From Credit Assignment to Entropy Regularization: Two New Algorithms for Neural Sequence Prediction

Zihang Dai*, Qizhe Xie*, Eduard Hovy

Language Technologies Institute, Carnegie Mellon University (*: equal contribution)

Background Information

For any ground-truth **sequence** pair $(\mathbf{x}^\star, \mathbf{y}^\star)$, training objective:

$$\mathrm{KL}\Big(\underbrace{P^\star(\mathbf{Y}\mid\mathbf{x}^\star,\mathbf{y}^\star)}_{\text{target distribution}} \,\Big\|\, \underbrace{P_\theta(\mathbf{Y}\mid\mathbf{x}^\star)}_{\text{model distribution}}\Big)$$

- Maximum Likelihood Estimation (MLE) target

$$P^\star_{\text{MLE}}(\mathbf{Y}=\mathbf{y}\mid\mathbf{x}^\star,\mathbf{y}^\star) = \delta_{\mathbf{y}=\mathbf{y}^\star}$$

  - Exposure bias: delta distribution → no exploration
  - Metric bias: train/test metric discrepancy

- Reward Augmented Maximum Likelihood (RAML) target

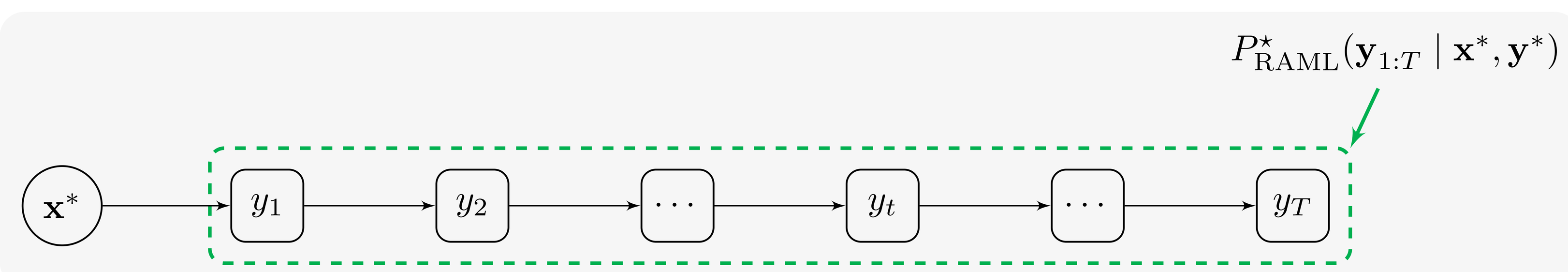
$$P^\star_{\text{RAML}}(\mathbf{Y}=\mathbf{y}\mid\mathbf{x}^\star,\mathbf{y}^\star) = \frac{\exp\left(R(\mathbf{y};\mathbf{y}^\star)/\tau\right)}{\sum_{\mathbf{y}'\in\mathcal{Y}}\exp\left(R(\mathbf{y}';\mathbf{y}^\star)/\tau\right)}$$

  - Incorporate test metric $R(\cdot;\cdot)$ into the training objective
  - Assign probabilities to similar but non-identical sequences
  - $\lim_{\tau\to 0} P^\star_{\text{RAML}}(\mathbf{Y}\mid\mathbf{x}^\star,\mathbf{y}^\star) = P^\star_{\text{MLE}}(\mathbf{Y}\mid\mathbf{x}^\star,\mathbf{y}^\star)$

Better Credit Assignment for RAML

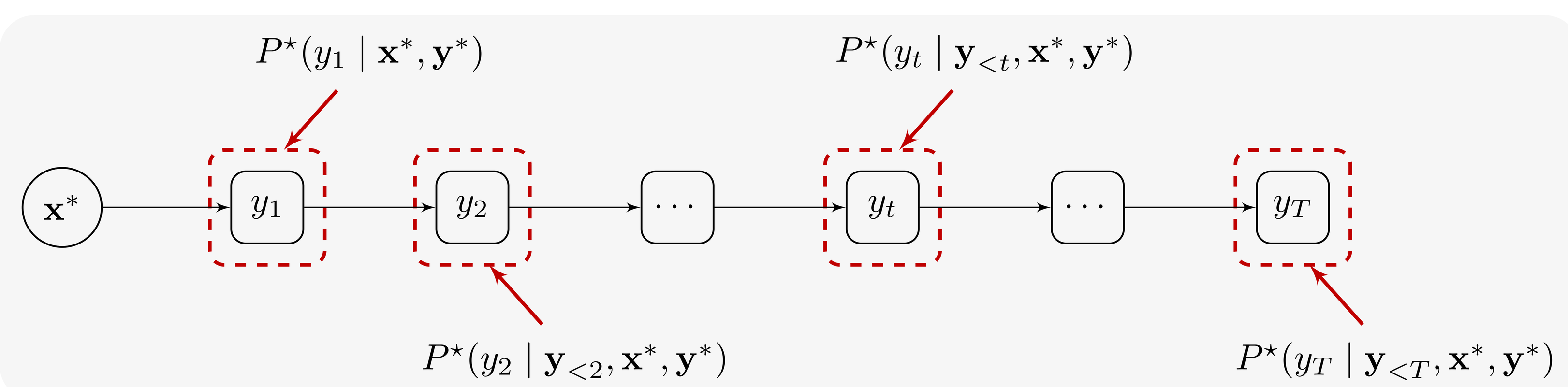
"Coarse" and **"Inefficient" Credit Assignment** of RAML:

- Target distribution is based on the reward of an "**entire sequence**" (coarse)
- Exponential sequence space (inefficient)



Sequence-level Credit Assignment

Token-level Credit Assignment



Find the token-level target distribution $P^\star(\mathbf{y}_t\mid\mathbf{y}_{<t},\mathbf{x}^\star,\mathbf{y}^\star)$, such that

$$P^\star_{\text{RAML}}(\mathbf{y}_{1:T}\mid\mathbf{x}^\star,\mathbf{y}^\star) = \prod_{t=1}^{T}\underbrace{P^\star(y_t\mid\mathbf{y}_{<t},\mathbf{x}^\star,\mathbf{y}^\star)}_{\textbf{Proposed Method}}$$

Token-level Distribution & Entropy-Regularized RL

Theorem 1. The token-level target distribution has the form

$$P^\star(y_t\mid\mathbf{y}_{<t},\mathbf{x}^\star,\mathbf{y}^\star) = \frac{\exp\left(Q^\star(\mathbf{y}_{<t},y_t)/\tau\right)}{\sum_{w\in\mathcal{W}}\exp\left(Q^\star(\mathbf{y}_{<t},w)/\tau\right)}, \tag{1}$$

where $Q^\star(\mathbf{y}_{<t},y_t)$ is the optimal soft-Q function defined by the **entropy-regularized** MDP

$$Q^\star(\underbrace{\mathbf{y}_{<t}}_{\textbf{State}},\underbrace{y_t}_{\textbf{Action}}) = \underbrace{\left[R(\mathbf{y}_{\le t}) - R(\mathbf{y}_{<t})\right]}_{\substack{\textbf{Immediate reward}\\ r(\mathbf{y}_{<t},y_t)}} + \underbrace{\tau\log\sum_{w'\in\mathcal{W}}\exp\left(Q^\star(\mathbf{y}_{\le t},w')/\tau\right)}_{\substack{\textbf{Value after taking step } y_t\\ V^\star(\mathbf{y}_{\le t})}}. \tag{2}$$

- Intuitively, a larger optimal Q value $\implies$ a higher target probability $P^\star$
- Using **reinforcement learning** to solve the MDP in Eqn. (2) gives both $Q^\star$ and $P^\star$
- $Q^\star$ can depend on both $\mathbf{x}^\star$ and $\mathbf{y}^\star$, i.e. $Q^\star(\mathbf{y}_{<t},y_t;\mathbf{x}^\star,\mathbf{y}^\star)$

Algorithm 1: Value Augmented Maximum Likelihood (VAML)

1. Solve the MDP in Eqn. (2) with **Soft Q-Learning**:

$$\min_\phi \left\| Q_\phi\left(\mathbf{y}_{<t},y_t;\mathbf{y}^\star\right) - \left[r(\mathbf{y}_{<t},y_t) + \tau\log\sum_{w'\in\mathcal{W}}\exp\left(Q_\phi\left(\mathbf{y}_{\le t},w';\mathbf{y}^\star\right)/\tau\right)\right]\right\|^2.$$

2. Minimize the **token-level KL** divergence based on the VAML target:

$$\min_\theta \mathrm{KL}\Big(P^\star_{\text{VAML}}\left(Y_t\mid\mathbf{y}_{<t},\mathbf{y}^\star\right) \,\Big\|\, P_\theta\left(Y_t\mid\mathbf{y}_{<t},\mathbf{x}^\star\right)\Big), \text{ with } P^\star_{\text{VAML}} = \frac{\exp\left(Q_\phi(\mathbf{y}_{<t},y_t;\mathbf{y}^\star)/\tau\right)}{\sum_{w\in\mathcal{W}}\exp\left(Q_\phi(\mathbf{y}_{<t},w;\mathbf{y}^\star)/\tau\right)}.$$

Algorithm 2: Entropy-Regularized Actor Critic (ERAC)

- Critic: trained to **evaluate** the Q-value of the current policy
- Actor: trained to **improve** the policy given the critic

For trajectory $\mathbf{y}$ from the current policy $P_\theta(\mathbf{Y}\mid\mathbf{x}^\star) = \prod_{t=1}^{T}\pi_\theta(Y_t\mid\mathbf{y}_{<t},\mathbf{x}^\star)$

$$\textbf{Critic: } \min_\phi \left\| Q_\phi\left(\mathbf{y}_{<t},y_t;\mathbf{y}^\star\right) - \overbrace{\left[r(\mathbf{y}_{<t},y_t) + \tau\underbrace{\mathcal{H}(\pi_\theta(Y_{t+1}\mid\mathbf{y}_{\le t}))}_{\text{Future Entropy}} + \sum_{w'\in\mathcal{W}}\pi_\theta\left(w'\mid\mathbf{y}_{\le t},\mathbf{x}^\star\right)Q_{\bar\phi}\left(\mathbf{y}_{\le t},w';\mathbf{y}^\star\right)\right]}^{\text{Target } Q\text{-value based on the target network } Q_{\bar\phi}}\right\|^2$$

$$\textbf{Actor: } \max_\theta \underbrace{\sum_{w\in\mathcal{W}}\pi_\theta(w\mid\mathbf{y}_{<t},\mathbf{x}^\star)Q_\phi\left(\mathbf{y}_{<t},w';\mathbf{y}^\star\right)}_{\text{Current value estimate}} + \tau\underbrace{\mathcal{H}(\pi_\theta(Y_t\mid\mathbf{y}_{<t}))}_{\text{Current Entropy}}$$

Stability techniques:

- Target network $Q_{\bar\phi}$ with delayed parameters $\hat\phi$
- Smooth Q-values by minimizing their "variances", i.e.,

$$\min_\phi \lambda_{\text{var}}\sum_{w\in\mathcal{W}}\left[Q_\phi(y_{<t},w;\mathbf{y}^\star) - \bar{Q}_\phi(y_{<t};\mathbf{y}^\star)\right], \quad \text{where } \bar{Q}_\phi(y_{<t};\mathbf{y}^\star) = \frac{1}{|\mathcal{W}|}\sum_{w'\in\mathcal{W}}Q_\phi(y_{<t},w';\mathbf{y}^\star).$$

Experiments

Machine Translation:

Comparison with existing works

Algorithm	BLEU
MIXER (Ranzato et al., 2015)	20.73
BSO (Wiseman and Rush, 2016)	27.9
Q(BLEU) (Li et al., 2017)	28.3
AC (Bahdanau et al., 2016)	28.53
RAML (Ma et al., 2017)	28.77
VAML	28.94
ERAC	**29.36**

- Dataset: IWSLT 2014 de-en
- Architecture: a seq2seq model with the dot-product attention
- Average performance of 9 different runs

Comparison with direct baselines

Algorithm	MT (w/o input feeding)			MT (w/ input feeding)		
	Mean	Min	Max	Mean	Min	Max
MLE	27.01 ± 0.20	26.72	27.27	28.06 ± 0.15	27.84	28.22
RAML	27.74 ± 0.15	27.47	27.93	28.56 ± 0.15	28.35	28.80
VAML	**28.16 ± 0.11**	**28.00**	**28.26**	**28.84 ± 0.10**	**28.62**	**28.94**
AC	28.04 ± 0.05	27.97	28.10	29.05 ± 0.06	28.95	29.16
ERAC	**28.30 ± 0.06**	**28.25**	**28.42**	**29.31 ± 0.04**	**29.26**	**29.36**

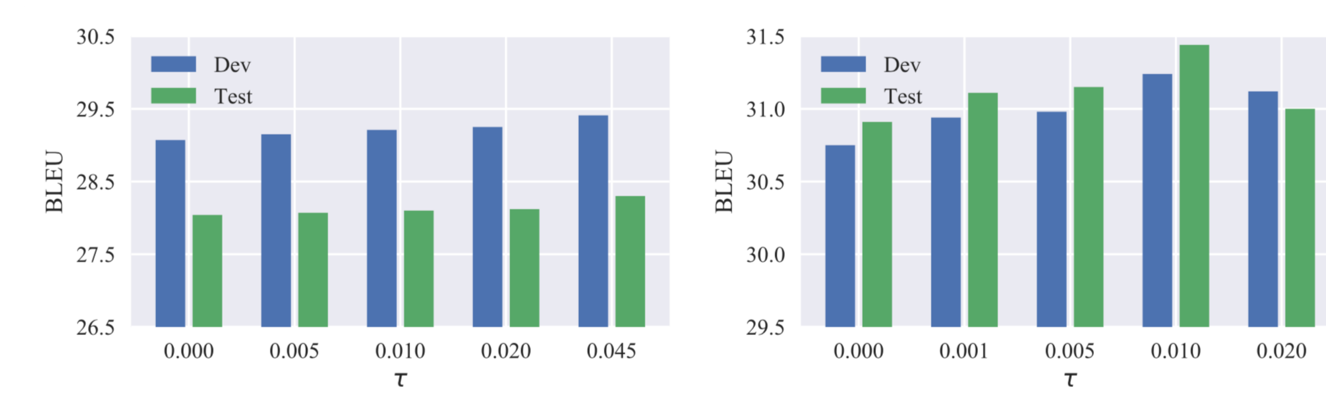
Image Captioning:

Algorithm	Image Captioning		
	Mean	Min	Max
MLE	29.54 ± 0.21	29.27	29.89
RAML	29.84 ± 0.21	29.50	30.17
VAML	**29.93 ± 0.22**	**29.51**	**30.24**
AC	30.90 ± 0.20	30.49	31.16
ERAC	**31.44 ± 0.22**	**31.07**	**31.82**

- Dataset: MSCOCO
- Architecture: the NIC model with a pretrained 101-layer ResNet encoder
- Evaluation metric: BLEU-4

Ablation study of ERAC:

Performance with different levels of entropy



(a) Machine translation (b) Image captioning

Importance of stability techniques

λ_{var} \ β	0.001	0.01	0.1	1
0	27.91	26.27†	28.88	27.38†
0.001	**29.41**	29.26	29.32	27.44

- $\beta = 1$: no target network
- $\lambda_{\text{var}} = 0$: no smoothing technique
- † indicates excluding extreme values due to divergence

Code available at: https://github.com/zihangdai/ERAC-VAML