.7	114.0
	Ours	**35.9**	**27.2**	**56.1**	**117.5**
Up-Down [6]	XE	36.2	27.0	56.4	113.5
	MIXER∗ [9]	35.5	27.2	56.5	115.3
	SCST [13]	36.3	**27.7**	56.9	120.1
	Ours	**37.4**	**27.7**	**57.0**	**123.1**