Zero-Shot Cross-Lingual Transfer of Neural Machine Translation with Multilingual Pretrained Encoders

Guanhua Chen[1]*, **Shuming Ma**[2], **Yun Chen**[3]†, **Li Dong**[2]
Dongdong Zhang[2], **Jia Pan**[1], **Wenping Wang**[4,1], **Furu Wei**[2]
[1]The University of Hong Kong; [2]Microsoft Research
[3]Shanghai University of Finance and Economics; [4]Texas A&M University
{ghchen,jpan,wenping}@cs.hku.hk, yunchen@sufe.edu.cn,
{shumma, lidong1, dozhang, fuwei}@microsoft.com

Abstract

Previous work mainly focuses on improving cross-lingual transfer for NLU tasks with a multilingual pretrained encoder (MPE), or improving the performance on supervised machine translation with BERT. However, it is under-explored that whether the MPE can help to facilitate the cross-lingual transferability of NMT model. In this paper, we focus on a zero-shot cross-lingual transfer task in NMT. In this task, the NMT model is trained with parallel dataset of only one language pair and an *off-the-shelf* MPE, then it is directly tested on zero-shot language pairs. We propose SixT, a simple yet effective model for this task. SixT leverages the MPE with a two-stage training schedule and gets further improvement with a position disentangled encoder and a capacity-enhanced decoder. Using this method, SixT significantly outperforms mBART, a pretrained multilingual encoder-decoder model explicitly designed for NMT, with an average improvement of 7.1 BLEU on zero-shot any-to-English test sets across 14 source languages. Furthermore, with much less training computation cost and training data, our model achieves better performance on 15 any-to-English test sets than CRISS and m2m-100, two strong multilingual NMT baselines.

1 Introduction

Multilingual pretrained encoders (MPE) such as mBERT (Wu and Dredze, 2019), XLM (Conneau and Lample, 2019), and XLM-R (Conneau et al., 2020) have shown remarkably strong results on zero-shot cross-lingual transfer mainly for natural language understanding (NLU) tasks, including named entity recognition (NER), question answering (QA) and natural language inference (NLI). These methods jointly train a Transformer (Vaswani et al., 2017) encoder to perform masked language modeling task in multiple languages. The pretrained model is then fine-tuned on a downstream NLU task using labeled data in a single language and evaluated on the same task in other languages. With this pretraining and fine-tuning approach, the MPE is able to generalize to other languages that even do not have labeled data. Given that MPE has achieved great success in cross-lingual NLU tasks, a question worthy of research is how to perform zero-shot cross-lingual transfer in the NMT task by leveraging the MPE. Some work (Zhu et al., 2020; Yang et al., 2020; Weng et al., 2020; Imamura and Sumita, 2019) explores approaches to improve NMT performance by incorporating monolingual pretrained Transformer encoder such as BERT (Devlin et al., 2019). However, simply replacing the monolingual pretrained encoder in previous studies with MPE does not work well for cross-lingual transfer of NMT (see baselines in Table 2). Others propose to fine-tune the encoder-decoder-based multilingual pretrained model for cross-lingual transfer of NMT (Liu et al., 2020; Lin et al., 2020). It is still unclear how to conduct cross-lingual transfer for NMT model with existing multilingual pretrained encoders such as XLM-R.

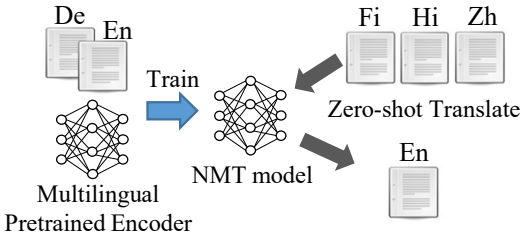


Figure 1: In the zero-shot cross-lingual NMT transfer task, the model is trained with parallel dataset of only one language pair (such as De-En) and a multilingual pretrained encoder. The trained model is tested on many-to-one language pairs (like Fi/Hi/Zh-En) in a zero-shot manner. Monolingual text of the to-be-tested source languages is not available in this task.

* Contribution during internship at Microsoft Research.
† Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 15–26
November 7–11, 2021.

In this paper, we focus on a Zero-shot cross-lingual(X) NMT Transfer task (ZeXT, see Figure 1), which aims at translating multiple unseen languages by leveraging an MPE. Different from unsupervised or multilingual NMT, only an MPE and parallel dataset of one language pair such as German-English are available in this task. The trained model is directly tested on many-to-one test sets in a zero-shot manner.

We propose a Simple cross-lingual(X) Transfer NMT model (SixT) which can directly translates languages unseen during supervised training. We initialize the encoder and decoder embeddings of SixT with the XLM-R and propose a two-stage training schedule that trades off between supervised performance and transferability. At the first stage, we only train the decoder layers, while at the second stage, all model parameters are jointly optimized except the encoder embedding. We further improve the model by introducing a position disentangled encoder and a capacity-enhanced decoder. The position disentangled encoder enhances cross-lingual transferability by removing residual connection in one of the encoder layers and making the encoder outputs more language-agnostic. The capacity-enhanced decoder leverages a bigger decoder than vanilla Transformer base model to fully utilize the labelled dataset. Although trained with only one language pair, the SixT model alleviates the effect of 'catastrophic forgetting' (Serra et al., 2018) and can be transferred to unseen languages. SixT significantly outperforms mBART with an average improvement of 7.1 BLEU on zero-shot any-to-English translation across 14 source languages. Furthermore, with much less training computation cost and training data, the SixT model gets better performance on 15 any-to-English test sets than CRISS and m2m-100, two strong multilingual NMT baselines.[1]

2 Problem Statement

The zero-shot cross-lingual NMT transfer task (ZeXT) explores approaches to enhance the cross-lingual transferability of NMT model. Given an MPE and parallel dataset of a language pair l_s-to-l_t, where l_s and l_t are supported by the MPE, we aim to train an NMT model that can be transferred to multiple unseen language pairs l_z^i-to-l_t, where $l_z^i \neq l_s$ and l_z^i is supported by the MPE. The learned NMT model is directly tested between the unseen language pairs l_z^i-to-l_t in a zero-shot manner. Different from multilingual NMT (Johnson et al., 2017), unsupervised NMT (Lample et al., 2018) or zero-resource NMT through pivoting (Chen et al., 2017, 2018), neither the parallel nor monolingual data in the language l_z^i is directly accessible in the ZeXT task. The model has to rely on the off-the-shelf MPE to translate from language l_z^i. The challenge to this task is how to leverage an MPE for machine translation while preserving its cross-lingual transferability. In this paper, we utilize XLM-R, which is jointly trained on 100 languages, as the off-the-shelf MPE.

The ZeXT task calls for approaches to efficiently build a many-to-one NMT model that can translate from 100 languages supported by XLM-R with parallel dataset of only one language pair. The trained model could be useful for translating resource-poor languages. It can further extend to scenarios where datasets of more language pairs are available. In addition, while currently the cross-lingual transferability of different MPEs is mainly evaluated on cross-lingual NLU tasks, the ZeXT task provides a new perspective for the evaluation, which can hopefully facilitate the research on MPEs.

3 Approach

3.1 Initialization and Fine-tuning Strategy

For downstream tasks like cross-lingual NLI/QA, only an output layer is added to the pretrained encoder at the fine-tuning stage. In contrast, an entire decoder is added on top of the MPE when the model is adapted to NMT task. The conventional strategy that fine-tunes all parameters reduces the cross-lingual transferability in the pretrained encoder due to the catastrophic forgetting effect. Therefore, we make *an empirical exploration* on how to initialize and fine-tune the NMT model with an MPE. The NMT model can be divided into four parts in our method: encoder embedding, encoder layers, decoder embedding, and decoder layers. With an MPE, each part can be trained with one of the following methods, namely,

- **Rand**: randomly initialized and trained;
- **Fix**: initialized from the MPE and fixed;
- **FT**: initialized from the MPE and trained.

We compare different fine-tuning strategies for these modules in a greedy manner. Starting from vanilla Transformer where all parts are randomly initialized, we explore the best training method for

[1]The code is available at https://github.com/ghchen18/emnlp2021-sixt.

ID	Strategy	Es	Fi	Hi	Zh	Avg.
(1)	Vanilla Transformer (BaseDec)	0.6	0.5	0.1	0.2	0.4
Encoder embedding:						
(2)	(1) + FT encoder embed	2.5	1.8	1.0	1.3	1.65
(3)	(1) + Fix encoder embed	2.4	1.5	1.3	1.7	1.73
Encoder layers:						
(4)	(3) + FT encoder layers	11.6	7.4	5.9	4.3	7.3
(5)	(3) + Fix encoder layers	17.9	10.1	6.0	5.2	9.8
Decoder embedding:						
(6)	(5) + FT decoder embed	18.7	10.3	7.1	6.3	10.6
(7)	(5) + Fix decoder embed	20.2	12.3	**7.8**	6.3	11.6
Decoder layers:						
(8)	(7) + Fix decoder layers	2.1	2.2	0.8	1.0	1.5
(9)	(7) + FT decoder layers	20.0	12.3	7.3	**6.9**	11.6
(10)	(7) + Rand BigDec	**20.7**	**13.7**	7.7	6.8	**12.2**

Table 1: BLEU results of different initialization and fine-tuning strategies on zero-shot any-to-English language pairs. Starting from vanilla Transformer where all parts are randomly initialized (Strategy (1)), we initialize the encoder embedding (Strategy (2)-(3)), the encoder layers (Strategy (4)-(5)), the decoder embedding (Strategy (6)-(7)) and the decoder layers (Strategy (8)-(10)) with MPE sequentially. Each time we compare the strategy of 'FT' and 'Fix' which fine-tunes the corresponding module or keeps it fixed, respectively. Since Strategy (8)-(9) use a larger decoder than the rest ones due to decoder layer initialization, we add Strategy (10) whose decoder size is the same as Strategy (8)-(9) for fair comparison. The best BLEU is bold and underlined.

the encoder embedding, the encoder layers, the decoder embedding, and the decoder layers, sequentially. The details of experimental settings are in the Section 4.1. From the results shown in Table 1, we observe that it is the best to initialize the encoder embedding, the encoder layers and the decoder embedding with XLM-R and keep their parameters frozen, while randomly initializing the decoder layers (see Figure 2). More discussions are in the Section 4.2.

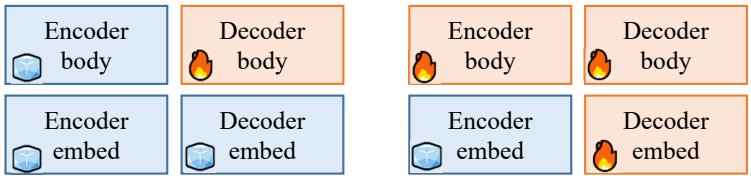
Two-stage training Since we freeze the encoder and only train the decoder layers, the model is able to perform translation while preserving the transferability of the encoder. However, freezing most of the parameters limits the capacity of the NMT model, especially when the training data goes large. Therefore, we propose a second training stage to further improve the translation performance by jointly fine-tuning all parameters except encoder embedding of the NMT.[2] Since the decoder has been well adapted to the encoder at the first stage, we expect the model can be slightly fine-tuned to improve the translation capacity without losing the transferability of the encoder.

[2]According to our preliminary experiment, the average BLEU is 0.2 lower when the encoder embedding is also learned at the second stage. Besides, freezing encoder embedding leads to higher computational efficiency.

3.2 Model

The training strategy and generalization objective of our model are different from vanilla Transformer. This motivates us to propose a new model that can further improve on zero-shot translations. The proposed model consists of a position disentangled encoder and a capacity-enhanced decoder, which aims at enhancing the cross-lingual transferability of the encoder and fully utilizing the labelled data, respectively.

Position disentangled encoder The representations from XLM-R initialized encoder have a strong positional correspondence to the source sentence. The word order information inside is language-specific and may hinder the cross-lingual transfer from supervised source language to unseen languages. Inspired by Liu et al. (2021), we propose to relax this structural constraint and make the encoder outputs less position- and language-specific. More specifically, at the second stage, we remove the residual connection after the self-attention sublayer in one of the encoder layers i



(1) Training at the first stage (2) Training at the second stage

Figure 2: The best strategy for training NMT model for ZeXT task. The blue icy blocks are initialized with an MPE and frozen, while the red fiery blocks are initialized randomly or from the first stage.

during training and inference.[3] The other encoder layers remain the same. The hidden states in this i^{th} encoder layer are calculated as the following pseudo code:

```
h[i] = SelfAttn(h[i-1])
h[i] = LayerNorm(h[i]) # No residual
  connection here
h[i] = h[i] + LayerNorm(FFN(h[i]))
```

where `SelfAttn` is the encoder self-attention sublayer, `FFN` is the feed-forward sublayer and `LayerNorm` is the layer normalization. Liu et al. (2021) aim at training a language-agnostic encoder for NMT using parallel corpus from scratch. Compared with them, our method shows that it's possible to make a pretrained multilingual encoder more language-agnostic by relaxing the position constraint during fine-tuning.

Capacity-enhanced decoder Some previous work (Zhu et al., 2020; Yang et al., 2020) incorporates BERT into NMT and configures the decoder size as Vaswani et al. (2017). For example, to train an NMT on Europarl De-En training dataset, the default decoder configuration is Transformer base (Gu et al., 2018; Currey et al., 2020). However, our model relies more on the decoder to learn from the labeled data, as the encoder is mainly responsible for cross-lingual transfer. This is also reflected in our training strategy: at the first stage only the decoder parameters are optimized, while at the second stage the encoder is only slightly fine-tuned to preserve its transferability. Therefore, the model capacity of SixT is smaller than vanilla Transformer with the same size. We propose to apply a capacity-enhanced decoder that has larger dimension of feed forward network, more layers and more attention heads at both the first and second training stages. The improvement brought by the big decoder is not simply because of more model parameters. More discussions are in the Section 4.2.

4 Experiments

4.1 Setup

Dataset We focus on the any-to-English translations for the ZeXT task. The Europarl-v7 German and English is used as training set. We evaluate the cross-lingual transfer abilities of NMT models on a variety of languages from different language groups[4]: German group (De, Nl), Romance group (Es, It, Ro), Uralic and Baltic group (Et, Fi, Lv), Indo-Aryan group (Hi, Ne) and Chinese (Zh). A concatenation of Fr-En and Cs-En validation dataset which are from different language groups is used as validation dataset for all any-to-English translation tasks. The details of the datasets are in the appendix. Note that none of the monolingual dataset of the tested source languages is available in all experiments.

Model settings We use the XLM-R base model as the off-the-shelf MPE. The model is implemented on `fairseq` toolkit (Ott et al., 2019). We set Transformer encoder the same size as the XLM-R base model. For the decoder, we use the same hyper-parameter setting as the encoder. We denote model with such configuration as SixT and use this configuration for our NMT models through the paper unless otherwise stated. The encoder-decoder attention modules are randomly initialized. We remove the residual connection at the 11-th (penultimate) encoder layer, which is selected on the validation dataset.

For the empirical exploration in Table 1, we use two model configurations. For Strategy (1)–(7) where decoder layers are trained from scratch, we use a smaller decoder denoted as BaseDec. This model configuration is denoted as SixT small. For the rest strategies, we follow the configuration of

[3]Different from Liu et al. (2021), we keep the layer normalization module after the self-attention sublayer for slightly better validation performance.

[4]We refer to the language group information in Table 1 of Fan et al. (2020).

SixT and denote its decoder as BigDec. Table 12 in Appendix presents the details of different model configurations.

Training and evaluation The Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ is used for training. We use label smoothing with value 0.1. The learning rate is 0.0005 and warmup step is 4000 at the first stage. For the second stage, we set the learning rate as 0.0001 and do not use warmup. All the drop-out probabilities are set to 0.3. We use eight GPUs and the batch size is set as 4096 tokens per GPU. Maximum updates number is 200k for the first stage and 30k for the second stage. We use beam search (beam size is 5) and do not tune length penalty. We evaluate the results with sacrebleu[5]. If not specified, the best checkpoint is selected by zero-shot cross-lingual transfer performance on the validation set for all experiments. We refer the reader to Section B in Appendix for more training details.

Baselines We compare our model with vanilla Transformer and five conventional methods to apply pretrained Transformer encoder on NMT task. The pretrained encoders in these methods are replaced with XLM-R base for fair comparison.

- Vanilla Transformer. The encoder is with the same size of XLM-R base, the decoder uses the size of BaseDec. All model parameters are randomly initialized.
- +XLM-R fine-tune encoder (Conneau and Lample, 2019). The encoder is initialized with XLM-R. All parameters are trained.
- +XLM-R fine-tune all (Conneau and Lample, 2019). All parameters except those of cross attention module are initialized with XLM-R and directly fine-tuned.
- +XLM-R as encoder embedding (Zhu et al., 2020). The XLM-R output is leveraged as the encoder input of the NMT. The XLM-R model is fixed during training.
- +Recycle XLM-R for NMT (Imamura and Sumita, 2019). The method initializes the encoder with XLM-R and only trains decoder at the first step. Then all are trained at the second step.
- XLM-R fused model (Zhu et al., 2020). The XLM-R output is fused into encoder and decoder separately with attention mechanism. The encoder embedding is initialized from XLM-R to facilitate transfer. The parameters of XLM-R are frozen during training.

[5]BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a +version.1.5.0

4.2 Results

The results of the empirical exploration in the Section 3.1 are shown in Table 1. Since Strategy (8)–(9) use a larger decoder than the rest ones, we add Strategy (10) whose decoder size is the same as Strategy (8)–(9) for fair comparison. Overall, we observe that it is best to use a big decoder and initialize the decoder embedding and all encoder parameters with XLM-R, and to train the decoder layers from scratch (Strategy (10)).

To verify the effect of a capacity enhanced decoder in the ZeXT task, we train vanilla Transformer with the same size of Strategy (7) (with BaseDec) and Strategy (10) (with BigDec) using the same training corpus.[6] The vanilla Transformer model with BaseDec and BigDec obtains a BLEU score of 23.5 and 22.9 on the De-En test set, respectively. The big decoder improves the performance of SixT, but fails to improve that of vanilla Transformer. This proves the effectiveness of BigDec to improve the zero-shot translation performance of our model.

Table 2 illustrates the performance of the proposed SixT comparing with the baselines. SixT gets 18.3 average BLEU and improves over the best baseline by 5.4 average BLEU, showing that SixT successfully learns to translate while preserving the cross-lingual transferability of XLM-R. For all language pairs, SixT obtains better transferring scores. In contrast, vanilla Transformer can hardly transfer and the other baselines do not well transfer to the distant languages. In addition to zero-shot performance, SixT also achieves the best result on De-En test set. Note that the best checkpoint is selected with zero-shot validation set for all methods.

Previous work (Conneau et al., 2020; Hu et al., 2020) mainly uses XLM-R for cross-lingual transfer on NLU tasks. The experiments demonstrate that XLM-R can be also utilized for zero-shot neural machine translation if it is fine-tuned properly. We leave the exploration of cross-lingual transfer using XLM-R for other NLG tasks as the future work.

4.3 Ablation Study

We conduct an ablation study with the proposed SixT on the Europarl De-En training set, as shown

[6]We use De-En validation dataset this time.

Model	De	Nl	Es	It	Ro	Fi	Lv	Et	Hi	Ne	Zh	Avg.
Vanilla Transformer	22.6	0.6	0.6	0.2	0.7	0.5	0.2	0.4	0.1	0.1	0.2	0.4
+ XLM-R fine-tune encoder	17.9	22.9	14.1	15.9	13.5	7.9	6.6	6.6	5.7	4.5	5.6	10.3
+ XLM-R as encoder embedding	22.7	25.7	13.9	16.1	11.4	7.2	5.9	6.2	4.4	2.6	3.9	9.7
+ Recycle XLM-R for NMT	23.0	28.3	17.5	21.8	16.3	9.2	7.1	8.3	5.1	4.0	4.5	12.2
+ XLM-R fused model	23.6	24.9	10.7	10.0	10.5	6.1	4.0	5.1	6.0	3.4	6.1	8.7
+ XLM-R fine-tune all	20.2	28.3	17.2	21.4	17.2	11.3	8.4	8.7	6.1	5.0	5.8	12.9
Our proposed SixT	**26.4**	**38.6**	**22.9**	**32.0**	**23.9**	**15.8**	**12.0**	**14.5**	**8.6**	**6.1**	**8.1**	**18.3**

Table 2: BLEU comparison between SixT and the baselines on zero-shot any-to-English language pairs. The Avg. column is the average BLEU over all zero-shot language pairs. The best BLEU score is bold and underlined.

ID	TwoStage	BigDec	Resdrop	De	Nl	Es	It	Ro	Fi	Lv	Et	Hi	Ne	Zh	Avg.
(1)	×	×	×	23.7	32.5	20.2	25.7	20.3	12.3	9.2	10.9	7.8	5.4	6.3	15.1
(2)	✓	×	×	26.3	36.4	19.0	24.4	21.6	15.3	10.9	12.8	7.1	4.6	6.8	15.9
(3)	×	✓	×	26.4	32.7	20.7	26.1	21.5	13.7	9.4	11.5	7.7	5.0	6.8	15.5
(4)	✓	✓	×	**27.3**	37.8	22.4	31.0	23.3	15.1	11.5	13.9	8.3	5.8	7.6	17.7
(5)	✓	×	✓	25.7	36.4	19.4	25.8	22.5	**15.9**	11.1	13.3	7.6	5.2	**8.1**	16.5
(6)	✓	✓	✓	26.4	**38.6**	**22.9**	**32.0**	**23.9**	15.8	**12.0**	**14.5**	**8.6**	**6.1**	**8.1**	**18.3**

Table 3: Ablation study of the SixT trained on Europarl De-En. We compare models with different combinations of the second training stage (TwoStage), the capacity-enhanced decoder (BigDec), and the position disentangled encoder (Resdrop). If using Resdrop, TwoStage is required because Resdrop is applied at the second training stage. Note that the model of ID (1) corresponds to the Strategy (7) in Table 1 and ID (6) corresponds to SixT . The best BLEU score is bold and underlined.

in Table 3. Overall, SixT obtains the best zero-shot translation results, demonstrating the importance of all three components. From the results of (1) to (3), TwoStage and BigDec along improve the zero-shot translation performance by 0.8 and 0.4 average BLEU over (1), respectively. However, combining them together brings a significant improvement of 2.6 average BLEU over (1). This indicates that TwoStage and BigDec are complementary to each other, thus it is important to use them together. The results of (6)→(5) confirms our claim: without using BigDec, the performance of SixT drops by 1.8 average BLEU. We also observe that the supervised task (De-En) improves with TwoStage and BigDec (from results of (1) to (4)) while degrades with Resdrop (see results of (2)→(5) and (4)→(6)). This is expected since Resdrop helps to build a more language-agnostic encoder. Although Resdrop degrades supervised performance, it improves zero-shot translation. The zero-shot performance is related with both supervised performance and model transferability. By either enhancing the supervised performance (with TwoStage and BigDec) or the model transferability (with Resdrop), the overall performance of zero-shot translation can be improved.

5 Analysis

Comparison with multilingual NMT In this part, we compare SixT with mBART (Liu et al., 2020), CRISS (Tran et al., 2020) and m2m-100 (Fan et al., 2020) on any-to-English test sets. mBART is a strong pretrained multilingual encoder-decoder based Transformer explicitly designed for NMT. We follow their setting and directly fine-tune all model parameters on WMT19 De-En training set. CRISS and m2m-100 are the state-of-the-art unsupervised and supervised multilingual NMT models, respectively. The CRISS model is initialized with the mBART model and iteratively fine-tuned on 1.8 billion sentences covering 90 language pairs. m2m-100 is trained with 7.5 billion parallel sentences across 2200 translation directions. The results of CRISS and m2m-100 are listed as reference, because CRISS and m2m-100 are many-to-many NMT models whose performance may degrade due to the competitions among different target languages (Aharoni et al., 2019; Zhang et al., 2020), while SixT is a many-to-one NMT model. The official m2m-100 model has three sizes: small (418M parameters), base (1.2B parameters) and large (12B parameters). The results of m2m-100

Model	# Sents	German		Romance			Uralic			Indo-Aryan				East Asian			Avg.
		De	Nl	Es	Ro	It	Fi	Lv	Et	Hi	Ne	Si	Gu	Zh	Ja	Ko	
mBART	0.04B	27.4	43.3	24.7	28.2	29.8	18.8	14.2	15.7	12.3	9.6	7.2	10.3	8.3	6.0	21.1	18.4
CRISS	1.8B	28.8	47.0	**32.2**	**35.4**	48.9	23.9	18.6	23.5	**23.1**	**14.7**	**14.4**	**19.0**	13.4	7.9	24.8	25.0
m2m-100	7.5B	28.0	48.5	30.0	34.1	**50.0**	24.9	**19.9**	**25.8**	21.9	3.7	10.6	0.4	**19.5**	**11.5**	**32.7**	24.1
SixT	0.04B	**33.8**	**54.7**	30.1	33.9	43.0	**26.3**	17.7	25.7	17.5	14.4	12.2	17.3	13.4	10.7	31.2	**25.5**

Table 4: Comparison with mBART, CRISS and m2m-100 on any-to-English test sets. Here we implement SixT with SixT large. mBART follows the original paper (Liu et al., 2020) for fine-tuning. '# Sents' is the number of sentences in the NMT training set. The best BLEU score is bold and underlined. 'Avg.' is the average BLEU across all language pairs.

	Train set	German		Romance			Uralic			Indo-Aryan				East Asian			Avg.
		De	Nl	Es	Ro	It	Fi	Lv	Et	Hi	Ne	Si	Gu	Zh	Ja	Ko	
Vanilla	WMT19 De-En	33.7	3.0	3.6	3.4	1.7	1.6	1.2	1.8	0.1	0.1	0.2	0.2	0.3	0.7	0.3	3.5
	CCAligned Es-En	6.8	5.5	32.5	6.4	17.3	2.0	1.7	2.5	0.3	0.1	0.2	0.2	0.8	0.8	0.4	5.2
	WMT19 Fi-En	1.3	0.7	1.3	1.8	0.6	21.7	0.6	1.7	0.2	0.1	0.1	0.2	0.2	0.4	0.3	2.1
	WAT21 Hi-En	0.6	0.9	0.2	0.5	0.6	0.4	0.3	0.5	21.5	3.6	0.1	0.2	0.1	0.1	0.3	2.0
	WMT18 Zh-En	0.2	0.2	0.3	0.3	0.2	0.3	0.1	0.3	0.1	0.1	0	0.1	22.3	0.3	0.1	1.7
SixT	WMT19 De-En	**31.8**	**47.7**	24.6	18.9	36.5	**28.4**	**14.6**	**18.1**	9.5	6.4	7.2	9.5	9.6	6.8	22.4	**19.5**
	CCAligned Es-En	19.9	38.2	**33.0**	**30.9**	**47.0**	15.2	11.5	12.7	6.9	4.2	3.4	5.6	7.6	4.0	12.8	16.9
	WMT19 Fi-En	18.9	28.4	19.5	21.1	25.1	22.8	11.7	16.7	7.5	6.1	6.1	7.3	8.2	5.1	15.2	14.6
	WAT21 Hi-En	19.0	38.0	20.1	20.7	34.3	15.2	11.5	14.6	**24.3**	**16.7**	**9.6**	**17.8**	8.3	5.9	**23.9**	18.7
	WMT18 Zh-En	20.0	31.8	21.2	21.8	28.2	15.1	11.4	13.4	10.6	7.4	8.1	8.2	**19.9**	**7.1**	20.2	16.3

Table 5: The BLEU results of SixT with training data of different language pairs. The best BLEU of each test set with SixT model is bold and underlined. 'Avg.' is the average BLEU across all language pairs.

(small) model are reported.

To compare with these models, we train a many-to-one SixT large model with WMT19 German-English training data, which only consists of 41 million sentences pairs. It only requires a pre-trained XLM-R large model and do not contain any data in other languages. We remove the residual connection after the self-attention sublayer of the 23-th (penultimate) encoder layer. The dataset and model configuration details are in Table 9 and 12 in the appendix.

From the results in Table 4, the SixT large model is significantly better than mBART and slightly better than CRISS and m2m-100. The averaged BLEU across all languages is 7.1, 0.5 and 1.4 higher than mBART, CRISS and m2m-100[7], respectively. The SixT model has larger model size, nevertheless, the results of SixT are impressive given that SixT does not use any monolingual or parallel texts except German-English training data. The performance gain over mBART shows that with proper fine-tuning strategy, the pretrained multilingual encoder has better cross-lingual transfer ability on NMT tasks. In addition, with large-scale German-English parallel data, the SixT model transfers well to distant resource-poor languages like Ne and Si, which indicates a promising approach to translate resource-poor languages. The SixT performance might be further improved with the data of more languages pairs. We leave this as future work.

Train set	# Sents	Vanilla	SixT
Europarl De-En	1.9M	23.1	26.4
WAT21 Hi-En	3.5M	26.1	24.3
WMT16 De-En	4.5M	30.9	31.2
WMT19 Fi-En	4.8M	22.5	22.8
CCAligned Es-En	20M	37.5	33.0
WMT18 Zh-En	23M	22.3	19.9
WMT19 De-En	41M	33.7	31.8

Table 6: Comparison with vanilla Transformer on the supervised translation direction. The '# Sents' column is the number of sentence pairs of the dataset.

Language transfer v.s. language distance In this part, we explore the relationship between the cross-lingual transfer performance and the language distance. We train the SixT models on different supervised language pairs including De-En, Es-En, Fi-En, Hi-En and Zh-En, and then directly apply them to all test sets, as seen in Table 5.[8]

[7]The 1.2B m2m-100 model is larger than our model (737M parameters) and gets 2.2 more average BLEU than SixT.

[8]The details of the datasets are in the appendix.

Train set	# Sents	German		Romance			Uralic			Indo-Aryan				East Asian			Avg.
		De	Nl	Es	Ro	It	Fi	Lv	Et	Hi	Ne	Si	Gu	Zh	Ja	Ko	
Europarl-v7	1.9M	26.4	38.6	22.9	23.9	32.0	15.8	12.0	14.5	8.6	6.1	5.6	7.5	8.1	4.7	15.1	16.1
WMT19	41M	31.8	47.7	24.6	18.9	36.5	28.4	14.6	18.1	9.5	6.4	7.2	9.5	9.6	6.8	22.4	19.5

Table 7: The BLEU results of SixT with training data of different sizes for any-to-English translation. '# Sents' is the number of parallel sentences in the training set. 'Avg.' is the average BLEU across all language pairs.

We observe that the cross-lingual transfer generally works better when the SixT model is trained on source languages in the same language family. The performance on Ko-En is one exception, where Hi-En achieves the best transfer performance. We also notice that the vocabulary overlapping (even character overlapping) between Hindi and Korean is low, showing that significant vocabulary sharing is not a requirement for effective transfer. When trained on 3.5 million Hi-En sentence pairs, SixT obtains promising results on the Ne-En and Si-En translation, with a BLEU score of 16.7 and 9.6, respectively. As comparison, The vanilla Transformer supervised with FLoRes training set only receives 14.5 and 7.2 BLEU score (Liu et al., 2020) on the same test sets. Therefore, another approach to translate resource-poor languages is to train SixT on similar high-resource language pairs.

As a comparison, we train vanilla Transformer configured as Transformer big[9] without MPE initialization with the same training sets and validation sets. The poor zero-shot cross-lingual performance of vanilla Transformer indicates that the XLM-R initialized encoder is essential and can produce language-agnostic representations.

Performance on the supervised language pair To study whether the SixT model gains the cross-lingual transfer ability at the cost of performance degradation on the supervised language pair, we compare the vanilla Transformer big model[10] and SixT model on the supervised translation task. The performance of SixT is lower than that of vanilla Transformer when more than 20M parallel sentences are available, but it gets better performance with fewer parallel sentences. The Hindi-to-English is an exception where SixT has lower BLEU. When large amount of bi-text data is given, the SixT model size is expected to be increased to fully digest the bi-text. For example, if we replace SixT with SixT large and train SixT large on WMT19 De-En, we get 33.8 BLEU on De-En test set (see Table 4), which is comparable of 33.7 BLEU obtained by vanilla Transformer.

[9] i.e. 'transformer_wmt_en_de_big' configuration in the `fairseq` toolkit.

[10] The validation dataset of the supervised language pair is used.

Train set	En-De	Fr-De	Cs-De	Ru-De	Nl-De	Avg.
WMT16	25.7	18.5	14.4	29.0	39.0	25.2
WMT19	26.7	20.1	15.6	31.4	42.3	27.4

Table 8: The BLEU results of SixT for any-to-German translation. 'Avg.' denotes the average BLEU across all zero-shot language pairs.

Performance vs. training corpus size To examine the relationship between cross-lingual transfer ability and training data size, we compare the zero-shot BLEU scores of SixT models trained on Europarl De-En and WMT19 De-En. The results are shown in Table 7. It shows that increasing training data size can consistently improve the zero-shot translation performance. For instance, SixT trained with WMT19 improves over SixT trained with Europarl-v7 by 3.4 average BLEU.

Performance with other target language To build many-to-one NMT model with other target language, we train two SixT models on WMT16 En-De and WMT19 En-De, respectively. We use Fi-De as validation language pair and Fr/Cs/Ru/Nl-De as test language pairs. From the results shown in Table 8, SixT can obtain reasonable transferring scores to unseen source languages when target language is not English. Again, the results confirm that the cross-lingual transfer ability improves with larger training data.

6 Related Work

Zero-shot cross-lingual transfer learning Multilingual pretrained models, such as mBERT (Wu and Dredze, 2019), XLM-R (Conneau et al., 2020), mBART (Liu et al., 2020), and mT5 (Xue et al., 2021), have achieved success on zero-shot cross-lingual transfer for various NLP tasks. The models are pretrained on large-scale multilingual corpora with a shared vocabulary. After pretrained, it is

fine-tuned on labeled data of downstream tasks in one language and directly tested in other languages in a zero-shot manner. While multilingual pretrained models with encoder-decoder-based architecture (Liu et al., 2020; Chi et al., 2020) work well on cross-lingual transfer for NLG tasks, multilingual pretrained encoders (Wu and Dredze, 2019; Conneau and Lample, 2019; Conneau et al., 2020) are mainly applied to cross-lingual NLU tasks (Hu et al., 2020). In this work, we explore how to fine-tune an off-the-shelf multilingual pretrained encoder for zero-shot cross-lingual transfer in neural machine translation, a typical NLG task.

Pretrained models for NMT Some previous works (Imamura and Sumita, 2019; Conneau and Lample, 2019; Yang et al., 2020; Weng et al., 2020; Ma et al., 2020; Zhu et al., 2020) explore methods to integrate pretrained language encoders into the NMT model to improve supervised translation performance. For instance, Zhu et al. (2020) propose BERT-fused model, in which they first use BERT to extract representations for an input sentence, and then fuses the representations into both the encoder and decoder via the attention mechanism. Another line of works (Liu et al., 2020; Song et al., 2019; Lin et al., 2020) propose novel encoder-decoder-based multilingual pretrained language models and fine-tune such models for NMT. For example, Liu et al. (2020) propose mBART, an encoder-decoder-based Transformer explicitly designs for NMT and demonstrate that mBART can be fine-tuned for supervised and zero-shot NMT. Different from them, we leverage MPE for zero-shot translation instead of supervised translation. Among the previous works, Wei et al. (2021) is the most similar with ours. They fine-tune their MPE on NMT with a two-stage strategy. However, their work focuses on improving the MPE for a more universal representation across languages and lacks in-depth study of cross-lingual NMT. In contrast, we aim at leveraging an MPE for machine translation while preserving its ability of cross-lingual transfer.

7 Conclusion

In this paper, we focus on the zero-shot cross-lingual NMT transfer (ZeXT) task which aims at leveraging an MPE for machine translation while preserving its ability of cross-lingual transfer. In this task, only a multilingual pretrained encoder such as XLM-R and one parallel dataset such as German-English are available. We propose SixT for this task, which enables zero-shot cross-lingual transfer for NMT by making full use of the labelled data and enhancing the transferability of XLM-R. Extensive experiments demonstrate the effectiveness of SixT. In particular, SixT outperforms mBART, a pretrained encoder-decoder-based model explicitly designed for NMT. It also gets better performance than CRISS and m2m-100, two strong multilingual NMT models, on 15 any-to-English test sets with less training data and training computation cost.

Acknowledgements

This project was supported by National Natural Science Foundation of China (No. 62106138) and Shanghai Sailing Program (No. 21YF1412100). Wenping Wang and Jia Pan acknowledge the support from Centre for Transformative Garment Production. We thank the anonymous reviewers for their insightful feedbacks on this work.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of NAACL*, pages 3874–3884.

Yun Chen, Yang Liu, Yong Cheng, and Victor OK Li. 2017. A teacher-student framework for zero-resource neural machine translation. In *Proceedings of ACL*, pages 1925–1935.

Yun Chen, Yang Liu, and Victor OK Li. 2018. Zero-resource neural machine translation with multi-agent communication game. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zewen Chi, Li Dong, Furu Wei, Wenhui Wang, Xian-Ling Mao, and Heyan Huang. 2020. Cross-lingual natural language generation via pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7570–7577.

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 *Proceedings of ACL*, pages 8440–8451, Online.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Anna Currey, Prashant Mathur, and Georgiana Dinu. 2020. Distilling multiple domains for neural machine translation. In *Proceedings of EMNLP*, pages 4500–4511.

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

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *arXiv preprint arXiv:2010.11125*.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of EMNLP*, pages 3622–3631.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 4411–4421.

Kenji Imamura and Eiichiro Sumita. 2019. Recycling a pre-trained BERT encoder for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 23–31.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

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

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of ACL*, pages 66–75.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of EMNLP*.

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

Danni Liu, Jan Niehues, James Cross, Francisco Guzmán, and Xian Li. 2021. Improving zero-shot translation by disentangling positional information. In *Proceedings of ACL*, pages 1259–1273.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

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. *arXiv preprint arXiv:2012.15547*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL*.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. 2018. Overcoming catastrophic forgetting with hard attention to the task. In *Proceedings of ICML*, volume 80, pages 4548–4557.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training. In *Proceedings of NeurIPS*.

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 *NeurIPS*, pages 5998–6008.

Xiangpeng Wei, Yue Hu, Rongxiang Weng, Luxi Xing, Heng Yu, and Weihua Luo. 2021. On learning universal representations across languages. *Proceedings of ICLR*.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo. 2020. Acquiring knowledge from pre-trained model to neural machine translation. In *Proceedings of AAAI*, pages 9266–9273.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. In *Proceedings of EMNLP-IJCNLP*, pages 833–844.

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 *Proceedings of NAACL*, pages 483–498.

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

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

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *Proceedings of ICLR*.

A Dataset

The dataset is from WMT translation task, CCAligned corpus[11], WAT21 translation task[12], Flores test set[13] and Tatoeba test sets[14]. We use the first 20M sentence pairs in the Es-En CCAligned corpus as training set. For experiments of Table 5, the validation set for De-En, Es-En and Fi-En are the concatnation of Fr-En and Cs-En validation set. We use Ta-En and Zh-En as the validation set for Hi-En and Zh-En, respectively. More details are in Table 9 to Table 11.

To be compatible with XLM-R model, all texts are tokenized with the same XLM-R sentencepiece (Kudo, 2018) model. The <bos> token is added at the beginning of each source sentence while <eos> token is appended at the end when the NMT model initializes encoder with XLM-R. The source sentence length is limited within 512 tokens.

B Model and Training Details

The encoder of SixT is the same size of XLM-R model. We compare models with different decoder configurations in the paper, the details are in the Table 12. For all models, the dimension of decoder hidden states equals that of encoder hidden states. The number of attention heads is set as 16 for the decoder of SixT large model, so that the dimension of hidden states can be divided by the number of attention heads. We use separate encoder and decoder embeddings. We tie the decoder input and output embeddings. The source vocabulary uses the same 250k vocabulary of XLM-R, while the target vocabulary is generated from the training corpus. All experiments are done with 8 GPUs.

We compare SixT large with CRISS, m2m-100 and mBART in the Table 4. We use the official model checkpoints of mBART[15] (611M parameters), CRISS[16] (680M parameters) and m2m-100[17]

[11] http://www.statmt.org/cc-aligned/

[12] http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/indic_wat_2021.tar.gz

[13] https://github.com/facebookresearch/flores/raw/master/data/flores_test_sets.tgz

[14] https://object.pouta.csc.fi/Tatoeba-Challenge/test-v2020-07-28.tar

[15] https://github.com/pytorch/fairseq/blob/master/examples/mbart

[16] https://github.com/pytorch/fairseq/tree/master/examples/criss

[17] https://github.com/pytorch/fairseq/tree/master/examples/m2m_100

Type	Lang	Source	# Sents
Training set	De-En	Europarl v7	1.9M
Training set	De-En	WMT16	4.5M
Training set	De-En	WMT19	41M
Training set	Es-En	CCAligned	20M
Training set	Fi-En	WMT19	4.8M
Training set	Hi-En	WAT21	3.5M
Training set	Zh-En	WMT18	23M
Valid set	Cs-En	Newstest 14	3003
Valid set	Es-En	Newstest 10	2489
Valid set	Fi-En	Newstest 19	1996
Valid set	Fr-En	Newstest 14	3003
Valid set	Hi-En	Newsdev 14	520
Valid set	Ta-En	WAT21	2390
Valid set	Zh-En	Newstest 17	2001

Table 9: Training and valid set for any-to-English translation. The '# Sents' column is the number of sentence pairs of the dataset.

Lang	Source	Lang	Source
De-En	Newstest 14	Ko-En	Tatoeba
Es-En	Newstest 13	Lv-En	Newstest 17
Et-En	Newstest 18	Ne-En	Flores
Fi-En	Newstest 16	Nl-En	Tatoeba
Gu-En	Newstest 19	Ro-En	Newstest 16
Hi-En	Newstest 14	Si-En	Flores
It-En	Tatoeba	Zh-En	Newstest 18
Ja-En	Newstest 20		

Table 10: Test sets for any-to-English translation.

Type	Lang	Source
Training set	En-De	WMT16
Training set	En-De	WMT19
Valid set	Fi-De	Tatoeba
Test set	Cs-De	Newstest 19
Test set	En-De	Newstest 14
Test set	Fr-De	Newstest 19
Test set	Nl-De	Tatoeba
Test set	Ru-De	Tatoeba

Table 11: Dataset used for English-to-German translation in Section 5.

Model	H_d	H_{enc}^{ff}	L_{enc}	A_{enc}	H_{dec}^{ff}	L_{dec}	A_{dec}
Transformer base	512	2048	6	8	2048	6	8
Transformer big	1024	4096	6	16	4096	6	16
SixT small	768	3072	12	12	2048	6	8
SixT	768	3072	12	12	3072	12	12
SixT large	1024	4096	24	16	3072	12	16

Table 12: Model configurations for different models. The 'A' column is the number of attention heads.

(418M parameters). The training hyper-parameters of SixT large model are the same with that in Section 4.1.

C Language Code

The information of the languages used in this paper is listed in the Table 13.

ISO	Language	Family
cs	Czech	Slavic
de	German	Germanic
en	English	Germanic
es	Spanish	Romance
et	Estonian	Uralic
fi	Finnish	Uralic
fr	French	Romance
gu	Gujarati	Indo-Aryan
hi	Hindi	Indo-Aryan
it	Italian	Romance
ja	Japanese	Japonic
ko	Korean	Koreanic
lv	Latvian	Baltic
ne	Nepali	Indo-Aryan
nl	Dutch	Germanic
ro	Romanian	Romance
ru	Russian	Slavic
si	Sinhala	Indo-Aryan
ta	Tamil	Dravidian
zh	Chinese	Chinese

Table 13: The information of the languages used in this paper.