On the Copying Problem of Unsupervised NMT: 
A Training Schedule with a Language Discriminator Loss

Yihong Liu\(^\circ\), Alexandra Chronopoulou\(^\circ\), Hinrich Schütze\(^\circ\), and Alexander Fraser\(^\circ\)
\(^\circ\)Center for Information and Language Processing, LMU Munich
\(^\circ\)Munich Center for Machine Learning (MCML)
{yihong, achron, fraser}@cis.lmu.de

Abstract
Although unsupervised neural machine translation (UNMT) has achieved success in many language pairs, the copying problem, i.e., directly copying some parts of the input sentence as the translation, is common among distant language pairs, especially when low-resource languages are involved. We find this issue is closely related to an unexpected copying behavior during online back-translation (BT). In this work, we propose a simple but effective training schedule that incorporates a language discriminator loss. The loss imposes constraints on the intermediate translation so that the translation is in the desired language. By conducting extensive experiments on different language pairs, including similar and distant, high and low-resource languages, we find that our method alleviates the copying problem, thus improving the translation performance on low-resource languages.

1 Introduction
UNMT (Lample et al., 2018; Artetxe et al., 2018) is a new and effective approach for tackling the scarcity of parallel data. Typically, a cross-lingual pretrained language model (PLM) (Peters et al., 2018; Devlin et al., 2019) is trained on two languages and then used to initialize the model for the UNMT task (Conneau and Lample, 2019; Song et al., 2019; Yang et al., 2020; Liu et al., 2020). However, when it comes to low-resource languages, especially when translating between distant language pairs, UNMT often yields very poor results (Neubig and Hu, 2018; Guzmán et al., 2019; Marchisio et al., 2020). One of the major problems that lead to low translation quality is the copying problem or off-target problem (Kim et al., 2020; Yang et al., 2020). That is: the trained model does not translate but copies some words or even the whole sentence from the input as the translation.

We find the copying problem is closely related to an unexpected behavior in BT (Sennrich et al., 2016): the model does not translate into the correct intermediate language but simply copies tokens from the source language. To address this problem, this work proposes a simple but effective method that can be integrated into the standard UNMT training. We leverage a language discriminator to detect the language of the intermediate translation generated in BT and backpropagate the gradients to the main model. In this way, we can provide implicit supervision to the model. We find that by adding such a training objective, the copying problem can be largely alleviated, especially for low-resource languages. Noticeably, we do not introduce any language-specific architectures into the main model. To the best of our knowledge, this is the first work that introduces a language discriminator loss to force the intermediate translations in BT to be in the correct language. The contributions of our work are as follows:
(1) We explore the reasons behind the copying problem in UNMT and propose a training schedule with a language discriminator loss.
(2) We evaluate our method on many languages, including high- and low-resource, and similar and distant language pairs.
(3) We carry out an analysis, showing the proposed method can reduce the copying ratio, especially on small-size datasets and distant language pairs.
(4) We make our code publicly available. \(^1\)

2 Problem Statement & Approach
2.1 Copying Problem
The copying problem is also known as an off-target translation issue in multilingual NMT especially zero-shot scenario (Gu et al., 2019; Yang et al., 2021; Chen et al., 2023). One important task in zero-shot NMT is to let the model translate into the correct language given so many target languages. Our motivation in UNMT is similar, while each

\(^1\)https://github.com/yihongL1U/xlm_lang_dis
Figure 1: A view of the UNMT architecture. The weights of the final fully connected layer (block $F$) are tied with the weight of the embedding layer (block $E$).

UNMT model often specifically deals with two languages, therefore only two translation directions are considered. Although adding language tags (Wu et al., 2021) is effective in addressing the copying problem in multilingual NMT, it is not a standard process in UNMT. This is because a language embedding is often added to each token embedding (Conneau and Lample, 2019; Song et al., 2019; Liu et al., 2022). Language embeddings have similar functions to language tags: providing information about the language of each token. Unfortunately, language embeddings turn out to be not very effective in addressing the copying problem, especially for low-resource or distant language pairs (Kim et al., 2020). Thus, in this work, we explore why the copying problem occurs and how we can alleviate it in UNMT. We analyze the problem from two perspectives:

**Architecture perspective.** In UNMT, the weight of the final fully connected layer (for obtaining the logits of each word in the vocabulary) is often tied to the weight of a cross-lingual embedding layer, as shown in Figure 1. That is, the representations of tokens from two languages are shared in the same space. Although this setting is arguably a better starting point for most modern NMT models, it unfortunately also allows the models to generate a token in an unexpected language at any time step. Furthermore, because of an autoregressive decoder, errors can easily accumulate, as the tokens initially generated by the model highly influence the generation of the subsequent tokens. In contrast to this setting, using separate word look-up tables or separate decoders for involved languages can address the problem (Lample et al., 2018; Liu et al., 2022). However, such a setting can be harmful for learning cross-lingual knowledge and largely increase the number of parameters. In this view, it is desired to keep the structure simple (no language-specific architecture) while preventing the model from decoding in a copying way.

**Objective perspective.** Typically, a UNMT model is trained by denoising autoencoding (DAE) (Vincent et al., 2008) and online back-translation (BT) (Sennrich et al., 2016) objectives. In DAE objective, even though the model is trained to denoise on two languages simultaneously, there is no guarantee that the model can transfer the cross-lingual information that might improve translation between the two languages. In fact, Song et al. (2019) empirically find that a pretrained encoder-decoder model with DAE objective can even perform worse than the model without it because DAE encourages the model to perform the copying. In comparison with DAE, BT is arguably more important, as it tries to directly optimize the translation. However, we find that BT can also “fail” during training. That is, the model can take the shortcut, i.e., copy the input sentence as the intermediate translation and then copy it again for
the reconstruction. By taking such a shortcut, the loss of BT can quickly decrease while the copying ratio (Liu et al., 2021), a metric to measure the percentage of generated tokens that are copied from the input, keeps increasing and reaches a high-value plateau, as shown in Figure 2. This indicates that: because of no constraints on the intermediate translation, the model can always choose the easiest shortcut for BT, which finally corrupts the model’s translation capability.

### 2.2 A Language Discriminator Loss

To avoid such an unexpected copying behavior in BT, our intuition suggests that forcing the intermediate generation to be in the correct language would be helpful. Instead of forcing all tokens, we could simply force the first token to be in the correct language, because the first generated token will influence the generation of all the subsequent tokens. Next, the problem is how to force the first generated token to be in the desired target language. An equivalent question would be: how can we force the output vector of the decoder at the first time step to be closer to the embedding of a token in the target language? The answer might be trivial. We could use a trained language discriminator (LD), which is a classifier, to classify the first-time-step output vectors of the decoder and then backpropagate the gradients to the main model (encoder and decoder).

In this way, the model knows which intermediate language it should generate for the first-time-step token, therefore preventing the copying behavior.

For training LD, we could use the first-time-step outputs of the decoder in DAE steps. The LD is trained to predict the language of the first-time-step outputs by minimizing the cross entropy loss:

\[
L_{LD} = \mathbb{E}_{x \sim D_{tr}}[p(l|LD(O_l))] \tag{1}
\]

where LD is the language discriminator, O_l are the first-time-step outputs generated by \(\text{Dec}(\text{Enc}(x, l), l)\) and l denotes the language (either src or tgt). Notably, \(L_{LD}\) only backpropagates to the language discriminator in the DAE step. In this way, the discriminator is able to distinguish representations from different languages.

In the BT process, the language discriminator is fixed and \(L_{LD}\) loss is only used to update the main model so it learns to differentiate representations from different languages. Taking src-tgt-src BT for example, the loss is as follows:

\[
L_{LD} = \mathbb{E}_{x \sim D_{tr}}[p(tgt|LD(O_{tgt}))] \tag{2}
\]

where \(O_{tgt}\) are the first-time-step outputs generated in the src-to-tgt step, i.e., \(\text{Dec}(\text{Enc}(x, src), tgt)\). The language discriminator does not have to be used for the next step in BT, i.e., tgt-to-src translation, because there are already ground-truth src-language sentences as supervision. All we need to do is to make sure the intermediate translation is in the correct language. We use a weight \(\lambda_{LD}\) to control the contribution of the LD loss to the final loss that is used to update the parameters of the main model. It is easy to note that the larger the weight, the model will be more focusing on the task of distinguishing representations from different languages.

This training schedule is similar to the adversarial loss (Goodfellow et al., 2014) used by Lample et al. (2018), where they trained a discriminator to make the outputs of the encoder language-agnostic, aiming to improve the cross-linguality of a shared encoder. Our aim, however, is different: we want to enable the decoder to generate distinguishable outputs which correctly correspond to the language that the model is expected to generate in the BT process. Algorithm 1 presents the training schedule in detail.

**Algorithm 1: Training Schedule**

**Input:** pretrained encoder \(\text{Enc}\) and decoder \(\text{Dec}\), language discriminator \(\text{LD}\), source and target monolingual data \(D_{src}, D_{tgt}\), maximum finetuning steps \(T\) and coefficient \(\lambda_{LD}\);

**Output:** Finetuned encoder \(\text{Enc}\) and decoder \(\text{Dec}\);

```
1 t ← 0;
2 while not converged or t < T do
3     // for src language do DAE and BT:
4     B_{src} ← sample batch from \(D_{src}\);
5     // DAE step (below)
6     ˜B_{src}, O_{src} ← generate reconstructions and first-time-step outputs from \(\text{Dec}(\text{Enc}(\text{noise}(B_{src}), src), src)\);
7     detach O_{src} from the compute graph ;
8     \(\theta_{Enc}, \theta_{Dec} \leftarrow \arg \min L_{DAE}(B_{src}, ˜B_{src})\);
9     \(\theta_{LD} \leftarrow \arg \min L_{LD}(O_{src}, src)\);
10    // BT step (below)
11    freeze \(\theta_{LD}\);
12    ˜B_{tgt}, O_{tgt} ← generate tgt-language translations and first-time-step outputs from \(\text{Dec}(\text{Enc}(B_{src}, src), tgt)\);
13    ˜B_{src} ← generate src-language back-translations from \(\text{Dec}(\text{Enc}(B_{tgt}, tgt), src)\);
14    \(\theta_{Enc}, \theta_{Dec} \leftarrow \arg \min L_{BT}(B_{src}, ˜B_{src}) + \lambda_{LD} L_{LD}(O_{tgt}, tgt)\);
15    // for tgt language do the same as above
16    t ← t + 1;
17 end
18 return \(\text{Enc}\) and \(\text{Dec}\);
```
3 Experiments

3.1 Setups

Multi30K (Elliott et al., 2016, 2017)\(^2\). The officially provided train, validation and test sets in English (En), German (De) and French (Fr) are used. Similar to Lample et al. (2018), we only use the caption of each image, and we split the train and validation sets into monolingual corpora by only using one-half of the data for a language.

WMT (Barrault et al., 2019). We select 50M sentences for high-resource languages: English (En), French (Fr), German (De), Russian (Ru) and Chinese (Zh) (14M available) and all available monolingual sentences for low-resource language: Gujarati (Gu) (3M), Kazakh (Kk) (4M). We report the results on newtest2014 for En-Fr pair, newtest2016 for En-De pair, newtest2018 for En-Ru pair and newtest2019 for the remaining language pairs.

Pretrained Models We use cross-lingual pre-trained language model (xlm-mlm-ende-1024 and xlm-mlm-enfr-1024) from HuggingFace\(^3\) (Wolf et al., 2020) to initialize a shared encoder (parameters are fixed) in Multi30K experiments. In those experiments, we randomly initialize a shared decoder because Multi30k is so small that a randomly initialized decoder can work already very well based on our preliminary experiments. For WMT experiments, we pretrain our own cross-lingual language models using the code base of XLM\(^4\) and use the pretrained models to initialize both the encoder and decoder for UNMT task.\(^5\)

3.2 Analysis on Multi30K

To figure out how the LD loss could influence the performance, we use six different weights for it: 0, 0.01, 0.1, 1, 10 and 100. When the weight equals 0, the UNMT training will not consider the LD loss at all and this setting would then be exactly the same as the vanilla (i.e., DAE + BT) UNMT. The results are shown in Table 2. In addition to BLEU scores (Papineni et al., 2002), we also compute copying ratios (Liu et al., 2021) for each listed direction.

The general trend shows that: when \(0 \leq \lambda_{LD} \leq 1\), the BLEU scores increase and the copying ratios decrease when increasing the weight, suggesting the copying problem is alleviated by introducing the LD loss. However, when \(\lambda_{LD} > 1\), the BLEU

\(^2\)https://github.com/multi30k/dataset
\(^3\)https://github.com/huggingface
\(^4\)https://github.com/facebookresearch/XLM
\(^5\)Details of hyperparameters and relevant information of all the models are shown in Section A.2 in the Appendix.
Table 1: Examples of translations from the model trained on Multi30K dataset (En-De pair) with different weights $\lambda_{LD}$ for language discriminator loss. We do not use beam search to generate these translations.

<table>
<thead>
<tr>
<th>$\lambda_{LD}$</th>
<th>Source input</th>
<th>Model output</th>
<th>Reference output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>a man in an orange hat staring at something.</td>
<td>ein mann mit einem orangerobenen hut, der etwas anstarrt.</td>
<td></td>
</tr>
<tr>
<td>0.01</td>
<td>a man in an orange hat staring at something.</td>
<td>ein mann in einem odergefarbenen hut spielt bei etwas.</td>
<td>ein boston terrier läuft auf grünen gras vor einem weißen zaun.</td>
</tr>
<tr>
<td>0.1</td>
<td>a man in an orange hat staring at something.</td>
<td>ein mann in einem odergefarbenen hut spielt bei etwas.</td>
<td>ein boston terrier läuft auf grünen gras vor einem weißen zaun.</td>
</tr>
<tr>
<td>1</td>
<td>a man in an orange hat staring at something.</td>
<td>ein mann in einem odergefarbenen hut spielt bei etwas.</td>
<td>ein boston terrier läuft auf grünen gras vor einem weißen zaun.</td>
</tr>
<tr>
<td>10</td>
<td>a man in an orange hat staring at something.</td>
<td>ein mann in einem odergefarbenen hut spielt bei etwas.</td>
<td>ein boston terrier läuft auf grünen gras vor einem weißen zaun.</td>
</tr>
<tr>
<td>100</td>
<td>a man in an orange hat staring at something.</td>
<td>ein mann in einem odergefarbenen hut spielt bei etwas.</td>
<td>ein boston terrier läuft auf grünen gras vor einem weißen zaun.</td>
</tr>
</tbody>
</table>

Table 2: BLEU scores and copying ratios (inside parentheses) of models trained with different weights $\lambda_{LD}$ on Multi30K dataset. When the weight $\lambda_{LD} = 0$, the model degenerates to the vanilla UNMT model.

<table>
<thead>
<tr>
<th>Models</th>
<th>En - De</th>
<th>De - En</th>
<th>En - Fr</th>
<th>Fr - En</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.22 (85%)</td>
<td>0.19 (84%)</td>
<td>0.14 (88%)</td>
<td>0.10 (85%)</td>
</tr>
<tr>
<td>0.01</td>
<td>15.78 (42%)</td>
<td>22.04 (24%)</td>
<td>24.73 (24%)</td>
<td>22.15 (25%)</td>
</tr>
<tr>
<td>0.1</td>
<td>25.91 (14%)</td>
<td>28.46 (15%)</td>
<td>39.72 (6%)</td>
<td>37.50 (7%)</td>
</tr>
<tr>
<td>1</td>
<td>27.96 (12%)</td>
<td>30.05 (12%)</td>
<td>42.74 (5%)</td>
<td>39.02 (6%)</td>
</tr>
<tr>
<td>10</td>
<td>24.35 (14%)</td>
<td>25.60 (13%)</td>
<td>41.26 (5%)</td>
<td>37.61 (6%)</td>
</tr>
<tr>
<td>100</td>
<td>20.66 (12%)</td>
<td>26.74 (10%)</td>
<td>30.65 (5%)</td>
<td>32.10 (7%)</td>
</tr>
</tbody>
</table>

scores decrease while copying ratios remain at the same level with the increase of the weight. This indicates that the model is over-emphasizing distinguishing the outputs when the weights are large. Therefore, moderate weights, e.g., 1, might be optimal if we want to alleviate the copying problem while achieving good translation performance.

When $\lambda_{LD} = 0$, poor BLEU scores are obtained because of the copying problem. We see that all copying ratios in Table 2 are very high: more than 80% for all directions. Example translations from the translation model for En-De pair in Table 1 show that when $\lambda_{LD} = 0$, the MT system simply copies the input sentences. It is very clear that with the increase of the weight, it becomes less likely for the model to copy the words from the source input as the output translation. However, when the weight is too large, e.g., $\lambda_{LD} = 100$, there are obvious mistakes made by the translation model. For example, “man” in English is wrongly translated to “frau” (means woman) in German, “a” is wrongly translated into “eine” since boston terrier is a masculine instead of a feminine noun. Moderate weights, e.g., $\lambda_{LD} = 1$, achieves the best performance while obtaining fewer errors.

To figure out how the LD loss influences the representations, i.e., the first-time-step output vectors generated by the decoder, we visualize these vectors in 2D by using principal component analysis (PCA), as shown in Figure 3. The visualization verifies the relationship between the output and the occurrence of the copying problem. src2tgt and tgt2tgt first-time-step outputs should be close to each other in the subspace as they are both used to directly generate target-language sentences. However, in Fig. 3 (a), when $\lambda_{LD} = 0$, src2tgt and src2src are located together while tgt2src and tgt2tgt are together. In contrast, when LD loss is imposed, e.g., $\lambda_{LD} = 1$ (Fig. 3 (d)), the outputs are distributed as we expect: src2tgt and tgt2tgt are located together and tgt2src and src2src together.
### 3.3 Main Results on WMT

As the proposed LD is helpful to alleviate the copying problem in Multi30K experiments when the weight $\lambda_{LD}$ is moderate, we further conduct experiments on WMT datasets, which are much larger than Multi30K. We use $\lambda_{LD} = 1$ as default.

#### High-resource language pairs. We report the results on Table 3 and average copying ratios for each language pair in Table 4. Firstly, we observe that there is a slight decrease in BLEU scores for En-De and En-Ru pair. Different from Table 2 where we see that the vanilla models suffer from the copying problem, the vanilla models in Table 3 perform fairly well on En-De and En-Ru. The copying ratios of each pair are also below 20%. We therefore speculate that the size and complexity of the training data can influence the effectiveness of the language discriminator, as it can easily distinguish the decoder outputs in Multi30K because the size is small and each sentence has a similar and simple structure. The copying problem does not severely impact the BLEU scores of these language pairs when training on WMT data, presumably because of the much larger dataset sizes. When the two languages are more distant, however, the copying problem can occur even if considerable training data is there: XLM baseline has a copying ratio of 29% on En-Zh pair. XLM (+LD) can improve results by 0.44 and 0.41 in En-Zh and Zh-En directions, and decrease the copying ratio by 5%, which indicates that the LD loss can improve the translation where the copying problem is obvious.

#### Low-resource language pairs. En-Kk and En-Gu represent two very distant pairs that include low-resource languages. We report the BLEU scores in Table 5 and average copying ratios in Table 4. From the results, we first see that the performance of all considered UNMT systems is rather poor. This is because they are all distant pairs and unsupervised training cannot learn enough cross-lingual information. We find the copying problem overwhelming, with 57% and 68% copying ratios on En-Kk and En-Gu pair respectively. By using the proposed LD loss, we see a consistent increase in BLEU scores and an evident decrease in average copying ratios (15% decrease on En-Kk and 14% on En-Gu pair respectively). This shows the incorporation of LD loss can significantly alleviate the copying problem. On the other hand, we attribute the weak translation quality to the already poor performance of the vanilla UNMT models, which cannot be largely improved simply by alleviating the copying problem. Decreasing copying ratios does not necessarily lead to a correct translation. Because of the unsupervised nature of the task, it can still be extremely hard for the model to learn enough cross-lingual information that is useful to perform good translation. Table 6 shows some examples, we notice that XLM (+ LD) generates sentences in the correct language, but the semantics of the output sentences is not that related to the original ones, indicating that lower copying ratios do not necessarily induce better translation quality.

---

**Table 3:** BLEU scores of the XLM baseline and the same model enhanced with the LD loss on high-resource language pairs. The scores of baseline are obtained by reproducing the published code (Conneau and Lample, 2019).

<table>
<thead>
<tr>
<th>Models</th>
<th>En - De</th>
<th>De - En</th>
<th>En - Fr</th>
<th>Fr - En</th>
<th>En - Ru</th>
<th>Ru - En</th>
<th>En - Zh</th>
<th>Zh - En</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline</td>
<td>20.51</td>
<td>25.99</td>
<td>22.87</td>
<td>25.88</td>
<td>14.10</td>
<td>16.92</td>
<td>6.36</td>
<td>4.28</td>
</tr>
</tbody>
</table>

**Table 4:** The copying ratio for each language pair of XLM baselines and LD model. The average of the ratios of two directions for a language pair is reported. The translations used to compute the ratios are the same as translations for BLEU used in Table 3 and Table 5.

<table>
<thead>
<tr>
<th>Models</th>
<th>En-De</th>
<th>En-Fr</th>
<th>En-Ru</th>
<th>En-Zh</th>
<th>En-Kk</th>
<th>En-Gu</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>18%</td>
<td>23%</td>
<td>11%</td>
<td>29%</td>
<td>57%</td>
<td>68%</td>
</tr>
<tr>
<td>(+ LD)</td>
<td>19%</td>
<td>25%</td>
<td>11%</td>
<td>24%</td>
<td>42%</td>
<td>52%</td>
</tr>
<tr>
<td>$\Delta$</td>
<td>+1%</td>
<td>+2%</td>
<td>-0%</td>
<td>-5%</td>
<td>-15%</td>
<td>-14%</td>
</tr>
</tbody>
</table>

**Table 5:** BLEU scores of the XLM baseline and the same model enhanced with the LD loss on low-resource language pairs. The scores of baseline (512) are copied from (Kim et al., 2020). Same as the setting for high-resource languages, we reproduced XLM with 1024-dim embeddings to obtain the scores for baseline (1024).

<table>
<thead>
<tr>
<th>Models</th>
<th>En-Kk</th>
<th>Kk-En</th>
<th>En-Gu</th>
<th>Gu-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline (512)</td>
<td>0.80</td>
<td>2.00</td>
<td>0.60</td>
<td>0.60</td>
</tr>
<tr>
<td>XLM baseline (1024)</td>
<td>1.80</td>
<td>1.59</td>
<td>2.12</td>
<td>0.54</td>
</tr>
<tr>
<td>XLM (+ LD)</td>
<td>2.03</td>
<td>1.70</td>
<td>3.55</td>
<td>0.64</td>
</tr>
</tbody>
</table>

---
<table>
<thead>
<tr>
<th>Model</th>
<th>Source input</th>
<th>Model output</th>
<th>Reference output</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline</td>
<td>Негізі, менің қарсылығым жоқ.</td>
<td>Негізі, менің қарсылығым жоқ.</td>
<td>Actually, I have no objection.</td>
</tr>
<tr>
<td>XLM (+LD)</td>
<td>Бұл сома алты еуроға тең.</td>
<td>The сома алты еуроға тең.</td>
<td>This amount equals to six euro.</td>
</tr>
<tr>
<td>XLM baseline</td>
<td>Олардың көпшілігі ауыл шаруашылығы</td>
<td>Their family life has changed.</td>
<td>Most of them are in agricultural area.</td>
</tr>
<tr>
<td>XLM (+LD)</td>
<td>саласында болып отыр.</td>
<td>Their family members have been</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>in the area for the past two years.</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Examples of translations from Kazakh to English by XLM baseline (1024) and XLM (+LD) in Table 5. The examples show XLM (+LD) suffers fewer the copying problem but it can generate incorrect tokens that do not match the semantics of the input sentence.

Based on the high- and low-resource translation experiments, our insights are as follows: the UNMT models can (easily) learn a lot of cross-lingual information on similar and high-resource languages and thus the copying problem is less obvious. Under such a case, additionally using LD loss can divert the focus of the training. However, on distant pairs involving low-resource languages, models would struggle to learn enough cross-lingual information and therefore the copying problem is obvious. In such a case, although involving LD loss cannot provide additional cross-lingual knowledge, it can alleviate the copying problem thus improving the performance to a certain extent.

4 Discussion

From the Multi30K and WMT experiments, we verify the ability of the LD loss to alleviate the copying problem by showing consistently lower copying ratios. However, the performance in terms of BLEU scores on these two datasets shows slightly different trends: we improve translation quality on Multi30K a lot by reducing the copying ratios; whereas we do not see a prominent improvement on WMT even if copying ratios are largely reduced. This discrepancy can be explained as follows. Two main issues are preventing the model from achieving good performance: (1) lacking cross-lingual alignment information that is useful for learning translation (2) no clear guidance on which language to translate into. The experiments on the small dataset Multi30K indicate that issue (1) is not the major obstacle when two similar languages are considered, e.g., En and Fr. In such a case, it is the issue (2) that prevents the model from performing the actual translation. This is why large improvements are achieved by simply adding the LD loss when training a model on Multi30k (note that the language discriminator does not provide any additional cross-lingual information but only acts as an implicit supervision). In the case of distant language pairs including low-resource languages, e.g., En-Gu and En-Kk in our WMT experiments, both issues (1) and (2) prohibit the model from learning to translate accurately. Although the copying problem is alleviated, as shown in Table 6, this does not guarantee a correct or even good translation quality. We therefore expect future research could explore using a more powerful baseline model, e.g., including static cross-lingual embeddings to improve the cross-linguality (Chronopoulou et al., 2021), which might further improve the performance for distant language pairs including low-resource languages.

5 Conclusion

In this paper, we find that the copying problem in UNMT is closely related to the lack of constraints on the intermediate translation in the BT process. To address this issue, we propose an LD loss to give additional supervision to the first-time-step output vectors generated by the decoder in the BT process. We find that the method can alleviate the copying problem by correcting the wrong behavior in BT. In addition, through extensive experiments on different language pairs (including low-resource languages and distant pairs), we discover that the method can consistently improve the performance of distant language pairs.

6 Limitations and Risks

Our training schedule introduces a language discriminator loss to impose constraints on the intermediate translation in the back-translation period. The experimental results suggest that our method can alleviate the copying problem when the involved languages are distant language pairs or lack training data. However, for language pairs that are not distant, and especially high-resource languages, our model does not show improvement over the baseline. Due to time and resource limitations, we do not further explore whether the optimal weight
for the language discriminator loss can have a connection with the size of the dataset and the involved language pairs. For example, for WMT En-De or En-Fr pairs, the languages are not distant language pairs and therefore we might obtain better results if the weights are slightly smaller. We believe that future research could explore this direction: to adapt the weight to different language pairs and the size of the training data. In addition, we do not conduct hyperparameter search for other hyperparameters, instead directly using suggested values.

In this work, we propose a novel training schedule that tries to address the copying problem, which is common among distant language pairs in UNMT. We experiment with high-resource languages English, German, French, Russian and Chinese, and low-resource languages including Gujarati and Kazakh. The training data we use is monolingual text extracted from online newspapers and released for the WMT series of shared tasks. As far as we know, all the monolingual corpora do not contain any metadata and therefore it would be unlikely that anyone can use the concerned data to attribute to specific individuals.

Acknowledgements

We would like to thank the anonymous reviewers. This work was funded by the European Research Council (grant #740516) and by the German Research Foundation (DFG, grant FR 2829/4-1).

References


ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.


A Appendix

A.1 Scores of Other Metrics

In addition to BLEU scores, we also compute other scores in other metrics, such as CHRF (Popović, 2015) in Table 9 and Table 7, COMET (Rei et al., 2020) in Table 10 and Table 8, and confidence interval of BLEU scores (Koehn, 2004) in Table 11, Table 12 and Table 13. The translations used for computing the scores are the same as the translations used to compute the BLEU scores in Table 3 and Table 5.

To quantify the copying problem, we use the copying ratio proposed by Liu et al. (2021), which is defined as follows:

\[
\text{Ratio} = \frac{\sum_{i=1}^{I} \text{count}(\text{copying tokens})}{\sum_{i=1}^{I} \text{count}(\text{tokens})}
\]

(3)

where \( I \) denotes the number of the total sentences in the test set, copying tokens are those tokens in the translation which are directly copied from the source language and the denominator is the total number of tokens in the generated translations. This metric will directly reflect the degree of the copying behavior of the translation model. The higher the copying ratio, the model tends to perform more copying instead translation. We report the average of the copying ratios of the two translation directions for each language pair in Table 4.

We could see that the copying problem of the XLM baseline models is very obvious in low-resource language pairs, i.e., En-Kk and En-Gu. When the language discriminator loss is introduced, the copying ratios decrease by more than 10%. We also notice that XLM (+LD) has a less obvious copying problem than the baseline in En-Zh pair, a distant language pair. For other language pairs, the copying problem is not that severe and therefore introducing the language discriminator loss does not much change the ratios.

A.2 Model Details

In Section 3.2, we use the pretrained XLM models from HuggingFace6 (Wolf et al., 2020) (xlm-mlm-enfr-1024, xlm-mlm-ende-1024) to initialize a shared encoder and randomly initialize a shared decoder. A single embedding layer (containing the words/subwords of both the source and target languages) from the pretrained encoder is used. The weight of the final fully connected layer is tied with the embedding layer. The parameters of the encoder are fixed except for this embedding layer which is also used by the decoder. The embedding size is 1024 and the hidden size of the decoder is 512. The decoder has 8 heads and 3 layers. We follow the denoising autoencoding hyperparameter settings used by Lample et al. (2018) and the training schedule of Liu et al. (2022), i.e., firstly fine-tuning the models with only DAE loss and LD loss for the language discriminator for the first 2 epochs, then fine-tuning the models with all losses (including the BT) for the rest of the epochs. We set the batch size to 32 and use Adam optimizer (Kingma and Ba, 2015) with an initial learning rate of 0.0001. We stop the training when the model does not improve the BLEU scores on the validation set for 5 epochs. We do not use beam search to generate translations for Multi30K.

In Section 3.3, we pretrain all our own cross-lingual language models of each language pair based on XLM code base7 (Conneau and Lample, 2019). Then the encoder and decoder are both initialized with the same cross-lingual pretrained model. The recommended hyperparameters for the model architecture are used, i.e., 1024 for the embedding size, 4096 for the hidden size, 8 heads and 6 layers for the transformer blocks. We follow the recommended pretraining as well as UNMT fine-tuning hyperparameters from XLM. We only change the hyperparameter \( \text{tokens_per_batch} \) to 250 to adapt to small- or moderate memory GPUs. We generate the translations by using beam search of size 5. These translations are used to compute the scores in all the WMT-related experiments.

For the language discriminator, we simply use a feed-forward neural network (FFNN). The language discriminator has two hidden layers and each layer has the same dimension as the embedding, i.e., 1024, for both Multi30K and WMT-related experiments. The output dimension is two which corresponds to the number of language domains we want to classify into, as we have two languages involved in the training for each model.

---

6https://github.com/huggingface

7https://github.com/facebookresearch/XLM
<table>
<thead>
<tr>
<th>Models</th>
<th>En-Kk</th>
<th>Kk-En</th>
<th>En-Gu</th>
<th>Gu-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline</td>
<td>8.85</td>
<td>7.61</td>
<td>7.95</td>
<td>4.76</td>
</tr>
<tr>
<td>XLM (+ LD)</td>
<td><strong>11.78</strong></td>
<td><strong>10.09</strong></td>
<td><strong>11.71</strong></td>
<td><strong>7.12</strong></td>
</tr>
</tbody>
</table>

Table 7: CHR F scores (Popović, 2015) of the XLM UNMT baseline as well as the XLM model with the language discriminator on low-resource language pairs (the translations used are the same as used in Table 5 for BLEU scores).

<table>
<thead>
<tr>
<th>Models</th>
<th>En-Kk</th>
<th>Kk-En</th>
<th>En-Gu</th>
<th>Gu-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline</td>
<td>-1.41</td>
<td>-1.10</td>
<td>-1.40</td>
<td>-1.90</td>
</tr>
<tr>
<td>XLM (+ LD)</td>
<td><strong>-1.14</strong></td>
<td><strong>-1.04</strong></td>
<td><strong>-0.91</strong></td>
<td><strong>-1.68</strong></td>
</tr>
</tbody>
</table>

Table 8: COMET scores (Rei et al., 2020) of the XLM UNMT baseline as well as the XLM model with the language discriminator on low-resource language pairs (the translations used are the same as used in Table 5 for BLEU scores). We use wmt20-comet-da model to evaluate the translations.
### Table 9: CHRF scores (Popović, 2015) of the XLM UNMT baseline as well as the XLM model with the language discriminator on high-resource language pairs (the translations used are the same as used in Table 3 for BLEU scores).

<table>
<thead>
<tr>
<th>Models</th>
<th>En-De</th>
<th>De-En</th>
<th>En-Fr</th>
<th>Fr-En</th>
<th>En-Ru</th>
<th>Ru-En</th>
<th>En-Zh</th>
<th>Zh-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline</td>
<td>45.09</td>
<td>48.20</td>
<td>44.99</td>
<td>49.93</td>
<td>34.75</td>
<td>38.56</td>
<td>16.11</td>
<td>19.08</td>
</tr>
<tr>
<td>XLM (+ LD)</td>
<td>44.42</td>
<td>48.20</td>
<td>42.94</td>
<td>50.50</td>
<td>34.39</td>
<td>36.56</td>
<td>16.74</td>
<td>20.45</td>
</tr>
</tbody>
</table>

### Table 10: COMET scores (Rei et al., 2020) of the XLM UNMT baseline as well as the XLM model with the language discriminator on high-resource language pairs (the translations used are the same as used in Table 3 for BLEU scores). We use wmt20-comet-da model to evaluate the translations.

<table>
<thead>
<tr>
<th>Models</th>
<th>En-De</th>
<th>De-En</th>
<th>En-Fr</th>
<th>Fr-En</th>
<th>En-Ru</th>
<th>Ru-En</th>
<th>En-Zh</th>
<th>Zh-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline</td>
<td>-0.19</td>
<td>-0.22</td>
<td>-0.04</td>
<td>0.19</td>
<td>-0.34</td>
<td>-0.22</td>
<td>-0.43</td>
<td>-0.78</td>
</tr>
<tr>
<td>XLM (+ LD)</td>
<td>-0.22</td>
<td>-0.23</td>
<td>-0.04</td>
<td>0.21</td>
<td>-0.37</td>
<td>-0.33</td>
<td>-0.36</td>
<td>-0.81</td>
</tr>
</tbody>
</table>

### Table 11: 95% confidence interval for the BLEU scores of the XLM UNMT baseline as well as the XLM model with the language discriminator on En-De and En-Fr pair (the translations used are the same as used in Table 3 for BLEU scores). Differences between bold results are statistically significant under $p = 0.05$. For the statistical test, we use paired bootstrap resampling (Koehn, 2004).

<table>
<thead>
<tr>
<th>Models</th>
<th>En-De</th>
<th>De-En</th>
<th>En-Fr</th>
<th>Fr-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline</td>
<td>20.53±0.59</td>
<td>25.96±0.66</td>
<td>22.85±0.72</td>
<td>25.89±0.57</td>
</tr>
<tr>
<td>XLM (+ LD)</td>
<td>20.42±0.61</td>
<td>25.84±0.63</td>
<td>21.18±0.76</td>
<td>26.92±0.59</td>
</tr>
</tbody>
</table>

### Table 12: 95% confidence interval for the BLEU scores of the XLM UNMT baseline as well as the XLM model with the language discriminator on En-Ru and En-Zh pair (the translations used are the same as used in Table 3 for BLEU scores). Differences between bold results are statistically significant under $p = 0.05$. For the statistical test, we use paired bootstrap resampling (Koehn, 2004).

<table>
<thead>
<tr>
<th>Models</th>
<th>En-Ru</th>
<th>Ru-En</th>
<th>En-Zh</th>
<th>Zh-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline</td>
<td>14.08±0.48</td>
<td>16.93±0.51</td>
<td>6.34±0.34</td>
<td>4.28±0.28</td>
</tr>
<tr>
<td>XLM (+ LD)</td>
<td>13.48±0.45</td>
<td>16.11±0.51</td>
<td>6.80±0.37</td>
<td>4.69±0.31</td>
</tr>
</tbody>
</table>

### Table 13: 95% confidence interval for the BLEU scores of the XLM UNMT baseline as well as the XLM model with the language discriminator on En-Kk and En-Gu pair (the translations used are the same as used in Table 3 for BLEU scores). Differences between bold results are statistically significant under $p = 0.05$. For the statistical test, we use paired bootstrap resampling (Koehn, 2004).

<table>
<thead>
<tr>
<th>Models</th>
<th>En-Kk</th>
<th>Kk-En</th>
<th>En-Gu</th>
<th>Gu-En</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLM baseline</td>
<td>1.80±0.37</td>
<td>1.58±0.48</td>
<td>2.13±0.31</td>
<td>0.54±0.17</td>
</tr>
<tr>
<td>XLM (+ LD)</td>
<td>2.04±0.45</td>
<td>1.69±0.49</td>
<td>3.56±0.41</td>
<td>0.64±0.20</td>
</tr>
</tbody>
</table>