Appendix for: Cross-model Back-translated Distillation for Unsupervised Machine Translation

Xuan-Phi Nguyen [1 2] **Shafiq Joty** [1 3] **Thanh-Tung Nguyen** [1 2] **Wu Kui** [2] **Ai Ti Aw** [2]

1. Appendix

In the following supplementary material, we first provide the full mathematical derivations of the loss function $\mathcal{L}$ presented in the paper (§1.1). Then, we provide the generalized version of our method cross-model back-translated distillation, or GCBD, and measure its effectiveness in the IWSLT English-German, German-English, English-French and French-English unsupervised tasks (§1.2). In addition, we investigate why ensemble knowledge distillation (Freitag et al., 2017), which boosts the performance in a supervised setup, fails to do so in an unsupervised setup where we replace the supervised agents used in the method with the UMT agents (§1.3). Finally, in §1.5, we provide a comparison between unsupervised models and supervised counterparts to provide a perspective of how far unsupervised machine translation research has progressed.

1.1. Derivations of negative log likelihood $\mathcal{N}(\theta_\alpha, \theta_\beta)$

In this section, we provide the complete mathematical derivations of the loss function $\mathcal{L}$ in the paper. Recalling that we are supposed to maximize the log probabilities of the variables x_s, y_t, z_s, x_t, y_s and z_t according to the sampling process in Figure 1 and the graphical model in Figure 2. Otherwise speaking, we seek to minimize the following negative log likelihood:

$$\mathcal{J}(\theta) = -\log P_\theta(x_s, y_t, z_s) - \log P_\theta(x_t, y_s, z_t) \quad (1)$$

Then we can expand the first term as follows:

$$\begin{aligned}\log P_\theta(x_s, y_t, z_s) &= \log \frac{P_\theta(x_s, y_t, z_s)}{P_\theta(x_s, y_t)} P_\theta(x_s, y_t) \\ &= \log P_\theta(z_s|x_s, y_t) + \log P_\theta(x_s, y_t) \\ &= \log P_\theta(z_s|x_s, y_t) + \log \frac{P_\theta(x_s, y_t)}{P_\theta(y_t)} P_\theta(y_t) \\ &= \log P_\theta(z_s|x_s, y_t) + \log P_\theta(x_s|y_t) + \log P_\theta(y_t)\end{aligned} \quad (2)$$

Since z_s is independent from x_s given y_t according to the graphical model (fig. 1), we have $P_\theta(z_s|x_s, y_t) = P_\theta(z_s|y_t)$, then Eq. 2 can be reduced to:

$$\begin{aligned}\log P_\theta(x_s, y_t, z_s) = \log P_\theta(z_s|y_t) + \log P_\theta(x_s|y_t) \\ + \log P_\theta(y_t)\end{aligned} \quad (3)$$

Alternatively, the first term can also be express as follows:

$$\begin{aligned}\log P_\theta(x_s, y_t, z_s) &= \log P_\theta(z_s|x_s, y_t) + \log P_\theta(x_s, y_t) \\ &= \log P_\theta(z_s|y_t) + \log P_\theta(y_t, x_s) \\ &= \log \frac{P_\theta(y_t|z_s)P_\theta(z_s)}{P_\theta(y_t)} + \log \frac{P_\theta(y_t, x_s)}{P_\theta(x_s)} P_\theta(x_s) \\ &= \log P_\theta(y_t|z_s) + \log P_\theta(z_s) - \log P_\theta(y_t) \\ &\quad + \log P_\theta(y_t|x_s) + \log P_\theta(x_s)\end{aligned} \quad (4)$$

After that, we expand the second term in similar fashion, which we yield:

$$\begin{aligned}\log P_\theta(x_t, y_s, z_t) = \log P_\theta(z_t|y_s) + \log P_\theta(x_t|y_s) \\ + \log P_\theta(y_s)\end{aligned} \quad (5)$$

$$\begin{aligned}\log P_\theta(x_t, y_s, z_t) = \log P_\theta(y_s|z_t) + \log P_\theta(y_s|x_t) \\ + \log P_\theta(z_t) + \log P_\theta(x_t) - \log P_\theta(y_s)\end{aligned} \quad (6)$$

Then, by adding up Eq. 3, 4, 5 and 6 together, and then divide it by 2, we will derive the negative log likelihood of Eq. 1 as:

$$\begin{aligned}\mathcal{J}(\theta) = \frac{1}{2}[&-\log P_\theta(y_t|z_s) - \log P_\theta(y_t|x_s) \\ &- \log P_\theta(z_s|y_t) - \log P_\theta(x_s|y_t) - \log P_\theta(y_s|z_t) \\ &- \log P_\theta(y_s|x_t) - \log P_\theta(z_t|y_s) - \log P_\theta(y_s|x_t) \\ &- \log P_\theta(x_s) - \log P_\theta(z_s) - \log P_\theta(x_t) - \log P_\theta(z_t)]\end{aligned} \quad (7)$$

---

[1]Nanyang Technological University [2]Institute for Infocomm Research (I[2]R), A*STAR [3]Salesforce Research Asia. Correspondence to: Xuan-Phi Nguyen <nguyenxu002@e.ntu.edu.sg>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

$$\mathbb{X}_s \dashrightarrow x_s \xrightarrow[s\to t]{\theta_\alpha} y_t \xrightarrow[t\to s]{\theta_\beta} z_s \xRightarrow[(z_s,y_t),(y_t,z_s)]{(x_s,y_t),(y_t,x_s)} \theta$$

$$\mathbb{X}_t \dashrightarrow x_t \xrightarrow[t\to s]{\theta_\alpha} y_s \xrightarrow[s\to t]{\theta_\beta} z_t \xRightarrow[(y_s,z_t),(z_t,y_s)]{(y_s,x_t),(x_t,y_s)} \theta$$

Figure 1: The sampling process of $x_s, x_t, y_s, y_t, z_s, z_t$.

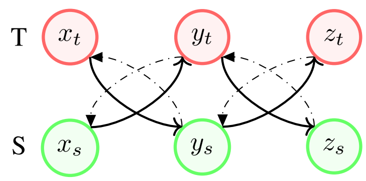

Figure 2: Graphical model representation of CBD. The model (θ) is trained on all parallel pairs (shown as directed links): $(x_s, y_t), (y_t, x_s), (y_t, z_s), (z_s, y_t), (x_t, y_s), (y_s, x_t), (y_s, z_t), (z_t, y_s)$.

Considering the generation process of x_s, y_s, z_s, x_t, y_t and z_t, we minimize the following CBD loss function $\mathcal{L}$:

$$\mathcal{L}_\theta(\theta_\alpha, \theta_\beta) = \mathop{\mathbb{E}}_{\substack{z_s\sim P(\cdot|y_t,\theta_\beta), y_t\sim P(\cdot|x_s,\theta_\alpha), x_s\sim\mathbb{X}_s \\ z_t\sim P(\cdot|y_s,\theta_\beta), y_s\sim P(\cdot|x_t,\theta_\alpha), x_t\sim\mathbb{X}_t}} [\mathcal{J}(\theta)] \quad (8)$$

where $\theta_\alpha, \theta_\beta \in \Theta$ are input parameters, which are specified in the CBD training procedure described in the main paper. Note θ_α is used to sample y_t, y_s from x_s, x_t while θ_β is used to back-translate y_t, y_s to z_s, z_t respectively.

It is note-worthy that in practice, we do not explicitly optimize the non-conditional terms $P_\theta(x_s)$, $P_\theta(x_t)$, $P_\theta(z_s)$ and $P_\theta(z_t)$. The reason is that the MT model θ is built as a strictly cross-lingual model, which means that it can only translate from one language to another, and possibly vice versa. It is not, how, equipped to train an explicit language model that only aims to optimize non-conditional log probabilities. We did attempt to pseudo-optimize them by using denoising-autoencoding strategy in the preliminary experiments. The results, however, yield no difference and sometimes under-performance. We conjecture that this is due to technical difficulty in forcing a single-language model loss upon a cross-lingual model for the sole purpose of improving machine translation. We put this in our future work.

1.2. Generalized version

In this section, we describe a generalized version of our CBD, which involves multiple UMT agents instead of just two. Then, we test this method in the IWSLT experiments to demonstrate its effectiveness and characteristics. Specifically, in addition to the input monolingual data $\mathbb{X}_s$ and $\mathbb{X}_t$ of languages s and t and the supervised model θ, we introduce another hyper-parameter n to indicate the number of unsupervised agents used to perform cross-model back-translation. The generalized cross-model back-translated distillation (GCBD) strategy is presented in Algorithm 1. In this method, instead of training only two agents, the method trains a set of n UMT agents $\Theta = \{\theta_1, ..., \theta_n\}$. During training, we iteratively select two orderly distinct agents θ_i and θ_j from Θ and use them to perform cross-model back-translation and train the model θ.

Algorithm 1 Generalized Cross-model Back-translated Distillation (GCBD): Given monolingual data $\mathbb{X}_s$ and $\mathbb{X}_t$ of languages s and t, and hyper-parameter n, return a UMT model with parameters θ.

```
for i ∈ 1, ..., n do
    Train UMT agent with parameters θ_i
end for
Initialize MT model θ (randomly or with pretrained model)
while until convergence do
    for i ∈ 1, ..., n do
        for j ∈ 1, ..., n where j ≠ i do
            θ ← θ − η∇_θ L(θ_α = θ_i, θ_β = θ_j)
        end for
    end for
end while
return θ
```

To evaluate GCBD in comparison with CBD, we conduct experiments with the IWSLT'13 English-French (En-Fr) and IWSLT'14 English-German (En-De) tasks. The setup for these experiments are identical to the IWSLT experiment in the main paper, except that we vary the hyper-parameter $n = (2, 3, 4)$ to determine the optimal number of agents. The results are reported in Table 2. As it can be seen, increasing the number of agents n to 3 adds an additional $0.4 - 1.0$ BLEU improvement compared to the standard CBD. Moreover, using 4 UMT agents does not improve the performance over using just 3 UMT, despite that this setup still outperforms the standard CBD. The results indicate that increasing the system complexity further is not always optimal and diminishing return is observed as we add more agents to the system.

Table 1: BLEU scores on the unsupervised IWSLT'13 English-French (En-Fr) and IWSLT'14 English-German (En-De) tasks with varying number of agents n of GCBD.

Method	En-Fr	Fr-En	En-De	De-En
NMT	29.6	30.7	15.8	19.1
+ GCBD ($n = 2$) (CBD)	31.8	31.8	18.4	21.7
+ GCBD ($n = 3$)	**32.8**	**32.1**	**19.2**	**22.2**
+ GCBD ($n = 4$)	32.3	32.0	19.1	21.9

Table 2: Percentage of tri-gram repetitions in the synthetic data generated by ensemble knowledge distillation (Freitag et al., 2017), compared to those created by CBD; and the respective test BLEU scores in the *base* WMT'14 En-Fr, WMT'16 En-De and En-Ro unsupervised tasks.

Method	En-Fr	Fr-En	En-De	De-En	En-Ro	Ro-En
% tri-gram repetition						
Ens-Distil	30.3%	34%	73%	76%	43%	86%
CBD	10^{-3}%	10^{-2}%	10^{-2}%	10^{-1}%	10^{-2}%	10^{-2}%
BLEU on test set						
Ens-Distil	17.3	20.0	3.5	3.7	1.2	1.1
CBD	26.6	25.7	16.6	20.5	18.1	17.8

1.3. Analysis of degeneration in ensemble knowledge distillation

Ensemble knowledge distillation (Freitag et al., 2017) has been used to enhance supervised machine translation. It uses multiple strong (supervised) teachers to generate synthetic parallel data from both sides of the parallel corpora by averaging the decoding probabilities of the teachers at each step. The synthetic data are then used to train the student model. Having seen its effectiveness in the supervised setup, we apply this same tactic to unsupervised MT tasks by replacing the supervised teachers with unsupervised MT agents. However, the method surprisingly causes drastic performance drop in the WMT'14 En-Fr, WMT'16 En-De and En-Ro unsupervised MT tasks.

By manual inspection, we found that many instances of the synthetic data are incomprehensible and contain repetitions, which is a degeneration behavior. We then quantitatively measure the percentage of sentences in the synthetic data containing tri-gram repetitions by counting the number of sentences where a word/sub-word is repeated at least three consecutive times. As reported in the main paper, from 30% to 86% of the synthetic data generated by the ensemble knowledge distillation (Ens-Distil) method are incomprehensible and contain repetitions. Relative to the performance of CBD, the performance drop in ensemble distillation is also more dramatic for language pairs with higher percentage of degeneration (En-Ro and En-De). This explains why the downstream student model fails to learn from these corrupted data. The results indicate that UMT agents are unable to jointly translate through ensembling strategy the monolingual data that they were trained on. This phenomenon may require further research to be fully understood. On the other hand, with less than 0.1% tri-gram repetitions, CBD generates little to no repetitions, which partly explains why it is able to improve the performance.

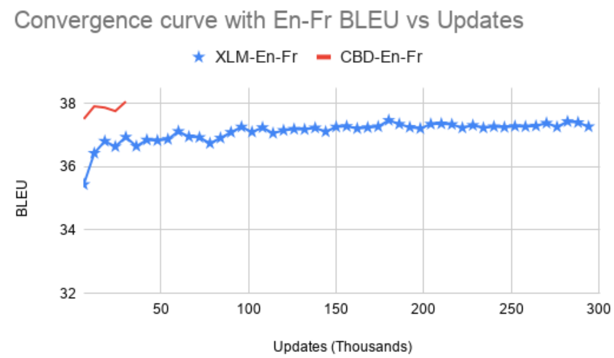

Figure 3: Convergence speed of CBD in comparison with baseline XLM, represented by the test BLEU score of the *WMT En-Fr* task after a given number of training updates.

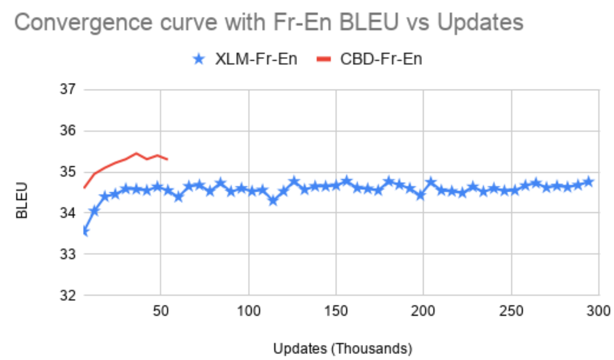

Figure 4: Convergence speed of CBD in comparison with baseline XLM, represented by the test BLEU score of the *WMT Fr-En* task after a given number of training updates.

1.4. Convergence curves of CBD compared with the baselines

This section provides extra convergence curve charts for all 6 of the language pairs in the large scale WMT English-French (Figure 3 & Figure 4), WMT English-German (Figure 5 & Figure 6) and WMT English-Romanian (Figure 7 & Figure 8) tasks. As it can be seen from the charts, CBD converges rapidly and outperforms the baselines with little additional resources, given the pretrained models provided by Conneau & Lample (2019) and Song et al. (2019).

1.5. Comparison with supervised MT

In this section, we compare the performances of the CBD method, along with previous SOTA unsupervised models, with the standard supervised Transformer model (Ott et al., 2018) to present a perspective of how much progress the field of unsupervised machine translation has made. More specifically, we use the provided Transformer models *pretrained* on the parallel WMT'14 English-French and

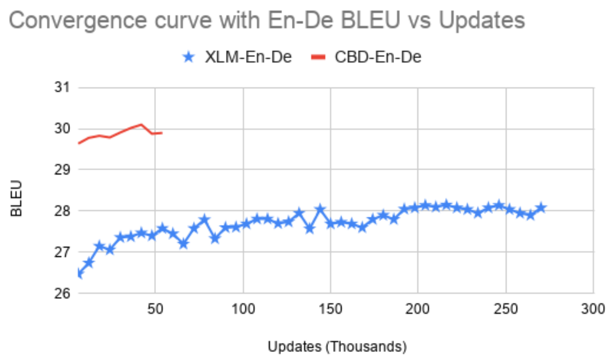

Figure 5: Convergence speed of CBD in comparison with baseline XLM, represented by the test BLEU score of the *WMT En-De* task after a given number of training updates.

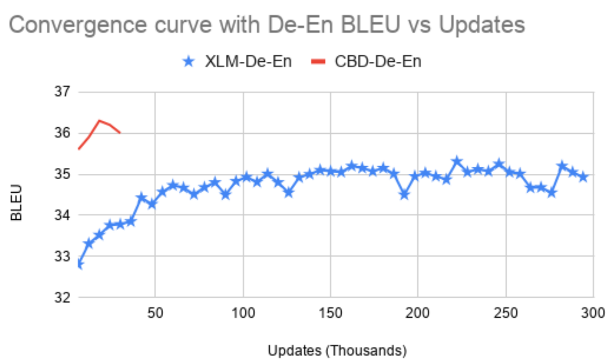

Figure 6: Convergence speed of CBD in comparison with baseline XLM, represented by the test BLEU score of the *WMT De-en* task after a given number of training updates.

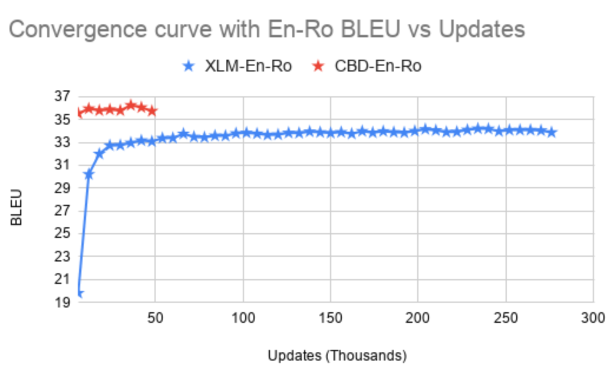

Figure 7: Convergence speed of CBD in comparison with baseline MASS, represented by the test BLEU score of the *WMT En-Ro* task after a given number of training updates.

English-German datasets and evaluate them on the WMT'14 En-Fr and WMT'16 En-De test sets, as similarly done for unsupervised counterparts. The results are presented in Table 3. As it can be seen, unsupervised MT models have made significant advancement throughout multiple

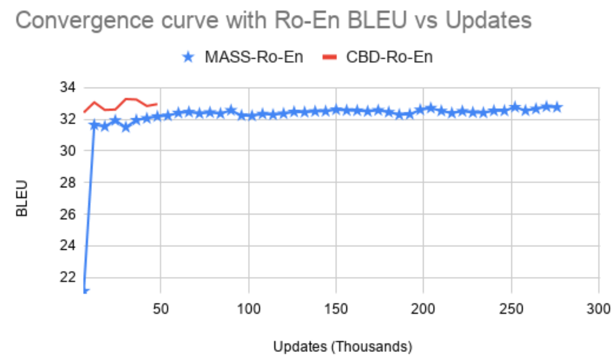

Figure 8: Convergence speed of CBD in comparison with baseline MASS, represented by the test BLEU score of the *WMT Ro-En* task after a given number of training updates.

Table 3: BLEU scores on the WMT'14 English-French (En-Fr) and WMT'16 English-German (En-De) tasks of unsupervised MT methods (MASS and CBD), in comparison to supervised MT method (Ott et al., 2018).

Method	**En-Fr**	**En-De**
Unsupervised MT		
XLM (Conneau & Lample, 2019)	33.4	26.4
MASS (Song et al., 2019)	37.5	28.3
CBD	38.2	30.1
Supervised MT		
Transformer (Ott et al., 2018)	43.2	33.0

iterations and refinement (Conneau & Lample, 2019; Song et al., 2019). However, while the CBD method further improve the performance, it still lags behind the supervised MT model (Ott et al., 2018) by around 3 to 5 BLEU points.

References

Conneau, A. and Lample, G. Cross-lingual language model pretraining. In Wallach, H., Larochelle, H., Beygelzimer, A., dÁlché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 7059–7069. Curran Associates, Inc., 2019.

Freitag, M., Al-Onaizan, Y., and Sankaran, B. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*, 2017.

Ott, M., Edunov, S., Grangier, D., and Auli, M. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation (WMT)*, 2018.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*, 2019.