How (Not) to Train Your Generative Model: Scheduled Sample, Likelihood, Adversary?

Ferenc Huszar

Balderton Capital LLP, London, UK

ferenc.huszar@gmail.com

Contribution

- Present a critique of scheduled sampling --- the **objective function** underlying scheduled sampling is **improper** and leads to an **inconsistent** learning algorithm.
- Revisit the problems that scheduled sampling was meant to address, and present an **alternative interpretation**.
- Introduce a generalization of **adversarial** training, and show how such method can interpolate between maximum likelihood training and our ideal training objective.

Scheduled Sampling

- For the *n*-th symbol we draw from a Bernoulli distribution with parameter ε to decide whether we keep the original symbol or use one generated by the model
- If we decided to replace the symbol, we use the current model RNN to output the predictive distribution of the next symbol given the current prefix, and sample from this predictive distribution
- We add to the training loss the log predictive probability of the real *n*-th symbol, given the prefix (the prefix at this point may already contain generated characters)
- Depending on the coinflip above, the original or simulated character is added to the prefix and we continue with the recursion

Critique to Scheduled Sampling

- Only consider a sequence of length 2 --- s = [x_1 x_2]

$$\begin{aligned} D_{ML}[P\|Q] &= KL[P\|Q] \\ &= KL[P_{x_1}\|Q_{x_1}] + \mathbb{E}_{z\sim P_{x_1}} KL[P_{x_2|x_1=z}\|Q_{x_2|x_1=z}] \end{aligned}$$

$$\begin{aligned} D_{alternative}[P\|Q] &= KL[P_{x_1}\|Q_{x_1}] + \mathbb{E}_{y\sim P_{x_1}} \mathbb{E}_{z\sim Q_{x_1}} KL[P_{x_2|x_1=y}\|Q_{x_2|x_1=z}] \\ &= KL[P_{x_1}\|Q_{x_1}] + \mathbb{E}_{z\sim Q_{x_1}} KL[P_{x_2}\|Q_{x_2|x_1=z}] \end{aligned}$$

$$D_{SS}[P\|Q] = KL[P_{x_1}\|Q_{x_1}] + \epsilon \mathbb{E}_{z\sim P_{x_1}} KL[P_{x_2|x_1=z}\|Q_{x_2|x_1=z}] + (1-\epsilon) \mathbb{E}_{z\sim Q_{x_1}} KL[P_{x_2}\|Q_{x_2|x_1=z}]$$

$$\begin{aligned} D_{SS}[P\|Q] &= KL[P_{x_1}\|Q_{x_1}] + \mathbb{E}_{z\sim P_{x_1}} KL\left[\epsilon P_{x_1|x_1=z} + \frac{Q_{x_1}(z)}{Q_{x_1}(z)} P_{x_2} \middle\| Q_{x_2|x_1=z}\right] + C_{P,\epsilon} \\ &= KL\left[P_{x_1}\left(\epsilon P_{x_2|x_1} + (1-\epsilon)\frac{Q_{x_1}P_{x_2}}{P_{x_1}}\right) \middle\| Q_{x_1,x_2}\right] + C_{P,\epsilon} \end{aligned}$$

- As ε change from 1 -> 0, the global optimum is between the true joint P and the factorized distribution $P_{x1}P_{x2}$

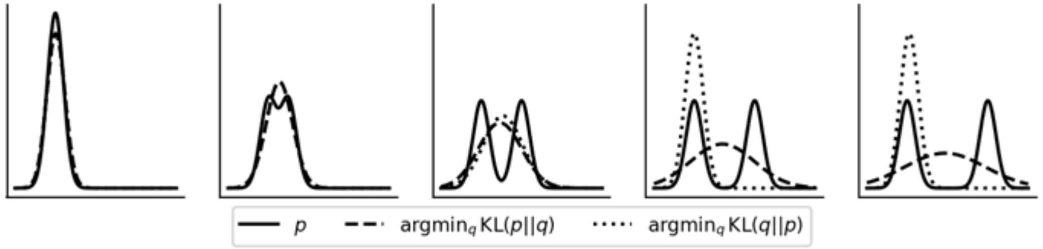
Two Assumption and A Conclusion

- Perceived quality of each sample is related to the
  - *surprisal* – log $Q_{human}(x)$

- Human observer has learnt an accurate model of the natural distribution of stimuli.
  - $Q_{human}(x) = P(x)$

- Safer is better!

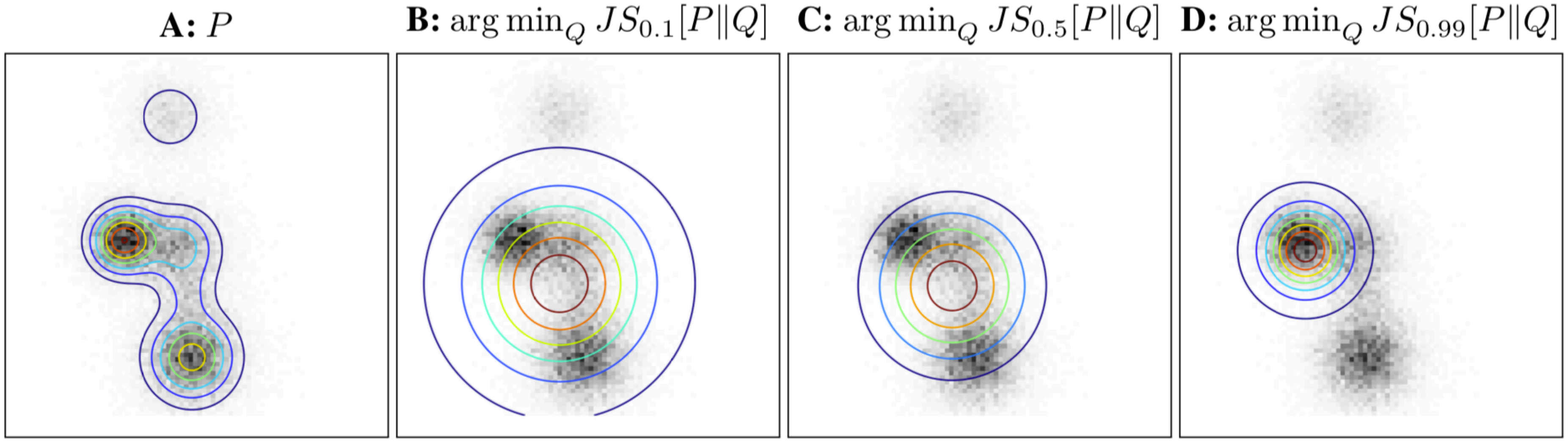
KL(q||p) and KL(p||q)



- Use KL(p||q) as an example
  - $KL(p||q) = \sum_s p(s)\, log \frac{p(s)}{q(s)}$
  - $q(s) > 0$ and $p(s)$ -> 0, KL -> 0, make the model generate some samples that do not locate on the data distribution.
- We need to minimize KL(q||p), but
  - It is only well-defined when P is positive and bounded in the full support of Q
  - P is an empirical distribution of samples in reality
  - Q is a smooth probabilistic model in reality

Generalized Adversarial Training

$$JS_{\pi}[P\|Q] = \pi \cdot KL[P\|\pi P + (1-\pi)Q] + (1-\pi)KL[Q\|\pi P + (1-\pi)Q]$$



A: P

B: $\arg\min_Q JS_{0.1}[P\|Q]$

C: $\arg\min_Q JS_{0.5}[P\|Q]$

D: $\arg\min_Q JS_{0.99}[P\|Q]$

Conclusion

- Maximum likelihood **should not be used as the training objective** if the end goal is to draw realistic samples from the model.
- Scheduled sampling, designed to overcome the shortcomings of maximum likelihood, **fails to address the fundamental problems**.
- We theorize that $KL[Q||P]$ could be used as an **idealized objective function**, but it is impractical to use in practice.
- We propose the generalized **Jensen-Shannon divergence** as a promising, more tractable objective function

CoT: Cooperative Training for Generative Modeling of Discrete Data

Sidi Lu, Lantao Yu, Siyuan Feng, Yaoming Zhu, Weinan Zhang, Yong Yu

Motivation & Contribution

- To exploit the supervision signal from the discriminator, most previous models leverage REINFORCE to address the non-differentiable problem of sequential discrete data, which introduces high variance and makes the model training quite unstable.

- To deal with such a problem, this paper propose a novel approach called Cooperative Training (CoT) to improve the training of sequence generative models.

Limitation of MLE and SeqGAN

- MLE
    - $KL(p||q) = \sum_s p(s)\, log \frac{p(s)}{q(s)}$
    - $q(s) > 0$ and $p(s)$ -> 0, KL -> 0, make the model generate some samples that do not locate on the data distribution.

- SeqGAN
    - High variance, which relies on pre-training via Maximum Likelihood Estimation
    - Mode collapse, which is cause by the reverse KL divergence.

Methodology

- CoT coordinately trains a generative module G, and an auxiliary predictive module M , called mediator, for guiding G in a cooperative fashion.
- The M is going to simulation the distribution of $\frac{1}{2}(p_{data} + G_\theta)$

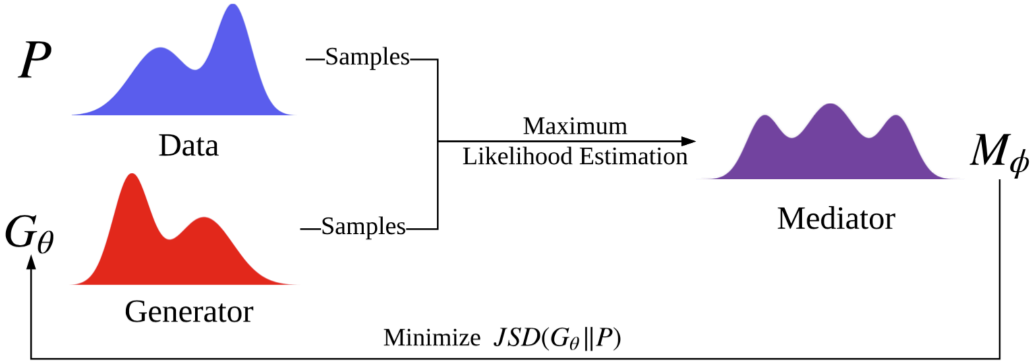


Figure 1. Process of Cooperative Training.

Algorithm Derivation_objective for mediator

- Mediator M_ϕ, which is a density function that estimates a mixture distribution of the learned generative distribution G_θ and target latent distribution p_{data}.

$$M_\phi \simeq \frac{1}{2}(p_{\text{data}} + G_\theta).$$

Lemma 1 (Mixture Density Decomposition)

$$\begin{aligned}
&\nabla_\phi J_m(\phi) \\
=&\nabla_\phi KL(M^* \| M_\phi) \\
=&\nabla_\phi \mathop{\mathbb{E}}_{s\sim M^*}\left[\log \frac{M^*(s)}{M_\phi(s)}\right] \\
=&\nabla_\phi \Big(- \mathop{\mathbb{E}}_{s\sim M^*}[\log M_\phi(s)]\Big) \\
=&\nabla_\phi \frac{1}{2}\Big(\mathop{\mathbb{E}}_{s\sim G_\theta}[-\log(M_\phi(s))] + \mathop{\mathbb{E}}_{s\sim p_{data}}[-\log(M_\phi(s))]\Big)
\end{aligned}$$

- The objective $J_m(\phi)$ for the mediator M parameterized by ϕ therefore becomes

$$J_m(\phi) = \frac{1}{2}\Big(\mathop{\mathbb{E}}_{s\sim G_\theta}[-\log(M_\phi(s))] + \mathop{\mathbb{E}}_{s\sim p_{\text{data}}}[-\log(M_\phi(s))]\Big)$$

Algorithm Derivation_generator object

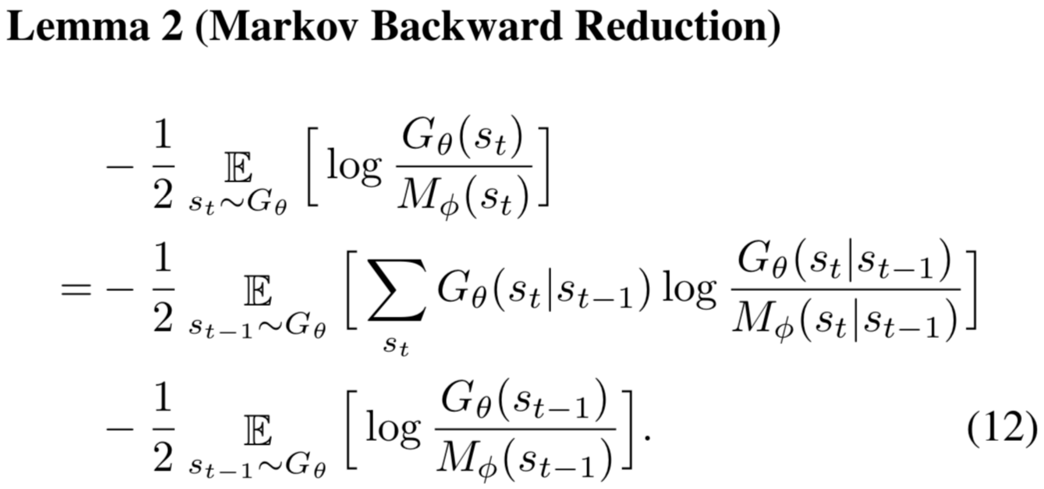
- The mediator is exploited to optimize an estimated Jensen-Shannon divergence for G_θ

$$
\begin{aligned}
& J_g(\theta) \\
= & -J\hat{S}D(G_\theta \| p_{\text{data}}) \\
= & -\frac{1}{2}\left[KL(G_\theta \| M_\phi) + KL(p_{\text{data}} \| M_\phi)\right] \\
= & -\frac{1}{2} \mathop{\mathbb{E}}_{s \sim G_\theta}\left[\log \frac{G_\theta(s)}{M_\phi(s)}\right] - \frac{1}{2} \mathop{\mathbb{E}}_{s \sim p_{\text{data}}}\left[\log \frac{p_{\text{data}}(s)}{M_\phi(s)}\right]
\end{aligned}
$$

- When calculating $\nabla\theta\ J_g\ (\theta)$, the second term has no effect on the final results. Thus, we could use this objective instead

$$
J_g(\theta) = -\frac{1}{2} \mathop{\mathbb{E}}_{s \sim G_\theta}\left[\log \frac{G_\theta(s)}{M_\phi(s)}\right]
$$

Algorithm Derivation—markov backward reduction



Lemma 2 (Markov Backward Reduction)

$$
\begin{aligned}
&-\frac{1}{2}\underset{s_t\sim G_\theta}{\mathbb{E}}\left[\log\frac{G_\theta(s_t)}{M_\phi(s_t)}\right]\\
=&-\frac{1}{2}\underset{s_{t-1}\sim G_\theta}{\mathbb{E}}\left[\sum_{s_t}G_\theta(s_t|s_{t-1})\log\frac{G_\theta(s_t|s_{t-1})}{M_\phi(s_t|s_{t-1})}\right]\\
&-\frac{1}{2}\underset{s_{t-1}\sim G_\theta}{\mathbb{E}}\left[\log\frac{G_\theta(s_{t-1})}{M_\phi(s_{t-1})}\right]. \qquad (12)
\end{aligned}
$$

- After recursively using Equation 12， we can get

$$J_g(\theta)=\sum_{t=0}^{n-1}\underset{s_t\sim G_\theta}{\mathbb{E}}\left[\pi_g(s_t)^\top(\log\pi_m(s_t)-\log\pi_g(s_t))\right]$$

- Up to now, we are still not free from REINFORCE, as the objective incorporates expectation over the learned distribution G_θ

$$
\begin{aligned}
&\nabla_\theta J_g(\theta)\\
=&\nabla_\theta\left(\sum_{t=0}^{n-1}\underset{s_t\sim G_\theta}{\mathbb{E}}\left[\pi_g(s_t)^\top(\log\pi_m(s_t)-\log\pi_g(s_t))\right]\right)
\end{aligned}
$$

Algorithm Derivation_factorizing the cumulative gradient

$$
\begin{aligned}
&\nabla_\theta J_{g,t}(\theta) \\
=&\nabla_\theta \left[\mathop{\mathbb{E}}_{s_t \sim G_\theta} \pi_g(s_t)^\top (\log \pi_m(s_t) - \log \pi_g(s_t)) \right] \\
=&\nabla_\theta \left[\sum_{s_t} G_\theta(s_t)(\pi_g(s_t)^\top (\log \pi_m(s_t) - \log \pi_g(s_t))) \right] \\
=&\sum_{s_t} \nabla_\theta \left[G_\theta(s_t)(\pi_g(s_t)^\top (\log \pi_m(s_t) - \log \pi_g(s_t))) \right]
\end{aligned}
$$

Let

$$L(s_t) = \pi_g(s_t)^\top (\log \pi_m(s_t) - \log \pi_g(s_t)),$$

then

$$
\begin{aligned}
&\nabla_\theta J_{g,t}(\theta) \\
=&\sum_{s_t} (\nabla_\theta G_\theta(s_t) L(s_t) + G_\theta(s_t) \nabla_\theta L(s_t)) \\
=&\sum_{s_t} G_\theta(s_t) \left(\nabla_\theta \log G_\theta(s_t) L(s_t) + \nabla_\theta L(s_t) \right) \\
=&\mathop{\mathbb{E}}_{s_t \sim G_\theta} \nabla_\theta [\text{stop\_gradient}(L(s_t)) \log G_\theta(s_t) + L(s_t)]
\end{aligned}
$$

$$
\begin{aligned}
&\nabla_\theta J^\gamma_{g,t}(\theta) \\
=&\mathop{\mathbb{E}}_{s_t \sim G_\theta} \nabla_\theta \left[\gamma(\text{stop\_gradient}(L(s_t)) \log G_\theta(s_t)) + L(s_t) \right]
\end{aligned}
$$

Experiment---universal sequence modeling in synthetic Turing test

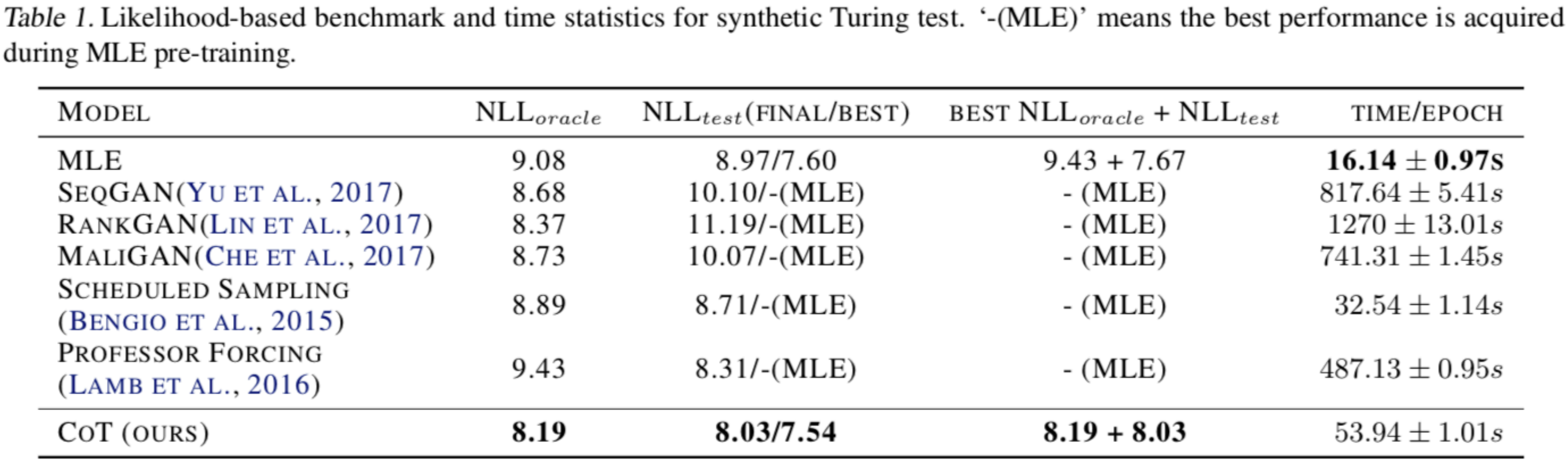
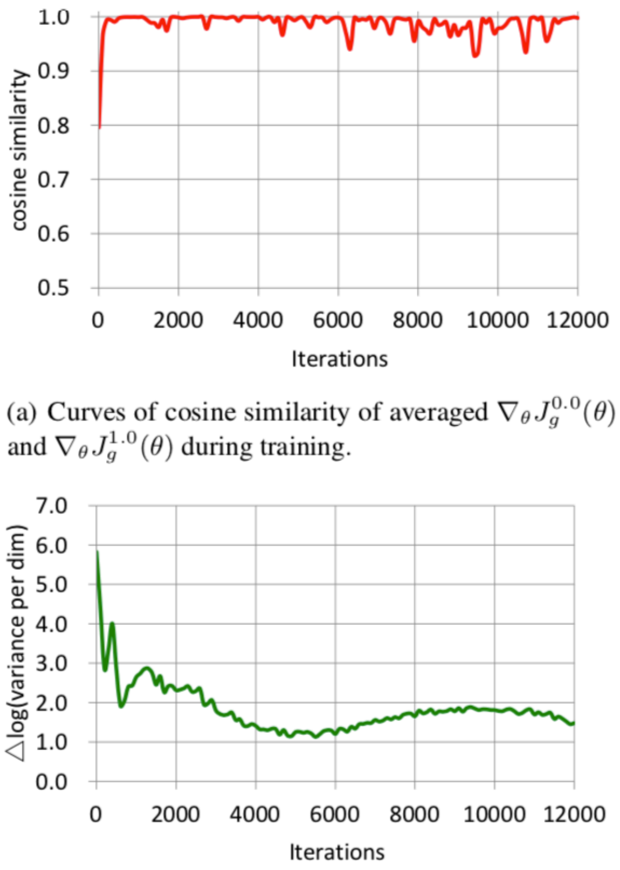


Table 1. Likelihood-based benchmark and time statistics for synthetic Turing test. '-(MLE)' means the best performance is acquired during MLE pre-training.

MODEL	NLL_{oracle}	NLL_{test}(FINAL/BEST)	BEST NLL_{oracle} + NLL_{test}	TIME/EPOCH
MLE	9.08	8.97/7.60	9.43 + 7.67	**16.14 ± 0.97s**
SEQGAN(YU ET AL., 2017)	8.68	10.10/-(MLE)	- (MLE)	$817.64 \pm 5.41s$
RANKGAN(LIN ET AL., 2017)	8.37	11.19/-(MLE)	- (MLE)	$1270 \pm 13.01s$
MALIGAN(CHE ET AL., 2017)	8.73	10.07/-(MLE)	- (MLE)	$741.31 \pm 1.45s$
SCHEDULED SAMPLING (BENGIO ET AL., 2015)	8.89	8.71/-(MLE)	- (MLE)	$32.54 \pm 1.14s$
PROFESSOR FORCING (LAMB ET AL., 2016)	9.43	8.31/-(MLE)	- (MLE)	$487.13 \pm 0.95s$
COT (OURS)	**8.19**	**8.03/7.54**	**8.19 + 8.03**	$53.94 \pm 1.01s$

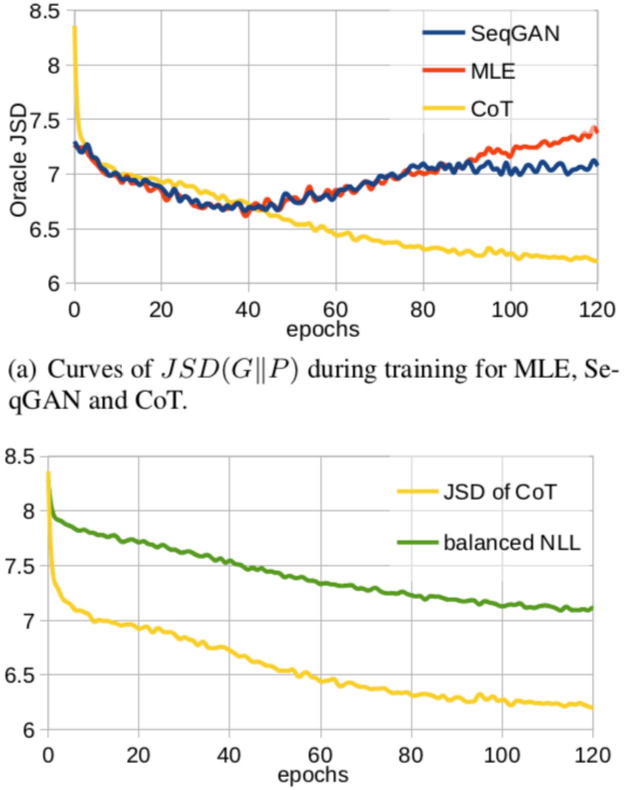
Experiment---universal sequence modeling in synthetic Turing test



(a) Curves of cosine similarity of averaged $\nabla_\theta J_g^{0.0}(\theta)$ and $\nabla_\theta J_g^{1.0}(\theta)$ during training.

(b) Curves of log variance reduction per dimension of $\nabla_\theta J_g^{0.0}(\theta)$ compared to $\nabla_\theta J_g^{1.0}(\theta)$

Figure 3. Empirical study on bias and variance comparison.



(a) Curves of $JSD(G\|P)$ during training for MLE, SeqGAN and CoT.

(b) Curves of balanced NLL and real JSD. Both results are from synthetic data experiments.

Figure 4. Training progress curves indicated by different values.

For quality evaluation, we evaluated BLEU on a small batch

Experiment --- Zero-prior Long & Diverse Text Generation

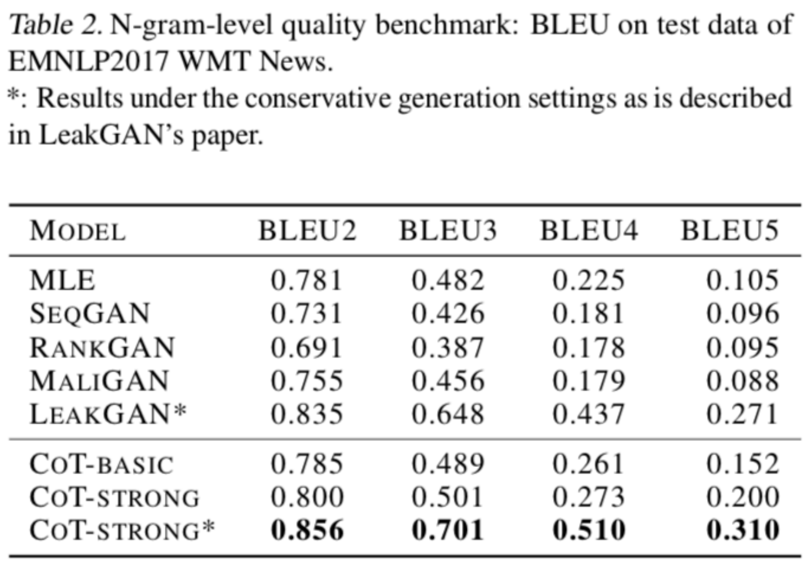


Table 2. N-gram-level quality benchmark: BLEU on test data of EMNLP2017 WMT News.
*: Results under the conservative generation settings as is described in LeakGAN's paper.

Model	BLEU2	BLEU3	BLEU4	BLEU5
MLE	0.781	0.482	0.225	0.105
SeqGAN	0.731	0.426	0.181	0.096
RankGAN	0.691	0.387	0.178	0.095
MaliGAN	0.755	0.456	0.179	0.088
LeakGAN*	0.835	0.648	0.437	0.271
CoT-basic	0.785	0.489	0.261	0.152
CoT-strong	0.800	0.501	0.273	0.200
CoT-strong*	**0.856**	**0.701**	**0.510**	**0.310**

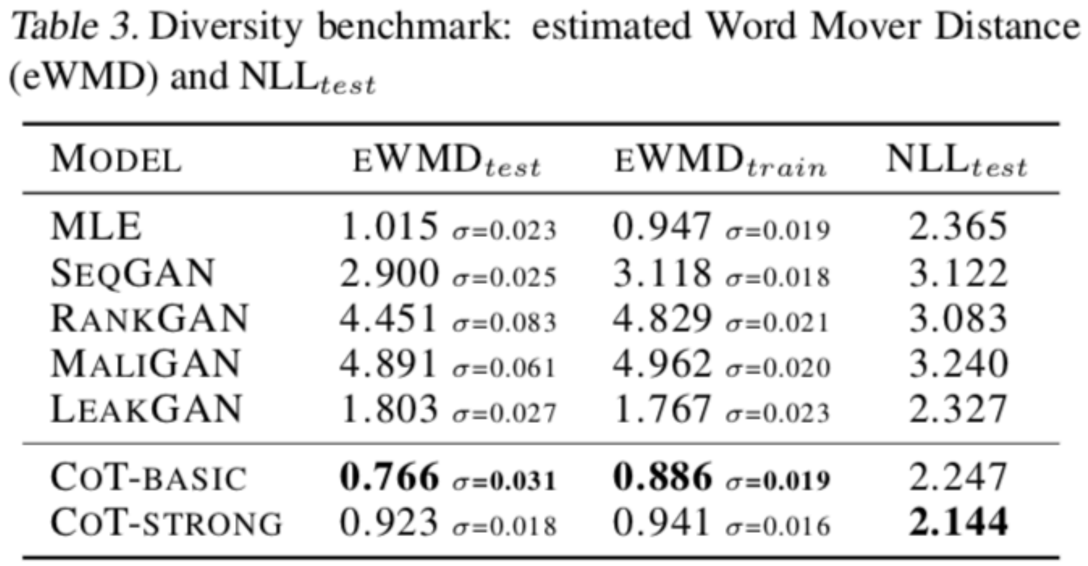


Table 3. Diversity benchmark: estimated Word Mover Distance (eWMD) and NLL$_{test}$

Model	eWMD$_{test}$	eWMD$_{train}$	NLL$_{test}$
MLE	1.015 σ=0.023	0.947 σ=0.019	2.365
SeqGAN	2.900 σ=0.025	3.118 σ=0.018	3.122
RankGAN	4.451 σ=0.083	4.829 σ=0.021	3.083
MaliGAN	4.891 σ=0.061	4.962 σ=0.020	3.240
LeakGAN	1.803 σ=0.027	1.767 σ=0.023	2.327
CoT-basic	**0.766 σ=0.031**	**0.886 σ=0.019**	2.247
CoT-strong	0.923 σ=0.018	0.941 σ=0.016	**2.144**

Conclusion

- Propose a novel approach called Cooperative Training (CoT) to improve the training of sequence generative models.
- Achieve independent success without the necessity of pre-training via maximum likelihood estimation or involving REINFORCE.
- Achieve superior performance on sample quality, diversity, as well as training stability.

Improving Sequence-to-Sequence Learning via Optimal Transport

Liqun Chen[1], Yizhe Zhang[2], Ruiyi Zhang[1], Chenyang Tao[1], Zhe Gan[3], Haichao Zhang[4], Bai Li[1], Dinghan Shen[1], Changyou Chen[5], Lawrence Carin[1]

[1]Duke University, [2]Microsoft Research, [3]Microsoft Dynamics 365 AI Research
[4]Baidu Research, [5]SUNY at Buffalo
{liqun.chen}@duke.edu

Motivation & Contribution

- Standard MLE training considers a word-level objective, predicting the next word given the previous ground-truth partial sentence.
- This procedure focuses on modeling local syntactic patterns, and may fail to capture long-range semantic structure.

- This paper imposes global sequence-level guidance via new supervision based on optimal transport, enabling the overall characterization and preservation of semantic features.

Semantic Matching with Optimal Transport

- OT distance on discrete domain

$$\mathcal{L}_{\text{ot}}(\boldsymbol{\mu}, \boldsymbol{\nu}) = \min_{\mathbf{T} \in \Pi(\mathbf{u}, \mathbf{v})} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{T}_{ij} \cdot c(\boldsymbol{x}_i, \boldsymbol{y}_j) = \min_{\mathbf{T} \in \Pi(\mathbf{u}, \mathbf{v})} \langle \mathbf{T}, \mathbf{C} \rangle$$

- Use IPOT algorithm to calculate the

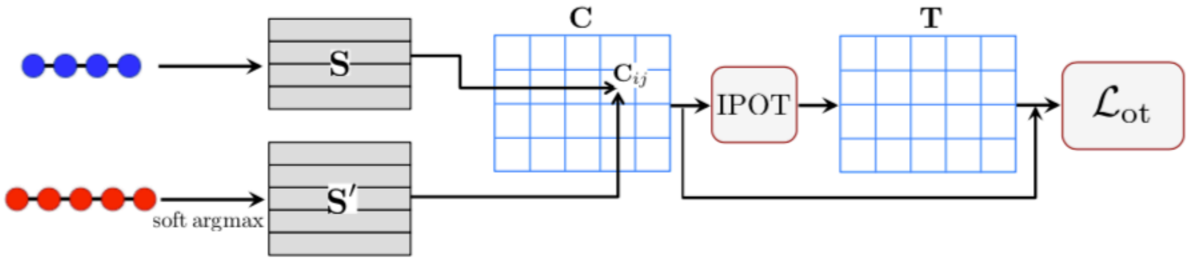
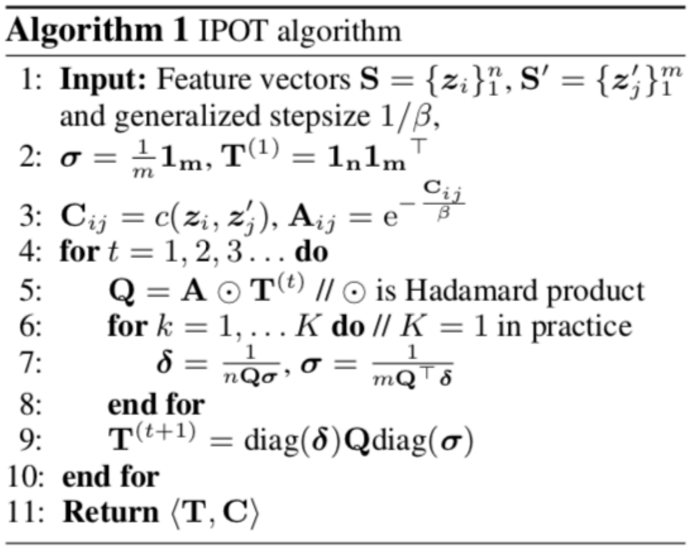
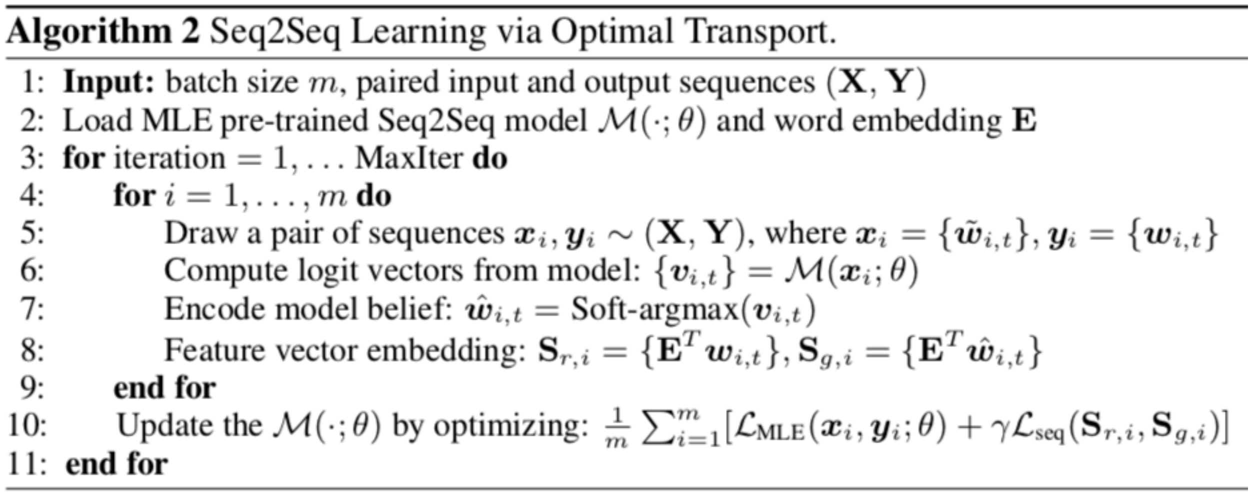


Figure 2: Schematic computation graph of OT loss.



Algorithm 1 IPOT algorithm

```
Input: Feature vectors $\mathbf{S} = \{\boldsymbol{z}_i\}_1^n, \mathbf{S}' = \{\boldsymbol{z}'_j\}_1^m$ and generalized stepsize $1/\beta$,
$\boldsymbol{\sigma} = \frac{1}{m}\mathbf{1_m}, \mathbf{T}^{(1)} = \mathbf{1_n}\mathbf{1_m}^\top$
$\mathbf{C}_{ij} = c(\boldsymbol{z}_i, \boldsymbol{z}'_j), \mathbf{A}_{ij} = \mathrm{e}^{-\frac{\mathbf{C}_{ij}}{\beta}}$
for $t = 1, 2, 3 \dots$ do
    $\mathbf{Q} = \mathbf{A} \odot \mathbf{T}^{(t)}$ // $\odot$ is Hadamard product
    for $k = 1, \dots K$ do // $K = 1$ in practice
        $\boldsymbol{\delta} = \frac{1}{n\mathbf{Q}\boldsymbol{\sigma}}, \boldsymbol{\sigma} = \frac{1}{m\mathbf{Q}^\top\boldsymbol{\delta}}$
    end for
    $\mathbf{T}^{(t+1)} = \mathrm{diag}(\boldsymbol{\delta})\mathbf{Q}\mathrm{diag}(\boldsymbol{\sigma})$
end for
Return $\langle \mathbf{T}, \mathbf{C} \rangle$
```



Algorithm 2 Seq2Seq Learning via Optimal Transport.

```
Input: batch size $m$, paired input and output sequences $(\mathbf{X}, \mathbf{Y})$
Load MLE pre-trained Seq2Seq model $\mathcal{M}(\cdot; \theta)$ and word embedding $\mathbf{E}$
for iteration $= 1, \dots$ MaxIter do
    for $i = 1, \dots, m$ do
        Draw a pair of sequences $\boldsymbol{x}_i, \boldsymbol{y}_i \sim (\mathbf{X}, \mathbf{Y})$, where $\boldsymbol{x}_i = \{\tilde{\boldsymbol{w}}_{i,t}\}, \boldsymbol{y}_i = \{\boldsymbol{w}_{i,t}\}$
        Compute logit vectors from model: $\{\boldsymbol{v}_{i,t}\} = \mathcal{M}(\boldsymbol{x}_i; \theta)$
        Encode model belief: $\hat{\boldsymbol{w}}_{i,t} = \text{Soft-argmax}(\boldsymbol{v}_{i,t})$
        Feature vector embedding: $\mathbf{S}_{r,i} = \{\mathbf{E}^T \boldsymbol{w}_{i,t}\}, \mathbf{S}_{g,i} = \{\mathbf{E}^T \hat{\boldsymbol{w}}_{i,t}\}$
    end for
    Update the $\mathcal{M}(\cdot; \theta)$ by optimizing: $\frac{1}{m}\sum_{i=1}^{m}[\mathcal{L}_{\text{MLE}}(\boldsymbol{x}_i, \boldsymbol{y}_i; \theta) + \gamma \mathcal{L}_{\text{seq}}(\mathbf{S}_{r,i}, \mathbf{S}_{g,i})]$
end for
```

Experiment

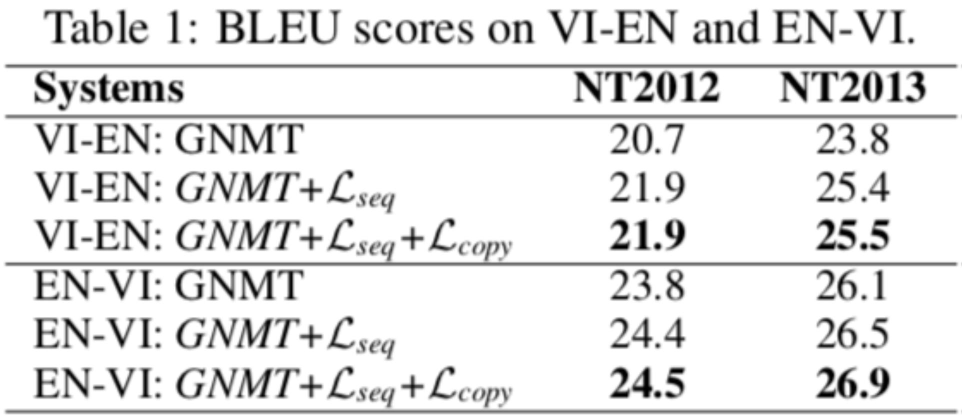


Table 1: BLEU scores on VI-EN and EN-VI.

Systems	NT2012	NT2013
VI-EN: GNMT	20.7	23.8
VI-EN: $GNMT+\mathcal{L}_{seq}$	21.9	25.4
VI-EN: $GNMT+\mathcal{L}_{seq}+\mathcal{L}_{copy}$	**21.9**	**25.5**
EN-VI: GNMT	23.8	26.1
EN-VI: $GNMT+\mathcal{L}_{seq}$	24.4	26.5
EN-VI: $GNMT+\mathcal{L}_{seq}+\mathcal{L}_{copy}$	**24.5**	**26.9**

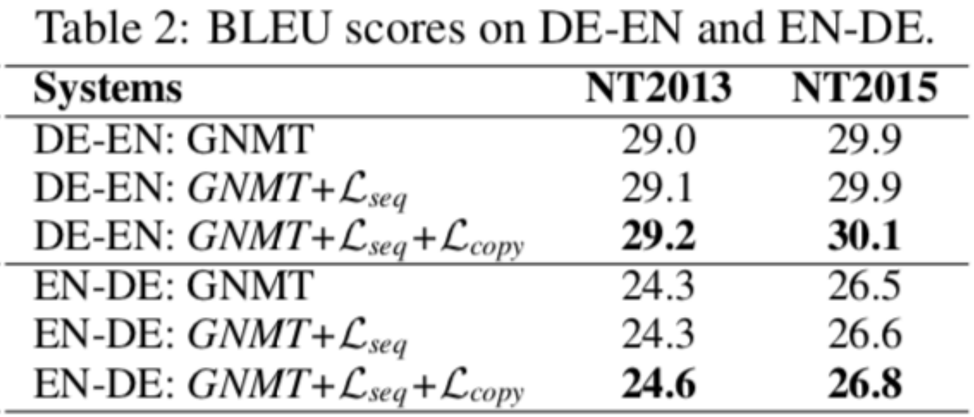


Table 2: BLEU scores on DE-EN and EN-DE.

Systems	NT2013	NT2015
DE-EN: GNMT	29.0	29.9
DE-EN: $GNMT+\mathcal{L}_{seq}$	29.1	29.9
DE-EN: $GNMT+\mathcal{L}_{seq}+\mathcal{L}_{copy}$	**29.2**	**30.1**
EN-DE: GNMT	24.3	26.5
EN-DE: $GNMT+\mathcal{L}_{seq}$	24.3	26.6
EN-DE: $GNMT+\mathcal{L}_{seq}+\mathcal{L}_{copy}$	**24.6**	**26.8**

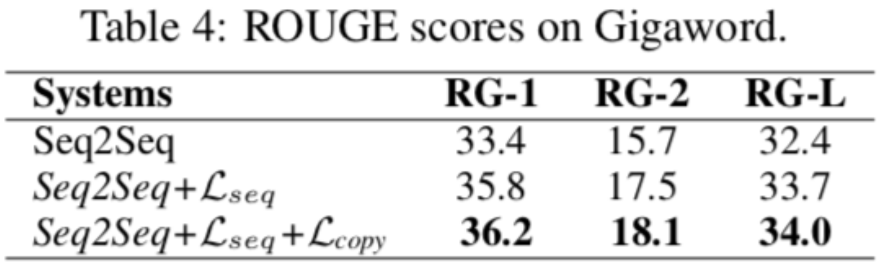


Table 4: ROUGE scores on Gigaword.

Systems	RG-1	RG-2	RG-L
Seq2Seq	33.4	15.7	32.4
$Seq2Seq+\mathcal{L}_{seq}$	35.8	17.5	33.7
$Seq2Seq+\mathcal{L}_{seq}+\mathcal{L}_{copy}$	**36.2**	**18.1**	**34.0**

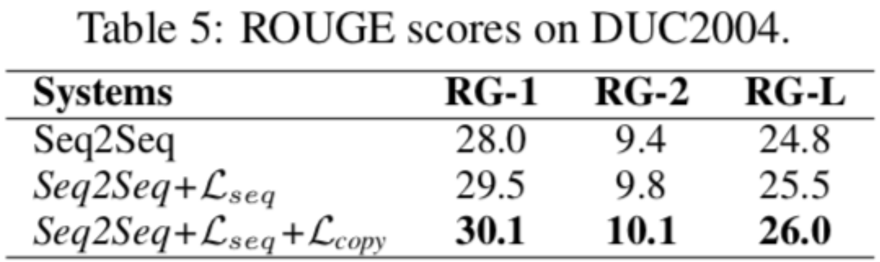
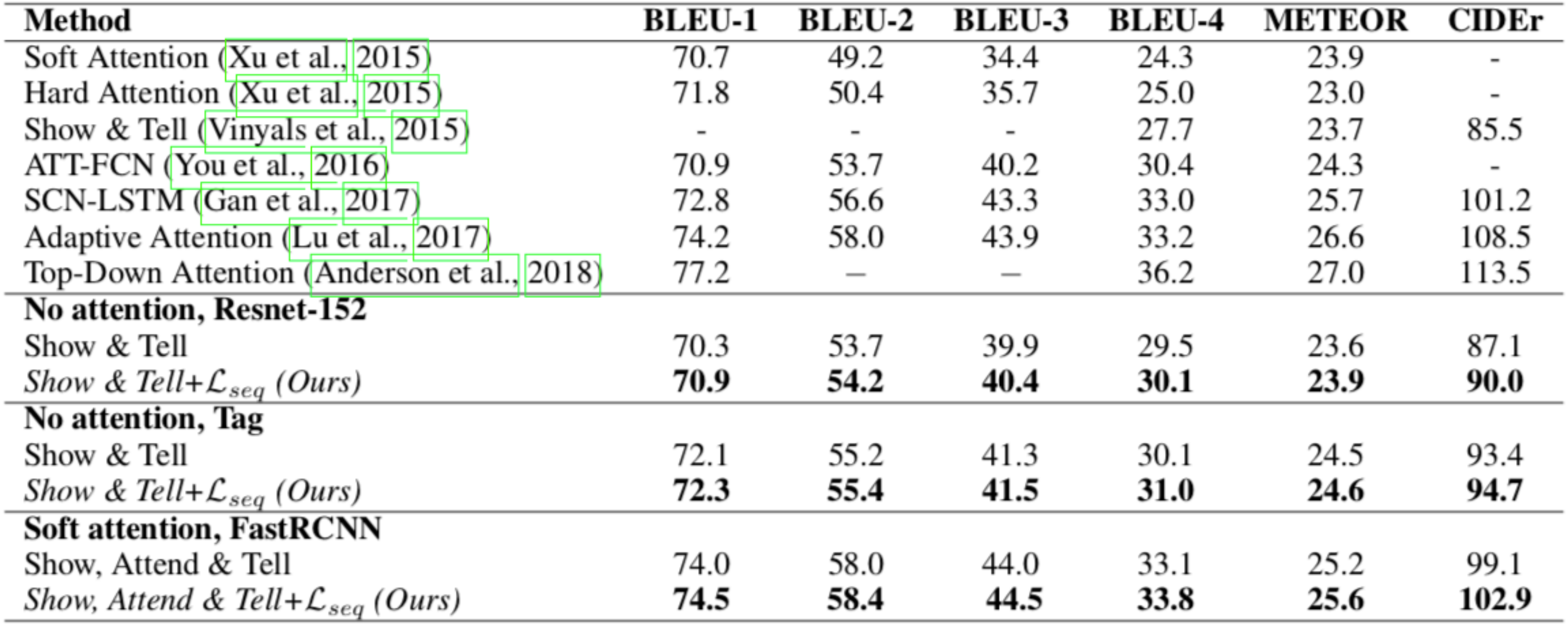


Table 5: ROUGE scores on DUC2004.

Systems	RG-1	RG-2	RG-L
Seq2Seq	28.0	9.4	24.8
$Seq2Seq+\mathcal{L}_{seq}$	29.5	9.8	25.5
$Seq2Seq+\mathcal{L}_{seq}+\mathcal{L}_{copy}$	**30.1**	**10.1**	**26.0**

Experiment

Table 6: Results for image captioning on the COCO dataset.



Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr
Soft Attention (Xu et al., 2015)	70.7	49.2	34.4	24.3	23.9	-
Hard Attention (Xu et al., 2015)	71.8	50.4	35.7	25.0	23.0	-
Show & Tell (Vinyals et al., 2015)	-	-	-	27.7	23.7	85.5
ATT-FCN (You et al., 2016)	70.9	53.7	40.2	30.4	24.3	-
SCN-LSTM (Gan et al., 2017)	72.8	56.6	43.3	33.0	25.7	101.2
Adaptive Attention (Lu et al., 2017)	74.2	58.0	43.9	33.2	26.6	108.5
Top-Down Attention (Anderson et al., 2018)	77.2	–	–	36.2	27.0	113.5
No attention, Resnet-152						
Show & Tell	70.3	53.7	39.9	29.5	23.6	87.1
Show & Tell+$\mathcal{L}_{seq}$ (Ours)	**70.9**	**54.2**	**40.4**	**30.1**	**23.9**	**90.0**
No attention, Tag						
Show & Tell	72.1	55.2	41.3	30.1	24.5	93.4
Show & Tell+$\mathcal{L}_{seq}$ (Ours)	**72.3**	**55.4**	**41.5**	**31.0**	**24.6**	**94.7**
Soft attention, FastRCNN						
Show, Attend & Tell	74.0	58.0	44.0	33.1	25.2	99.1
Show, Attend & Tell+$\mathcal{L}_{seq}$ (Ours)	**74.5**	**58.4**	**44.5**	**33.8**	**25.6**	**102.9**

Conclusion

- This work is motivated by the major deficiency in training Seq2Seq models: that the MLE training loss does not operate at sequence-level.
- This work propose the usage of optimal transport as a sequence-level loss to improve Seq2Seq learning.
- By applying this new method to machine translation, text summarization, and image captioning, this paper demonstrate that our proposed model can be used to help improve the performance compared to strong baselines.