On the Copying Behaviors of Pre-Training for Neural Machine Translation

Xuebo Liu[1*], Longyue Wang[2], Derek F. Wong[1], Liang Ding[3],
Lidia S. Chao[1], Shuming Shi[2] and Zhaopeng Tu[2]

[1]NLP[2]CT Lab, Department of Computer and Information Science, University of Macau
[2]Tencent AI Lab [3]The University of Sydney
`nlp2ct.xuebo@gmail.com`, `{derekfw,lidiasc}@um.edu.com`
`{vinnylywang,shumingshi,zptu}@tencent.com`
`ldin3097@sydney.edu.au`

Abstract

Previous studies have shown that initializing neural machine translation (NMT) models with the pre-trained language models (LM) can speed up the model training and boost the model performance. In this work, we identify a critical side-effect of pre-training for NMT, which is due to the discrepancy between the training objectives of LM-based pre-training and NMT. Since the LM objective learns to reconstruct a few source tokens and copy most of them, the pre-training initialization would affect the copying behaviors of NMT models. We provide a quantitative analysis of copying behaviors by introducing a metric called *copying ratio*, which empirically shows that pre-training based NMT models have a larger copying ratio than the standard one. In response to this problem, we propose a simple and effective method named *copying penalty* to control the copying behaviors in decoding. Extensive experiments on both in-domain and out-of-domain benchmarks show that the copying penalty method consistently improves translation performance by controlling copying behaviors for pre-training based NMT models. Source code is freely available at `https://github.com/SunbowLiu/CopyingPenalty`.

1 Introduction

Self-supervised pre-training (Devlin et al., 2019; Song et al., 2019), which acquires general knowledge from a large amount of unlabeled data to help *better* and *faster* learning downstream tasks, has an intuitive appeal for neural machine translation (NMT; Bahdanau et al., 2015; Vaswani et al., 2017). One direct way to utilize pre-trained knowledge is initializing the NMT model with a pre-trained language model (LM) before training it on parallel data (Conneau and Lample, 2019; Liu et al.,

LM Pre-Training: $\mathcal{L}_{\mathrm{PT}} = -\log P(\mathbf{x}|\tilde{\mathbf{x}})$	
Source	Military _ Field Marshal Hussein _ _ in attendance.
Target	==Military== ruler ==Field== ==Marshal== ==Hussein== Tantawi was ==in== ==attendance==.

NMT Training: $\mathcal{L}_{\mathrm{NMT}} = -\log P(\mathbf{y}|\mathbf{x})$	
Source	Military ruler Field Marshal Hussein Tantawi was in attendance.
Target	Der Militärführer Feldmarschall ==Hussein== ==Tantawi== war anwesend.

Table 1: Training objective gap between Seq2Seq LM pre-training and NMT training. LM learns to reconstruct a few source tokens and copy most of them, while NMT learns more translation rather than copying. Underlines denote artificial noises, and ==highlights== indicate expected copying tokens.

2020). As a range of surface, syntactic and semantic information has been encoded in the initialized parameters (Jawahar et al., 2019; Goldberg, 2019), they are expected to bring benefits to NMT models and hence the translation quality.

However, there is a discrepancy between the training objective of sequence-to-sequence LM pre-training and that of NMT training. As shown in Table 1, LM learns to reconstruct all source tokens with some noises, while NMT learns to translate most source tokens and copy few of them. Knowles and Koehn (2018) and Liu et al. (2020) show that LM pre-training requires to copy ∼65% of tokens, while NMT training only needs to copy <10%. We believe that unexpected knowledge can be propagated to the NMT model via pre-training, which may bias NMT models to mistakenly copy source tokens to the target side. For example, the source word "Field Marshal" might be mistakenly copied to the target side by pre-training based NMT models, since such copying behaviors can be learned in

the pre-training stage.

In this paper, we first validate the change of copying behaviors in NMT models initialized with the pre-training weights. To this end, we propose a metric named *copying ratio* to quantitatively measure the extent of copying behaviors of NMT models. Experimental results on the WMT14 En-De data show that the NMT model with pre-training improves translation performance at the cost of introducing more copying predictions. Analyses on model training show that the NMT model with pre-training attempts to forget the copying behaviors transferred from pre-training, while the vanilla NMT model learns in the opposite way. Due to the dominating copying behaviors in the pre-training, the copying ratio of pre-training based NMT model (i.e., 10.8%) is much higher than that of the vanilla NMT model (i.e., 9.3%). Extensive analyses show that higher copying ratios severely hurt sentence fluency and word accuracy in translations, particularly for the translation of proper nouns, establishing the necessity for controlling the copying behaviors of NMT models.

To tackle this problem, we propose a simple and effective *copying penalty* to control the copying behaviors in inference, which requires no modification to model architectures and training algorithms. Specifically, we introduce a new regularizing term to the prediction at each time step, which guides the model to copy source tokens only when the model is highly confident. Experimental results on the WMT14 English-German and the OPUS German-English benchmark demonstrate that the proposed approach can significantly control copying behaviors in NMT models, making the model more accurately generate copying tokens.

Our contributions are summarized as follows:

- We reveal a critical side-effect of pre-training for NMT, where pre-training introduces more copying behaviors into NMT outputs.

- We propose a simple and effective *copying penalty* to further improve the performance of NMT models with pre-training by controlling copying behaviors in generated translation.

- We find that the domain containing a large number of copying tokens (e.g., the IT) benefits more from the proposed copying penalty.

2 Observing Copying Behavior Changes

The fact is that some source words are excessively copied by NMT models from the source to the target side instead of being translated, which leads to a high copying ratio in NMT outputs. In this section, we first propose a metric to measure the copying ratio of model predictions. Second, we quantitatively investigate the effect of pre-training on NMT in the perspective of copying behaviors. We expect to provide more evidence for controlling the copying behaviors of NMT models.

2.1 Experimental Setup

Data We conducted experiments on the widely-used WMT14 English-German benchmark. We used the processed data provided by Vaswani et al. (2017), which consists of 4.5M sentence pairs.[1] We used all the training data for model training. The validation set is newstest2013 of 3,000 examples and the test set is newstest2014 of 3,003 examples.

Models and Settings We implemented all the models by the open-sourced toolkit fairseq (Ott et al., 2019).[2] We used 8 V100 GPUs for the experiments. We mainly compared two models: 1) RANDOM, which is a vanilla NMT model whose weights are randomly initialized without pre-training; and 2) PRETRAINED, an NMT model using the weights of pre-trained mBART.cc25[3] for parameter initialization, which has shown its usability and reliability for translation tasks (Tran et al., 2020; Tang et al., 2020).

For the training of RANDOM, we used the Transformer *big* setting of Ott et al. (2018b) with a huge training batch size of 460K tokens.[4] For PRETRAINED, we fine-tuned on the pre-trained mBART.cc25 with a training batch size of 131K tokens. The hyperparameters keep the same with RANDOM except the 0.2 label smoothing, 2500 warm-up steps, and 1e-4 maximum learning rate.

Evaluation For each model, we selected the checkpoint with the lowest perplexity on the validation set for testing. The beam size is 5 and the length penalty is 0.6. In addition to report-

---

	Source	Military ruler Field *Marshal* Hussein Tantawi was in attendance.
	Target	Der Militärführer Feldmarschall Hussein Tantawi war anwesend.
	RANDOM	Anwesend war der Militärmachthaber Feldmarschall Hussein Tantawi.
	PRETRAINED	Militärischer Feldherr *Marshal* Hussein Tantawi war anwesend.

Table 2: Translation from English to German. The words in color denote the copying tokens of which blue denotes right copies and *red* denotes copying errors.

	Performance		Copying		
	BLEU	**TER**	**Ratio**	**CER**	**#S**
ORACLE	-	-	8.5%	-	9
RANDOM	28.3	60.7	9.3%	17.4	20
PRETRAINED	29.4	59.4	10.8%	27.6	50

Table 3: Results on the WMT14 En-De test set. OR-ACLE denotes the statistics on the reference. "#S" denotes the number of instances whose overlaps between the source and target exceeding 50% (Ott et al., 2018a). Although PRETRAINED gains better model performance than RANDOM, it also excessively copies tokens from the source.

ing the commonly-used 4-gram BLEU score (Papineni et al., 2002), we also report Translation Error Rate (TER) (Snover et al., 2006) to better capture the translation performance of unigrams, which more directly reflects the copying behaviors of NMT models. Both the scores are calculated by `sacrebleu` (Post, 2018) with de-tokenized text and unmodified references.[5],[6]

2.2 Copying Ratio

Ratio To measure the extent of the copying behaviors in NMT models, we calculate the ratio of copying tokens in translation outputs:

$$\text{Ratio} = \frac{\sum_{i=1}^{I} \text{count(copying token)}}{\sum_{i=1}^{I} \text{count(token)}} \quad (1)$$

where I denotes the total number of sentences in the test set. We count the number of "copying token" by comparing each input and output sentence pair. The denominator is the total number of tokens in output sentences. In general, higher Ratio values indicate more copying behaviors produced by the NMT model, and vice versa.

Copying Error Rate (CER) To further analyze the copying problem in NMT models, we propose to calculate the rate of incorrect copying tokens among all copied ones:

$$\text{CER} = \frac{\sum_{i=1}^{I} \text{count(copying error)}}{\sum_{i=1}^{I} \text{count(copying token)}} \quad (2)$$

where we count the number of "copying error" by checking whether the copying tokens are included in its reference sentence. The CER is expected to be zero, which indicates that all copying tokens are correct. Table 2 gives an example. In experiments,

Ratio and CER are computed based on words rather than sub-words. We further filter all punctuations, which are similar across different languages.

Model Performance We compare the performance and copying behaviors of the final models in Table 3. The results show that although PRETRAINED improves the overall performance in terms of BLEU and TER scores, it tends to generate more copying errors, limiting its further improvement. In the following part, we probe into the essence of the copying behaviors via carefully designed experiments.

2.3 Learning Curves of Copying Behaviors

We analyze copying behaviors in learning dynamics. Specifically, we translate the test set using intermediate checkpoints at different training steps, and then compute corresponding Ratio and CER values. We compare RANDOM and PRETRAINED, and plot their learning curves in Figure 1.

Ratio Two models behave quite differently in the early stages of training. Taking Step 100 for instance, PRETRAINED copies 89% tokens while RANDOM does not generate any copying tokens. This demonstrates that the copying habit in the pretrained model is transferred to NMT models. As training proceeds, the copying behaviors of PRETRAINED are heavily suppressed, resulting in a rapid drop in Ratio. On the contrary, RANDOM is able to quickly learn copying from scratch, leading to an upward trend. After learning curves become stable, PRETRAINED performs more copying behaviors than RANDOM (10.8% vs. 9.3% Ratio).

CER In general, the results of CER show similar trends to those observed in Ratio. In the beginning, the CER of PRETRAINED is extremely high (i.e.,

---

[5] BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.4.14

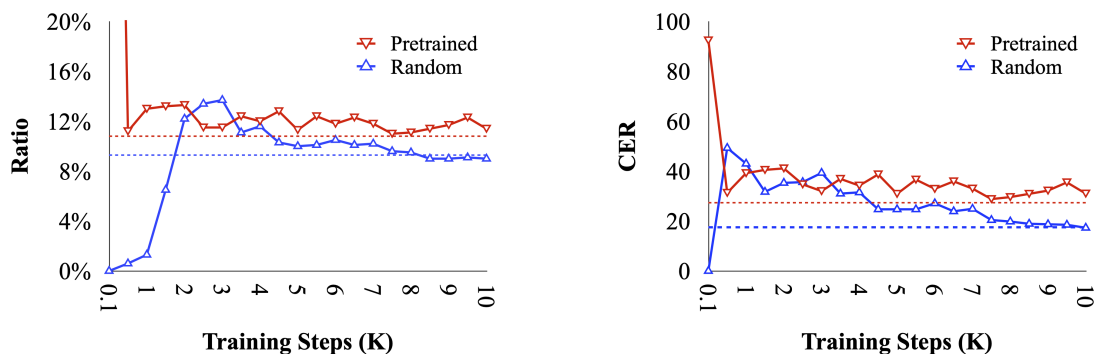[6] TER+lang.en-de+tok.tercom-nonorm-punct-noasian-uncased+version.1.4.14

Figure 1: Copying ratio (left) and CER score (right) of RANDOM and PRETRAINED at different training steps. Reference lines report the values of the final models gaining the lowest validation perplexities. PRETRAINED gains 89.4% copying ratio at 0.1K step, which is omitted for better display clarity. PRETRAINED learns to forget its copying behaviors by reducing the copying ratio from 89.4% to 10.8% and CER from 92.4 to 27.6, while RANDOM learns copying from scratch by increasing the copying ratio and CER from 0 to 9.3% and 17.4, respectively.

	PPL	Rand>Pre		Rand<Pre	
		Ratio	CER	Ratio	CER
RANDOM	60.8	10.0%	18.5	8.7%	17.4
PRETRAINED	95.1	9.5%	14.9	12.5%	39.7

Table 4: Sentence perplexity on the test set. PRETRAINED's translation of worse perplexity (i.e., "Rand<Pre") also contains the higher Ratio (12.5%) and CER (39.7) scores.

92.4), revealing that most of the copying tokens are incorrect. The reason behind this phenomenon is that pre-trained models are accustomed to copying source words but the habit is overly transferred to the downstream translation models. The interesting finding is that RANDOM also makes more mistakes on copying at the early training stage. Finally, the error rate of PRETRAINED is much higher than that of RANDOM (27.6 vs. 17.4), showing that pretrained models indeed expose harmful knowledge to NMT models.

Learning curves of two kinds of models perform in opposite ways: RANDOM learns copying from scratch while PRETRAINED tries to forget this behavior. As a result, PRETRAINED copies more source tokens than RANDOM and suffers severe copying errors. This motivates us to further investigate the effects of copying behaviors on NMT models in terms of translation quality.

2.4 Effect of Copying Ratio

Sentence Fluency The copying tokens from the source usually contain some tokens that do not belong to the target language, which might hurt the fluency of generated translations. Starting from this

intuition, we use an external language model (Ng et al., 2019)[7] trained on in-domain data to evaluate the fluency of translation outputs. As shown in Table 4, Random achieves a much better perplexity than mBART (60.8 vs. 95.1 PPL), demonstrating that the NMT model with pre-training generates less fluent sentences than that trained from scratch.

To take a closer look at the fluency gap, we divide outputs of each model into two subsets: sentences with better or worse perplexity by comparing RANDOM and PRETRAINED. As seen, the fluency of translation is related to copying ratio and errors. The sentences with higher Ratio and CER scores tend to be less fluent. Taking PRETRAINED's worse subset for example (Rand<Pre), it gains a 12.5% Ratio and 39.7 CER scores, confirming that excessive copying behaviors lead to negative effects in terms of translation fluency.

Word Accuracy We also give a word-level analysis by bucketing copying tokens according to part-of-speech (POS) tags and calculate Ratio and CER in each type. In experiment, we employ Stanford POS tagger with the `german-ud.tagger` model to automatically label output sentences (Toutanova et al., 2003). Table 5 lists the results. The "Oracle" (Row 1) denotes the statistics by comparing the source input and its reference. As seen, most copying operations should occur in the type of proper noun (PROPN). This type occupies 5.7% Ratios, followed by adposition (ADP), numeral (NUM), noun (NOUN), and other types (Others).

Compared with RANDOM, we observe that the

---

[7] https://dl.fbaipublicfiles.com/fairseq/models/lm/wmt19.de.tar.gz

	Total		PROPN		ADP		NUM		NOUN		Others	
	Ratio	CER*	Ratio	CER*	Ratio	CER*	Ratio	CER*	Ratio	CER*	Ratio	CER*
ORACLE	8.5%	-	5.7%	-	1.1%	-	0.8%	-	0.5%	-	0.3%	-
RANDOM	9.3%	17.4	6.3%	14.5	1.3%	25.8	0.9%	14.2	0.5%	21.3	0.3%	44.4
PRETRAINED	10.8%	27.6	7.5%	27.3	1.3%	24.9	0.9%	14.0	0.6%	21.0	0.5%	63.8
Δ	**+1.5%**	**+10.2**	**+1.2%**	**+12.8**	0%	-0.9	0%	-0.2	+0.1%	+0.3	+0.2%	**+19.4**

Table 5: Copying behaviors by part-of-speech (POS) bucket on the WMT14 En-De task. "Oracle" denotes the statistics in the reference. "CER*" denotes only using the copying tokens belonging to each POS category for the CER calculation. Δ denotes the changes from the RANDOM to PRETRAINED, in which the significant ones are **bold**. Most of the copying tokens are found in translating proper nouns (PROPN) in PRETRAINED.

increase of Ratio for PRETRAINED mainly attributes to copying PROPN words (+1.2%). In addition, PRETRAINED generates more copying errors (+10.2), especially on PROPN and Others types (+12.8 and +19.4). These results reveal that it is necessary to pay more attention to proper nouns on controlling copying behaviors for NMT.

3 Controlling Copying Behaviors

Based on the above experiments, we prove that pre-training indeed changes the copying behaviors of NMT models, hurting the sentence fluency and word accuracy of generated translations. To alleviate this issue, we propose a simple and effective method *copying penalty* to make the copying behaviors in NMT controllable.

3.1 Copying Penalty (CP)

To control copying behaviors in NMT, an intuitive way is generating copying tokens only when the model is of high confidence. To this aim, we propose to modify the probability distribution predicted by the NMT model, decreasing the predicting probability of the tokens also occurred in the source (i.e., weakening the model confidence of making copying predictions). In this way, for those predictions are wavering between copying and translating, the model is more likely to translate them, and thus only those confident copying tokens will be retained. Specifically, during inference, the predicting probability of t-th time step is as follows:

$$P(y_j|y_{<j}, \mathbf{x}) \in \mathbb{R}^{\mathcal{V}} = \mathrm{softmax}(\mathbf{y}_t) \quad (3)$$

where $P(y_j|y_{<j}, \mathbf{x})$ denotes the probability over the whole target vocabulary and $\mathbf{y}_t$ denotes the decoder output of t-th time step. The search algorithm (e.g., beam search) will take this probability distribution as a candidate to find the final translation of the source sentence.

Copying penalty regularizes the prediction probability of each time step by element-wisely multiplying a new constraint $\mathrm{CP} \in \mathbb{R}^{\mathcal{V}}$:

$$\mathrm{CP} = \begin{cases} 1, & y_j \notin \mathbf{x}/\mathcal{C}_{\mathrm{punc}} \\ \alpha, & y_j \in \mathbf{x}/\mathcal{C}_{\mathrm{punc}} \end{cases} \quad (4)$$

where α is a hyperparameter to control the penalty which can be tuned on the development set, similar to length penalty (Wu et al., 2016). $\mathbf{x}/\mathcal{C}_{\mathrm{punc}}$ denotes the set of sources tokens excluding punctuation and eos, which means that the prediction probabilities of punctuation and eos will not be penalized. For those predictions not belonging to the source, their probabilities keep the same. But for those predictions that are copied from the source, their probabilities will be α times as large/small as before and the model will be more/less likely to choose them as candidates for searching.

The proposed method is simple and effective: 1) It does not change the model architecture and does not need any additional model training, thus no parameters needed to be newly introduced; 2) Its implementation only requires some low-cost matrix operations during model inference, slightly slowing the decoding speed; and 3) It can significantly control the overall copying ratio of the model predictions, making the model accurately generate copying tokens, as shown in the following sections.

Effect of Copying Penalty Figure 2 depicts the changes of the copying ratio and CER scores when setting CP to different values on the test data. When setting CP smaller than 1 (i.e., punishing copying tokens), only those confident copying predictions will be made, and thus reducing both the copying ratio and CER scores. Conversely, setting CP larger than 1 makes the model generate more copying tokens even some of them are of low confidence, thus both the copying ratios and CER scores increase. Similar to length penalty (Wu et al., 2016), we also
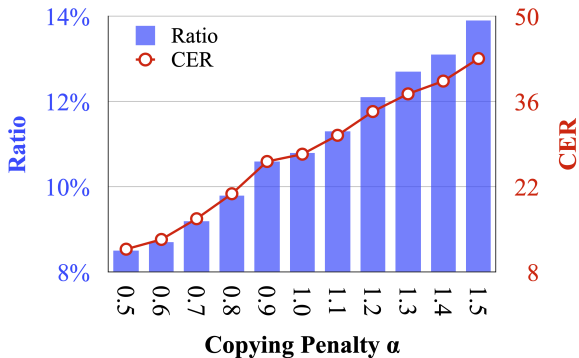
Figure 2: Copying ratios and CER scores by different copying penalties in PRETRAINED. When setting CP smaller than 1 (i.e., penalizing copying), both the copying ratios and CER scores decrease, and vice versa.

tuned CP on the development data and found that setting CP to 0.7 wins the best BLEU score. Therefore, we used this value for decoding test data in the following experiments.

Empirical results also show that the copying penalty is very efficient. When evaluated by a single 32GB V100 GPU card, the inference speed of PRETRAINED is about 612 token/s, and that of the model with CP is 607 token/s. The extra latency of the copying penalty is negligible.

3.2 Main Results

Table 6 lists the overall results of the model performance and copying behavior. By looking at the part of evaluating the all test data of 3,003 sentences, the results first confirm the effectiveness of PRETRAINED that can consistently improve the model performance in terms of BLEU and TER scores. However, the introduction of pre-trained knowledge also brings more copying properties to the model that increases the copying ratio, copying errors, and the number of copying sentences at the same time. Thanks to the introduction of the copying penalty, the model successfully alleviates the copying errors (i.e., reducing the CER score from 27.6 to 16.8), making them be on par with RANDOM, and thus further improve the BLEU and TER scores over the strong PRETRAINED.

To better understand how copying behaviors affect model performance, we split the test data into two subsets: HasCopy and NoCopy. One intuitive assumption is that copying errors would significantly hurt the performance of the NoCopy data since every copying token in the translation is a copying error. The results confirm our assumption that PRETRAINED can only improve limited model

| | Performance | | Copying | | |
	BLEU	TER	Ratio	CER	#S
All (3,003 sentences)					
ORACLE	-	-	8.5%	-	9
RANDOM	28.3	60.7	9.3%	17.4	**20**
PRETRAINED	29.4	59.4	10.8%	27.6	50
+CP	**29.6**	**59.0**	**9.2%**	**16.8**	**20**
HasCopy (1,774 sentences)					
ORACLE	-	-	12.5%	-	9
RANDOM	29.7	59.4	13.4%	13.8	**15**
PRETRAINED	30.9	57.8	14.8%	20.7	32
+CP	**31.0**	**57.5**	**13.0%**	**13.2**	17
NoCopy (1,229 sentences)					
ORACLE	-	-	0%	-	0
RANDOM	25.4	62.9	**1.2%**	-	5
PRETRAINED	26.2	62.7	2.7%	-	18
+CP	**26.7**	**62.0**	**1.2%**	-	**3**

Table 6: Overall results on the WMT14 En-De test set. "Oracle" denotes the statistics on the reference. "+CP" denotes using the proposed copying penalty method for inference. "#S" denotes the number of instances whose overlaps between the source and target exceeding 50%. "HasCopy" denotes evaluating on the sampled test set containing copies between the source and target, while "NoCopy" denotes evaluating on the remained set without any copying. CER score is not applicable in No-Copy as all the copying tokens are copying errors.

performance on the NoCopy data (e.g., improving the TER scores from 62.9 to 62.7), but with the copying penalty, the copying errors less occur in PRETRAINED (i.e., reducing the copying ratio from 2.7 to 1.2 and copying sentences from 18 to 3) and thus, better model performance.

Sentence Fluency The copying penalty improves sentence fluency. In §2.4, we show that the perplexity of PRETRAINED (95.1) is worse than that of RANDOM (60.9). However, after introducing the copying penalty into PRETRAINED, the perplexity gets a significant drop from 95.1 to 62.3, which can be on par with RANDOM. This confirms our hypothesis that more copying behaviors hurt NMT in terms of translation fluency, and controlling copying behaviors can make the model generate fluent outputs.

Word Accuracy The copying penalty enhances the translation of PROPN. As shown in Table 7, the copying penalty improves the translations of proper nouns, reducing the copying ratio from 7.5%

	#Num	Total		PROPN	
		Ratio	CER	Ratio	CER*
All	3,003	10.8%	27.6	7.5%	27.3
+CP		9.2%	16.8	6.3%	15.1
Tgt-Ori	1,500	10.1%	17.5	6.6%	15.6
+CP		9.4%	12.6	6.1%	10.1
Src-Ori	1,503	11.4%	34.4	8.3%	34.8
+CP		9.1%	20.3	6.4%	18.9

Table 7: Copying behaviors of the source original and target original text in PRETRAINED. "#Num" denotes the total number of sentences in each test set. The translation of source original text contains more copying tokens, and CP can reduce the copying ratio.
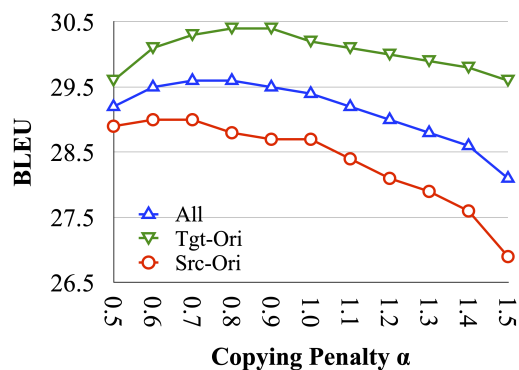

Figure 3: BLEU scores of different copying penalties in PRETRAINED. Penalizing copying (i.e., $\alpha < 1$) brings benefits to the translations of various sources. Translating source original sentences is more sensitive to copying behaviors, leading to a larger score degradation when encouraging copying (i.e., $\alpha > 1$).

to 6.3% and the CER score from 27.3 to 15.1.

To make headway into the translation of proper nouns, we further investigate the translations from various sources that usually show the large difference in the number of proper nouns (Lembersky et al., 2011). Specifically, we investigate the two kinds of sentence pairs in the WMT14 En-De test set: 1) the *source original text* (Src-Ori) that originated in English and was human-translated into German; 2) the *target original text* (Tgt-Ori) was translated in the opposite direction, originating in German with manual translation into English.

Zhang and Toral (2019) conclude that Tgt-Ori is artificially easier to translate, resulted in inflated scores for NMT models. Our results of PRE-TRAINED positively support this conclusion that translating Tgt-Ori makes fewer copying errors and this might be a reason why it can win a better translation performance. However, by looking at the last row, Src-Ori suffers from serious copying errors especially in translating proper nouns, making it harder to translate. Encouragingly, the copying penalty nicely reduces the copying ratios and copying errors in translating both Src-Ori and Tgt-Ori. These results further reveal the importance of controlling copying behaviors in NMT models since translating source original text is the core task of most NMT systems (Graham et al., 2020).

The above results have shown that copying errors worsen the translation of Src-Ori. To support the claim, we further investigate the effects of varying degrees of copying errors in the translations of Src-Ori and Tgt-Ori. Figure 3 shows the change of BLEU scores with different copying penalties. Clearly, the translation of Src-Ori is more sensitive to copying errors and thus the BLEU scores get a sharp degradation when setting the copying penalty

greater than 1, which verifies our claim.

3.3 Out-of-domain Robustness

Improving out-of-domain (OOD) robustness is one of the benefits of pre-training for NLP tasks (Hendrycks et al., 2020; Tu et al., 2020), but the OOD sentences usually contain some low-frequency proper nouns which are hard to translate (Ding et al., 2021). In this part, we take the first step towards understanding how pre-training affects the OOD robustness of NMT models.

Setup We followed Müller et al. (2020) to preprocess all the used data sets.[8] We served the medical domain as the training domain (i.e., using the data from the medical domain for model training and validation), which consists of 1.1M training examples and 2,000 validation examples. The test set of the medical domain contains 1,691 examples, while the test sets of the IT, Koran, law, and subtitle domains are with 2,000 examples respectively. For training RANDOM, we used the Transformer *base* setting with 32K batch size. The model dropout is set to 0.3, while the dropouts for attention and inner-FFN activation are set to 0.2. For PRETRAINED, apart from using 32K batch size, the other hyperparameters follow the training of the WMT14 English-German task. The beam size is set to 5 and the length penalty is set to 1.4. We evaluated the model performance on the OOD test sets from IT, Koran, Law, and Subtitles domains,

---

[8] https://github.com/ZurichNLP/domain-robustness/blob/master/scripts/preprocessing/preprocess_de_en.sh

	InD	OutD				
	Med.	**Avg.**	**IT**	**Kor.**	**Law**	**Sub.**
Existing	61.5	11.7	17.1	1.1	25.3	3.4
+Reg.	60.8	13.1	-	-	-	-
Random	60.5	11.4	20.5	1.1	20.5	3.4
Pretrained	63.1	17.6	29.8	2.4	31.0	7.3
+CP	**63.2**	**18.3**	**31.5**	**2.5**	**31.1**	**7.9**

Table 8: BLEU scores on the OPUS De-En translation task trained on the in-domain medical data. "Existing" and "+Reg." denote the results of baseline and regularization method from Müller et al. (2020). CP can significantly improve the translation performance of the IT domain that needs to copy more tokens from the source.

and the averaged BLEU scores can be seen as the OOD robustness of each NMT model.

Results Table 8 lists the results. Clearly, Pretrained substantially improves the performances of in-domain translation and OOD robustness, increasing the in-domain BLEU scores from 60.5 to 63.1 and the OOD BLEU scores from 11.4 to 17.6 respectively. The copying penalty can further improve the OOD robustness of Pretrained that consistently improves the model performance of each OOD test set. The copying penalty can even remarkably enhance Pretrained in translating the sentences from the IT domain (when setting CP to 1.2). One possible reason is that the IT domain needs to copy more tokens from the source sentence than translating sentences from other domains, thus the copying penalty can play a greater role and bring a significant performance boost. This also verifies the effectiveness of the copying penalty.

4 Related Work

4.1 Pre-Training for NMT

Recently, pre-training has been shown useful for transferring general knowledge to specific downstream tasks, including text classification, question answering and natural language inference (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019). Compared with training from scratch, fine-tuning a pre-trained model on downstream datasets usually pushes state-of-the-art performances, while reducing computational and labeling costs.

Previous studies mainly investigate the effect of pre-training on NMT from two perspectives: 1) *knowledge extraction*, where a fixed pre-trained model is used to encode input sequences into fea-

tures which are then fed into NMT models; and 2) *parameter initialization*, where part/all of the parameters of an NMT model are initialized by a pre-trained model and then training the model on downstream datasets (i.e., parallel corpus).

About knowledge extraction, Yang et al. (2020a) and Zhu et al. (2020) explore enhancing encoder and decoder representations by leveraging pre-trained BERT models (Devlin et al., 2019). In addition, Chen et al. (2020) distill the soft labels from BERT to improve predictions for NMT. These methods are effective but costly because the novel NMT architecture needed to be carefully designed and the computation graph has to store the parameters of both the pre-trained model and NMT model.

About parameter initialization, pre-trained models in different architectures have been studied. For the pre-trained model whose architecture is similar to Transformer encoder (e.g., BERT) or decoder (e.g., GPT (Radford et al., 2018)), the parameters of encoder and decoder can be independently initialized (Conneau and Lample, 2019; Rothe et al., 2020). For the pre-trained model building upon the encoder-decoder architecture (Sutskever et al., 2014), all the model parameters can be directly inherited by NMT, which is easy to use and effective (Song et al., 2019; Lewis et al., 2020; Lin et al., 2020; Yang et al., 2020b).

In general, most previous works focus on designing novel pre-training methods and architectures to boost the model performance of NMT, but the understanding of pre-training for NMT is still limited. This paper improves pre-training for NMT by first understanding its weakness in copying behavior, revealing the importance of further identifying the side-effect from pre-training.

4.2 Copying Behaviors of NMT

It is a common behavior in Seq2Seq models to copy source tokens to the target sentences, especially in monolingual generation tasks. For example, Gu et al. (2016) propose a copying mechanism to explicitly help model learn copying predictions, showing its effectiveness in the tasks of dialogue and summarization.

The copying behaviors also exist in NMT, particularly in languages that share some alphabets (e.g., English and German). Koehn and Knowles (2017) observe that subword-based NMT (Sennrich et al., 2016) outperforms statistical machine translation when translating/copying unknown words.

Knowles and Koehn (2018) find that NMT is able to translate source words in specific contexts via copying (e.g., personal names followed by "Mrs."), and even these are unknown words. However, too many copying signals (i.e., source and target sentences are identical) in training data may lead to one potential threat: NMT models prefer copying source tokens instead of translating them, resulting in performance degradation (Ott et al., 2018a; Khayrallah and Koehn, 2018).

This paper broadens the understanding of copying behaviors in NMT models. We observe that the translation of proper nouns in the source original text contains more copying tokens, which sheds light upon future works.

5 Conclusion and Future Work

We find that NMT models with pre-training are prone to generate more copying tokens. We introduce a copying ratio and a copying error rate to quantitatively analyze copying behaviors in NMT evaluation. In addition, a simple and effective copying penalty is proposed to enhance the copying behaviors during model inference. Experimental results prove the effectiveness of the copying penalty, which can effectively control copying behaviors and improve the overall model performance, especially for the domains (e.g., the IT) where much copying is needed. Extensive analyses reveal that translating proper nouns in source original text generates more copying tokens, providing a direction for the following works on controlling copying behaviors of NMT models.

In the future, we would like to test the effectiveness of the copying penalty on the NMT models with other powerful pre-trained models, and explore more kinds of discrepancies between LM pre-training and NMT training which can be investigated to improve the performance of NMT models. It is also worthwhile to adapt it to other Seq2Seq tasks that need to make a large number of copying predictions, e.g., text summarization and grammar error correction (Liu et al., 2021).

Acknowledgement

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2020. Distilling knowledge learned in BERT for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7893–7905, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liang Ding, Longyue Wang, Xuebo Liu, Derek F. Wong, Dacheng Tao, and Zhaopeng Tu. 2021. Rejuvenating low-frequency words: Making the most of parallel data in non-autoregressive translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical power and translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 72–81, Online. Association for Computational Linguistics.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, Online. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Rebecca Knowles and Philipp Koehn. 2018. Context and copying in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3034–3041, Brussels, Belgium. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Gennadi Lembersky, Noam Ordan, and Shuly Wintner. 2011. Language models for machine translation: Original vs. translated texts. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 363–374, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663, Online. Association for Computational Linguistics.

Xuebo Liu, Longyue Wang, Derek F. Wong, Liang Ding, Lidia S. Chao, and Zhaopeng Tu. 2021. Understanding and improving encoder layer fusion in sequence-to-sequence learning. In *International Conference on Learning Representations*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mathias Müller, Annette Rios, and Rico Sennrich. 2020. Domain robustness in neural machine translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018a. Analyzing uncertainty in neural machine translation. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3953–3962. PMLR.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018b. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Citeseer.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. In *Advances in Neural Information Processing Systems*, volume 33, pages 2207–2219. Curran Associates, Inc.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Weinan Zhang, Yong Yu, and Lei Li. 2020a. Towards making the most of bert in neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9378–9385.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020b. CSP:code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2624–2636, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 73–81, Florence, Italy. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating BERT into neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.