Bridging the Gap between Training and Inference for Neural Machine Translation

Wen Zhang, Yang Feng, Fandong Meng, Di You, Qun Liu

ACL 2019, Best Long Paper Award

Presented by Kiki Ng, 2020-02-24

UNIVERSITY OF WATERLOO

Content

- Problems in NMT | Exposure Bias, Overcorrection
- Proposed Methods | Oracle Word Selection
- Key Experiments & Analysis | NIST (Zh→En), WMT'14 (En→De)
- Significance & Discussion

UNIVERSITY OF WATERLOO

Problems in NMT

1. Exposure Bias
2. Overcorrection

Exposure Bias

(Ranzato et al., 2015)

A discrepancy / "gap": predicted words are drawn from different distribution at training and inference respectively

Exposure Bias

- During training, what are fed as the context to the model?

Exposure Bias

- During training, what are fed as the context to the model?
    - Ground-truth words ("Data Distribution")
- Inferencing?

Exposure Bias

- During training, what are fed as the context to the model?
  - Ground-truth words ("Data Distribution")
- Inferencing?
  - Model-predicted words ("Model Distribution")

Exposure Bias

- During training, what are fed as the context to the model?
  - Ground-truth words ("Data Distribution")
- Inferencing?
  - Model-predicted words ("Model Distribution")

… We want a *correction*

Overcorrection

- Example:

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

Overcorrection

- Example:

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

Overcorrection

- Example:

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

Overcorrection

- Example:

Wrong! "by" should be the right choice

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

(Larger sentence likelihood)

Overcorrection

- Example:

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

Overcorrection

- Example:

Wrong! "the rule" should be the right choice

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

(Effect from "by")

Overcorrection

- Example:

Wrong! "the rule" should be the right choice

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

(Effect from "by")

Overcorrection

- Example:

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

("with" is fed as a context word)

"Overcorrection Recovery" (OR)

Overcorrection

- Example:

… What is a proper way to feed both ground-truth words and predicted words?

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

("with" is fed as a context word)

"Overcorrection Recovery" (OR)

Proposed Methods (in short)

An adjust to the training process...

Proposed Methods (in short)

An adjust to the training process...

#1: Oracle Word Selection

- Select *oracle words* from its predicted words

Proposed Methods (in short)

An adjust to the training process...

#1: Oracle Word Selection

- Select *oracle words* from its predicted words
- Sample as context from the oracle words and ground-truth words

Proposed Methods (in short)

An adjust to the training process...

#2: Sample with Decay

Proposed Methods (in short)

An adjust to the training process...

#2: Sample with Decay

(Scheduled Sampling, Bengio et al.)

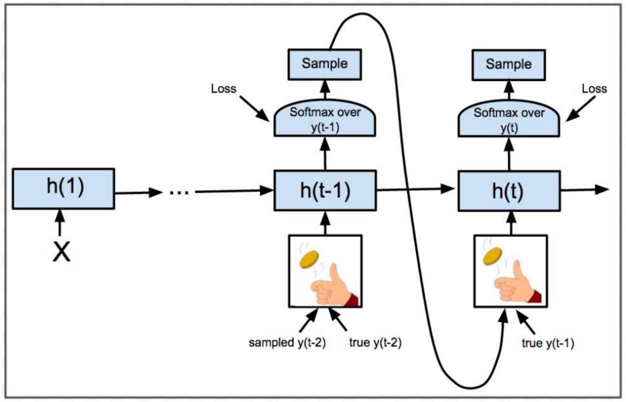


Figure 1: Illustration of the Scheduled Sampling approach, where one flips a coin at every time step to decide to use the true previous token or one sampled from the model itself.

Proposed Methods (in short)

An adjust to the training process!

#2: Sample with Decay

- The probability of sampling ground-truth words *decays* with the training process
- Decreasing guidance

(Scheduled Sampling, Bengio et al.)

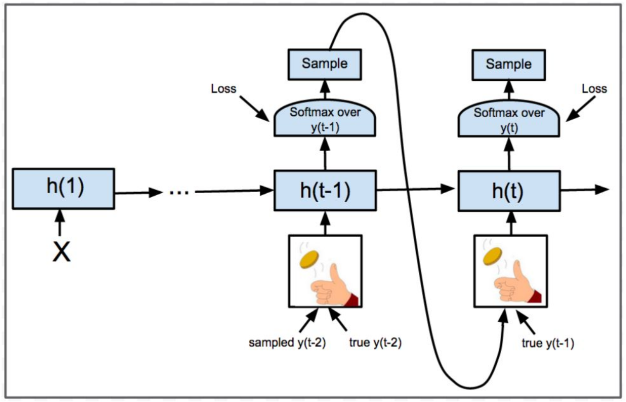


Figure 1: Illustration of the Scheduled Sampling approach, where one flips a coin at every time step to decide to use the true previous token or one sampled from the model itself.

Proposed Methods (in short)

An adjust to the training process!

#2: Sample with Decay

(Scheduled Sampling, Bengio et al.)

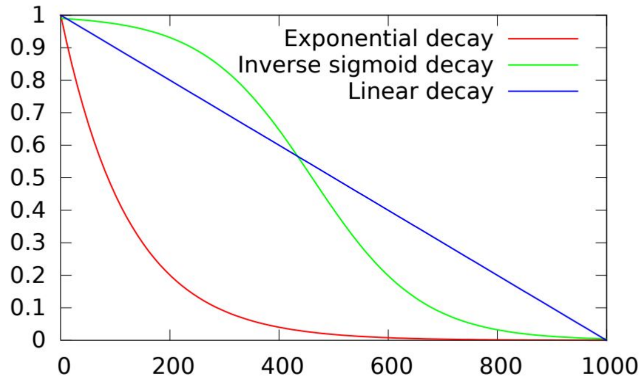


Figure 2: Examples of decay schedules. x: index of the mini-batches

Inverse sigmoid decay

$$p = \frac{\mu}{\mu + \exp\left(e/\mu\right)}$$

e: index of the epoches*

Proposed Methods (in short)

Verify the approach on

- RNN-based NMT Model
- the Transformer Model

Proposed Methods

An RNN-based NMT Model example (Bahdanau et al., 2015)

source sequence and the observed translation are $\mathbf{x} = \{x_1, \cdots, x_{|\mathbf{x}|}\}$ and $\mathbf{y}^* = \{y_1^*, \cdots, y_{|\mathbf{y}^*|}^*\}$.

Encoder. A bidirectional Gated Recurrent Unit (GRU) (Cho et al., 2014) is used to acquire two sequences of hidden states, the annotation of x_i is $h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i]$. Note that e_{x_i} is employed to represent the embedding vector of the word x_i.

$$\overrightarrow{h}_i = \mathbf{GRU}(e_{x_i}, \overrightarrow{h}_{i-1}) \quad (1)$$

$$\overleftarrow{h}_i = \mathbf{GRU}(e_{x_i}, \overleftarrow{h}_{i+1}) \quad (2)$$

Attention. The attention is designed to extract source information (called source context vector). At the j-th step, the relevance between the target word y_j^* and the i-th source word is evaluated and normalized over the source sequence

$$r_{ij} = \mathbf{v}_a^T \tanh\left(\mathbf{W}_a s_{j-1} + \mathbf{U}_a h_i\right) \quad (3)$$

$$\alpha_{ij} = \frac{\exp\left(r_{ij}\right)}{\sum_{i'=1}^{|\mathbf{x}|} \exp\left(r_{i'j}\right)} \quad (4)$$

The source context vector is the weighted sum of all source annotations and can be calculated by

$$c_j = \sum_{i=1}^{|\mathbf{x}|} \alpha_{ij} h_i \quad (5)$$

Decoder. The decoder employs a variant of GRU to unroll the target information. At the j-th step, the target hidden state s_j is given by

$$s_j = \mathbf{GRU}(e_{y_{j-1}^*}, s_{j-1}, c_j) \quad (6)$$

The probability distribution P_j over all the words in the target vocabulary is produced conditioned on the embedding of the previous ground truth word, the source context vector and the hidden state

$$t_j = g\left(e_{y_{j-1}^*}, c_j, s_j\right) \quad (7)$$

$$o_j = \mathbf{W}_o t_j \quad (8)$$

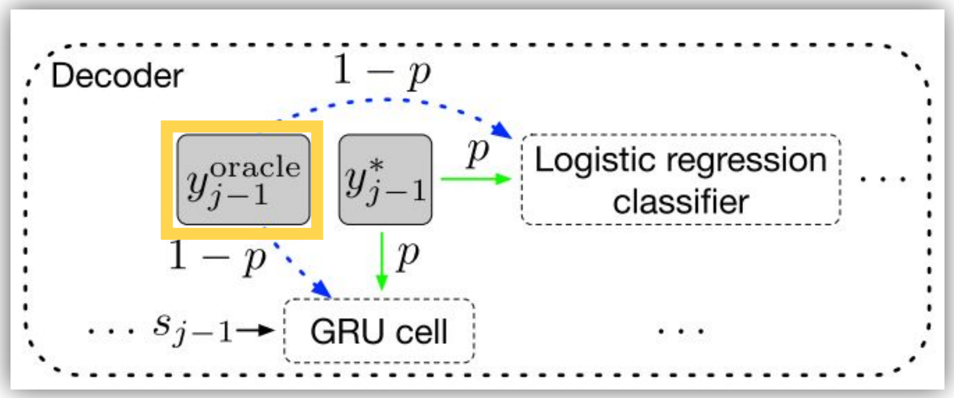
$$P_j = \mathrm{softmax}\left(o_j\right) \quad (9)$$

where g stands for a linear transformation, $\mathbf{W}_o$ is used to map t_j to o_j so that each target word has one corresponding dimension in o_j.

Proposed Methods

#1: Oracle Word Selection

At step *j*



(… with several strategies)

#1 Oracle Word Selection

Word Level Oracle (WO)

At step *j-1*

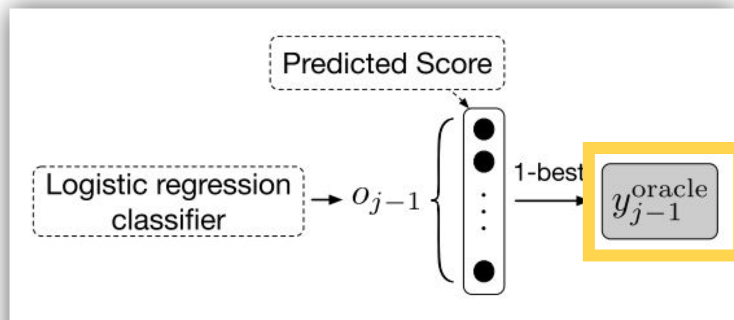
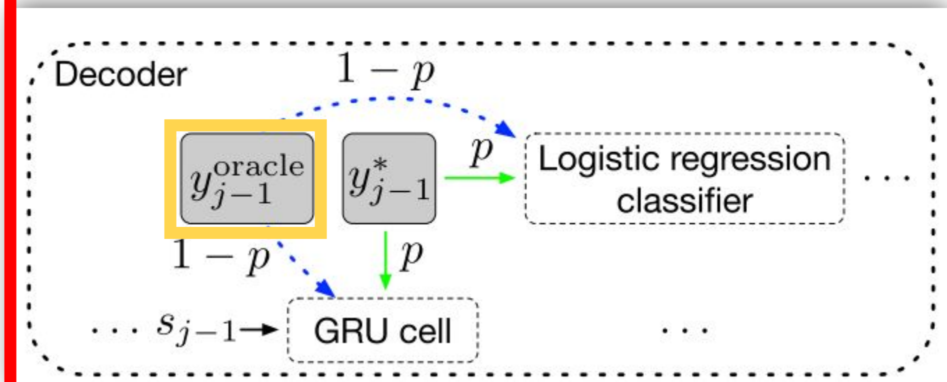


Figure 2: Word-level oracle without noise.

At step *j*



#1 Oracle Word Selection

WO with *Gumbel Noise*

At step *j-1*

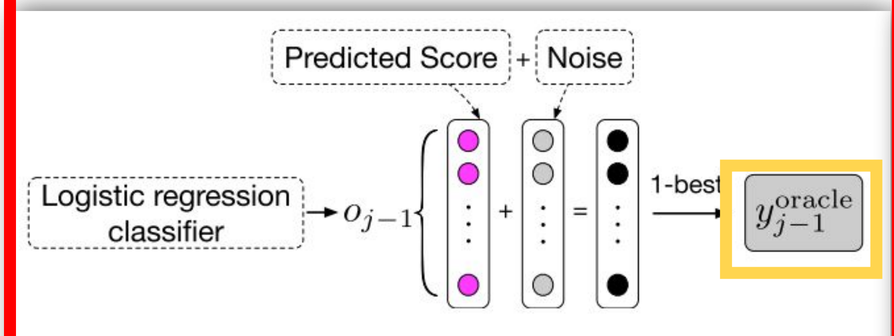
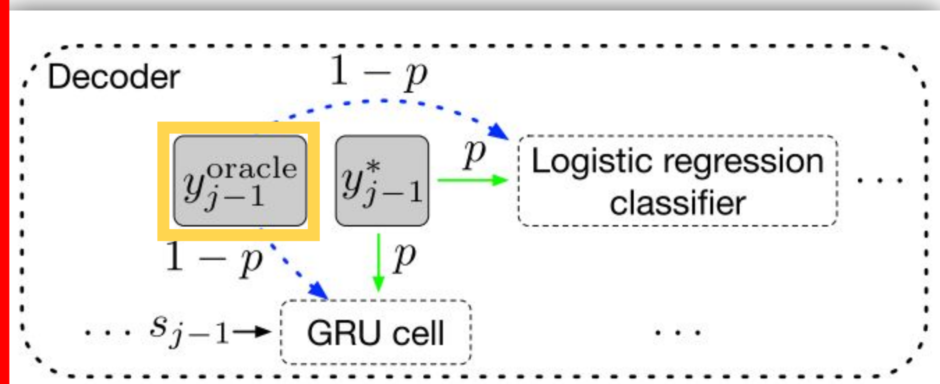


Figure 3: Word-level oracle with Gumbel noise.

At step *j*



UNIVERSITY OF WATERLOO

#1 Oracle Word Selection

WO with *Gumbel Noise*

At step *j-1*

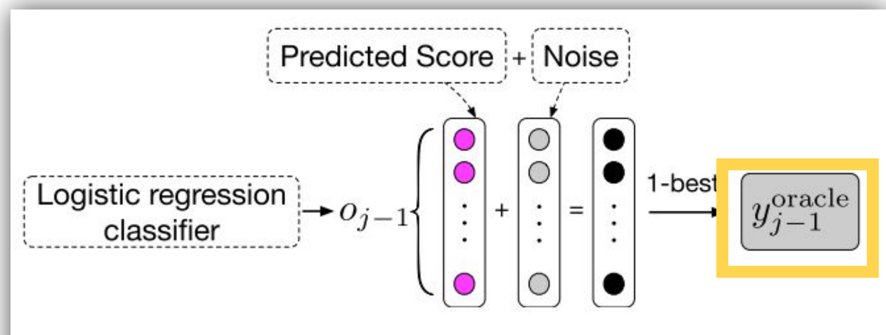


Figure 3: Word-level oracle with Gumbel noise.

Gumbel Noise?

in Figure 3, then softmax function is performed, the word distribution of y_{j-1} is approximated by

$$\eta = -\log\left(-\log u\right) \tag{10}$$

$$\tilde{o}_{j-1} = \left(o_{j-1} + \eta\right)/\tau \tag{11}$$

$$\tilde{P}_{j-1} = \operatorname{softmax}\left(\tilde{o}_{j-1}\right) \tag{12}$$

where η is the Gumbel noise calculated from a uniform random variable $u \sim \mathcal{U}(0,1)$, τ is temperature. As τ approaches 0, the softmax function is similar to the argmax operation, and it becomes uniform distribution gradually when $\tau \rightarrow \infty$.

#1 Oracle Word Selection

WO with *Gumbel Noise*

At step *j-1*

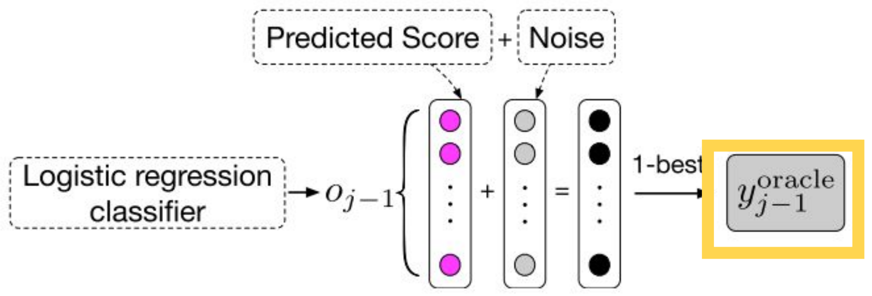


Figure 3: Word-level oracle with Gumbel noise.

Gumbel Noise?

- Rough idea:
  the *Gumbel-max* trick helps
  one sample from categorical distribut
  ion given log-probabilities without
  leaving log space

- Added here as regularization
  to make the selection more robust

#1 Oracle Word Selection

Sentence Level Oracle (SO)

- Not only to select the oracle word at the word level
- Try to select *an oracle sentence* first

#1 Oracle Word Selection

Sentence Level Oracle (SO)

- At step *j-1*:
    1. Get *k*-best candidate translations* using beam-search
    2. Rank with BLEU, select the highest as the *oracle sentence*
    3. Pick the *(j-1)-th* word as the oracle word

#1 Oracle Word Selection

Sentence Level Oracle (SO)

- At step *j-1*:
    1. Get *k*-best candidate translations* using beam-search
    2. Rank with BLEU, select the highest as the *oracle sentence*
    3. Pick the *(j-1)-th* word as the oracle word

* *Force Decoding*:
Candidates are forced to have the same length as the ground-truth sentence

Training Objective

Decoder. The decoder employs a variant of GRU to unroll the target information. At the j-th step, the target hidden state s_j is given by

$$s_j = \mathbf{GRU}(e_{y^*_{j-1}}, s_{j-1}, c_j) \tag{6}$$

The probability distribution P_j over all the words in the target vocabulary is produced conditioned on the embedding of the previous ground truth word, the source context vector and the hidden state

$$t_j = g\left(e_{y^*_{j-1}}, c_j, s_j\right) \tag{7}$$

$$o_j = \mathbf{W}_o t_j \tag{8}$$

$$P_j = \mathrm{softmax}\left(o_j\right) \tag{9}$$

where g stands for a linear transformation, $\mathbf{W}_o$ is used to map t_j to o_j so that each target word has one corresponding dimension in o_j.

3.3 Training

After selecting y_{j-1} by using the above method, we can get the word distribution of y_j according to Equation (6), (7), (8) and (9). We do not add the Gumbel noise to the distribution when calculating loss for training. The objective is to maximize the probability of the ground truth sequence based on maximum likelihood estimation (MLE). Thus following loss function is minimized:

$$\mathcal{L}\left(\theta\right) = -\sum\nolimits_{n=1}^{N}\sum\nolimits_{j=1}^{|\mathbf{y}^n|}\log P_j^n\left[y_j^n\right] \tag{16}$$

where N is the number of sentence pairs in the training data, $|\mathbf{y}^n|$ indicates the length of the n-th ground truth sentence, P_j^n refers to the predicted probability distribution at the j-th step for the n-th sentence, hence $P_j^n\left[y_j^n\right]$ is the probability of generating the ground truth word y_j^n at the j-th step.

Key Experiments

Translation Tasks:

- NIST Chinese → English (Zh→En)
- WMT'14 English →German (En→De)

Key Experiments

- NIST Chinese → English (Zh→En)

Systems	Architecture	MT03	MT04	MT05	MT06	Average
	Existing end-to-end NMT systems					
Tu et al. (2016)	Coverage	33.69	38.05	35.01	34.83	35.40
Shen et al. (2016)	MRT	37.41	39.87	37.45	36.80	37.88
Zhang et al. (2017)	Distortion	37.93	40.40	36.81	35.77	37.73
	Our end-to-end NMT systems					
	RNNsearch	37.93	40.53	36.65	35.80	37.73
Scheduled Sampling (Bengio et al.)	+ SS-NMT	38.82	41.68	37.28	37.98	38.94
	+ MIXER	38.70	40.81	37.59	38.38	38.87
this work	+ OR-NMT	**40.40**‡†*	**42.63**‡†*	**38.87**‡†*	**38.44**‡	**40.09**
	Transformer	46.89	47.88	47.40	46.66	47.21
	+ word oracle	47.42	48.34	47.89	47.34	47.75
	+ sentence oracle	**48.31***	**49.40***	**48.72***	**48.45***	**48.72**

Table 1: Case-insensitive BLEU scores (%) on Zh→En translation task. "‡", "†", "⋆" and "∗" indicate statistically significant difference (p<0.01) from RNNsearch, SS-NMT, MIXER and Transformer, respectively.

Key Experiments

- WMT'14 English → German (En → De)

Systems	newstest2014
RNNsearch	25.82
+ SS-NMT	26.50
+ MIXER	26.76
+ OR-NMT	**27.41**‡
Transformer (base)	27.34
+ SS-NMT	28.05
+ MIXER	27.98
+ OR-NMT	**28.65**‡

Table 3: Case-sensitive BLEU scores (%) on En→De task. The "‡" indicates the results are significantly better ($p<0.01$) than RNNsearch and Transformer.

Result Analysis

- Factor analysis on Oracle Word Selection

Systems	Average
RNNsearch	37.73
+ word oracle	38.94
+ noise	39.50
+ sentence oracle	39.56
+ noise	**40.09**

Table 2: Factor analysis on Zh→En translation, the results are average BLEU scores on MT03∼06 datasets.

Result Analysis

- Convergence (Left),
  Sentence Length (Middle)
  Gumbel Noise Factor (Right)

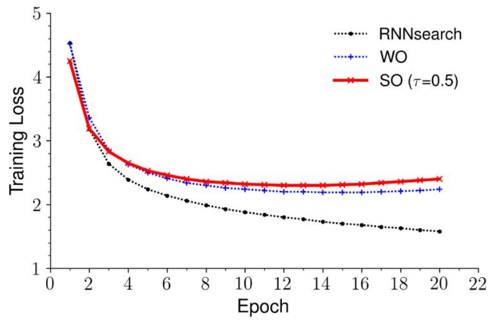


Figure 4: Training loss curves on Zh→En translation with different factors. The black, blue and red colors represent the RNNsearch, RNNsearch with word-level oracle and RNNsearch with sentence-level oracle systems respectively.

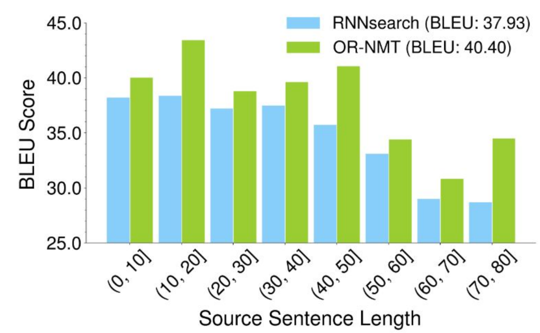


Figure 7: Performance comparison on the MT03 test set with respect to the different lengths of source sentences on the Zh→En translation task.

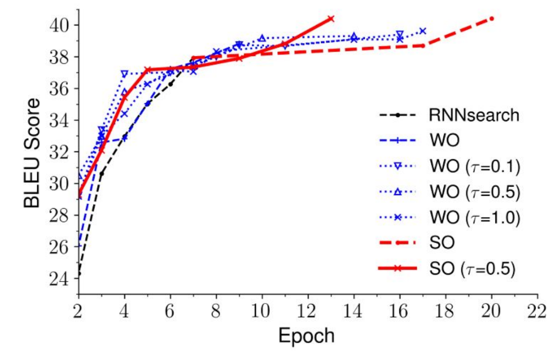


Figure 6: Trends of BLEU scores on the MT03 test set with different factors on the Zh→En translation task.

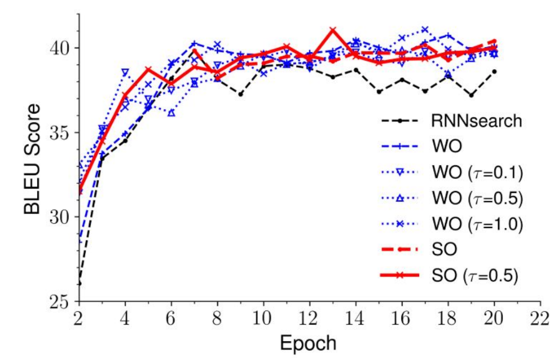


Figure 5: Trends of BLEU scores on the validation set with different factors on the Zh→En translation task.

UNIVERSITY OF WATERLOO

Result Conclusion

- Mitigate the gap between training and inference by:
  - feeding as context the oracle word / groudtruth word with a sampling scheme
  - Sampling the context word with decay from the ground truth words
- Verified the effectiveness with strong baseline models
- Sentence-level oracle show superiority over the Word-level oracle

UNIVERSITY OF WATERLOO

Significance

- Justify the effectiveness thoroughly with detailed analysis

- Easy to adopt

  (Github: https://github.com/ictnlp/OR-NMT)

Significance

- Justify the effectiveness thoroughly
  with detailed analysis

- Easy to adopt
  (Github: https://github.com/ictnlp/OR-NMT)

- (… Application on my project: Cantonese-Chinese Translation Task)

Discussion

- A comparison to Bengio's work

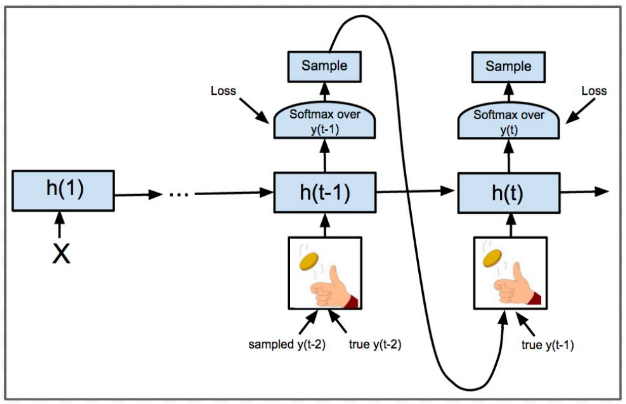
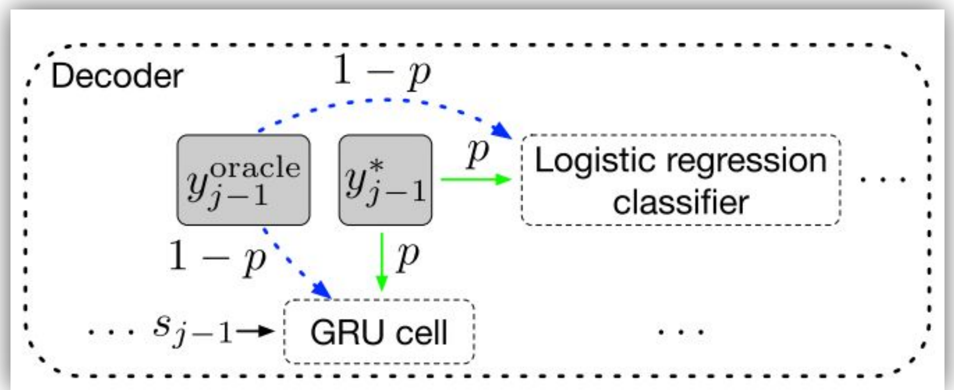


Figure 1: Illustration of the Scheduled Sampling approach, where one flips a coin at every time step to decide to use the true previous token or one sampled from the model itself.



UNIVERSITY OF WATERLOO

Reference

Zhang, W., Feng, Y., Meng, F., You, D. and Liu, Q., 2019. Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. ICLR 2015.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 1171–1179. Curran Associates, Inc.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. arXiv preprint arXiv:1511.06732.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. 2015. Improving multi-step prediction of learned time series models. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelli- gence, AAAI'15, pages 3024–3030. AAAI Press.