GANLM: Encoder-Decoder Pre-training with an Auxiliary Discriminator

Jian Yang[1 *], **Shuming Ma**[2], **Li Dong**[2], **Shaohan Huang**[2], **Haoyang Huang**[2],
Yuwei Yin[3], **Dongdong Zhang**[2], **Liqun Yang**[1 †], **Furu Wei**[2], **Zhoujun Li**[1]
[1]State Key Lab of Software Development Environment, Beihang University
[2]Microsoft Research Asia; [3]The University of Hong Kong
{jiaya, lqyang, lizj}@buaa.edu.cn;
{shumma, lidong1, shaohanh, haohua, dozhang, fuwei}@microsoft.com;
yuweiyin@hku.hk

Abstract

Pre-trained models have achieved remarkable success in natural language processing (NLP). However, existing pre-training methods underutilize the benefits of language understanding for generation. Inspired by the idea of Generative Adversarial Networks (GANs), we propose a GAN-style model for encoder-decoder pre-training by introducing an auxiliary discriminator, unifying the ability of language understanding and generation in a single model. Our model, named as GANLM, is trained with two pre-training objectives: replaced token detection and replaced token denoising. Specifically, given masked source sentences, the generator outputs the target distribution and the discriminator predicts whether the target sampled tokens from distribution are incorrect. The target sentence is replaced with misclassified tokens to construct noisy previous context, which is used to generate the gold sentence. In general, both tasks improve the ability of language understanding and generation by selectively using the denoising data. Extensive experiments in language generation benchmarks show that GANLM with the powerful language understanding capability outperforms various strong pre-trained language models (PLMs) and achieves state-of-the-art performance.[1]

1 Introduction

The pre-training-then-fine-tuning paradigm has been proven successful in many natural language processing tasks (Devlin et al., 2019; Liu et al., 2019; Schick and Schütze, 2021). While there are various pre-training approaches for the encoder-only architectures (Clark et al., 2020; Conneau et al., 2020), the encoder-decoder pre-training is underexplored, which is essential for natural language generation. To pre-train the entire encoder-decoder model, BART (Lewis et al., 2020) proposes a denoising language model objective and T5 (Raffel et al., 2020) pre-trains the models with a span corruption objective. Furthermore, mBART (Liu et al., 2020) and mT5 (Xue et al., 2021) extend them to be multilingual pre-trained language models.

Unlike most encoder-decoder pre-training methods that simply apply sequence-to-sequence tasks on a single encoder-decoder architecture, we explore the approaches to pre-train the model in a GAN-style manner with an auxiliary discriminator. GAN (Goodfellow et al., 2014) performs well on both text and image generation tasks by combining the generator and discriminator. It aims to improve the ability of the generator to produce high-quality samples, which is important for the encoder-decoder pre-training when transferred to downstream generation tasks. Similarly, MaskGAN (Fedus et al., 2018) shows the GAN-like training can improve the quality of the autoregressive language model. Therefore, it is intuitive to leverage GAN to empower the encoder-decoder pre-training by unifying language understanding and generation.

In this work, we propose a pre-training framework GANLM, using GAN-style learning to im-

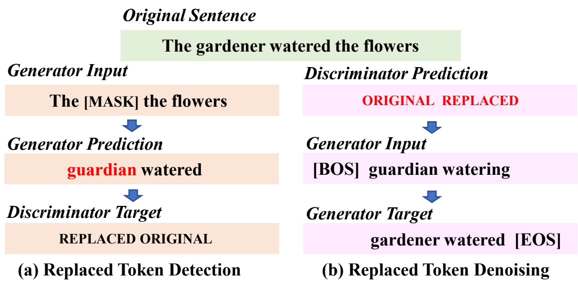


Figure 1: A pre-training sample of our method, where replaced token detection (discriminator) and replaced token denoising (generator) are used for pre-training. The discriminator classifies each generated token into REPLACED or ORIGINAL, where REPLACED denotes the predicted token is different from the gold token. The red fonts denote incorrect predictions.

---

*Contribution during internship at Microsoft Research Asia.

†Corresponding author.

[1]https://github.com/CSJianYang/GanLM

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 9394–9412
July 9-14, 2023

prove the transferability of pre-trained language models for the natural language generation. Specifically, the encoder reads the masked source sentence and the generator obtains target distribution. Then, the discriminator distinguishes whether each token sampled from the target distribution matches the target gold sentence (replaced token detection). The misclassified tokens by discriminator are regarded as hard tokens for the generator to predict accurately. We replace original tokens in the target sentence with misclassified sampled ones to construct the noisy previous context for predicting the target sentence (replaced token denoising). In Figure 1, the generator predicts the masked words "guardian watered", where the incorrect token "guardian" and correct token "watered" are both misclassified into `REPLACED` and `ORIGINAL` by the discriminator. Next, we resample a different token "watering" from the generated distribution. Consequently, the target tokens "gardener watered" are replaced with the sampled tokens "guardian watering" to construct the noisy sample. The generator predicts the next word conditioned on previous noisy tokens (replaced token denoising). Through combing two tasks, GANLM strengthen generation performance with the enhanced language understanding capability from the replaced token detection task.

Our method is effective for text generation and can be extended to natural language understanding tasks. We pre-train GANLM model on large-scale monolingual corpora and evaluate the performance of our pre-trained English model GANLM and multilingual model GANLM-m on various downstream tasks, including text summarization, machine translation, and data-to-text generation. Experimental results demonstrate that our method substantially outperforms previous pre-trained encoder and sequence-to-sequence models on generation tasks. Our method is further tested on GLUE (Wang et al., 2019) and XNLI (Conneau et al., 2018) to validate the transferability of our pre-trained model. Analytic experiments emphasize the importance of the discriminator in both the pre-training and fine-tuning stage, leading to better performance.

2 GANLM

2.1 Model Overview

Our GAN-style pre-trained model comprises a generator ($\mathcal{G}$) and discriminator ($\mathcal{D}$), which are both encoder-decoder frameworks and conditioned on the same encoder (Enc). In Figure 2, the encoder reads the masked sentence and the generator decoder obtains the target distribution. Then the discriminator decoder distinguishes whether each token in the sampled target sentence matches the gold reference. Tokens in the target gold sentence are randomly replaced with misclassified ones by the discriminator to construct the noisy sample, which is fed into the generator decoder to predict the target sentence (replaced token denoising).

2.2 Masked Sequence Generator

Given a monolingual sentence $x = (x_1, \dots, x_n)$ with n words from the dataset D_k of language $L_k \in L_{all} = \{L_1, \dots, L_K\}$ ($|L_{all}| = K$), some random spans of contiguous tokens in x are corrupted as the source sentence, which is denoted as $x^{src} = (x_1, \dots, x_{\backslash u:v}, \dots, x_n)$. $x_{\backslash u:v}$ is a masked span of $x_{u:v}$, where the fragment from position u to v is corrupted by `[MASK]`. Given x^{src}, the generator predicts the original identities of the masked tokens $x^{trg} = (x_{\backslash 1}, \dots, x_{u:v}, \dots, x_{\backslash n})$ autoregressively:

$$x_t^{trg} = \text{Enc-Dec}(x^{src}, x_{1:t-1}^{trg}; \{\theta_\mathcal{E}, \theta_\mathcal{G}\}) \tag{1}$$

where $\theta_\mathcal{E}$ and $\theta_\mathcal{G}$ denote the encoder and decoder parameters of the generator. Enc-Dec denotes an encoder-decoder model. The generator predicts the next position t token x_t^{trg} based on previous tokens.

The training objective of sequence-to-sequence masked language modeling (S2S-MLM) on the dataset D_k of language L_k is defined as:

$$\mathcal{L}_\mathcal{G} = \mathbb{E}_{x \sim D_k}\left[\log P_G(x^{trg}|x^{src}; \{\theta_\mathcal{E}, \theta_\mathcal{G}\})\right] \tag{2}$$

where x^{src} and x^{trg} are derived from x.

2.3 Replaced Token Detection

The generator outputs the distribution of each target token and we create a sampled sentence $\hat{x}^{trg}$ by randomly sampling tokens from the distribution. The discriminator distinguishes whether each token in $\hat{x}^{trg}$ is replaced compared to x^{trg}. Given the target distribution $P_G(x_t^{trg}|x^{src})$ ($x_t^{trg} \in x^{trg}$) from the generator, we construct $\hat{x}^{trg}$ for the discriminator:

$$\begin{aligned} \hat{x}^{trg} &= \text{REPLACE}(x^{trg}; x'_t) \\ w.r.t.\ x'_t &\sim P_G(x_t^{trg}|x^{src}) \wedge x_t^{trg} \in x^{trg} \end{aligned} \tag{3}$$

where $\text{REPLACE}(\cdot)$ replaces target t-th position unmasked token in x^{trg} with the sampled token x'_t from the generated distribution $P_G(x_t^{trg}|x^{src})$.

Given the source sentence x^{src} and the encoder $\theta_\mathcal{E}$, the decoder of the discriminator $\theta_\mathcal{D}$ obtains a sequence of hidden representations $H_d =$

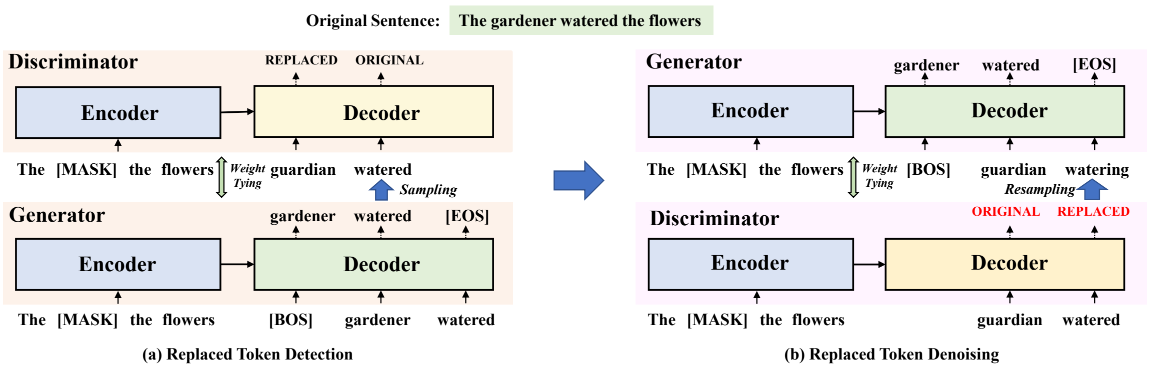


Figure 2: Overview of GANLM, including (a) replaced token detection and (b) replaced token denoising. The encoder reads the source sentence and the generator obtains target distribution, where the generator and discriminator are supervised by the gold labels in (a). The discriminator distinguishes whether the sampled tokens "guardian watered" are replaced (both tokens are misclassified in this example). For the correct predicted token "watered", we obtain a different token "watering" by resampling. The target tokens are replaced with the misclassified tokens to construct the noisy input, which are used to predict the gold sentence "gardener watered `[EOS]`" in (b).

$(h_1, \dots, h_n)$ by feeding the sampled sentence $\hat{x}^{trg}$ to the discriminator decoder:

$$H_d = \text{Enc-Dec}(x^{src}, \hat{x}^{trg}; \{\theta_{\mathcal{E}}, \theta_{\mathcal{D}}\}) \tag{4}$$

where $\theta_{\mathcal{E}}$ and $\theta_{\mathcal{D}}$ denote the encoder and decoder parameters of the discriminator. The decoder of the discriminator $\theta_{\mathcal{D}}$ adopts the bidirectional language model to classify each input token by extracting the past and future representations.

Given the representations H_d, the discriminator classifies sampled tokens $\hat{x}^{trg}$ into the `REPLACED` or `ORIGINAL` label with a sigmoid function σ:

$$V = \sigma(H_d W_d) \tag{5}$$

where $W_d \in R^{d_e \times 2}$ is the matrix projects the token representations to two categories (`REPLACED` or `ORIGINAL`) and d_e is the model hidden size.

The training objective of the replaced token detection task for the discriminator is:

$$\mathcal{L}_{\mathcal{D}} = \mathbb{E}_{x \sim D_k}[\mathbb{I}(\hat{x}^{trg} = x^{trg}) \log V + \mathbb{I}(\hat{x}^{trg} \neq x^{trg}) \log(1 - V)] \tag{6}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

2.4 Replaced Token Denoising

Although our model structure is similar to GAN, the generator is trained with maximum likelihood rather than the standard GAN objective due to the difficulty of the GAN training in NLP. We replace tokens in x^{trg} with misclassified tokens by discriminator to construct the noisy previous context x_f^{trg}. If the sampled token $\hat{x}_t^{trg} = x_t$ is labeled with `ORIGINAL`, we will resample the token x'_t ($x'_t \neq x_t$) from target distribution as the misclassified token to modify x_t in x^{trg}. When $\hat{x}_t^{trg} = x'_t$ ($x'_t \neq x_t$) is labeled with `REPLACED`, the miscalssified token x'_t directly replaces x_t in the target sentence. Given the target sentence x^{trg} and generated probabilities P_G, we replace tokens in x^{trg} with sampled tokens as the previous noisy context:

$$\begin{aligned} x_f^{trg} &= \text{REPLACE}(x^{trg}; \hat{x}_t^{trg}) \\ w.r.t.\ \hat{x}_t^{trg} &\sim P_G(x_t^{trg} | x^{src}) \wedge t \in v \end{aligned} \tag{7}$$

where $v = \{v_1, \dots, v_p\}$ ($|v| = p$) denotes the positions in x^{trg} of the misclassified tokens.

The training objective of the replaced token denoising ($\mathcal{DG}$) task based on the source sentence x^{src} and target noisy context x_f^{trg} is described as:

$$\mathcal{L}_{\mathcal{DG}} = \mathbb{E}_{x \sim D_{L_k}}[-\log P(x^{trg} | x^{src}, x_f^{trg}; \{\theta_{\mathcal{E}}, \theta_{\mathcal{D}}\})] \tag{8}$$

where x^{trg} is predicted by the previous noisy tokens x_f^{trg} instead of previous gold context.

2.5 Multi-task Learning

Given multilingual corpora $D_{all} = \{D_1, \dots, D_K\}$ of K languages, the pre-trained model with parameters $\{\theta_{\mathcal{E}}, \theta_{\mathcal{G}}, \theta_{\mathcal{D}}\}$ is jointly trained over K languages to optimize the combined self-supervised objective as below:

$$\mathcal{L}_{\mathcal{P}} = \mathbb{E}_{L_k \in L_{all}}[\mathcal{L}_{\mathcal{G}} + \lambda \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{DG}}] \tag{9}$$

where $\lambda = 10.0$ is the discriminator weight and $L_{all} = \{L_1, \dots, L_K\}$. To improve model efficiency, a tiny discriminator decoder (4 layers) is adopted to help the generator decoder (12 layers).

3 Discriminator-enhanced Fine-tuning

To fully utilize the pre-trained parameters, we keep the auxiliary discriminator in downstream generation tasks (discriminator-enhanced fine-tuning) to enhance the generator, where both the pre-trained generator and discriminator are recycled. Given the annotated corpus D_s of K languages, the pre-trained model $\{\theta_{\mathcal{E}}, \theta_{\mathcal{D}}, \theta_{\mathcal{G}}\}$ is optimized by:

$$\mathcal{L}_{\mathcal{F}} = \mathbb{E}_{x,y\sim D_s}[\mathcal{L}_{\mathcal{G}} + \lambda\mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{D}\mathcal{G}}] \qquad (10)$$

where x and y are the parallel pair from D_s. The objective in the fine-tuning stage use the original pair x and y without S2S-MLM. The generator $\{\theta_{\mathcal{E}}, \theta_{\mathcal{G}}\}$ are kept for inference by throwing out the discriminator decoder $\theta_{\mathcal{D}}$. Alternatively, the discriminator ($\mathcal{D}$: $\{\theta_{\mathcal{E}}, \theta_{\mathcal{D}}\}$) or generator ($\mathcal{G}$:$\{\theta_{\mathcal{E}}, \theta_{\mathcal{G}}\}$) can also be separately fine-tuned on the downstream task.

4 Experiment Setting

4.1 Pre-training Details

Model Configuration In the experiments, we adopt a sequence-to-sequence base-setting Transformer architecture with 768 hidden size, 3072 FFN (feed-forward network) dimension, 12 attention heads, and 12 encoder/decoder layers. The maximum sequence length of learned positions embeddings in the encoder/decoder is set as 1024. All token embedding matrices and output projection matrix parameters are shared for model efficiency.

Dataset Following the previous work (Liu et al., 2019), our English pre-trained model GANLM is trained on 160GB English monolingual data from BookCorpus, CC-News, OpenWebText, and CC-Stories. In addition, we pre-train GANLM-m with 6TB multilingual data as the pioneering work (Ma et al., 2021), which is a combination of CC100, CC-Net, and Wikipedia, covering 100 languages. All texts are tokenized by SentencePiece (Kudo and Richardson, 2018) and encoded by the dictionary from XLM-R (Conneau et al., 2020).

Optimization For S2S-MLM, we randomly mask 15% of the words in each instance with an average span length of 3 (Raffel et al., 2020). For the replaced token detection, we set the discriminator weight $\lambda = 10.0$. We adopt Adam (Kingma and Ba, 2015) with a learning rate of 3e-4 and 10K warm-up steps for pre-training. The model is trained on 128 NVIDIA A100 GPUs (40GB) from scratch and each batch contains 8K samples. The English pre-trained model GANLM and multilingual model GANLM-m are trained for 500K steps. Specifically, all methods in Table 1 are pre-trained with 500K steps for a fair comparison.

4.2 Downstream Tasks

Monolingual Summarization **CNN / DailyMail** (See et al., 2017) is an abstractive summarization dataset aiming at generating a concise summary from an English news article in CNN and DailyMail. As a popular abstractive summarization dataset, **XSum** (Narayan et al., 2018) compresses a BBC news article to a short one-sentence summary.

Multilingual Summarization To test the capability of our multilingual pre-trained model, a large-scale multilingual dataset named **WikiLingua** (Ladhak et al., 2020) of 18 languages from WikiHow is used to evaluate multilingual abstractive summarization systems.

Bilingual Translation For the bilingual task, we use the **WMT-14 English-German**, **WMT-14 English-French**, and **WMT-16 English-Romanian** dataset for evaluation. WMT-14 En-De from WMT consists of 4.5M sentence pairs and the newstest2014 is used as the test set. WMT-14 En-Fr is a large-scale dataset containing nearly 41M sentence pairs and newstest2014 is adopted for evaluation. WMT-16 En-Ro is comprised of original parallel sentences and back-translation data.

Multilingual Translation **IWSLT-17** of 5 languages and **WMT-10** of 11 languages are utilized for multilingual translation. For IWSLT-17, English (En), German (De), Italian (It), Dutch (Nl), and Romanian (Ro) corpora are downloaded from the IWSLT-2017 benchmark. We use dev2010 for validation and tst2017 for test. For WMT-10, we use the parallel data of 11 languages from the WMT benchmark for evaluation (Wang et al., 2020).

Data-to-Text Generation Data-to-text generation accepts multiple triplets and produces a description. WebNLG (Gardent et al., 2017) contains parallel DBpedia triple sets and short texts. The En-En direction contains 17K triple sets and 45K short texts and the En-Ru direction contains 7K triple sets and 19K texts in Russian. The ROUGE scores on the valid set are reported for a fair comparison with the previous work (Gehrmann et al., 2021).

ID	Model	Pre-training Objective	Summarization	Translation		
			RG-1/RG-2/RG-L	$\text{Avg}_{En\rightarrow X}$	$\text{Avg}_{X\rightarrow En}$	Avg_{all}
①	Transformer w/o Pretraining	-	32.36/11.46/25.47	21.4	25.5	23.5
②	BERT/mBERT (Devlin et al., 2019)	Masked Language Model	36.93/15.00/29.62	26.4	29.6	28.0
③	ELECTRA (Clark et al., 2020)	Replaced Token Detection	43.02/19.94/34.83	29.1	32.8	30.3
④	BART (Lewis et al., 2020)/mBART (Liu et al., 2020)	Denoising Autoencoder	44.13/21.04/36.02	30.3	33.3	31.4
⑤	T5 (Raffel et al., 2020)/mT5 (Xue et al., 2021)	Span Corruption	44.22/21.06/36.12	30.4	33.6	31.7
⑥	**GANLM/GANLM-m (ours)**	Replaced Token Detection + Replaced Token Denoising	**45.36/21.98/36.84**	**31.2**	**34.2**	**32.8**
⑦	⑥ - Discriminator-enhanced Fine-tuning	Replaced Token Detection + Replaced Token Denoising	44.74/21.47/36.40	31.1	34.0	32.6
⑧	⑦ - Replaced Token Denoising	Replaced Token Detection	44.28/21.14/36.24	30.6	33.6	32.1

Table 1: Comparison of different pre-training objectives. Particularly, all methods in this table use the base-setting model and are pre-trained with 500K steps on the same corpora for a fair comparison. We report ROUGE scores for abstractive text summarization (XSum) and BLEU scores for multilingual machine translation (IWSLT-17).

Model	#Corpus	XSum RG-1/RG-2/RG-L	CNN / DailyMail RG-1/RG-2/RG-L
PTRNET (See et al., 2017)	-	28.10/8.02/21.72	39.53/17.28/36.38
MASS (Song et al., 2019)	-	39.75/17.24/31.95	42.12/19.50/39.01
BERTSUMABS (Liu, 2019)	16GB	38.76/16.33/31.15	41.72/19.39/38.76
RoBERTa (Liu et al., 2019)	160GB	42.19/19.22/34.23	41.28/19.11/38.57
ERNIE-GEN (Xiao et al., 2020)	16GB	-	42.30/19.92/39.68
T5 (Raffel et al., 2020)	750GB	-	42.05/20.34/39.40
UniLM (Dong et al., 2019)	16GB	-	43.08/20.43/40.34
UniLMv2 (Bao et al., 2020)	160GB	44.00/21.11/36.08	43.16/20.42/40.14
RoBERTa + *s2s-ft* (Bao et al., 2021)	160GB	43.39/20.55/35.63	42.28/20.21/39.87
UniLMv2 + *s2s-ft* (Bao et al., 2021)	160GB	44.37/21.54/36.61	43.89/21.05/41.02
GANLM (ours)	160GB	**45.36/21.98/36.84**	**44.15/21.12/41.32**

Table 2: Abstractive summarization results on the test set of CNN / DailyMail, and XSum. The evaluation metric is the F1 score of ROUGE (RG) scores.

Model	En	Zh	Avg_{18}
Transformer (Vaswani et al., 2017)	35.9/13.3/29.6	32.1/16.2/26.6	29.9/10.7/25.0
XLM-R (Conneau et al., 2020)	41.4/17.6/34.5	42.2/23.8/34.9	37.5/16.0/31.2
mBART (Liu et al., 2020)	44.2/20.0/32.1	44.8/25.8/37.6	40.1/18.2/33.7
GANLM-m (ours)	**44.7/20.6/37.8**	**45.7/26.4/38.0**	**40.5/18.6/34.0**

Table 3: Results of our method and other baselines on multilingual abstractive summarization. We report the RG-1/RG-2/RG-L (ROUGE) F1 scores of the 18 WikiLingua languages and the average scores.

4.3 Fine-tuning Details

Abstractive Summarization During fine-tuning, we use the Adam (Kingma and Ba, 2015) optimizer with an initial learning rate of 1e-4 and the batch size is set as 2048 tokens on 8 V100 GPUs. The models are trained with the label smoothing cross-entropy with a smoothing ratio of 0.1.

Neural Machine Translation For the large-scale multilingual dataset WMT-10, our pre-trained model is fine-tuned on 32 V100 GPUs with a learning rate of 3e-4. For all bilingual translation tasks and the IWSLT-2017 benchmark, we adopt Adam with a learning rate of 1e-4 and set the batch size as 2048 tokens on 8 V100 GPUs.

Data-to-text Generation We use Adam with a learning rate of {8e-5,1e-4} and set the batch size as 16 sentences on the WebNLG dataset.

5 Comparing Pre-training Objectives

To verify the potential of our pre-training task under a fair comparison, we re-implement previous pre-training tasks and pre-trains baselines on the same corpora with 500K steps, including BERT/mBERT (Devlin et al., 2019), ELECTRA (Clark et al., 2020), BART (Lewis et al., 2020)/mBART (Liu et al., 2020), and T5 (Raffel et al., 2020)/mT5 (Xue et al., 2021). Table 1 reports the ROUGE and BLEU points on the summarization dataset XSum and multilingual translation dataset IWSLT-17. All models have 12 encoder and 12 decoder layers with a hidden size of 768. We observe that the encoder-decoder pre-trained model (T5/mT5) outperforms the pre-trained encoder (ELECTRA, BERT/mBERT), which corroborates the encoder-decoder pre-training is more beneficial to the downstream generation task. Experiments ⑥∼⑧ show the importance of the discriminator and replaced token denoising. Experiment ⑧ demonstrates that only the replaced token detection task can still bring improvement through strengthening the encoder shared by both generator and discriminator. Besides, the replaced token detection task is also helpful to downstream language understanding tasks with a powerful encoder. Lastly, the results verify that fine-tuning with the help of the pre-trained auxiliary discriminator further improves performance.

6 Results of GANLM

The English pre-trained model GANLM is evaluated on the abstractive text summarization task with the ROUGE (Lin, 2004) scores.

XSum As shown in Table 2, the pre-training methods achieve significant improvements over the strong baseline PTRNET without pre-training. The sequence-to-sequence pre-trained model such as

UniLMv2 + *s2s-ft* outperforms other pre-training baselines, where the pseudo-masked technique is applied to the fine-tuning stage. Our method beats all pre-training baselines by a large margin with the discriminator-enhanced fine-tuning strategy. It emphasizes the importance of the fine-tuning strategy for the performance of downstream tasks.

CNN / DailyMail Our method is also evaluated on the CNN / DailyMail dataset in Table 2. The comparisons further indicate that our method obtains strong performance on generation by leveraging the discriminator.

7 Results of GanLM-m

To evaluate the multilingual pre-trained model GanLM-m, we report the BLEU (Papineni et al., 2002) scores for machine translation and ROUGE (Lin, 2004) scores for text summarization and data-to-text generation.

WikiLingua Table 3 reports the average ROUGE scores of 18 WikiLingua languages. The large improvement over other pre-training method demonstrate the summarization ability of our GanLM-m.

WMT14 En-De The results on the bilingual translation are presented at Table 4. We observe that the proposed GanLM outperforms all previous works in the high-resource machine translation scenario ($>$ 4M sentence pairs).

WMT14 En-Fr We further conduct experiments on the WMT14 En-Fr bilingual translation task. Table 4 GanLM-m shows that GanLM-m still brings significant improvement to the downstream task with large-scale machine translation fine-tuning data ($>$ 40M sentence pairs).

WMT16 En-Ro For the low-resource setting ($<$ 1M sentence pairs), there is an average gain of +4 BLEU points compared to the Transformer baseline in Table 5. With the same back-translation data, GanLM-m further improves the model performance and still beats other baselines.

WMT-10 For the multilingual translation, we compare GanLM-m with the strong multilingual pre-trained models in Table 7 and Table 6, such as mBART (Liu et al., 2020). It is notable our method outperforms large pre-trained model mBART with 1024 hidden size by a large margin (+1$\sim$2 BLEU points). Plus, there is a +1.5 BLEU gain over XLM-R, whose encoder and decoder are initialized by the cross-lingual pre-trained encoder (Ma et al., 2020).

Model	En→De	De→En	En→Fr	Fr→En
Transformer (Vaswani et al., 2017)	27.8	30.7	38.2	37.4
mBERT (Devlin et al., 2019)	28.0	30.8	38.0	37.8
XLM-R (Conneau et al., 2020)	29.4	31.4	39.5	38.7
mBART (Conneau et al., 2020)	29.5	33.2	42.0	39.2
mT5 (Conneau et al., 2020)	28.8	32.1	39.8	38.6
GanLM-m (ours)	**30.6**	**34.0**	**42.9**	**39.8**

Table 4: Comparison with other pre-training approaches on the WMT14 En-De and WMT14 En-Fr benchmark.

Model	En→Ro	Ro→En	Ro→En (+BT)
Transformer (Vaswani et al., 2017)	34.0	33.3	36.4
XLM (Conneau and Lample, 2019)	-	35.6	38.5
MASS (Song et al., 2019)	-	-	39.1
BART (Lewis et al., 2020)	-	-	38.0
BART-En (Liu et al., 2020)	36.0	35.8	37.4
BART-Ro (Liu et al., 2020)	37.6	36.8	38.1
XLM-R (Conneau et al., 2020)	35.6	35.8	-
mBART (Liu et al., 2020)	37.7	37.8	38.8
mT5 (Liu et al., 2020)	37.1	37.2	38.0
GanLM-m (ours)	**38.3**	**38.0**	**39.3**

Table 5: Comparison with other pre-training methods on the WMT16 En-Ro benchmark.

WebNLG Table 8 presents the performance on the data-to-text generation task, showing that GanLM outperforms multilingual sequence-to-sequence pre-training baselines mBART and mT5 by +2 ROUGE-L points on both languages.

8 Analysis

Ablation Study To analyze the effect of the proposed pre-training and fine-tuning strategies, we conduct an ablation study of each component of our method in Table 9. Experiment ④ and ⑥ verify the merits of the replaced token detection and replaced token denoising. Furthermore, experiment ⑦ shows that our model with the replaced token denoising task obtains the best performance by jointly fine-tuning generator ($\mathcal{G}$) and discriminator ($\mathcal{D}$).

Low-resource Setting To further analyze the performance of GanLM-m given different sizes of downstream parallel data, we randomly extract K percentage of the whole sentence pairs as the fine-tuned parallel data from the full WMT-16 En→Ro training data. We set $K = \{10\%, 20\%, \dots, 100\%\}$ and compare our method with the Transformer baseline model. Figure 3 shows the BLEU points of our pre-trained multilingual model and the baseline. When the parallel data size is small, the baseline without pre-trained model produces unsatisfactory results. Similarly, in Figure 3(a), GanLM fine-

En→X test sets		#Params	Fr	Cs	De	Fi	Lv	Et	Ro	Hi	Tr	Gu	Avg_{10}
1→1	BiNMT (Vaswani et al., 2017)	242M/10M	36.3	22.3	40.2	15.2	16.5	15.0	23.0	12.2	13.3	7.9	20.2
1→N	MNMT (Vaswani et al., 2017)	242M	34.2	20.9	40.0	15.0	18.1	20.9	26.0	14.5	17.3	13.2	22.0
	mBART (Liu et al., 2020)	611M	33.7	20.8	38.9	14.5	18.2	20.5	26.0	15.3	16.8	12.9	21.8
	XLM-R (Conneau et al., 2020)	362M	34.7	21.5	40.1	15.2	18.6	20.8	26.4	15.6	17.4	14.9	22.5
	GANLM (ours)	430M	**36.0**	**22.4**	**42.1**	**16.5**	**19.7**	**21.5**	**27.0**	**17.4**	**18.6**	**16.3**	**23.8**
N→N	MNMT (Vaswani et al., 2017)	242M	34.2	21.0	39.4	15.2	18.6	20.4	26.1	15.1	17.2	13.1	22.0
	mBART (Liu et al., 2020)	611M	32.4	19.0	37.0	13.2	17.0	19.5	25.1	15.7	16.7	14.2	21.0
	XLM-R (Conneau et al., 2020)	362M	34.2	21.4	39.7	15.3	18.9	20.6	26.5	15.6	17.5	14.5	22.4
	GANLM-m (ours)	430M	**35.0**	**21.8**	**40.2**	**16.1**	**19.2**	**21.9**	**26.7**	**16.2**	**17.9**	**14.4**	**22.9**

Table 6: En→X evaluation results for bilingual (1→1), one-to-many (1→N), and many-to-many (N→N) models on WMT-10. The languages are ordered from high-resource languages (left) to low-resource languages (right).

X→En test sets		#Params	Fr	Cs	De	Fi	Lv	Et	Ro	Hi	Tr	Gu	Avg_{10}
1→1	BiNMT (Vaswani et al., 2017)	242M/10M	36.2	28.5	40.2	19.2	17.5	19.7	29.8	14.1	15.1	9.3	23.0
N→1	MNMT (Vaswani et al., 2017)	242M	34.8	29.0	40.1	21.2	20.4	26.2	34.8	22.8	23.8	19.2	27.2
	mBART (Liu et al., 2020)	611M	36.2	29.9	40.0	22.2	20.6	27.2	37.2	23.3	25.7	21.7	28.4
	XLM-R (Conneau et al., 2020)	362M	35.6	30.2	40.9	22.7	21.7	28.4	37.3	25.4	26.2	22.6	29.1
	GANLM (ours)	430M	**36.9**	**31.8**	**42.4**	**23.2**	**22.5**	**29.4**	**37.9**	**27.2**	**27.9**	**22.9**	**30.2**
N→N	MNMT (Vaswani et al., 2017)	242M	35.9	29.2	40.0	21.1	20.4	26.3	35.5	23.6	24.3	20.6	27.7
	mBART (Liu et al., 2020)	611M	34.8	28.9	39.4	20.7	20.2	25.8	35.9	22.5	25.0	21.9	27.5
	XLM-R (Conneau et al., 2020)	362M	35.7	30.3	41.0	22.2	21.3	28.1	37.0	25.4	26.1	21.9	28.9
	GANLM-m (ours)	430M	**37.0**	**31.1**	**42.4**	**22.7**	**22.5**	**28.1**	**37.1**	**25.3**	**26.9**	**22.7**	**29.6**

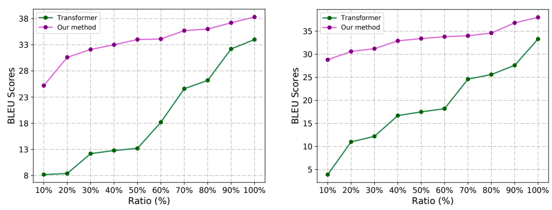
Table 7: X→En evaluation results for bilingual (1→1), one-to-many (1→N), and many-to-many (N→N) models on WMT-10. The languages are ordered from high-resource languages (left) to low-resource languages (right).

Model	En RG-1/RG-2/RG-L	Ro RG-1/RG-2/RG-L
mBART (Liu et al., 2020)	83.4/63.1/70.3	34.8/13.4/33.0
$\text{mT5}_{\text{small}}$ (Gehrmann et al., 2021)	78.8/59.2/67.2	29.7/10.5/28.4
mT5_{base} (Gehrmann et al., 2021)	82.3/62.1/69.7	33.0/12.7/31.3
GANLM-m (ours)	**83.8/63.9/71.2**	**35.2/15.0/33.4**

Table 8: Results on data-to-text generation (WebNLG).

ID	Method	$\mathcal{D}$	$\mathcal{G}$	Xsum RG-1/RG-2/RG-L
①	Transformer w/o Pre-training		✓	32.36/11.46/25.47
②	① + S2S-MLM		✓	44.44/21.25/36.22
③	② + Replaced Token Detection	✓		42.11/18.58/33.21
④	② + Replaced Token Detection		✓	44.28/21.14/36.24
⑤	④ + Replaced Token Denoising	✓		42.41/18.98/34.31
⑥	④ + Replaced Token Denoising		✓	44.74/21.47/36.40
⑦	④ + Replaced Token Denoising	✓	✓	**45.36/21.98/36.84**

Table 9: Ablation study of our method on the test set of the abstractive summarization benchmark XSum, where GANLM is fine-tuned on the downstream task with different pre-training and fine-tuning strategies.



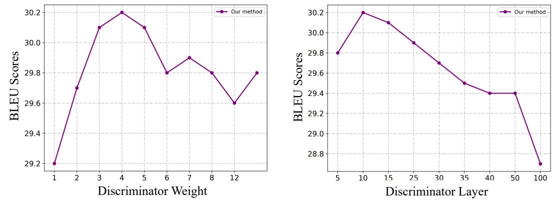
(a) En→Ro (b) Ro→En

Figure 3: Comparison between Transformer and our method on WMT-16 (a) En→Ro and (b) Ro→ En.

tuned on nearly half data (purple line, 50%) defeats the baseline trained on all pairs (green line, 100%), exemplifying the effectiveness of our method in low-resource scenarios.

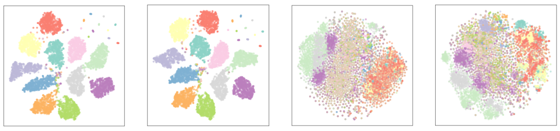
Discussion on Discriminator The weight value λ and layer number of the discriminator are key factors to our pre-training task. As shown in Figure 4, we vary discriminator weight in Figure 4(a) to find a balance between the generator and discriminator objective. To this end, we study the performance of GANLM with different λ, where λ ranges from 5.0 to 100.0. When the weight of the discriminator is 10.0, multiple pre-training tasks are balanced. Moreover, we find it more efficient to have a tiny discriminator ($3 \sim 6$ layers) in Figure 4(b).

Multilingual Representations We randomly select 1000 parallel sentences of each language in WMT-10 and visualize their representations (Maaten and Hinton, 2008) of the last two encoder layers in Figure 5 using our multilingual model fine-tuned on WMT-10 and the multilingual base-



(a) Discriminator Weight (b) Discriminator Layer

Figure 4: Effect of (a) discriminator weight and (b) Discriminator layer on the WMT14 En→De task.



(a) 11-th (b) 12-th (c) 11-th (d) 12-th

Figure 5: (a) and (b) are representations of the baseline from the 11-th and 12-th encoder layers while (c) and (d) are counterparts of the fine-tuned model. Each color denotes one language (11 languages in WMT-10).

line. The first hidden state of the encoder is adopted as the sentence representation. Compared to Figure 5(a) and 5(b) of the baseline, different languages become closer and likely to overlap with each other in Figure 5(c) and 5(d) of our method, demonstrating that our method effectively aligns representations of different languages to the shared space.

Massively Multilingual Translation We compare GANLM-m with the state-of-the-art multilingual NMT model M2M-124 (Goyal et al., 2021). M2M-124_{large} and DeltaLM + Zcode both have a large hidden size of 1024. Our pre-trained model is fine-tuned on the same training data as DeltaLM + Zcode (Yang et al., 2021). Compared to M2M-124_{large}, GANLM-m with fewer training data and only 430M parameters depends more on the transferability of the cross-lingual pre-training model. In Table 10, our method outperforms the DeltaLM + Zcode in zero-shot translation direction ($\text{Avg}_{X \to Y}$) by +1.5 BLEU points, benefiting from our pre-trained model in cross-lingual zero-shot transfer.

Comparison of Pre-training Cost Our English pre-trained model GANLM is trained for nearly 2 weeks on 128 A100 GPUs (40GB), with 500K training steps and a batch size of 8K sequences. Compared to the re-implemented T5 (Raffel et al., 2020), our method is only 0.5 times slower than T5 with the same training steps but gets a significant improvement on the machine translation, text

Model	#Params	$\text{Avg}_{X \to En}$	$\text{Avg}_{En \to Y}$	$\text{Avg}_{X \to Y}$
M2M-124_{base} (Goyal et al., 2021)	175M	15.43	12.02	5.85
M2M-124_{large} (Goyal et al., 2021)	615M	20.03	16.21	7.66
DeltaLM + Zcode (Yang et al., 2021)	711M	30.39	23.52	11.21
GANLM-m (ours)	430M	**30.70**	**24.83**	**13.65**

Table 10: Massively multilingual translation average results (102 × 101 translation directions) on the devtest sets of the flores benchmark.

Model	MNLI	SST-2	MRPC	RTE	QNLI	QQP	Avg_6
BERT (Devlin et al., 2019)	84.5	93.2	87.3	68.6	91.7	91.3	86.1
XLNet (Yang et al., 2019)	86.8	94.7	88.2	74.0	91.7	91.4	87.8
RoBERTa (Liu et al., 2019)	87.6	94.8	90.2	78.7	92.8	91.9	89.3
GANLM-m ($\mathcal{D}$)	89.0	94.7	**90.6**	83.2	93.9	91.7	90.5
GANLM-m ($\mathcal{G}$)	**89.3**	**95.0**	90.5	**85.0**	**94.2**	**92.0**	**91.0**

Table 11: Results of base-setting models on the valid set of GLUE. We report accuracy for classification tasks.

summarization, and data-to-text generation tasks.

Training of replaced token denoising To fully understand the training procedure of the replaced token denoising, we plot the training loss of sequence-to-sequence masked language modeling $L_{\mathcal{G}}$, replaced token detection, and replaced token denoising in Figure 6. Furthermore, we investigate how many tokens in the target sentence are replaced with the misclassified tokens by discriminator in Figure 7. We define p_r as the replaced rate in the target gold sentence. Nearly 7.5% tokens of the target previous tokens are replaced with the misclassified tokens to construct the noisy input samples for the generator decoder.

Language Understanding Our method can be easily extended to various downstream language understanding tasks. We use the GLUE benchmark (Wang et al., 2019) to estimate English pre-trained model GANLM and the XNLI dataset (Conneau et al., 2018) to evaluate the capability of the multilingual language understanding. Our method is tested on each language separately by fine-tuning generator ($\mathcal{G}$) or discriminator ($\mathcal{D}$) on the XNLI dataset. In Table 11, Our English pre-trained model performs better than RoBERTa. Additionally, our pre-trained model outperforms the previous cross-lingual pre-trained encoder XLM and pre-trained encoder-decoder model mT5 in Table 12.

9 Related Work

Pre-training for Generation Language modeling based on the self-supervised learning training objective and large-scale data has been widely used to acquire contextual representations. Pre-training a large Transformer encoder (Vaswani et al., 2017;

Models	En	De	Th	Tr	Vi	Avg$_{15}$
Fine-tuning on English training set (Cross-lingual zero-shot transfer)						
XLM (Conneau and Lample, 2019)	85.0	77.8	73.2	72.5	76.1	75.1
mT5 (Xue et al., 2021)	84.7	77.4	73.2	72.8	74.2	75.4
GANLM-m ($\mathcal{D}$)	85.0	78.6	**74.3**	**74.4**	**77.2**	**75.8**
GANLM-m ($\mathcal{G}$)	**86.3**	**79.0**	74.2	74.5	76.5	75.5
Fine-tuning on each training set (Translate-train)						
XLM (Conneau and Lample, 2019)	85.0	80.3	75.5	74.7	76.6	76.7
mT5 (Xue et al., 2021)	84.7	-	-	-	-	-
GANLM-m ($\mathcal{D}$)	85.0	80.7	76.9	74.4	79.1	77.9
GANLM-m ($\mathcal{G}$))	**86.3**	**80.8**	**77,4**	**74.5**	**79.2**	**78.0**
Fine-tuning on all training sets (Translate-train-all)						
XLM (Conneau and Lample, 2019)	85.0	80.3	76.0	75.6	78.5	77.8
mT5 (Xue et al., 2021)	82.0	77.7	75.0	74.8	74.5	75.9
GANLM-m ($\mathcal{D}$)	**87.3**	**83.1**	**80.3**	**79.9**	81.3	80.5
GANLM-m ($\mathcal{G}$)	87.2	82.7	79.8	79.6	**81.6**	**80.6**

Table 12: Analysis of multilingual classification on the XNLI test set. The evaluation metric is accuracy (%).

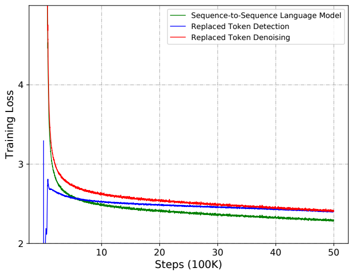


Figure 6: The training loss of sequence-to-sequence language modeling, replaced token detection, and replaced token denoising in the pre-training stage of our English pre-trained model GANLM.

Devlin et al., 2019; Joshi et al., 2019; Liu et al., 2019) with the masked language modeling (MLM) task brings significant improvement for various downstream natural language understanding (NLU) tasks. Many enhanced versions of MLM tasks (Joshi et al., 2019; Sun et al., 2019; Liu et al., 2019; Clark et al., 2020) are proposed to further enhance the capability of the pre-trained model. Besides, pre-training a Transformer decoder (Radford et al., 2018, 2019; Schick and Schütze, 2021) is beneficial for unconditional text generation. There have been numerous attempts for pre-training a sequence-to-sequence Transformer model by adding generative training objectives, such as MASS (Song et al., 2019) and BART (Lewis et al., 2020). Furthermore, T5 (Raffel et al., 2020) explores different pre-training tasks and proposes to corrupt consecutive span of tokens for pre-training. Different from previous works, our work focuses on leveraging the auxiliary discriminator ameliorate encoder-decoder pre-training on language generation tasks.

Multilingual Pre-training Inspired the success of pre-training in a single language such as English,

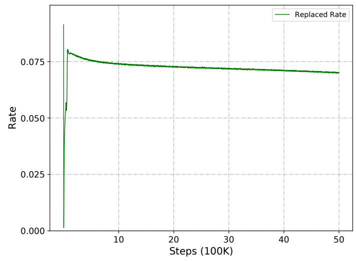


Figure 7: The replaced rate of the replaced token denoising task in the pre-training stage of our English pre-trained model GANLM

.

recent works (Conneau and Lample, 2019; Conneau et al., 2020; Yang et al., 2022a, 2020; Chi et al., 2021b; Yang et al., 2022b,c, 2021) aim to learn cross-lingual representations with different training objectives in multiple languages. For the sequence-to-sequence model, mBART (Liu et al., 2020) pre-trains a Transformer model by denoising training objective in multiple languages. mT5 (Xue et al., 2021) extends the span corruption task for multilingual training and mT6 (Chi et al., 2021a) amplify generation task by introducing a partially non-autoregressive objective. Along the line of research, different multilingual pre-trained models (Ma et al., 2020; Chi et al., 2020) are proposed to solve downstream cross-lingual generation tasks.

10 Conclusion

In this work, we introduce GANLM, a state-of-the-art pre-training encoder-decoder framework for both language generation and understanding tasks trained on large-scale corpora. Our GAN-style models are pre-trained with replaced token detection and replaced token denoising by introducing an auxiliary discriminator. Extensive experiments prove the effectiveness of GANLM on various language generation and translation benchmark datasets. We further verify the capability of the pre-trained model on multiple downstream understanding tasks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62276017, U1636211, 61672081), the 2022 Tencent Big Travel Rhino-Bird Special Research Program, and the Fund of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2021ZX-18).

References

Zeljko Agic and Ivan Vulic. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *ACL 2019*, pages 3204–3210.

Hangbo Bao, Li Dong, Wenhui Wang, Nan Yang, and Furu Wei. 2021. s2s-ft: Fine-tuning pretrained transformer encoders for sequence-to-sequence learning. *CoRR*, abs/2110.13640.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *ICML 2020*, volume 119, pages 642–652.

Zewen Chi, Li Dong, Shuming Ma, Shaohan Huang, Saksham Singhal, Xian-Ling Mao, Heyan Huang, Xia Song, and Furu Wei. 2021a. mT6: Multilingual pretrained text-to-text transformer with translation pairs. In *EMNLP 2021*, pages 1671–1683.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020. Cross-lingual natural language generation via pre-training. In *AAAI 2020*, pages 7570–7577.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021b. Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *NAACL 2021*, pages 3576–3588.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL 2020*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *NeurIPS 2019*, pages 7057–7067.

Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. *arXiv preprint arXiv:1809.05053*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL 2019*, pages 4171–4186.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeurIPS 2019*, pages 13042–13054.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. Ccaligned: A massive collection of cross-lingual web-document pairs. In *EMNLP 2020*, pages 5960–5969.

William Fedus, Ian J. Goodfellow, and Andrew M. Dai. 2018. Maskgan: Better text generation via filling in the _______. In *ICLR 2018*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from RDF data. In *INLG 2017*, pages 124–133.

Sebastian Gehrmann, Tosin P. Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondrej Dusek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Rubungo Andre Niyongabo, Salomey Osei, Ankur P. Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. *CoRR*, abs/2102.01672.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *CoRR*, abs/1406.2661.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2021. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *CoRR*, abs/2106.03193.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP 2018*, pages 66–71.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen R. McKeown. 2020. Wikilingua: A new benchmark dataset for cross-lingual abstractive summarization. *CoRR*, abs/2010.03093.

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 *ACL 2020*, pages 7871–7880.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *ACL 2004*, pages 74–81.

Yang Liu. 2019. Fine-tune BERT for extractive summarization. *CoRR*, abs/1903.10318.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pretraining for neural machine translation. *TACL*, 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. *CoRR*, abs/1907.11692.

Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, Alexandre Muzio, Saksham Singhal, Hany Hassan Awadalla, Xia Song, and Furu Wei. 2021. Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders. *CoRR*, abs/2106.13736.

Shuming Ma, Jian Yang, Haoyang Huang, Zewen Chi, Li Dong, Dongdong Zhang, Hany Hassan Awadalla, Alexandre Muzio, Akiko Eriguchi, Saksham Singhal, Xia Song, Arul Menezes, and Furu Wei. 2020. XLM-T: scaling up multilingual machine translation with pretrained cross-lingual transformer encoders. *CoRR*, abs/2012.15547.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *EMNLP 2018*, pages 1797–1807.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL 2002*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21:140:1–140:67.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *NAACL 2021*, pages 2339–2352.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. Ccmatrix: Mining billions of high-quality parallel sentences on the web. In *ACL 2021*, pages 6490–6500.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *ACL 2017*, pages 1073–1083.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *ICML 2019*, volume 97, pages 5926–5936.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced representation through knowledge integration. *ArXiv*, abs/1904.09223.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *LREC 2012*, pages 2214–2218.

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 *NIPS 2017*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR 2019*. OpenReview.net.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020. Multi-task learning for multilingual neural machine translation. In *EMNLP 2020*, pages 1022–1034.

Dongling Xiao, Han Zhang, Yu-Kun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE-GEN: An enhanced multi-flow pre-training and fine-tuning framework for natural language generation. *CoRR*, abs/2001.11314.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *NAACL 2021*, pages 483–498.

Jian Yang, Shaohan Huang, Shuming Ma, Yuwei Yin, Li Dong, Dongdong Zhang, Hongcheng Guo, Zhoujun Li, and Furu Wei. 2022a. CROP: zero-shot cross-lingual named entity recognition with multilingual labeled sequence translation. In *Findings of EMNLP 2022*, pages 486–496.

Jian Yang, Shuming Ma, Haoyang Huang, Dongdong Zhang, Li Dong, Shaohan Huang, Alexandre Muzio, Saksham Singhal, Hany Hassan, Xia Song, and Furu Wei. 2021. Multilingual machine translation systems from microsoft for WMT21 shared task. In *WMT 2021*, pages 446–455. Association for Computational Linguistics.

Jian Yang, Shuming Ma, Dongdong Zhang, Shuangzhi Wu, Zhoujun Li, and Ming Zhou. 2020. Alternating language modeling for cross-lingual pre-training. In *AAAI 2020*, pages 9386–9393.

Jian Yang, Yuwei Yin, Shuming Ma, Dongdong Zhang, Zhoujun Li, and Furu Wei. 2022b. High-resource language-specific training for multilingual neural machine translation. In *IJCAI 2022*, pages 4461–4467.

Jian Yang, Yuwei Yin, Shuming Ma, Dongdong Zhang, Shuangzhi Wu, Hongcheng Guo, Zhoujun Li, and Furu Wei. 2022c. UM4: unified multilingual multiple teacher-student model for zero-resource neural machine translation. In *IJCAI 2022*, pages 4454–4460.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *NeurIPS 2019*, pages 5754–5764.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *ACL 2020*, pages 1628–1639.

A Statistics of Datasets

WMT-14 En-De WMT-14 En-De consists of 4.5M sentence pairs. The validation set is devtest2014, and the test set is newstest2014.[2]

WMT-16 En-Fr WMT-14 En-Fr is a large-scale dataset containing nearly 41M sentence pairs, where newstest2014 is employed for evaluation.

WMT-16 En-Ro WMT-16 En-Ro is comprised of original parallel sentences and back-translation data. We use newsdev2016 for validation and newstest2016 for test. Following the previous work (Liu et al., 2020), we use the same back-translation data for a fair comparison.[3]

IWSLT-2017 We download English (En), German (De), Italian (It), Dutch (Nl), and Romanian (Ro) corpora from the IWSLT-2017 benchmark. The dev2010 is used for validation and tst2017 for test.[4]

WMT-10 Table 13 lists the detailed statistics of 10 language pairs from WMT-10, which is a collection of parallel data in different languages from the WMT datasets. The parallel data is paired with English and other 10 languages, including French (Fr), Czech (Cs), German (De), Finnish (Fi), Latvian (Lv), Estonian (Et), Romanian (Ro), Hindi (Hi), Turkish (Tr) and Gujarati (Gu). The corpora of the WMT benchmark, exclude WikiTiles, from the latest available year of each language are chosen. After removing the duplicated samples, we limit the size of each parallel language pair data up to 10 million by randomly sampling from the whole corpus. We adopt the same valid and test sets from the WMT benchmark as the previous work (Wang et al., 2020).

WikiLingua To test the capability of our multilingual pre-trained model, a large-scale multilingual dataset named **WikiLingua** (Ladhak et al., 2020) of 18 languages from WikiHow is used to evaluate multilingual abstractive summarization systems.[5]

[2] https://statmt.org/wmt14/translation-task.html
[3] https://www.statmt.org/wmt16/translation-task.html
[4] https://sites.google.com/site/iwsltevaluation2017/TED-tasks
[5] https://github.com/esdurmus/Wikilingua

Code	Language	#Bitext	Training	Valid	Test
Fr	French	10M	WMT15	Newstest13	Newstest15
Cs	Czech	10M	WMT19	Newstest16	Newstest18
De	German	4.6M	WMT19	Newstest16	Newstest18
Fi	Finnish	4.8M	WMT19	Newstest16	Newstest18
Lv	Latvian	1.4M	WMT17	Newsdev17	Newstest17
Et	Estonian	0.7M	WMT18	Newsdev18	Newstest18
Ro	Romanian	0.5M	WMT16	Newsdev16	Newstest16
Hi	Hindi	0.26M	WMT14	Newsdev14	Newstest14
Tr	Turkish	0.18M	WMT18	Newstest16	Newstest18
Gu	Gujarati	0.08M	WMT19	Newsdev19	Newstest19

Table 13: Statistics and sources of the training, valid, and test sets from WMT between English and other languages.

B Pre-training and Fine-tuning Details

Pre-training Hyper-parameters Table 14 summarizes the hyper-parameters for pre-training GANLM and GANLM-m The task-specific hyper-parameters for the downstream language generation and understanding tasks are in Table 15.

Abstractive Summarization During fine-tuning, we use the Adam (Kingma and Ba, 2015) optimizer with an initial learning rate of 1e-4 and the batch size is set as 2048 tokens on 8 V100 GPUs. The models are trained with the label smoothing cross-entropy with a smoothing ratio of 0.1. The last 5 checkpoints are averaged for evaluation.

Neural Machine Translation We adopt Adam with a learning rate of 1e-4 and set the batch size as 2048 tokens on 8 V100 GPUs for all bilingual translation tasks and the IWSLT-2017 benchmark. For the large-scale multilingual dataset WMT-10, our pre-trained model is fine-tuned on 32 V100 GPUs with a learning rate of 3e-4. For a fair comparison, we adopt the same architecture and model size as our pre-trained model.

Data-to-text Generation We use Adam with a learning rate of {8e-5,1e-4} and set the batch size as 16 sentences on the WebNLG dataset.

Multi-lingual Fine-tuning Following the previous work (Wang et al., 2020; Ma et al., 2021), we adopt a dynamic temperate-based sampling strategy to mitigate the unbalance of the multilingual corpora, where we gradually sample more pairs in low-resource languages with the number of epochs increasing. The temperature of the i-th epoch is calculated by:

$$\tau_i = \min(\tau_1, \tau_0 + \frac{i}{N}(\tau - \tau_0)) \quad (11)$$

Hyper-parameter	GanLM	GanLM-m
Number of Encoder Layers	12	12
Number of Generator Layers	12	12
Number of Discriminator Layers	4	4
Hidden size	768	768
FFN hidden size	3072	3072
Attention heads	12	12
Attention head size	64	64
Dropout	0.1	0.1
Attention Dropout	0.1	0.1
Warmup Steps	10k	10k
Peak Learning Rate	4e-4	5e-4
Batch Size	8K	8K
Weight Decay	0.01	0.01
Max Steps	500k	500k
Learning Rate Decay	Linear	Linear
Adam β_1	0.9	0.9
Adam β_2	0.98	0.98
Gradient Clipping	0.0	0.0

Table 14: Hyper-parameters for pre-training GanLM and GanLM-m.

where τ_0 is the initial temperature, τ_1 is the peak temperature, and N is the number of warm-up epochs. We set $\tau_0 = 1.0$, $\tau_1 = 5.0$, and $N = 5$ for all multilingual experiments for a fair comparison.

Given the temperature τ_i i-th epoch, we can calculate the real sampling ratio of the language L_k, where $L_k \in L_{all} = \{L_1, \dots, L_K\}$:

$$q_{L_k}(i) = \frac{p_{L_k}^{\frac{1}{\tau_i}}}{\sum_{j=1}^{K} p_{L_j}^{\frac{1}{\tau_i}}} \tag{12}$$

where $q_{L_k}(i)$ is the sampling ratio of the language L_k in the i-th epoch. p_{L_k} is the real data ratio of the language L_k in all languages. τ_i is the temperature of the i-th epoch, as described in Equation 11.

C Results on Downstream Task

GLUE For each classification task of the GLUE (Wang et al., 2019), we conduct 5 experiments with different seeds $\{1, 2, 3, 4, 5\}$ and report the average accuracy of 5 experiments.

XNLI We also conduct 5 experiments with different seeds $\{1, 2, 3, 4, 5\}$ and report the average accuracy of 5 experiments.

FLORES Since the corpora of $X \rightarrow Y$ are commonly scarce, the performance of low-resource translation direction $\text{Avg}_{X\rightarrow Y}$ mainly depends on the zero-shot cross-lingual transferability of the pre-trained model. Our model with the 12 encoder layers and 12 decoder layers significantly outperforms the previous state-of-the-art model M2M-124 with large model size. In Figure 8, we report the multilingual model initialized by our pre-trained model in all translation directions, where the languages are ordered alphabetically by the language code. Following the previous work (Yang et al., 2021), we use the same training data, including CCAligned (El-Kishky et al., 2020), CCMatrix (Schwenk et al., 2021), OPUS-100 (Zhang et al., 2020), JW300 (Agic and Vulic, 2019), Tatoeba (Tiedemann, 2012), WMT2021 news track[6], multilingual track data[7].

D Weight Sharing

Our pre-trained model includes the discriminator ($\mathcal{D} : \{\theta_{\mathcal{E}}, \theta_{\mathcal{D}}\}$) and generator ($\mathcal{G} : \{\theta_{\mathcal{E}}, \theta_{\mathcal{G}}\}$). We can use a 12-layer generator decoder $\theta_{\mathcal{G}}$ and a 4-layer tiny discriminator decoder $\theta_{\mathcal{D}}$ for replaced token denoising. We propose a weight sharing strategy to improve the model efficiency of the pre-training by sharing weights among the generator and decoder ($\theta_{\mathcal{D}} = \theta_{\mathcal{G}}$) by setting the discriminator generator and generator decoder as the same size (both 12 layers). Table 18 lists the results of different weight sharing strategies. It turns out the sharing decoder setting performs worse than not sharing. It is reasonable since the generator decoder is used for sequence generation whereas the discriminator decoder is a classifier.

[6] http://statmt.org/wmt21/translation-task.html
[7] http://data.statmt.org/wmt21/multilingual-task/

Task	Learning Rate	Warmup Steps	Batch Size	Weight Decay	Max Epoch	Gradient Clipping	Max Source Positions	Max Target Positions
Text Summarization								
CNN / DailyMail	1e-4	1000	2048 (Tokens)	0.0	16	0.0	608	160
XSum	1e-4	1000	2048 (Tokens)	0.0	16	0.0	720	48
WikiLingua	1e-4	1000	2048 (Tokens)	0.0	16	0.0	512	160
Machine Translation								
WMT14 En-De	1e-4	4000	2048 (Tokens)	0.0	50	0.0	512	512
WMT14 En-Fr	1e-4	4000	2048 (Tokens)	0.0	50	0.0	512	512
WMT14 En-Ro	1e-4	4000	2048 (Tokens)	0.0	16	0.0	512	512
IWSLT17	1e-4	4000	2048 (Tokens)	0.05	16	0.0	512	512
WMT10	3e-4	4000	2048 (Tokens)	0.0	8	0.0	512	512
Data-to-Text								
WebNLG	{2.5e-5, 5e-5}	1000	2048 (Tokens)	0.05	16	0.0	512	512
Natural Language Understanding								
XNLI	{2.5e-5, 5e-5}	4000	16 (Sentences)	0.05	30	1.0	512	512
GLUE	{1e-5, 2.5e-5, 5e-5}	4000	{8,16} (Sentences)	0.05	30	1.0	512	512

Table 15: Task-specific hyper-parameters for downstream language generation and understanding benchmarks.

Seed	MNLI	SST-2	MRPC	RTE	QNLI	QQP	Avg_6
Fine-tuning on Discriminator ($\mathcal{D}$)							
1	88.9	94.5	89.7	83.8	93.8	91.6	90.4
2	89.1	94.7	90.0	84.8	93.9	91.7	90.7
3	88.9	94.5	91.7	83.0	93.7	91.9	90.6
4	89.0	94.7	90.9	84.1	93.8	91.8	90.7
5	89.2	95.2	90.7	80.1	94.2	91.7	90.2
Avg	89.0	94.7	90.6	83.2	93.9	91.7	90.5
Fine-tuning on Generator ($\mathcal{G}$)							
1	89.2	95.1	90.4	85.6	94.1	91.9	91.0
2	89.1	95.2	90.9	85.6	94.3	92.1	91.2
3	89.2	95.0	90.4	84.5	94.1	91.9	90.9
4	89.4	95.1	90.9	84.8	94.1	92.1	91.1
5	89.6	94.8	89.7	84.5	94.2	91.8	90.8
Avg	89.3	95.0	90.5	85.0	94.2	92.0	91.0

Table 16: The accuracy scores of the base-setting models on the valid set of GLUE classification tasks.

Model	En	Ar	Bg	De	El	Es	Fr	Hi	Ru	Sw	Th	Tr	Ur	Vi	Zh	Avg_{15}
Cross-lingual zero-shot transfer (models fine-tune on English data only)																
mBERT	80.8	64.3	68.0	70.0	65.3	73.5	73.4	58.9	67.8	49.7	54.1	60.9	57.2	69.3	67.8	65.4
XLM	85.0	73.1	77.4	77.8	76.6	78.9	78.7	69.6	75.3	68.4	73.2	72.5	67.3	76.1	76.5	75.1
mT5-Small	79.6	65.2	71.3	69.2	68.6	72.7	70.7	62.5	70.1	59.7	66.3	64.4	59.9	66.3	65.8	67.5
mT5-Base	84.7	73.3	78.6	77.4	77.1	80.3	79.1	70.8	77.1	69.4	73.2	72.8	68.3	74.2	74.1	75.4
GANLM-m (D)	85.9	72.6	**78.6**	78.6	**76.6**	**80.7**	79.8	70.4	76.0	**64.4**	**74.3**	74.4	66.5	**77.2**	**75.9**	**75.5**
GANLM-m (G)	**86.3**	**73.2**	77.9	**79.0**	76.5	80.3	**80.4**	**70.8**	**76.7**	62.9	74.2	**74.5**	**66.6**	76.5	75.7	75.4
Translate-train (models fine-tune on English training data plus translations in all target languages)																
XLM	85.0	76.5	79.3	80.3	78.1	80.3	80.2	72.3	78.1	70.9	75.5	74.7	63.2	76.6	78.6	76.6
GANLM-m (D)	85.9	**76.9**	**79.9**	80.7	**79.5**	**81.6**	80.9	74.2	**78.7**	**71.8**	76.9	76.9	**65.8**	79.1	**80.0**	77.9
GANLM-m (G)	**86.3**	76.7	79.7	**80.8**	79.7	81.6	**82.0**	**74.6**	78.6	70.8	**77.4**	**77.1**	65.3	**79.2**	79.3	**77.9**
Translate-train (models fine-tune on English training data plus translations in all target languages)																
XLM	85.0	77.6	80.9	80.3	79.1	81.3	80.8	72.9	78.3	72.8	76.0	75.6	68.5	78.5	79.5	77.8
mT5-Small	69.5	63.7	67.5	65.7	66.4	67.5	67.3	61.9	66.4	59.6	63.9	63.5	60.4	63.3	64.5	64.7
mT5-Base	82.0	74.4	78.5	77.7	78.1	79.1	77.9	72.2	76.5	71.5	75.0	74.8	70.4	74.5	76.0	75.9
GANLM-m (D)	**87.3**	**78.3**	82.7	**83.1**	82.2	83.8	83.3	**77.3**	81.3	73.1	**80.3**	**79.9**	71.2	81.3	**81.8**	80.5
GANLM-m (G)	87.2	78.3	**83.3**	82.7	**82.3**	**84.0**	**83.6**	77.1	**81.4**	**74.5**	79.8	79.6	**71.3**	**81.6**	81.6	**80.6**

Table 17: XNLI accuracy scores for each language.

ID	#Params	Strategy	Xsum RG-1/RG-2/RG-L	WMT16 En-Ro En→Ro/Ro→En
①	390M	$\theta_{\mathcal{G}} = \theta_{\mathcal{D}}$	43.26/19.82/35.02	37.4/37.2
②	430M	$\theta_{\mathcal{G}} \neq \theta_{\mathcal{D}}$	**45.36/21.98/36.84**	**38.3/38.0**

Table 18: Evaluation results with different weight sharing strategies on the test set of the Xsum summarization task and WMT16 En-Ro translation task. Both generator decoder $\theta_{\mathcal{G}}$ and discriminator decoder $\theta_{\mathcal{D}}$ have 12 layers in Experiment ② by sharing decoder parameters.

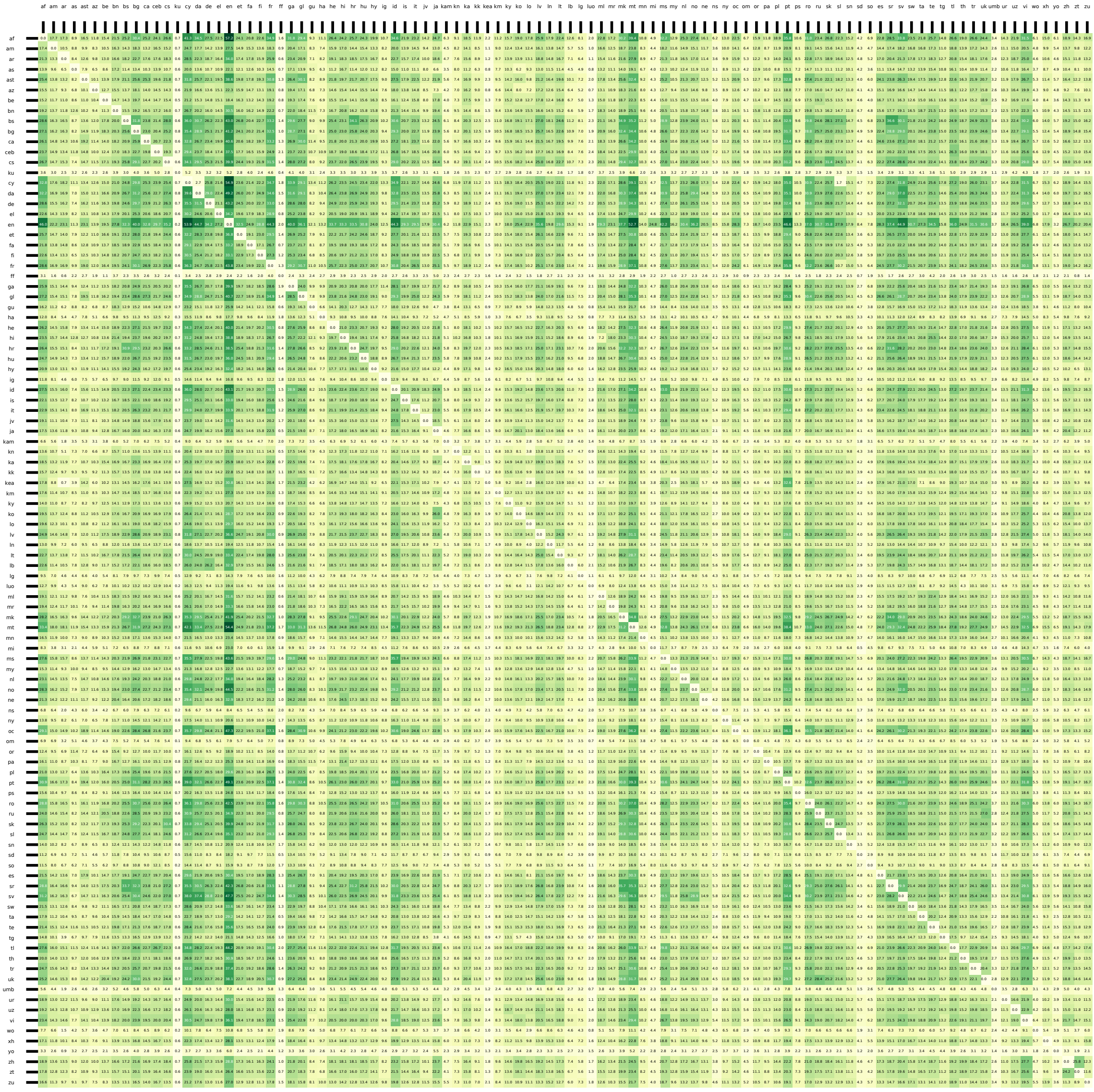


Figure 8: Evaluation results of our multilingual model on all translation directions on the FLORES-101 devtest set, where our model consists of 12 encoder and 12 decoder layers with a hidden size of 768. We fine-tune the multilingual encoder-decoder pre-trained model GANLM-m on the large-scale dataset. The language x in the i-th row and language y in the j-th column denotes the translation direction from the language x to language y. For example, the cell of the 1-st row (af) and the 3-nd column (ar) represents the result of the translation direction af→ar. The table shows the results of all translation directions of 102 languages.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 11

☑ A2. Did you discuss any potential risks of your work?
Section 12

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 4

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section 4

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☒ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

☒ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

☒ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.