Bandit Structured Prediction for Neural Seq2Seq Learning

Changzhi Sun

Bandit Structured Prediction for Neural Sequence-to-Sequence Learning

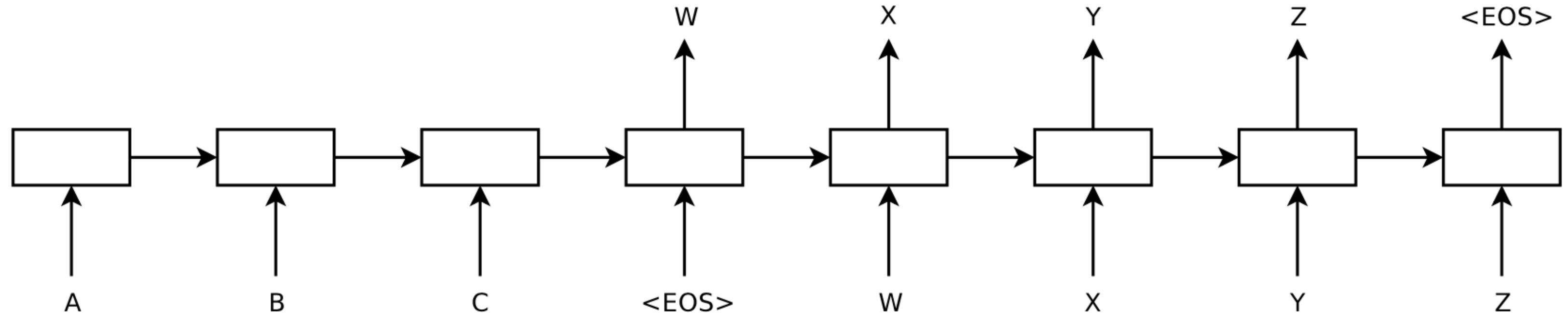
Julia Kreutzer[*] and **Artem Sokolov**[*] and **Stefan Riezler**[†,*]
[*]Computational Linguistics & [†]IWR, Heidelberg University, Germany
{kreutzer,sokolov,riezler}@cl.uni-heidelberg.de

Outline

1. Introduction
2. Neural Bandit Structured Prediction
3. Experiements
4. Conclusion

Introduction

- seq2seq learning



- data-hungry neural network model

NMT

- exposure bias
- reinforcement and imitation learning
  - learn from feedback
  - mismatch between word-level loss and sequence-level evaluation metric
  - mixture of the REINFORCE
- minimum risk training
- dual-learning

This paper

- French-to-English translation domain adaptation
- weak feedback
- control variates

Neural Bandit Structured Prediction

Algorithm 1 Neural Bandit Structured Prediction

Input: Sequence of learning rates γ_k

Output: Optimal parameters $\hat{\theta}$

1: Initialize θ_0
2: **for** $k = 0, \ldots, K$ **do**
3:     Observe $\mathbf{x}_k$
4:     Sample $\tilde{\mathbf{y}}_k \sim p_\theta(\mathbf{y}|\mathbf{x}_k)$
5:     Obtain feedback $\Delta(\tilde{\mathbf{y}}_k)$
6:     $\theta_{k+1} = \theta_k - \gamma_k \, s_k$
7: Choose a solution $\hat{\theta}$ from the list $\{\theta_0, \ldots, \theta_K\}$

Bandit Structured prediction

- bandit structured prediction is a stochastic optimization framework
- learning is performed from partial feedback
- this feedback is received in the form of task loss evaluation of a predicted output structure
- without having access to gold standard structures

Objectives

- Expected Loss(EL): expectation of a task loss $\Delta(\tilde{\mathbf{y}})$ over all input and output structures

$$L^{\mathrm{EL}}(\theta) = \mathbb{E}_{p(\mathbf{x})\, p_\theta(\tilde{\mathbf{y}}|\mathbf{x})} [\Delta(\tilde{\mathbf{y}})]$$

- Stochastic gradient:

$$s_k^{\mathrm{EL}} = \Delta(\tilde{\mathbf{y}}) \frac{\partial \log p_\theta(\tilde{\mathbf{y}}|\mathbf{x}_k)}{\partial \theta}$$

- Output structures $\tilde{\mathbf{y}}$ are sampled word by word

Sampling Structures

Algorithm 2 Sampling Structures

Input: Model θ, target sequence length limit T_y

Output: Sequence of words $\mathbf{w} = (w_1, \ldots, w_{T_y})$ and log-probability p

$w_0 = \text{START}, p_0 = 0$

$\mathbf{w} = (w_0)$

for $t \leftarrow 1 \ldots T_y$ **do**

  $w_t \sim p_\theta(w|\mathbf{x}, \mathbf{w}_{<t})$

  $p_t = p_{t-1} + \log p_\theta(w|\mathbf{x}, \mathbf{w}_{<t})$

  $\mathbf{w} = (w_1, \ldots, w_{t-1}, w_t)$

end for

Return $\mathbf{w}$ and p_T

Objectives

- Pairwise Preference Ranking(PR)

$$L^{\mathrm{PR}}(\theta) = \mathbb{E}_{p(\mathbf{x})\, p_\theta(\langle \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \rangle | \mathbf{x})} \left[\Delta(\langle \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \rangle) \right]$$

- Stochastic gradient:

$$\begin{aligned} s_k^{\mathrm{PR}} = & \Delta(\langle \tilde{\mathbf{y}}_i, \tilde{\mathbf{y}}_j \rangle) \\ & \times \left(\frac{\partial \log p_\theta(\tilde{\mathbf{y}}_i | \mathbf{x}_k)}{\partial \theta} + \frac{\partial \log p_\theta^{-}(\tilde{\mathbf{y}}_j | \mathbf{x}_k)}{\partial \theta} \right) \end{aligned}$$

Objectives

- Learn to rank $\tilde{\mathbf{y}}_i$ over $\tilde{\mathbf{y}}_j$ with pairwise feedback, either continuous (cont)

$$\Delta(\langle \mathbf{y}_i, \mathbf{y}_j \rangle) = \Delta(\mathbf{y}_j) - \Delta(\mathbf{y}_i)$$

- or binary (bin)

$$\Delta(\langle \mathbf{y}_i, \mathbf{y}_j \rangle) = \begin{cases} 1 & \text{if } \Delta(\mathbf{y}_j) > \Delta(\mathbf{y}_i) \\ 0 & \text{otherwise.} \end{cases}$$

- Draw negative sample $\tilde{\mathbf{y}}_j$ from distribution p_θ^-, one word per output structure (chosen randomly):

$$p_\theta^+(\tilde{y}_t = w_i | \mathbf{x}, \hat{\mathbf{y}}_{<t}) = \frac{\exp(o_{w_i})}{\sum_{v=1}^{V} \exp(o_{w_v})},$$
$$p_\theta^-(\tilde{y}_t = w_j | \mathbf{x}, \hat{\mathbf{y}}_{<t}) = \frac{\exp(-o_{w_j})}{\sum_{v=1}^{V} \exp(-o_{w_v})}$$

Algorithm 3 Sampling Pairs of Structures

Input: Model θ, target sequence length limit T_y

Output: Pair of sequences $\langle \mathbf{w}, \mathbf{w}' \rangle$ and their log-probability p

$p_0 = 0$
$\mathbf{w}, \mathbf{w}', \hat{\mathbf{w}} = (\text{START})$
$i \sim \mathcal{U}(1, T)$
for $t \leftarrow 1 \dots T_y$ **do**
    $\hat{w}_t = \arg\max_{w \in V} p_\theta^+(w|\mathbf{x}, \hat{\mathbf{w}}_{<t})$
    $w_t \sim p_\theta^+(w|\mathbf{x}, \hat{\mathbf{w}}_{<t})$
    $p_t = p_{t-1} + \log p_\theta^+(w_t|\mathbf{x}, \hat{\mathbf{w}}_{<t})$
    if $i = t$ **then**
        $w'_t \sim p_\theta^-(w|\mathbf{x}, \hat{\mathbf{w}}_{<t})$
        $p_t = p_t + \log p_\theta^-(w'_t|\mathbf{x}, \hat{\mathbf{w}}_{<t})$
    else
        $w'_t \sim p_\theta^+(w|\mathbf{x}, \hat{\mathbf{w}}_{<t})$
        $p_t = p_t + \log p_\theta^+(w'_t|\mathbf{x}, \hat{\mathbf{w}}_{<t})$
    end if
    $\mathbf{w} = (w_1, \dots, w_{t-1}, w_t)$
    $\mathbf{w}' = (w'_1, \dots, w'_{t-1}, w'_t)$
    $\hat{\mathbf{w}} = (\hat{w}_1, \dots, \hat{w}_{t-1}, \hat{w}_t)$
end for
Return $\langle \mathbf{w}, \mathbf{w}' \rangle$ and p_T

Control Variates

Augment a random variable X (here: $X = s_k$) by another random variable Y, the control variate. With $\bar{Y} = \mathbb{E}[Y]$, $X - \hat{c}\,Y + \hat{c}\,\bar{Y}$ is an unbiased estimator of $\mathbb{E}[X]$. Control variates with high Cov(X, Y) **reduce the variance** of the gradient estimate. Two choices here:

1. **Baseline (BL)** [2]:

$$Y_k = \nabla \log p_\theta(\tilde{\mathbf{y}}|\mathbf{x}_k) \frac{1}{k} \sum_{j=1}^{k} \Delta(\tilde{\mathbf{y}}_j).$$

2. **Score Function (SF)** [3]:

$$Y_k = \nabla \log p_\theta(\tilde{\mathbf{y}}|\mathbf{x}_k).$$

Experiments

Neural machine translation **domain adaptation**:

- Adapt a pre-trained model (Europarl, fr-en) to new domains (News Commentary and TED).
- **Simulated feedback** with GLEU on references
- Encoder-decoder architecture with attention
- Full-information baselines: maximum likelihood estimation on reference translations

Strategies for **handling of unknown words**:

1. attention-based replacement of UNKs for word-based models [4]
2. sub-word models with Byte-Pair-Encoding (BPE) [5]

Domain	**Version**	**Train**	**Valid.**	**Test**
Europarl	v.5	1.6M	2k	2k
News Commentary	WMT07	40k	1k	2k
TED	TED2013	153k	2k	2k

Table 1: Number of parallel sentences for training, validation and test sets for French-to-English domain adaptation.

Algorithm	**Train data**	**Iter.**	**EP**	**NC**	**TED**
MLE	EP	12.3M	31.44	26.98	23.48
MLE-UNK			31.82	28.00	24.59
MLE-BPE		12.0M	31.81	27.20	24.35

Table 2: Out-of-domain NMT baseline results (BLEU) on in- and out-of-domain test sets trained only on EP data.

Algorithm	Train data	Iter.	EP	NC
MLE	NC	978k	13.67	22.32
MLE-UNK			13.83	22.56
MLE-BPE		1.0M	14.09	23.01
MLE	EP→NC	160k	26.66	31.91
MLE-UNK			27.19	33.19
MLE-BPE		160k	27.14	33.31
Algorithm	**Train data**	**Iter.**	**EP**	**TED**
MLE	TED	2.2M	14.16	32.71
MLE-UNK			15.15	33.16
MLE-BPE		3.0M	14.18	32.81
MLE	EP→TED	460k	23.88	33.65
MLE-UNK			24.64	35.57
MLE-BPE		2.2M	23.39	36.23

Table 3: In-domain NMT baselines results (BLEU) on in- and out-of-domain test sets. The EP→NC is first trained on EP, then fine-tuned on NC. The EP→TED is first trained on EP, then fine-tuned on TED.

Algorithm	Iter.	EP	NC	Diff.
EL	317k	$30.36_{\pm 0.20}$	$29.34_{\pm 0.29}$	2.36
EL-UNK*	317k	$30.73_{\pm 0.20}$	$30.33_{\pm 0.42}$	2.33
EL-UNK**	349k	$30.67_{\pm 0.04}$	$30.45_{\pm 0.27}$	2.45
EL-BPE	543k	$30.09_{\pm 0.31}$	$30.09_{\pm 0.01}$	2.89
PR-UNK** (bin)	22k	$30.76_{\pm 0.03}$	$29.40_{\pm 0.02}$	1.40
PR-BPE (bin)	14k	$31.02_{\pm 0.09}$	$28.92_{\pm 0.03}$	1.72
PR-UNK** (cont)	12k	$30.81_{\pm 0.02}$	$29.43_{\pm 0.02}$	1.43
PR-BPE (cont)	8k	$30.91_{\pm 0.01}$	$28.99_{\pm 0.00}$	1.79
SF-EL-UNK**	713k	$29.97_{\pm 0.09}$	$30.61_{\pm 0.05}$	2.61
SF-EL-BPE	375k	$30.46_{\pm 0.10}$	$30.20_{\pm 0.11}$	3.00
BL-EL-UNK**	531k	$30.19_{\pm 0.37}$	$31.47_{\pm 0.09}$	3.47
BL-EL-BPE	755k	$29.88_{\pm 0.07}$	$31.28_{\pm 0.24}$	**4.08**

(a) Domain adaptation from EP to NC.

Table 4: Bandit NMT results (BLEU) on EP, NC and TED test sets. UNK* models involve UNK replacement only during testing, UNK** include UNK replacement already during training. For PR, either binary (bin) or continuous feedback (cont) was used. Control variates: average reward baseline (BL) and score function (SF). Results are averaged over two independent runs and standard deviation is given in subscripts. Improvements over respective out-of-domain models are given in the Diff.-columns.

Algorithm	**Iter.**	**EP**	**TED**	**Diff.**
EL	976k	$29.34_{\pm 0.42}$	$27.66_{\pm 0.03}$	4.18
EL-UNK*	976k	$29.68_{\pm 0.29}$	$29.44_{\pm 0.06}$	4.85
EL-UNK**	1.1M	$29.62_{\pm 0.15}$	$29.77_{\pm 0.15}$	5.18
EL-BPE	831k	$30.03_{\pm 0.43}$	$28.54_{\pm 0.04}$	4.18
PR-UNK** (bin)	14k	$31.84_{\pm 0.01}$	$24.85_{\pm 0.00}$	0.26
PR-BPE (bin)	69k	$31.77_{\pm 0.01}$	$24.55_{\pm 0.01}$	0.20
PR-UNK** (cont)	9k	$31.85_{\pm 0.02}$	$24.85_{\pm 0.01}$	0.26
PR-BPE (cont)	55k	$31.79_{\pm 0.02}$	$24.59_{\pm 0.01}$	0.24
SF-EL-UNK**	658k	$30.18_{\pm 0.15}$	$29.12_{\pm 0.10}$	4.53
SF-EL-BPE	590k	$30.32_{\pm 0.26}$	$28.51_{\pm 0.18}$	4.16
BL-EL-UNK**	644k	$29.91_{\pm 0.03}$	$30.44_{\pm 0.13}$	5.85
BL-EL-BPE	742k	$29.84_{\pm 0.61}$	$30.24_{\pm 0.46}$	**5.89**

(b) Domain adaptation from EP to TED.

Table 4: Bandit NMT results (BLEU) on EP, NC and TED test sets. UNK* models involve UNK replacement only during testing, UNK** include UNK replacement already during training. For PR, either binary (bin) or continuous feedback (cont) was used. Control variates: average reward baseline (BL) and score function (SF). Results are averaged over two independent runs and standard deviation is given in subscripts. Improvements over respective out-of-domain models are given in the Diff.-columns.

Conclusion

- Successful training of NMT with weak feedback
- Large improvements for domain adaptation, outperforming linear models
- Control variates improve generalization

Thanks Q&A