Language Models are Good Translators

Shuo Wang¹ Zhaopeng Tu² Zhixing Tan¹ Wenxuan Wang² Maosong Sun¹,³ Yang Liu¹,³,⁴
¹Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Center, Tsinghua University
³Beijing Academy of Artificial Intelligence ⁴Institute for AIR, Tsinghua University
¹wangshuo.thu@gamil.com
¹{zxtan, sms, liuyang2011}@tsinghua.edu.cn
²Tencent AI Lab
²{zptu, jwxwang}@tencent.com

Abstract

Recent years have witnessed the rapid advance in neural machine translation (NMT), the core of which lies in the encoder-decoder architecture. Inspired by the recent progress of large-scale pre-trained language models on machine translation in a limited scenario, we firstly demonstrate that a single language model (LM4MT) can achieve comparable performance with strong encoder-decoder NMT models on standard machine translation benchmarks, using the same training data and similar amount of model parameters. LM4MT can also easily utilize source-side texts as additional supervision. Though modeling the source- and target-language texts with the same mechanism, LM4MT can provide unified representations for both source and target sentences, which can better transfer knowledge across languages. Extensive experiments on pivot-based and zero-shot translation tasks show that LM4MT can outperform the encoder-decoder NMT model by a large margin.

1 Introduction

Recent years have witnessed the success of the neural machine translation (NMT) models [SVL14, BCB15, GAG+17, VSP+17], which translate texts in the source language into the target language with neural networks. A number of studies have directed their attention to designing more advanced NMT models. TRANSFORMER [VSP+17] is the most widely-used NMT model, which uses attention-based neural networks for both the encoder and the decoder. [WFB+19] propose an efficient NMT model using lightweight and dynamic convolutions. [SLL19] apply neural architecture search and find a better alternative to vanilla TRANSFORMER model. Although the core mechanism has evolved from RNN to self-attention and then other alternatives, the encoder-decoder architecture is still the dominating framework for NMT models.

Previous studies have shown that the boundary between encoder and decoder, in terms of the localization of the representation in the continuous space, is blurry for multilingual NMT [KBCF19]. [HTX+18] demonstrate that the standard NMT model benefits from weakening the boundary between encoder and decoder by sharing parameters of the two components. More recently, GPT-3 [BMR+20], which is a single language model (LM) pre-trained on huge amount of multilingual data, firstly shows some promising results of LM on machine translation when given in-context translation examples as prefixes. GPT-3 has several key limitations that prevent it from serving as the practical NMT model, including (1) GPT-3 fails for machine translation without in-context prefixes; (2) both the amount of training data and model parameters for GPT-3 are several orders of magnitude larger than those of standard NMT models, which is prohibitively expensive for many researchers and developers. Nevertheless, the surprising results still trigger us to think a research question: *can we really accomplish the machine translation task with a single language model?*

To answer this question, we explore the ability of language models for machine translation (i.e., LM4MT) with only limited parallel data, which is often used to train the standard encoder-decoder NMT models. Surprisingly, we find through experiments that the vanilla language model only marginally underperforms the encoder-decoder NMT models with comparable model sizes. Benefiting from the characteristic of LM4MT to generate both the source and target sentences in the same manner, we introduce an auto-encoding loss with a decaying schedule to help LM4MT better learn from source-language sentences. Experimental results show that the proposed LM4MT achieves comparable performance with or even better performance than its encoder-decoder counterpart on six machine translation benchmarks. For instance, on the benchmarking WMT14 English⇒German and English⇒French translation tasks, LM4MT achieves 29.3 BLEU and 42.9 BLEU, respectively. The additional source-side supervision can improve the model performance in two other aspects: (1) higher translation quality for source-original sentences, which are usually more complex and difficult to translate [ZT19]; and (2) better model robustness against missing word perturbations. These results reveal that the source-side supervision can help LM4MT better understand source texts.

Another appealing advantage of LM4MT is the unified representation for both source and target sentences, which might better transfer knowledge across languages. We empirically validate our hypotheses in two scenarios: (1) *pivot-based translation* where a pivot language serves as the transit station to transfer the knowledge from the source language to the target language [KPP⁺19]; and (2) *zero-shot translation*¹ for multilingual NMT model, which has a stricter requirement on the model representations to implicitly bridge between the language pairs unseen during training [JSL⁺17]. Experimental results show that LM4MT can outperform the encoder-decoder NMT model in all cases. We find the source-side auto-encoding loss is essential for LM4MT to perform zero-shot translation.

To sum up, the main contributions of this paper are listed as follows:

- We firstly demonstrate that a single language model can achieve comparable translation performance with the encoder-decoder NMT model of the same model size.
- Benefiting from the additional source-side supervision and unified representations across different languages, the proposed LM4MT can outperform the encoder-decoder NMT model in both pivot-based and zero-shot translation scenarios.

2 Related Work

Functionalities and Importance of Encoder In recent years, there has been a growing interest in understanding the functionalities of the encoder in encoder-decoder NMT models. For example, [TSN19] simplify the TRANSFORMER model to an encoder-free model and find that the encoder is crucial for NMT models to achieve good results. [WLX⁺19, WTSL20] show that enlarging the capacity of the encoder is more effective for improving translation performance than enlarging the decoder. However, these conclusions might only hold for the encoder-decoder architecture. In this paper, we rethink the importance of encoder in a new architecture and reveal that a single decoder can accomplish the translation task well.

Shared Encoder and Decoder [KBCF19] find that the boundary between encoder and decoder is blurry for multilingual NMT. In standard translation, there are some works that weaken the boundary between the encoder and decoder. For example, [HTX⁺18] share the parameters of the encoder and decoder, which coordinates the learning of hidden representations of the two components. However, their model still encodes source-language texts in the same way as vanilla encoder-decoder NMT models, ignoring the source-side supervision. The source and target sentences are still consumed in separate mechanisms. We take one step further and simplify the architecture into a simple decoder, where we can easily utilize the source-side supervision to help better understand source texts.

Language Model for Machine Translation Recently, the pre-trained language model GPT-3 [BMR⁺20] has shown encouraging results on machine translation tasks. GPT-3 is pre-trained on massive data with a huge amount of parameters, and requires in-context translation examples to achieve good translation performance. Our work, on the other hand, is the first attempt to systematically compare the ability of encoder-decoder models and LM (i.e., single decoder) across different translation scenarios, using the same training data and similar model sizes.

¹Zero-shot translation denotes translating between language pairs that do not exist in the training data.

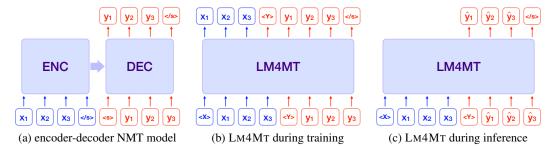


Figure 1: Illustration of LM4MT. For comparison, we also plot the encoder-decoder model. During training, we feed LM4MT with the concatenation of the source- and target-language sentences, which are explicitly separated by special language tags. At the inference time, the source-language text, together with the target-language tag, is used as the prefix for LM4MT to generate the translation.

3 Approach

3.1 Preliminaries

Language Model Given a monolingual sentence $\mathbf{y} = \{y_1, y_2, \dots, y_T\}$, the goal of language modeling is to estimate the joint probability $P(\mathbf{y})$, which is usually auto-regressively factorized as

$$P(\mathbf{y}) = \prod_{t=1}^{T} P(y_t | \mathbf{y}_{< t}), \tag{1}$$

where $\mathbf{y}_{< t} = \{y_1, y_2, \dots, y_{t-1}\}$ is the prefix before y_t . With this factorization, the problem is simplified to estimating each conditional factor. Standard neural language models [DYY⁺19] encode the context $\mathbf{y}_{< t}$ into a continuous vector, which is then multiplied by the word embedding matrix to obtain the logits. The logits are then used to compute the probability distribution over the next word through the Softmax function.

Encoder-Decoder NMT Model Given a source-language sentence $\mathbf{x} = \{x_1, x_2, \dots, x_S\}$, NMT models learn to predict the conditional probability of the corresponding target-language sentence \mathbf{y} :

$$P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} P(y_t|\mathbf{x}, \mathbf{y}_{< t}).$$
 (2)

Most previous NMT models use the encoder-decoder framework [BCB15, VSP $^+$ 17, LGG $^+$ 20]. The encoder is a feature extractor, which maps the source-language sentence $\mathbf x$ into a sequence of continuous representations $\mathbf r = \{\mathbf r_1, \mathbf r_2, \dots, \mathbf r_S\}$. The decoder is a conditional language model, which estimates the probability $P(\mathbf y|\mathbf r)$. In the widely used TRANSFORMER [VSP $^+$ 17] model, both the encoder and the decoder use attention networks, which are shown effective to learn contextualized representations [DCLT19]. To achieve good translation performance, NMT models are expected to not only extract effective source-language representations $\mathbf r$, but also be capable of generating a fluent target-language sentence that can recover the information conveyed in the source sentence.

3.2 Language Model for Machine Translation

Recently, several works find that huge language models pre-trained on large-scale data set can achieve good results on a number of natural language understanding and generation tasks [RWC⁺19, BMR⁺20, DQL⁺21, LZD⁺21], indicating the potential of language models to serve as a good feature extractor. However, for machine translation, it is still not well investigated whether a language model along can act as a source-language feature extractor and a target-language generator at the same time, especially without huge model size and large amount of training data. In this work, we aim to investigate the capability of language models to perform machine translation, which is a task that requires the abilities of both language understanding and generation.

From Equation (1) and (2), we find that objectives of language modeling and machine translation are quite similar, since the source-language sentence x can be seen as a special type of prefix. Inspired by this observation, we propose the language model for machine translation (i.e., LM4MT) that are trained to estimate the joint probability of the two sentence x and y:

$$P(\mathbf{x}, \mathbf{y}) = \prod_{s=1}^{S} P(x_s | \mathbf{x}_{< s}) \prod_{t=1}^{T} P(y_t | \mathbf{x}, \mathbf{y}_{< t}).$$
(3)

Specifically, we concatenate x and y into a sentence pair, and then use LM4MT to estimate the joint probability of such a sentence pair as if it is one sentence. To help the model better identify the boundary of sentences from different languages, we add a special language tag before each sentence. Figure 1b depicts an example for the training phrase of LM4MT. Experiments in Section 5.1 show that the added language tags are effective to improve the translation performance of LM4MT.

Training Just as standard language models, the training objective of LM4MT is to minimize the negative log-likelihood of the probability $P(\mathbf{x}, \mathbf{y})$:

$$-\log P(\mathbf{x}, \mathbf{y}) = \mathcal{L}^{AE} + \mathcal{L}^{MT} = -\sum_{s=1}^{S} \log P(x_s | \mathbf{x}_{< s}) - \sum_{t=1}^{T} P(y_t | \mathbf{x}, \mathbf{y}_{< t}), \tag{4}$$

where $\mathcal{L}^{\text{AE}} = -\sum_{s=1}^{S} \log P(x_s|\mathbf{x}_{< s})$, which is the auto-encoding loss, reflecting the ability of the model to reconstruct the source-language sentence \mathbf{x} . $\mathcal{L}^{\text{MT}} = -\sum_{t=1}^{T} P(y_t|\mathbf{x},\mathbf{y}_{< t})$, which is the machine translation loss that has been widely-used for NMT models. Compared to encoder-decoder NMT models, LM4MT is trained with an additional source-side auto-encoding loss \mathcal{L}^{AE} .

Intuitively, the explicit supervision induced by \mathcal{L}^{AE} may help LM4MT better understand the source-language sentences. Moreover, using \mathcal{L}^{AE} makes the modeling mechanisms of the source and target texts more similar, which may reduce the representation gap between source and target sentences. We find through experiments (Section 5.1) that simply adding \mathcal{L}^{AE} is not a good practice for machine translation, which might be caused by that \mathcal{L}^{AE} prevents the model to further minimize \mathcal{L}^{MT} at the end of the training. Inspired by the training strategy of unsupervised NMT [LOC⁺18, LC19], we multiply \mathcal{L}^{AE} with a decaying factor λ_d . Therefore, the training objective of LM4MT is

$$\mathcal{L}^{\text{LM4MT}} = \lambda_d \mathcal{L}^{\text{AE}} + \mathcal{L}^{\text{MT}}.$$
 (5)

As shown in Figure 2, we use a piece-wise linear decaying schedule for the factor λ_d . Through this strategy, we expect LM4MT to better learn from the source sentences at early stages and then focus more on the translation loss at the end of the training.

Inference Different from the training phrase during which LM4MT estimates the probability for both the source- and target-language sentences, we only let LM4MT predict the target-language tokens at the inference time. As shown in Figure 1c, LM4MT takes in source-language tokens and encodes them into hidden states in parallel. Then the proposed model generates each target-language token step by step.

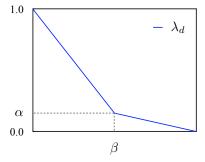


Figure 2: Decaying schedule for λ_d in Eq. (5). α and β are hyper-parameters.

4 Experimental Setup

Data To make a thorough comparison between the widely-used TRANSFORMER [VSP⁺17] model and our proposed LM4MT, we conducted experiments on datasets with different sizes: WMT16² English-Romanian (En-Ro), WMT14³ English-German (En-De) and English-French (En-Fr), which

²http://statmt.org/wmt16/translation-task.html

http://statmt.org/wmt14/translation-task.html

consist of 0.6M, 4.5M and 35.8M sentence pairs respectively. In En-Ro, we used news-dev2016 as the validation set and news-test2016 as the test set. In En-De and En-Fr, we used news-test2013 and news-test2014 as the validation and test sets, respectively. We applied BPE [SHB16] with 32K merge operations for En-De data, and with 40K merge operations for En-Ro and En-Fr data.

Model We used the encoder-decoder based TRANSFORMER model [VSP⁺17] as our baseline. We used models of two sizes, namely the TRANSFORMER-Base and TRANSFORMER-Big, both of which consist of a 6-layer encoder and a 6-layer decoder. The hidden sizes of TRANSFORMER-Base and TRANSFORMER-Big are 512 and 1024, respectively. We also list the results of some recent representative architectures for comparison, including TRANSFORMER with relative position representations [SUV18], scaling TRANSFORMER [OEGA18], layer-wise coordinated TRANSFORMER [HTX⁺18], dynamic convolutions [WFB⁺19], and evolved TRANSFORMER [SLL19].

The proposed LM4MT model consists of only a self-attention decoder. There are mainly three differences between the LM4MT model and the decoder of the vanilla TRANSFORMER [VSP+17]. Firstly, each LM4MT layer has only one type of attention network while the decoder in [VSP+17] has both self-attention and cross-attention networks. Secondly, we use pre-norm residual unit [WLX+19] in order to train deep language models. Thirdly, we follow [RWC+19] to use GELU activations [HG16] in LM4MT, which has been shown to be effective in language model training. Similar to TRANSFORMER, we also conduct experiments with base and big settings for LM4MT. The hidden size of LM4MT-Base is 512 and that of LM4MT-Big is set to 1024.

Training Details For En-Ro, we only trained base models since the training corpus was too small (0.6M sentence pairs) to learn big models. The dropout rate was set to 0.3 for both TRANSFORMER and LM4MT. We set weight decay to 1e−4 to overcome over-fitting. For En-De, the dropout rates were set to 0.1 for base models and 0.3 for big models. For En-Fr, the dropout rate was set to 0.1 for both base and big models. For all the pivot-based and multilingual translation models, the dropout rate was set to 0.1. For all the three language pairs, we used Adam [KB15] to optimize the model parameters, with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. We trained base models on mini-batches that contain approximately 32K target-language tokens for 150K steps. For big models, we followed [OEGA18] to use *larger batches*, which contain approximately 460K tokens, to further boost the performance of big models. When using the large batch size, we followed [WFB⁺19] to use the cosine learning rate schedule, where the learning rate was warmed up linearly to 1e-3 in the first 10K steps and then decayed to 1e-7 following a cosine rate within a single cycle. Big models were trained for 30K steps on large batches. We obtained the final model by averaging the last 5 checkpoints, which were saved at 1000-update and 500-update intervals for base and big models, respectively. As for the decaying schedule, we set α =0.1, β =37.5K for base models and α =0.1, β =22.5K for big models. All models were implemented on the top of fairseq toolkit. We conducted all the experiments on 8 Nvidia Telsa V100 32GB GPUs.

5 Experimental Results

5.1 Ablation Study

Effect of Language Tag In the training phase, we simply feed LM4MT with the concatenation of the source- and target-language sentences, which may make it difficult to identify the start position of the target-language sentence. In response to this problem, we use language tags to explicitly distinguish sentences of different languages, as shown in Figure 1b. Table 1 shows the impact of language tags, indicating the importance of explicit indicators of languages for LM4MT. In the following experiments we use language tags by default.

Table 1: Effect of language tag ("Tag") for 19-layer LM4MT with \mathcal{L}^{MT} loss.

Tag	BLEU
-	26.7
×	25.8
✓	26.3
	-

Effect of Layer Number Since TRANSFORMER has an additional encoder compared to LM4MT, we deepen the LM4MT to assimilate the parameter count. Table 2 shows the results of LM4MT

⁴https://github.com/pytorch/fairseg

with different depths. The hidden size of all models are 512. Surprisingly, the 6-layer LM4MT performs only 1.8 BLEU lower than the encoder-decoder baseline, although it has much fewer parameters (56.9M vs. 98.8M). Enlarging the LM4MT by adding layers is effective to improve the translation performance. When using comparable amount of parameters, the performance of LM4MT is 0.4 BLEU point lower than the Transformer baseline (26.3 vs. 26.7 for L19 LM4MT). In the following experiments, we use L19 as the default architecture for LM4MT.

Effect of Auto-Encoding Loss Another key difference between LM4MT and TRANS-FORMER is that we introduce an additional auto-encoding loss \mathcal{L}^{AE} (Equation (5)) for LM4MT to better understand the source sentence. To empirically validate the effect of the source-side \mathcal{L}^{AE} , we follow [YG19] to separately report the BLEU score on the source-original⁵ sentences on WMT14 En \Rightarrow De test set. Intuitively, the translation of source-original sentences requires a better understanding of source sentences to handle the relatively more complex sentence structures and more diverse contents [WTT⁺21].

Effect of Auto-Encoding Loss Another key difference between LM4MT and TRANS- \mathcal{L}^{MT} as the training loss.

Model	Arch.	Param.	BLEU
TRANSFORMER	L6-L6	98.8M	26.7
	L6	56.9M	24.9
LM4MT	L12	75.8M	25.6
	L18	94.7M	26.2
	L19	97.8M	26.3

As listed in Table 3, directly adding auto-encoding loss (" $\mathcal{L}^{AE}+\mathcal{L}^{MT}$ ") fails to improve translation performance. We plot the learning curves of validation perplexities for LM4MT trained with different objectives. We find that directly adding the auto-encoding loss inversely increase the validation perplexity in late training stages. Using the decaying schedule to adjust the weight of \mathcal{L}^{AE} can effectively improve translation performance and help training convergence. The performance improvement is mainly from better translation of source-original sentences ("o"), which confirms our claim that the auto-encoding loss can help to better understand the source sentences. In the following experiments, we train LM4MT models with decaying auto-encoding loss by default.

Table 3: Effect of auto-encoding loss \mathcal{L}^{AE} . λ_d denotes a dynamic weight decaying from 1 to 0 during training. "o" denotes source-original sentences in the test set, while "n-o" denotes the target-original sentences.

Model	Loss	Valid		Test	
1110401	2005	vuitu	all	0	n-o
TRANS.	\mathcal{L}^{MT}	26.7	27.8	27.4	28.1
	$\mathcal{L}^{ ext{MT}}$	26.3	27.3	26.5	27.6
LM4MT	$\mathcal{L}^{ ext{AE}} + \mathcal{L}^{ ext{MT}}$	26.0	26.9	26.6	27.0
	$\lambda_d \mathcal{L}^{\mathrm{AE}} + \mathcal{L}^{\mathrm{MT}}$	26.8	27.8	27.6	27.9

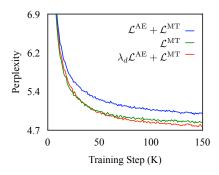


Figure 3: Learning curves of validation perplexity on WMT14 En⇒De.

5.2 Standard Translation

Translation Performance Table 4 lists the results on several WMT benchmarks with different data scales. LM4MT performs very competitively on all the six translation directions, demonstrating that our approach is applicable to low-, medium-, and high-resource language pairs. For instance, on the widely-used WMT14 En⇒De benchmark, LM4MT-BIG performs as well as our implemented TRANSFORMER-BIG, which is also comparable with several strong baselines. Note that our work is complementary to many previous works since LM4MT can use more advanced components, such as dynamic convolution [WFB+19] and evolved TRANSFORMER cell [SLL19]. Closely related to our work, LAYER-WISE COOR. [HTX+18] shares the parameters of the encoder and decoder, but they still use separate mechanisms to model the source- and target-language texts and ignore the source-side supervision. We take one step further and simplify the architecture into a simple decoder.

⁵Source-original texts are written by source-language native speakers, which are found to be more difficult to translate than human-translated texts [ZT19]. Since the En \Rightarrow De validation set contains six different original languages, we report the source-original results on the En \Rightarrow De test set.

Table 4: Translation performance on WMT16 En⇔Ro (0.6M sentence pairs), WMT14 En⇔De (4.6M sentence pairs) and WMT14 En⇔Fr (35.8M sentence pairs) test sets.

Model	WMT16	En⇔Ro	WMT14	En⇔De	WMT14	En⇔Fr
1720401	En⇒Ro	Ro ⇒ En	En⇒De	De⇒En	En⇒Fr	Fr⇒En
	Exist	ing Work				
TRANSFORMER-BASE [VSP+17]	-	-	27.3	-	38.1	-
TRANSFORMER-BIG [VSP+17]	-	-	28.4	-	41.0	-
RELATIVE POSITION [SUV18]	-	-	29.2	-	41.5	-
SCALE-TRANS. [OEGA18]	-	-	29.3	-	43.2	-
LAYER-WISE COOR. [HTX ⁺ 18]	34.4	-	29.0	-	-	-
DYNAMIC CONV. [WFB+19]	-	-	29.7	-	43.2	-
EVOLVED-TRANS. [SLL19]	-	-	29.5	-	41.3	-
	Our Im	plementat	ion			
TRANSFORMER-BASE	34.4	33.9	27.8	31.3	41.2	36.5
Lm4MT-Base	34.2	34.1	27.8	31.5	41.2	37.1
TRANSFORMER-BIG	-	-	29.3	33.0	43.0	39.0
LM4MT-BIG	-	-	29.3	33.2	42.9	39.5

Model Robustness To investigate the robustness of LM4MT to noisy inputs, we followed [SB18, ZZH $^+$ 20] to construct noisy test examples by omitting some words in the source-language sentences. Table 5 lists the results on WMT14 En \Leftrightarrow De test sets. It is evident that LM4MT is more robust than TRANSFORMER to missing word noise, and the performance improvement generally goes up with the increase of missing ratio. This may be attributed to the auto-encoding loss \mathcal{L}^{AE} , which induces source-side reconstruction supervision that can help LM4MT better "denoise" noisy inputs.

Table 5: Translation performance with missing words. All models use the base setting.

Missing Ratio	0%	30%	50%
WMT	<i>14 En⇒</i>	De	
TRANSFORMER LM4MT	27.8 27.8	19.1 20.2	11.5 13.6
WMT	14 De⇒	-En	
TRANSFORMER LM4MT	31.3 31.5	19.9 20.6	11.8 14.2

5.3 Pivot-Based Translation

LM4MT uses a unified decoder to represent the source- and target-language sentences, both of which are learned by causal self-attention networks. Accordingly, the representation gap between the source- and target-language sentences are much smaller in LM4MT than that in encoder-decoder models. We believe that the shared representation can help better transfer the knowledge between source- and target-language texts. We verify the research hypothesis in the pivot-based translation scenario [KPP+19], where the pivot language severs as the intermediate output to transfer the knowledge from the source language to the target.

Formally, we have parallel corpora $\mathcal{B}(X,Y)=\{\langle \mathbf{x}_n,\mathbf{y}_n\rangle\}_{n=1}^N$ and $\mathcal{B}(Y,Z)=\{\langle \mathbf{y}_m,\mathbf{z}_m\rangle\}_{m=1}^M$, and we aim to translate sentences of language X into language Z. Assume that parallel corpus $\mathcal{B}(X,Z)$ is not available. To achieve this goal, we train NMT models on the mixture of $\mathcal{B}(X,Y)$ and $\mathcal{B}(Y,Z)$. Following [JSL+17], we pend the target-language tag to the source sentence to indicate the translation direction for Transformer. For LM4MT, we use the tagging scheme illustrated in Figure 1b. We also train Transformer models using the same tagging scheme as LM4MT and find that only using the target-language tag works better for Transformer. At the inference time, we firstly translate the test data in language X into the pivot language Y, which is then translated to language Z. Intuitively, pivot-based translation tasks can benefit from LM4MT model that learns a shared representation across different languages.

In our experiments, we use English as the pivot language and aim to translate between De and Fr. We mix the WMT14 En-De and En-Fr corpora and the En-De corpus is upsampled to the same size with that of the En-Fr corpus. We use news-test2020 De⇔Fr to evaluate the translation

Table 6: Pivot-based (English as the pivot language) translation performance measured by BLEU score. Results are reported on WMT14 test sets for En⇔De and En⇔Fr, and WMT20 test sets for De⇔Fr. NMT model for each direction is trained on the mxiture of WMT14 En-De and WMT14 En-Fr training corpora. Improvements over the TRANSFORMER baseline are highlighted in red cells while deteriorations are represented in green cells. Deeper color indicates larger performance gap.

Model	D	e⇒En⇒F	'r	F	'r⇒En⇒D	e
1120001	De ⇒ En	En⇒Fr	De⇒Fr	Fr⇒En	En⇒De	Fr⇒De
TRANSFORMER-BASE	30.8	39.5	25.7	35.2	26.8	21.4
LM4MT-BASE	30.9	39.5	26.6	36.1	26.5	22.8
Transformer-Big	32.2	42.3	26.5	37.7	29.1	24.2
Lm4Mt-Big	32.5	41.9	27.7	38.1	29.0	25.1

Table 7: Translation performance in the multilingual setting. "Zero-Shot" means translating between unseen language pairs while "Multilingual" means translating between seen language pairs.

Model		Multilingual			Zero-Shot	
1110401	De⇒En	En⇒Fr	Fr⇒En	E n⇒ D e	De⇒Fr	Fr⇒De
TRANSFORMER-BASE	30.4	38.5	34.8	25.8	21.9	13.8
LM4MT-BASE	30.4	38.4	34.8	25.9	24.9	19.5
TRANSFORMER-BIG LM4MT-BIG	33.1	41.9	38.0	29.2	23.8	16.7
	33.9	41.1	37.8	28.6	29.5	24.9

performance. In each direction, all the models are trained only for the involved two intermediate directions. For instance, De⇒Fr models are trained on the mixture of De⇒En and En⇒Fr data. Table 6 shows the results of both base and big models. For pivot-based translation (i.e., De⇔Fr), LM4MT consistently outperforms the TRANSFORMER baseline by a large margin. To dispel the doubt that the superior performance of LM4MT might come from improvements accumulated in the two individual directions, we compute the performance gap between the two models in each direction, which is highlighted by color. The improvements in pivot-based translation directions are consistently larger than the two-step accumulated improvements, reconfirming the strength of LM4MT to exploit the shared knowledge between the source and target sides of parallel training data.

5.4 Zero-Shot Translation

Previous studies have shown that a single multilingual NMT model can enable zero-shot translation – translating between language pairs on which the NMT model has never been trained [JSL $^+$ 17, GWCL19]. Compared with pivot-based translation in Section 5.3, zero-shot translation requires no pivot language as the intermediate output, thus requires better model representations to implicitly bridge between zero-shot language pairs. We followed [JSL $^+$ 17] to conduct experiments with the multilingual setting. Specifically, we mix the bidirectional WMT14 En \Leftrightarrow De and En \Leftrightarrow Fr corpora to train a single multilingual NMT model, which is used to translate the zero-shot language pair of the WMT20 De \Leftrightarrow Fr test sets. Similarly to pivot-based experiments, we upsampled En \Leftrightarrow De corpora to the same size with that of the En \Leftrightarrow Fr corpora.

Table 7 lists the results. In zero-shot translation directions, TRANSFORMER performs much worse than LM4MT. These results demonstrate the superiority of LM4MT on zero-shot translation. By manually checking the generated translations, we found that the failed models suffer from the *off-target translation issue* (i.e., translating into a wrong target language), which is the major source of the inferior zero-shot performance [ZWTS20]. We follow [ZWTS20] to employ the langdetect library⁶ to detect the language of model outputs, and measure the translation-language accuracy for zero-shot cases. Table 8a lists the results, which provide empirical support for our findings. This is potentially caused by that TRANSFORMER tends to memorize translation directions seen in the

⁶https://github.com/Mimino666/langdetect

Table 8: Analyses of off-target translation issue.

(a) Translation-language accuracy for zero-shot.

(b) BLEU on sentences with the correct language.

Model	Zero	-Shot
1110401	De⇒Fr	Fr⇒De
TRANSFORMER-BASE LM4MT-BASE	88.3% 97.9 %	79.7% 97.6%
TRANSFORMER-BIG LM4MT-BIG	87.8% 98.8 %	76.5% 98.5 %

Model	Zero	-Shot
1/10401	De⇒Fr	Fr⇒De
TRANSFORMER-BASE	24.4	18.1
LM4MT-BASE	25.5	19.9
TRANSFORMER-BIG	26.1	21.8
LM4MT-BIG	29.4	25.7

Table 9: Effect of auto-encoding loss on zero-shot translation.

Model	Loss		Multil	Zero-Shot			
1120401	2000	De⇒En	En⇒Fr	Fr⇒En	En⇒De	De⇒Fr	Fr⇒De
TRANSBASE	$\mathcal{L}^{ ext{MT}}$	30.4	38.5	34.8	25.8	21.9	13.8
	$ \mathcal{L}^{ ext{MT}}$	29.5	37.9	34.2	25.4	-7.9	8.6
LM4MT-BASE	$\mathcal{L}^{ ext{AE}} + \mathcal{L}^{ ext{MT}}$	29.3	37.5	33.7	24.7	23.8	18.6
	$\lambda_d \mathcal{L}^{ ext{AE}} + \mathcal{L}^{ ext{MT}}$	30.4	38.4	34.8	25.9	24.9	19.5
TRANSBIG	\mathcal{L}^{MT}	33.1	41.9	38.0	29.2	23.8	16.7
	\mathcal{L}^{-}	33.5	40.9	37.0	28.6	19.3	21.8
Lm4MT-Big	$\mathcal{L}^{ ext{AE}} + \mathcal{L}^{ ext{MT}}$	33.0	40.5	37.1	27.6	28.8	23.3
	$\lambda_d \mathcal{L}^{ ext{AE}} + \mathcal{L}^{ ext{MT}}$	33.9	41.1	37.8	28.6	29.5	24.9

training data while LM4MT can better generalize into unseen directions. More detailed results on off-target mistakes can be found in Appendix (Tables 10 and 11).

To rule out the effect of the target language, we also evaluate the model performance on test examples that can be translated into the correct language by both TRANSFORMER and LM4MT. The results are shown in Table 8b, which demonstrate that LM4MT performs better than TRANSFORMER even when both of them can output the correct language in zero-shot cases. We attribute the improvement to the reason that LM4MT can map sentences that come from various languages into a more unified representation space, thus can better understand the input sentences.

To further investigate why LM4MT shows a strong performance in zero-shot translation, we train LM4MT using different loss functions. Table 9 shows the results, indicating that the source-language supervision \mathcal{L}^{AE} is crucial for LM4MT to achieve good performance. Training with \mathcal{L}^{AE} , LM4MT is optimized by gradients induced from both the source- and target-side tokens, while encoder-decoder NMT models are trained only with the target-side loss, which may increase the representation gap between the source and target texts. Moreover, \mathcal{L}^{AE} enables better understanding of source sentences, which may prevent LM4MT from capturing spurious correlations between input sentences and language tags, which has been found to be harmful to zero-shot translation [GWCL19].

6 Conclusion

We propose a novel LM4MT model to perform machine translation using only a single language model. Although LM4MT is more simplified than the standard encoder-decoder NMT model, it can achieve competitive results with several strong encoder-decoder NMT baselines on standard machine translation tasks. LM4MT shows superior performance in both pivot-based and zero-shot translation scenarios by better transferring knowledge across languages with a unified representation. One potential limitation of this work is that we only conduct experiments on parallel data. We believe that using additional monolingual data can further augment the performance of LM4MT, which can naturally consume monolingual texts without data augmentation methods. Another interesting direction is to investigate the effect of LM4MT on unsupervised NMT, which uses unified representations to bridge between language pairs without training signals from parallel data.

References

- [BCB15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.
- [BMR+20] T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, T. Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In NeurIPS, 2020.
- [DCLT19] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.
- [DQL⁺21] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. All nlp tasks are generation tasks: A general pretraining framework. *ArXiv*, abs/2103.10360, 2021.
- [DYY⁺19] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhut-dinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, 2019.
- [GAG⁺17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. In *ICML*, 2017.
- [GWCL19] Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. Improved zero-shot neural machine translation via ignoring spurious correlations. In *ACL*, 2019.
 - [HG16] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv*, abs/1606.08415, 2016.
- [HTX⁺18] Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Layerwise coordination between encoder and decoder for neural machine translation. In *NeurIPS*, 2018.
- [JSL⁺17] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 2017.
 - [KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In ICLR, 2015.
- [KBCF19] Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. Investigating multilingual NMT representations at scale. In *EMNLP*, 2019.
- [KPP+19] Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. Pivot-based transfer learning for neural machine translation between non-English languages. In EMNLP, 2019.
 - [LC19] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. In NeurIPS, 2019.
- [LGG+20] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. TACL, 8:726–742, 2020.
- [LOC⁺18] Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *EMNLP*, 2018.
- [LZD⁺21] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *ArXiv*, abs/2103.10385, 2021.

- [OEGA18] Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *WMT*, 2018.
- [RWC⁺19] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
 - [SB18] Harshil Shah and David Barber. Generative neural machine translation. In *NeurIPS*, 2018.
 - [SHB16] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, 2016.
 - [SLL19] David R. So, Chen Liang, and Quoc V. Le. The evolved transformer. In ICML, 2019.
 - [SUV18] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL*, 2018.
 - [SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014.
 - [TSN19] Gongbo Tang, Rico Sennrich, and Joakim Nivre. Understanding neural machine translation by simplification: The case of encoder-free models. In *RANLP 2019*, 2019.
- [VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [WFB⁺19] Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *ICLR*, 2019.
- [WLX⁺19] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In *ACL*, 2019.
- [WTSL20] Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. On the inference calibration of neural machine translation. In *ACL*, 2020.
- [WTT⁺21] Shuo Wang, Zhaopeng Tu, Zhixing Tan, Shuming Shi, Maosong Sun, and Yang Liu. On the language coverage bias for neural machine translation. In *Findings of ACL*, 2021.
 - [YG19] Philipp Koehn Yvette Graham, Barry Haddow. Translationese in machine translation evaluation. In *Arxiv*, 2019.
 - [ZT19] Mike Zhang and Antonio Toral. The effect of translationese in machine translation test sets. In *WMT*, 2019.
- [ZWTS20] Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL*, 2020.
- [ZZH⁺20] Zaixiang Zheng, Hao Zhou, Shujian Huang, Lei Li, Xin-Yu Dai, and Jiajun Chen. Mirror-generative neural machine translation. In *ICLR*, 2020.

A Appendix

Table 10: Target-language ratios for zero-shot cases. The expected language is highlighted in green.

Model	Model De⇒Fr			Fr⇒De		
Wiodei	De	En	Fr	Fr	En	De
TRANSFORMER-BASE	0.6%	9.8%	88.3%	4.0%	15.2%	79.7%
LM4MT-BASE	0.3%	1.4%	97.9 %	0.9%	0.1%	97.6 %
TRANSFORMER-BIG LM4MT-BIG	2.8%	8.6%	87.8%	12.7%	9.3%	76.5%
	0.1%	0.6%	98.8 %	0.2%	1.0%	98.5 %

Table 11: Example for zero-shot De⇒Fr translation in multilingual translation. The Transformer model often translates into other language (e.g. English rather than French or copying the source sentence), while our LM4MT suffers less from these mistakes.

Source	Im Süden und Osten Europas tun sich die ökologischen Parteien nach wie vor schwer.
Reference	Les partis écologiques ont du mal à percer dans le sud et dans l'est de l'Europe.
TRANSBASE	In the south and east of Europe, the environmental parties are still struggling.
LM4MT-BASE	Dans les pays de l'Europe du Sud et de l'Est, les partis écologiques restent difficiles.
TRANSBIG	In the south and east of Europe, the environmental parties are still struggling.
LM4MT-BIG	Dans les pays du sud et de l'est de l'Europe, les partis écologiques continuent de se heurter à des difficultés.
Source	Bundesverteidigungsministerin Ursula von der Leyen (CDU) soll neue Präsidentin der EU-Kommission werden
Reference	La ministre fédérale de la Défense Ursula von der Leyen (CDU) doit devenir la nouvelle présidente de la Commission europé.
TRANSBASE	Bundesverteidigungsministerin Ursula von der Leyen (CDU) soll zum neuen Präsidenten der EU-Kommission werden.
LM4MT-BASE	La ministre fédérale de la défense , Ursula von der Leyen ($\overline{\text{CDU}}$) , est censée devenir la nouvelle présidente de la Commission européenne.
TRANSBIG	Bundesverteidigungsministerin Ursula von der Leyen (CDU) soll neue Präsidentin der EU-Komission werden.
LM4MT-BIG	Le ministre fédéral de la défense Ursula von der Leyen (CDU) doit devenir la nouvelle présidente de la Commission européenne.