The NiuTrans Machine Translation Systems for WMT22

Weiqiao Shan[1], **Zhiquan Cao**[1], **Yuchen Han**[1], **Siming Wu**[1], **Yimin Hu**[1],
Jie Wang[1], **Yi zhang**[1], **Hang Cao**[1], **Baoyu Hou**[1], **Chenghao Gao**[1], **Xiaowen Liu**[1],
Tong Xiao[1,2], **Anxiang Ma**[1,2] **and Jingbo Zhu**[1,2]
[1]NLP Lab, School of Computer Science and Engineering,
Northeastern University, Shenyang, China
[2]NiuTrans Research, Shenyang, China
shanweiqiao96@gmail.com ,{xiaotong, maanxiang, zhujingbo}@mail.neu.edu.cn

Abstract

This paper describes the NiuTrans neural machine translation systems of the WMT22 General MT constrained task. We participate in four directions, including Chinese→English, English→Croatian, and Livonian↔English. Our models are based on several advanced Transformer variants, e.g., Transformer-ODE, Universal Multiscale Transformer (UMST). The main workflow consists of data filtering, large-scale data augmentation (i.e., iterative back-translation, iterative knowledge distillation), and specific-domain fine-tuning. Moreover, we try several multi-domain methods, such as a multi-domain model structure and a multi-domain data clustering method, to rise to this year's newly proposed multi-domain test set challenge. For low-resource scenarios, we build a multi-language translation model to enhance the performance and try to use the pre-trained language model (mBERT) to initialize the translation model.

1 Introduction

We participate in the WMT22 General MT task, including Chinese→English (ZH→EN), English→Croatian (EN→HR), and Livonian↔English (LIV↔EN) in four directions. All of our systems are built with constrained data sets. We adopt some methods that have been proven to work well in WMT over the past few years (Li et al., 2019; Zhang et al., 2020; Zhou et al., 2021). At the same time, we also adopt some new model structures (Li et al., 2022; Jiang et al., 2020), data clustering (Aharoni and Goldberg, 2020), initialization (Guo et al., 2020), and training methods (Liu et al., 2021), which are described in detail below.

For data preparation and augmentation, since filtering data could hurt the model performance on the general domain machine translation task, we apply several soft data filtering rules to preserve as much data as possible (Zhang et al., 2020; Zhou et al., 2021). To obtain the in-domain data, we use the open-source toolkit `XenC` (Rousseau, 2013) and specially try a domain clusters method based on the BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) model in the ZH→EN direction. We also use back-translation (Sennrich et al., 2016a) and knowledge distillation (Freitag et al., 2017) iteratively to increase the size of in-domain data, which has been proved effective in recent years (Zhang et al., 2020; Zhou et al., 2021).

For model architectures, our system is built on several Transformer variants, including Transformer-RPR, Transformer-DLCL (Wang et al., 2019), Transformer-ODE (Li et al., 2021), Transformer-UMST (Li et al., 2022), and Transformer-based model with domain mixing (Jiang et al., 2020). We build a wide and deep model based on the pre-norm structure (Wang et al., 2019) and relative position representation(RPR) (Shaw et al., 2018), inspired by the effectiveness of the deep model. Furthermore, we select four single models to build the ensemble model for better performance. Particularly, in the EN↔LIV direction, we build a multilingual machine translation system (Johnson et al., 2017) based on the above models.

For model initialization, training, and decoding strategies, we use nucleus sampling(Top-P) (Holtzman et al., 2020), top-k sampling(Top-K), and beam search as decoding methods in all languages. At the same time, we adopt scheduling sampling (Liu et al., 2021) in ZH→EN direction during fine-tuning. Furthermore, we attempt to initialize the translation model with the pre-trained language model based on lightweight adapter (Guo et al., 2020) in the EN↔LIV direction.

Based on the softer filtering rules and appropriate hypo-parameter settings, we achieve better results on the deep model than last year. In the ZH→EN direction, fine-tuning with the normal training and the scheduling sampling also obtain

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 366 - 374
December 7-8, 2022 ©2022 Association for Computational Linguistics

good results. Furthermore, we use an unsupervised multi-domain data clustering method and some simple domain classification methods. However, we find no significant domain differences in the constrained data. Initializing the translation model with the pre-trained model leads to poor performance in the EN↔LIV direction. It may be due to the sensitivity to the size of the training set.

The rest of the paper is organized as follows: In Section 2, we describe our system in detail, including the data preprocessing and filtering, model structure, back-translation and knowledge distillation, fine-tuning, and post-editing. In Section 3, we introduce our experimental settings and results according to different tasks and give a brief analysis. In Section 4, we summarize our work.

2 System Overview

In Figure 1, we describe the whole process of our system. We use three different colors to represent the different translation tasks. At the data preparation stage, we perform several data processing methods to obtain the training set. Then, we train several models with different structures and use back-translation(BT) and knowledge distillation(KD) iteratively based on ensemble model. Finally, we obtain our final submission based on fine-tuning and post editing.

2.1 Data Preprocessing and Filtering

In the word segmentation stage, we choose different word segmentation methods for the three languages according to the language characteristic. In ZH→EN, we use the `NiuTrans` (Xiao et al., 2012) word segmentation tool for both Chinese and English, which makes it easier for the model to align the words in the bilingual sentence. In EN↔LIV, we use `Reldi-Tokeniser`[1] for each language. In EN→HR, we use `Reldi-Tokeniser` for Croatian and `Niutrans` for English. Further, we apply BPE (Sennrich et al., 2016b) with 32K operations and not shared vocabulary in most language pairs. Specifically, in EN↔LIV, we use five languages, including EN, CS, LIV, ET, and LV, to build a multilingual translation system. We apply BPE with different operations for different languages, as shown in Table 1. Furthermore, we manually construct a dictionary based on `fast_align` (Dyer et al., 2013) to improve word-level alignment.

language	**operations**
EN	32K
CS	32K
LIV	10K
ET	10K
LV	10K

Table 1: Bpe operations in Livonian↔English

We mainly use the previous filtering method (Zhou et al., 2021). Nevertheless, we adopt softer filtering rules to improve the model performance on the general MT task as follows:

- Filter out sentences that contain long words over 40 characters and sentences that contain over 200 words.
- The word ratio between the source and target sentence must be in the range of [1/3, 3].
- Use Unicode to filter uncommon characters that never appear in previous years' test sets.
- Filter out the sentences which contain HTML tags or duplicated translations.

We use the same filtering rules for monolingual and bilingual data, and based on the filtering rules, we retain more data to do domain filtering further. Based on these filtering rules, we effectively reduce the <UNK> proportion on the previous years' newstest set, while retaining some longer sentences to meet the challenge of the general test set.

2.2 Model Architectures

In recent years, the deep model has been widely proven to be a very effective model structure (Wang et al., 2019; Zhang et al., 2020; Zhou et al., 2021), so we use a variety of deep models, including Transformer-RPR, Transformer-DLCL (Wang et al., 2019), and Transformer-ODE (Li et al., 2021). In addition, we use a new model structure, Transformer-UMST (Li et al., 2022), which uses multi-scale information to enhance the representation ability of model representation. The explicit information of the above model is shown in Table 2.

Transformer-RPR: Compare to Vanilla Transformer, we only increase the number of encoder layers and add RPR into the self-attention at each layer to efficiently consider the relative positions between different representations.

[1] https://github.com/clarinsi/reldi-tokeniser/blob/master/LICENSE

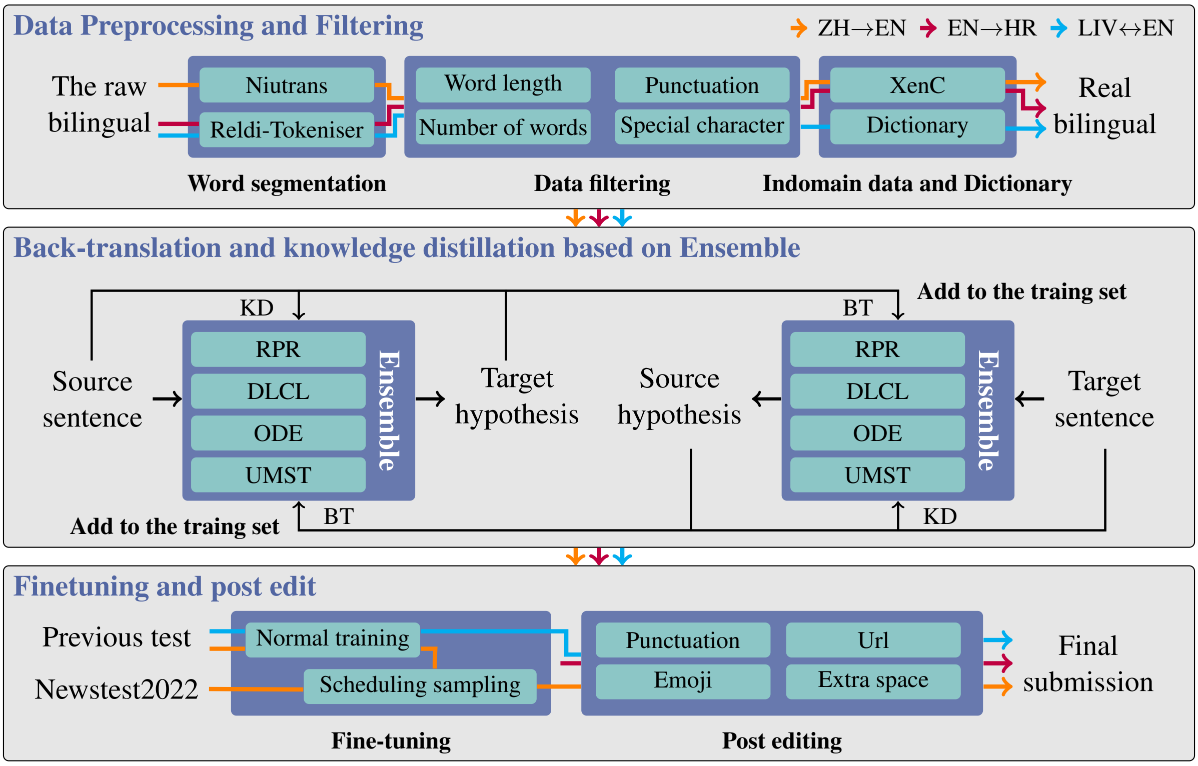

Figure 1: The whole process of our system.

Transformer-DLCL: Build a deeper network with dense inter-layer connections based on the vanilla Transformer, which can increase the information flow at the lower layer.

Transformer-ODE: Based on the relationship between numerical methods of Ordinary Differential Equations(ODEs) and Transformer, A more efficient Transformer calculation method can be obtained by solving ODEs.

Transformer-UMST: Enhance the representation ability of vanilla Transformer by importing sentence-level and word-level information to attention.

2.3 Back-Translation And Knowledge Distillation

Back-translation (Sennrich et al., 2016a) is a popular data augmentation method to improve the performance of machine translation models. We use iterative back-translation(Hoang et al., 2018) based on the in-domain monolingual data to alleviate the domain adaptation problem (Zhang et al., 2020). In addition to use pseudo data directly, we also try Tagged Back-translation (Caswell et al., 2019) in the EN→HR direction. Based on iterative back-translation, we utilize iterative knowledge distillation, which iteratively transforms knowledge (Zhou et al., 2021) from an ensemble model to sub-models based on sequential knowledge distillation (Hinton et al., 2015; Kim and Rush, 2016). We use the following steps for iterative back-translation and knowledge distillation:

1. Select the good quality monolingual data from the source language, filter the data closest to the single domain by XenC toolkit, and obtain bilingual pseudo-data by a forward translation.

2. Filter the data, mix the pseudo data with the training set, and train the back translation model.

3. Search for the best ensemble model combination among all existing models.

4. Use the ensemble model to translate the filtered monolingual data of the target language, and obtain the pseudo data.

We use the `newstest2021` to evaluate our model performance, and repeat steps 1-3(BT) or 2-4(KD) of the above process until the model performance no longer improves[2]. When performing steps 1

[2] It is worth noticing that we do a set of KD after a set of BT, these two methods are combined sequentially

and 4, we use various decoding methods, including beam search, Top-K, and Top-P. In the tasks of ZH→EN, EN→HR, the ratio of raw bilingual data to pseudo data in the training set was about $2 : 1$. In the EN↔LIV tasks, the size of pseudo data is much larger than the raw bilingual data.

2.4 Model Ensemble

The ensemble model can significantly improve the translation quality by considering the output of every single model. We search for the model ensemble combined with the highest BLEU score on the `newstest2021` and use the model ensemble repeatedly in knowledge distillation, back-translation, and fine-tuning. This ensures that we can obtain the optimal models at every stage.

2.5 Fine-tuning

A model trained on a large amount of data may not outperform a model trained on a small amount of in-domain data (Zhou et al., 2021). This phenomenon indicates a mismatch between the domain of the training set and the test set, which becomes an obstacle to performance improvement. For a specific domain, we adjust the size of the training datasets that are more focused on a single domain. However, We find it hard to separate bilingual data into multiple domains. In the case that the training set domain is inseparable, fine-tuning by domain is a reasonable way.

Fine-tuning the model with in-domain data is an effective way to alleviate the domain mismatch (Luong and Manning, 2015; Zeng et al., 2021; Tran et al., 2021). In the ZH→EN direction, we split the test set into four domains according to the domain label in the test set and fine-tune the model in the single domain for each of the four domains.

Take the news domain as an example, and fine-tuning process consists of the following three steps:

1. Translate sentences in the news domain to generate pseudo data by the best ensemble model on `newstest2021`.

2. Fine-tune all sub-models in the ensemble model with pseudo-data, `newstest2020`, and `newstest2021`.

3. Based on the data mentioned in the previous step and `test2022`, we utilize the scheduling sampling strategy (Liu et al., 2021) for fine-tuning further.

For the other three domains, we do not use Step 2 because we do not have any other in-domain data.

2.6 Post Editing

Post editing is a way to correct significant errors in the model translation. In all directions, we insist on using common rules to correct significant errors in the model translation. The errors include:

- Misalignment of symbols and emoji between source and target languages.

- The unnecessary space between Url, HTML, and the text in parenthesis.

In the final submission, this process corrects approximately 2% of all tokens in the test set (most of them are symbols such as extra Spaces between characters).

3 Experiment

3.1 Experiment Settings

The implementation of our models was based on `fairseq` (Ott et al., 2019), and the total data we used were shown in Table 3. In the ZH→EN direction, All models were trained on 8 RTX 2080Ti GPUs, and all other direction models were trained on 4 RTX 3090 GPUs. We used Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.997$ during training. Except for the base model, all of our models adopted the pre-norm structure (Wang et al., 2019). Following the idea of the work (Wang et al., 2019), we adopted the deep and wide model to increase the model capacity (Zhou et al., 2021; Wang et al., 2019). Under the GPU memory constraint, we accumulated the gradient four times and set the batch size to 2048 tokens.

For the deep model, we trained the model for 15 epochs at most. We set max learning rate = 0.002 and warmup step = 8000 for all deep models. All dropout probabilities were 0.1. Meanwhile, we also used FP16 to accelerate the training process. All experiments were evaluated on `newstest2021` using `SacreBLEU` (Post, 2018) in the EN→HR and the EN↔LIV directions. In the ZH→EN direction, we used `multi-bleu.perl`[3]. At last, we introduced a patience factor during decoding, which provided a more flexible decoding depth (Kasai et al., 2022). However, this method led to a significantly slower decoding speed. So we only applied this method to generate the final output.

[3] https://github.com/moses-smt/mosesdecoder

Model	depth	P&R	Head	Hidden Size	Filter Size	Batch size	update freq
BASE	6	✗	8	512	2048	4096	2
RPR	24	✓	8	512	4096	2048	4
DLCL	25	✓	8	512	4096	2048	4
ODE	6	✓	16	1024	4096	1024	16
ODE	12	✓	16	1024	4096	1024	16
UMST	24	✓	8	512	4096	2048	4

Table 2: Explicit information of model structure, P&R indicates whether to use the pre-Norm and relative position representation(RPR)

	Bilingual	Monolingual	
		EN	Other
ZH→EN	12.10M	8M	11M
EN→HR	46M	5M	20M
EN↔LIV	600	0.15M	0.04M

Table 3: The sentences we use in each direction after filtering(The M stands for million).

3.2 ZH→EN

For the ZH↔EN tasks, we only submit in the ZH→EN direction. We filter out the part of data from ParaCrawl, News Commentary V16, WikiMatrix, UN Parallel Corpus V1.0, and the CCMT Corpus as the training set. We use the filtering rules and XenC mentioned above for data filtering. We end up with 12 million raw bilingual data as the training set.

Regarding the multi-domain adaptation, we try an unsupervised data clustering method that uses the pre-trained model's hidden state to do domain classification in the training set. We also use TF-IDF to select keywords from the test set to represent each domain and then use these keywords to select in-domain sentences from constrained data. Unfortunately, the aforementioned methods show poor performance except in the news domain. We find no significant domain difference between the constrained bilingual data and constrained monolingual data we used.

Based on the training set, we train several deep models mentioned above. We use `newstest2020` as the valid set and `newstest2021` as the test set to modify the hyper-parameters and find the optimal ensemble combination. In addition, we also realize a multi-domain translation model which introduces layer-wise Domain Mixing into the vanilla Transformer. However, the model performs poorly on the inseparable domain data, so it is not included in our model ensemble.

For the first round of back-translation, we filter multiple groups' English monolingual data from the News crawl, News discussions, Europarl v10, News Commentary, Common Crawl, and Leipzig Corpora. The amount of data is about 4 million to 8 million sentences. We use the best ensemble model to translate monolingual data with Beam Search, Top-K, and Top-P decoding. By directly concatenating the raw training and pseudo data, we fine-tune the existing model and achieve +0.85 BLEU improvement after the first back-translation iteration, then achieve +0.59 BLEU improvement after the second back-translation iteration.

For knowledge distillation, we filter 3 million monolingual data from News crawl, News Commentary, Common Crawl, Extended Common Crawl, and Leipzig Corpora. We use the best ensemble model to translate the monolingual and obtain the pseudo data, and then fine-tune each sub-model in the ensemble model. We obtain the improvement of 0.16 BLEU points. We select the best four models to construct the ensemble model every time during back-translation and knowledge distillation in both directions.

For fine-tuning, we first do fine-tuning on the news domain to search the optimal hypo-parameters. We use `newstest2019`, `newstest2020` in both ZH→EN and EN→ZH directions as the training set, and obtain the optimal learning rate of 0.001 on `newstest2021`. We achieve +0.93 BLEU improvement in the ZH→EN direction. Then we add `newstest2021` to the training set for fine-tuning. In order to improve the performance of the model in a single domain, we divide the `test2022` into five domains according to the domain labels: news, social, conversational, e-commerce, and biomedical. Finally, we do domain adaptation separately in four domains except biomedical by fine-tuning the model with scheduling sampling.

At last, we use four single-domain models to generate translation in every single domain and use post-processing methods to correct the error in the translation, which brings us +0.81 BLEU improvement in the ZH→EN direction. Our experimental results are shown in Table 4.

3.3 EN→HR

For the EN→HR tasks, we choose ParaCrawl v9, Tilde MODEL corpus, WikiMatrix, and OPUS total of four parallel data corpora of about 90M. We choose all of the News Crawl and Leipzig Corpora for the Croatian monolingual data of about 20M. In order to strengthen the generalization of the model in the social, conversational, and e-commerce domains, we choose the Web and Wikipedia parts from Leipzig Corpora about 10M for the English monolingual data to distill our models.

In addition to the common data filtering process, we calculate the Levenshtein ratio of two adjacent sentences from sorted sentences to remove duplication sentences whose Levenshtein ratio are not less than 0.85. After the data filtering, about 46M sentence pairs are left to build our system. Additionally, we use a shared vocabulary and set the merge operations of BPE to 32K.

Since the domain of the official development set focuses on e-commerce and reviews, we make a general domain test set by ourselves to evaluate the model generalization ability better. To use the Croatian monolingual data, we implement tagged back-translation, which brings us +0.35 BLEU improvement on the official development set and +0.5 BLEU improvement on our test set. We also implement knowledge distillation to use English monolingual data, which brings us +0.3 BLEU improvement on the official development set and +0.14 BLEU improvement on our test set.

We use XenC to select 5M sentence pairs similar to the official development set from the original training set and then fine-tune each model for several epochs. However, we find that not only fine-tuning significantly reduces the model's generalization, but also has a slightly better performance on the official development set and significantly worse performance on our test set. Finally, we put all models together to search for the best ensemble greedily. This method brings us +0.51 BLEU improvement on the official development set and +0.25 BLEU improvement on our test set.

During post-processing, we use rules to adjust the order of punctuation, case inconsistencies and remove some extra spaces, which brings us +0.43 BLEU improvement on the official development set.

3.4 EN↔LIV

For the EN↔LIV tasks, we create a many-to-many multilingual submission for WMT2022. The multilingual submission includes seven language directions, which are CS→EN, ET→EN, LV→EN, EN↔LIV, ET→LIV, and LV→LIV. For CS→EN , we only use ParaCrawl v9 dataset and obtain 50M parallel data after cleaning. After the data filtering, we sample the top 10M data according to a language model trained with CS→EN data. For ET→EN, LV→EN, EN↔LIV, ET→LIV, and LV→LIV directions, we only use OPUS liv4ever v1 dataset, separately obtaining 956, 997, 540, 11420, 10786 parallel data after cleaning. We use the valid set and test set in OPUS liv4ever v1 data set as our valid set and test set. It is worth noting that we delete the same sentences in the test set and the train set.

We use a combination of multiple language directions to train the baseline model, including many-to-many and many-to-one, and find that models trained by all language directions data and many-to-many is 1 BLEU point higher on average than the model trained by several language directions data or many-to-one in the test set. We find that data in different language directions can provide semantic help to EN↔LIV model because CS, LV, ET and LIV are similar languages. So, we select all language directions data and many-to-many to train our model.

We also use pre-trained model for language modeling. Since the constrained track, we choose the AB-Net (Guo et al., 2020) model whose encoder and decoder are initialized with mBERT. However, the performance of AB-Net model was lower than that of the baseline model, so it is not included in our final submission results. The poor performance may be due that: first, the pre-trained model doesn't contain LIV, and second, the parallel data of EN↔LIV is too scarce. This leads to a big challenge to transform the knowledge of the pre-trained model into the EN↔LIV translation model.

Due to the lack of EN↔LIV parallel data, the model cannot capture the alignment information at the word level. Therefore, we make a parallel

System	ZH→EN	EN→HR	EN→LIV	LIV→EN
Baseline	24.27	31.28	4.1	6.57
Deep model	27.2	32.68	5.66	8.79
+ Dict	–	–	8.16	13.79
+ Iteratively BT	28.64	33.03	–	–
+ Iteratively KD	28.8	33.33	8.96	15.99
+ Fine-tuning	29.73	–	10.26	16.89
+ Ensemble	-	33.84	10.48	16.95
+ Post edit	30.54	34.27	–	–

Table 4: BLEU evaluation results on the WMT 2021 ZH→EN, EN↔LIV test sets and WMT 2021 EN→HR development sets.

dictionary of EN↔LIV. First, we use fast_align [4] tool to align the words on the EN↔LIV dataset, and then manually check and modify it. Finally, we obtain a parallel dictionary of EN↔LIV with a dictionary size of 3127. We mix the parallel dictionary and parallel data of EN↔LIV to obtain new parallel data. Then, we train the model by new parallel data and bring us +5 BLEU improvement in the LIV→EN direction and +2.5 BLEU improvement in the EN→LIV direction.

We also use iterative back-translation and iterative knowledge distillation to enhance the model. Since the many-to-many method, the back-translation implemented in the LIV→EN direction is the same as the knowledge distillation in the EN→LIV direction. During the back-translation on the EN→LIV direction, we use 40000 LIV monolingual data from OPUS liv4ever v1 data set. And then during the knowledge distillation on the EN→LIV direction, we use the test set in OPUS liv4ever v1 as in-domain data, and we use the XenC tool to sample 150000 EN monolingual data from Europarl v10 based on in-domain data. We generate pseudo data by using both post-ensemble and ensemble methods. We obtain the improvement of 2.2 BLEU points and 0.8 BLEU points in the back-translation (knowledge distillation) in LIV→EN and EN→LIV. After KD, we use the OPUS liv4ever v1 valid set to fine-tune our models for five epochs with the 0.0003 learning rate and obtain +0.9 and +1.3 BLEU improvement in the LIV→EN and EN→LIV directions.

3.5 Submission Results

The results of our best submissions in four directions this year are shown in Table 5. In the EN→HR direction, our system performed well trained on large amounts of bilingual data. In the EN↔LIV direction, our multilingual model performance is better than the model initialized by the pre-trained model(e.g., mBERT), indicating that the multilingual model has potential in the low resource language. In the ZH→EN direction, KD is not performing well enough in `newstest2021` as usual. On the one hand, this may be related to our data filtering method and the domain changes on the test set; on the other hand, it may be related to our stronger deep model.

Direction	Submission
ZH→EN	26.2
EN→HR	18.1
EN→LIV	12.3
LIV→EN	13.0

Table 5: Our final submission results in four directions.

4 Conclusion

This paper introduces our submissions on WMT22 in four directions. We train our system with constrained data in all directions. The system is constituted by the ensemble model based on multiple deep models.

For training data, we use a softer data filtering method to obtain more data and make the model more robust in the general domain. Based on this data, model performance is better than our last year's systems. We use iterative back-translation and knowledge distillation methods which have been proven to be very effective in the past. In addition, fine-tuning using both normal training and scheduling sampling also achieves good results.

In the ZH→EN direction, we mainly build the news domain translation model. Also, we try the multi-domain data clustering method and

[4] https://github.com/clab/fast_align

multi-domain adaptation method to build the multi-domain model. However, because the sentences in constrained data have no noticeable domain difference, the performance of the above method is not satisfactory. In the EN↔LIV direction, we try the multilingual model and initialization method, which initialize the translation model with the pre-trained model. We find that the multilingual model show more considerable potential than the model initialized with mBERT, even under minimal data.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7747–7763. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 1: Research Papers*, pages 53–63. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648. The Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Junliang Guo, Zhirui Zhang, Linli Xu, Hao-Ran Wei, Boxing Chen, and Enhong Chen. 2020. Incorporating BERT into parallel sequence decoding with adapters. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 18–24. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2020. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1823–1834. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Trans. Assoc. Comput. Linguistics*, 5:339–351.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Dragomir R. Radev, Yejin Choi, and Noah A. Smith. 2022. Beam decoding with controlled patience. *CoRR*, abs/2204.05424.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1317–1327. The Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bei Li, Quan Du, Tao Zhou, Shuhan Zhou, Xin Zeng, Tong Xiao, and Jingbo Zhu. 2021. ODE transformer: An ordinary differential equation-inspired model for neural machine translation. *CoRR*, abs/2104.02308.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266. Association for Computational Linguistics.

Bei Li, Tong Zheng, Yi Jing, Chengbo Jiao, Tong Xiao, and Jingbo Zhu. 2022. Learning multiscale transformer models for sequence generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 13225–13241. PMLR.

Yijin Liu, Fandong Meng, Yufeng Chen, Jinan Xu, and Jie Zhou. 2021. Scheduled sampling based on decoding steps for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3285–3296. Association for Computational Linguistics.

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.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign@IWSLT 2015, Da Nang, Vietnam, December 3-4, 2015*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. *Prague Bull. Math. Linguistics*, 100:73–82.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook ai's WMT21 news translation task submission. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 205–215. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1810–1822. Association for Computational Linguistics.

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. Niutrans: An open source toolkit for phrase-based and syntax-based machine translation. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 19–24. The Association for Computer Linguistics.

Xianfeng Zeng, Yijin Liu, Ernan Li, Qiu Ran, Fandong Meng, Peng Li, Jinan Xu, and Jie Zhou. 2021. Wechat neural machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 243–254. Association for Computational Linguistics.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, Bei Li, Tong Xiao, and Jingbo Zhu. 2020. The niutrans machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, pages 338–345. Association for Computational Linguistics.

Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, Laohu Wang, Jingnan Zhang, Canan Huang, Zhongxiang Yan, Chi Hu, Bei Li, Tong Xiao, and Jingbo Zhu. 2021. The niutrans machine translation systems for WMT21. In *Proceedings of the Sixth Conference on Machine Translation, WMT@EMNLP 2021, Online Event, November 10-11, 2021*, pages 265–272. Association for Computational Linguistics.