The NiuTrans Machine Translation Systems for WMT19

Bei Li[1], Yinqiao Li[1], Chen Xu[1], Ye Lin[1], Jiqiang Liu[1], Hui Liu[1], Ziyang Wang[1], Yuhao Zhang[1], Nuo Xu[1], Zeyang Wang[1], Kai Feng[1], Hexuan Chen[1], Tengbo Liu[1], Yanyang Li[1], Qiang Wang[1], Tong Xiao[12] and Jingbo Zhu[12]

[1]NLP Lab, Northeastern University, Shenyang, China

[2]NiuTrans Co.,Ltd., Shenyang, China

libei_neu@outlook.com, {xiaotong, zhujingbo}@mail.neu.edu.cn

Abstract

This paper described the submission of the NiuTrans neural machine translation systems for the WMT 2019 news translation tasks. We participated in 13 translation directions, including 11 supervised EN↔{ZH, DE, RU, KK, LT}, GU→EN tasks and unsupervised DE↔CS sub-tracks. Our systems were built on Deep Transformer and several back-translation methods. Iterative Knowledge Distillation and ensemble + reranking were also employed to obtain stronger models. Our unsupervised submissions were based on NMT enhanced by SMT. As a result, we achieved the highest BLEU scores in {KK↔EN, GU→EN} directions, ranked 2nd in {RU→EN, DE↔CS} and 3rd in {ZH→EN, LT→EN, EN→RU, EN↔DE} among all constrained submissions.

1 Introduction

Our NiuTrans team participated in 13 WMT19 shared news translation tasks, including 11 supervised and 2 unsupervised sub-tracks. We reused some effective approaches of our WMT18 submissions (Wang et al., 2018), including back-translation by beam search (Sennrich et al., 2016b), Byte Pair Encoding (Sennrich et al., 2016c) and further strengthened our systems by exploiting some new techniques in this year.

For our supervised task submissions, all language pairs shared similar model architectures and training flow. We proposed four novel deep Transformer architectures based on (Wang et al., 2019) as our baseline, which surpass the standard `Transformer-Big` significantly in terms of both translation quality and convergence speed.

As for the data augmentation aspect, we experimented several back-translation methods (Sennrich et al., 2016b), including beam search, unrestricted sampling and sampling-topk proposed by Edunov et al. (2018), to leverage the target-side monolingual data. We also applied Iterative Knowledge Distillation (Freitag et al., 2017) to leverage source-side monolingual data.

Our system also employed conventional combination methods including ensemble and feature-based re-ranking to further improve the translation quality. We proposed a simple greedy search algorithm to find the best ensemble combination effectively and efficiently. Hypothesis combination (Hassan et al., 2018) was also adopted to generate more diverse hypotheses for better reranking.

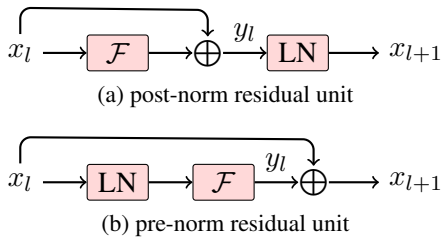
For unsupervised tasks, we mainly investigated the methodology of unsupervised SMT (Artetxe et al., 2019) and NMT (Lample and Conneau, 2019) to build our baselines, then presented a joint training strategy on top of these baselines to boost their performances.

This paper was structured as follows: we described the details of our novel Deep Transformer in Section 2, then in Section 3 we presented an overview of our universal training flow for all supervised language pairs and the unsupervised methods. The experiment settings and main results were shown in Section 4.

2 Deep Transformer

Neural machine translation models based on multi-layer self-attention (Vaswani et al., 2017) has shown strong results on several large-scale tasks. Enlarging the model capacity is an effective way to obtain stronger networks, including widening the hidden representation or deepening the model layers. Bapna et al. (2018) has shown that learning deeper networks is not easy for vanilla Transformer due to the gradient vanishing/exploding problem.

Wang et al. (2019) emphasized that the location of layer normalization played a vital role when



(a) post-norm residual unit

(b) pre-norm residual unit

Figure 1: Examples of pre-norm residual unit and post-norm residual unit. $\mathcal{F}$ = sub-layer, and LN = layer normalization.

training deep Transformer. In early versions of Transformer (Vaswani et al., 2017), layer normalization was placed after the element-wise residual addition (see Figure 1(a)). While in recent implementations (Vaswani et al., 2018), layer normalization was applied to the input of every sub-layer(see Figure 1(b)), which can provide a direct way to pass error gradient from top to bottom. In this way pre-norm Transformer is more efficient for training than post-norm (vanilla Transformer) when the model goes deeper. Remarkably, a dynamic linear combination of previous layers method[1] can further improve the translation quality. Note that we built our deep self-attentional counterparts in pre-norm way as default. In this section we described the details about our deep architectures as below:

Pre-Norm Transformer: In recent Tensor2Tensor implementations[2], layer normalization (Lei Ba et al., 2016) was applied to the input of every sub-layer which the computation sequence could be expressed as: *normalize→Transform→dropout→residual-add*. In this way we could successfully train a deeper pre-norm Transformer within comparable performance with `Transformer-Big` or even better, only one fourth training cost.

Pre-Norm Transformer-RPR: We found Transformer-RPR (Shaw et al., 2018) which simultaneously incorporating relative position information with sinusoidal position encodings for sequences in pre-norm style could outperform the pre-norm Transformer with the same encoder depth. We used clipping distance $k = 20$ with the unique edge representations per layer and head.

Pre-Norm Transformer-DLCL: The Transformer-DLCL employed direct links with all previous layers and offered efficient access to lower-level representations in a deep stack. An additional weight matrix $W_{l+1} \in R^{L \times L}$ was used to weight each incoming layer in a linear manner. This method can be formulated as:

$$\Psi(y_0, y_1 ... y_l) = \sum_{k=0}^{l} W_k^{l+1} LN(y_k) \quad (1)$$

Eq.1 provided a way to learn preference of layers in different levels of the stack, $\Psi(y_0, y_1 ... y_l)$ was the combination of previous layer representation. Furthermore, this method is model architecture free which we can integrate it with either pre-norm Transformer or pre-norm Transformer-RPR for further enhancement. The details can be seen in Wang et al. (2019).

3 System Overview

3.1 Data Filter

Previous work (Junczys-Dowmunt, 2018; Wang et al., 2018; Stahlberg et al., 2018) indicated that rigorous data filtering scheme is crucial, or it will lead to catastrophic loss in quality, especially in EN↔DE and EN↔RU. For most language pairs, we filter the training bilingual corpus with following rules:

- Normalize punctuation with Moses scripts except ZH ↔ EN language pair.
- Filter sentences longer than 100 words, or exceed 40 characters in a single word.
- Filter sentences which contain HTML tags or duplicated translations.
- Filter sentences which both source and target side are identical language.
- Filter sentences whose alignment scores obtained by fast-align[3] are lower than -6.
- The word ratio between source and target must not exceed 1:3 or 3:1.

After several data augmentation methods to leverage monolingual data in order to further boost translation quality, the same data filter strategy was employed.

[1] We called it as Transformer-DLCL in the subsequent sections

[2] https://github.com/tensorflow/tensor2tensor

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

3.2 Back Translation

Back-translation (Sennrich et al., 2016b) is an essential method to integrate the target side monolingual synthetic knowledge when building a state-of-the-art NMT system. Especially for low-resource tasks, it's indispensable to augment the training data by mixing the pseudo corpus with the parallel part, in that the target side lexicon coverage is insufficient, such as EN $\leftrightarrow$ {KK, GU} only consist of 0.11M and 0.5M bilingual data, respectively.

How to select the appropriate sentences from the abundant monolingual data is a crucial issue due to the limitation of equipment and huge overhead time. We adopted training a 5-gram language model based on the mixture of development set and bilingual-target side data to score the monolingual sentence. In addition, considering the impact of sequence length, we set a threshold range from 10 to 50.

Recent work (Edunov et al., 2018) has shown that different methods of generating pseudo corpus made discrepant influence on translation performance. Edunov et al. (2018) indicated that sampling or noisy synthetic data gives a much stronger training signal than data generated by beam or greedy search. This year we attempted several data augmentation method as below:

- Beam search: Generated target translation by beam search with beam 4.
- Sampling: Selected a word randomly from the whole distribution each step which increases the diversity of pseudo corpus compared with beam search, but low precision.
- Sampling Top-K: Selected a word in a restricted way that only top-K (we set K as 10) words can be chosen.

It's worthy noting that experimental results on different language pairs behaved inconsistent: Sampling is more helpful when it comes to low-resource problem like Kazakh, Gujarati and Lithuanian. Oppositely, we observed that language pairs with abundant parallel corpus like ZH$\leftrightarrow$EN are insensitive to sampling method, and slight improvement by restricted sampling which selected from top-10 candidates. We used different strategies to leverage monolingual resource for specific task which we will show detail description in Section4.

3.3 Greedy Based Ensemble

Ensemble decoding is an effective system combination method to boost machine translation quality via integrating the predictions of several single models at each decode step. It has been proved effective in the past few years' WMT tasks (Wang et al., 2018; Deng et al., 2018; Junczys-Dowmunt, 2018; Sennrich et al., 2016a). We enhanced the single model by employing deep self-attentional models. Note that the improvement is poor if the single models performed strong enough and no significant benefits from increasing the participant quantity. So it's necessary to utilize the models sufficiently to search a better combination on development set. We adopted an easily operable greedy-base strategy as below:

Algorithm 1 An Simple ensemble algorithm based on greedy search

Input:
a model list Ω_{cand} sorted by the development scores.

Output:
a final model list Φ_{final}.

1: **for all** $4\_model\_combination$ that $model \in top-8\ models$ **do**
2: Ensemble decoding to get the score
3: **end for**
4: Choose the best 4model combination as the initial Φ_{final} .
5: **repeat**
6: Shift the single model from the rest of Ω_{cand} to the Φ_{final} which performs better when combined with Φ_{final}.
7: **until** there is tiny improvement as increasing the model number

To ensure for the diversity among the candidate models, we constructed the single model from several perspectives, such as different initialization seed, training epochs, model sizes and network architectures described in Section 2. On the development set, this algorithm can consistently improve nearly 1-1.5 BLEU points over the best single model across all the language directions in which we have participated.

3.4 Iterative Knowledge Distillation

A natural idea to further boost the performance of the ensemble model obtained in Section 3.3 is to alternate Knowledge Distillation (Hinton et al.,

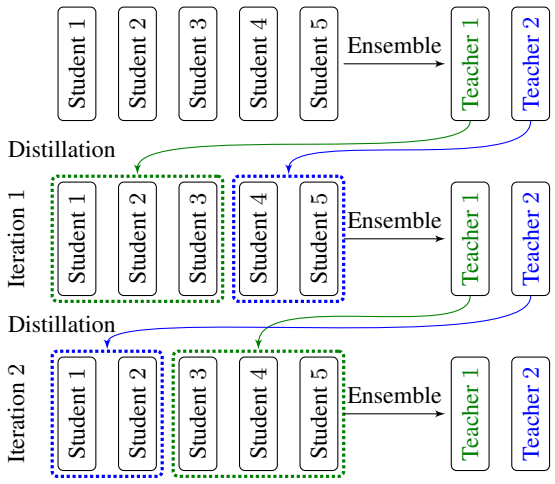


Figure 2: A simple example of Iterative Knowledge Distillation with 5 students, 2 teachers and 2 iterations

2015; Freitag et al., 2017) and Ensemble iteratively. The naive approach started with a list of single model candidates as the students and the best 4 models combination retrieved from Algorithm 1 as the teacher. Sequence-level Knowledge Distillation (Kim and Rush, 2016) was then applied to fine-tune each student model with additional source data. With these enhanced student models, a stronger 4 models combination can be produced through Algorithm 1. We iterated this process until less than 0.1 BLEU improvement on the validation set.

However, in the preliminary experiments we found that such iteration didn't yield good results as we expected. We attributed this phenomenon to the deficiency of model diversity, due to the fact that all students were collapsed to a similar optimum induced by the same teacher they learnt from, which limited the potential gain from iteration. To avoid this, in each step of the iteration, we split the candidates into 4 subsets randomly and assign each subset a distinct teacher model sampled from the top-4 models combinations, then fine-tuned each model within the same subset with its corresponding teacher model. Moreover, we added additional 2M source-side monolingual data in each step to better preserve the model diversity. Figure 2 shows an example.

3.5 Feature Reranking

This year we adopted an hypothesis combination strategy to pick up a potentially better translation from the N-best consisting of several different ensemble outputs. For example we generated 96 hypothesises by 8 different ensemble systems, and set the beam size as 12 during the decoding procedure instead of obtaining all 96 outputs from a single but best ensemble model. The oracle computed by sentence-level BLEU script on development set indicated that hypothesis combination achieved 5 BLEU points superior compared with the single ensemble output. Our reranking features would be described on five aspects as following:

Right-to-Left Models: NMT models generate target translations in a left to right fashion, so it's obviously that incorporating models which generate the target sentences in reverse order can be complementary (Stahlberg et al., 2018). We trained four deep Transformer-DLCL models of different hyper-parameter settings by reversing the target side sentence, followed by ensemble knowledge distillation method to enhance the single model performance. Experiment results showed that the accuracy of the reverse model was extremely necessary, or you may even get worse results.

Target-to-Source Models: Re-scoring between the hypothesis and the source input by target-to-source system. In addition Target-to-Source-Right-to-Left models were needed.

Language Model: We both used 5-gram language model and deep self-attention language model trained on target monolingual data.

Cross-lingual Sentence Similarity: We mixed the source-to-target and target-to-source training data about 1:1 to train a cross-lingual translation model, in order to compute the cosine similarity between the n-best hypothesis and source sentence-level vectors (Hassan et al., 2018) .

Sentence-Align Score: We used fast-align tool to evaluate the alignment probability between source and target.

Translation Coverage: A SMT phrase-table to obtain the top-50 translation for each source-to-target word pair. In this way, the translation coverage score can be easily gained with respect to the dual direction hits in dictionary with length normalization.

We rescored 96-best outputs generated by several ensemble systems using a rescoring model consisting of features above by K-batched MIRA (Cherry and Foster, 2012) algorithm which is widely used in Moses[4].

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

3.6 Unsupervised NMT

We also participated in the unsupervised translation task with only monolingual data provided by WMT organizer. We both attempted the unsupervised SMT and NMT, then combined them for better results. To train SMT models, the unsupervised tuning (Artetxe et al., 2019) was applied to further enhanced the unsupervised SMT system, which employed a small pseudo generated by target-to-source system to adjust weights of source-to-target system. We followed Artetxe et al. (2019) to exploit subword information into unsupervised SMT system, which adding two additional weights to the initial phrase-table. The new features employed a character-level similarity function instead of word translation probabilities, which are analogous to the lexical weightings.

For unsupervised NMT, the techniques we used were based on the recently proposed for unsupervised machine translation (Lample and Conneau, 2019), including proper initialization, leveraging a strong language model and iterative back-translation (Lample et al., 2018). Our systems were initialed by cross-lingual masked language model, which brought significant improvement than cross-lingual embedding method. After that, the standard NMT architecture can be trained by only leveraging monolingual data using combining denoising auto-encoding and iterative back-translation. We adopted two training strategies combining both NMT and SMT models to further enhance our unsupervised system:

- Generate the pseudo corpus by SMT and warmup the NMT models restricted in first 1000 training steps, then we used the pseudo corpus generated by NMT systems for the remained training.

- We mixed the pseudo corpus consisting of NMT and SMT outputs in 1:1 at the beginning, and we increased the ratio of NMT pseudo corpus iteratively until there was no significantly improvement on validation set.

4 Experiments and Results

For all supervised tasks, we used deep self-attentional models as our baseline, and we also experimented the shallow and wide counterparts to verify its effectiveness with the same training corpus. Preliminary experiments indicated that our deep models can even outperform the standard `Transformer-Big` by 0.7-1.3 BLEU scores on different language pairs. All of our experiments employed 25/30 encoder layers and 6 decoder layers, both embedding and hidden size have a dimension of 512, 8 heads for the self-attention and encoder-decoder attention mechanisms. We shared the target-side embedding and softmax matrix. All BLEU scores were reported with mteval-v13a.pl[5]. Next, we will show details for different language pairs in the following subsections.

4.1 Experiment setting

We implemented deep fashion models based on Tensor2Tensor, all models were trained on eight 1080Ti GPUs. We used the Adam optimizer with $\beta_1 = 0.97$, $\beta_2 = 0.997$ and $\epsilon = 10^{-6}$ as well as gradient accumulation due to the high GPU memory consumption. The training data was reshuffled after finishing each training epoch, and we batched sentence pairs by target-side sentences lengths, with 8192 tokens per GPU. Large learning rate and warmup-steps were chosen for faster convergence. We set max learning rate as 0.002 and warmup-steps as 8000 for most of language pairs including EN↔{ZH, RU, KK, LT}. Specifically in EN↔DE task, 16000 warmup-steps achieved better results. During training, we also employed label smoothing with a confidence score 0.9 and all the dropout probabilities were set to 0.1. Furthermore, we averaged the last 15 checkpoints of a single training process for all language pairs. The models were saved and validated every 20 minutes.

4.2 English ↔ Chinese

For ZH ↔ EN system, our parallel corpus included CWMT, wikititles-v1, NewsCommentary-v14, and 30% randomly sampled data from UN corpus. All parallel data were segmented by NiuTrans (Xiao et al., 2012) word segmentation toolkit. After the preprocessing, we trained BPE (Sennrich et al., 2016c) models with $32,000$ merge operations for both sides respectively.

For back-translation, we trained 25-layers transformer models using WMT18 (Wang et al., 2018) training data for both directions. We selected 10M NewsCrawl2018 monolingual data for ZH→EN and the combination of XinHua and XMU data

[5] https://github.com/mosessmt/mosesdecoder/blob/master/scripts/generic/mtevalv13a.pl

for EN→ZH. Experimental results from table 1 showed that generating the pseudo corpus by beam search brought significant improvement on *newstest2018* for ZH↔EN. Meanwhile, for EN→ZH system, additional pseudo corpus[6] by sampling-top10 could obtain +0.7 BLEU points on *newstest2018*, but exhibited negative impact on *newstest2019*.

For ZH→EN, we trained 12 models with different configurations, e.g., layers, batch size, filters, seed, etc. The best performance on our development set *newstest2018* gained +1.6 BLEU improvement than `Transformer-Base`, even +0.7 BLEU higher than `Transformer-Big`. Iterative Knowledge Distillation with 4 teachers, 3 iterations and 1 epoch per iteration gave +1.6 BLEU improvement over the best single model. To this end, almost +4 BLEU improvement was observed on *newstest2019*. Through greedy based ensemble algorithm we selected the best 8-model combination on *newstest2018* and boosted our system performance by +0.8 BLEU. Our reranking model contained 27 features, including 4 L2R-Ensemble, 4 R2L-Ensemble, 4 T2S-Ensemble, 4 T2S-R2L-Ensemble and other features mentioned in Section 3.5.

For EN→ZH, we used the same training setting to obtain our best system. The results after applying each component are reported in Table 1. Surprisingly, adding pseudo corpus hindered our system improvement on *newstest2019*, yet gained +3.7 BLEU improvement on *newstest2018*. One possible explanation is that the construction of test set in this year is different from those in previous years.

	EN-ZH		ZH-EN	
System	18test	19test	18test	19test
Base	38.3	35.7	24.2	-
+Beam	41.3	36.1	26.2	27.0
+S-TopK	42.0	35.9	-	-
Big	43.2	37.1	27.1	27.7
DLCL25RPR	43.9	38.2	27.8	29.1
+EKD	44.6	39.3	29.6	33.0
+Ensemble	45.1	39.8	30.4	34.0
+Reranking	45.6	39.9	30.9	34.2

Table 1: Results for EN↔ZH on official WMT test

[6]We mixed the sampling-topk corpus with the parallel one to fine-tune each single model

4.3 English ↔ German

Table 2 presents the BLEU scores on *newstest2018* and *newstest2019* for EN↔DE tasks. All parallel training data released were used and we adopted the dual conditional cross-entropy method (Junczys-Dowmunt, 2018) to filter out the noise data in ParaCrawl corpus, resulting in 10M bilingual sentences pairs. A joint BPE model was applied in both directions with 32,000 merge operations. Moreover, we selected shared vocabulary for both language pairs.

The target-side monolingual data played an important role in the success of this language pairs. We back-translated 10M monolingual in-domain data from the collection of NewsCrawl2016-2018 filtered by XenC (Rousseau, 2013). We observed that generating pseudo corpus via random sampling is much more effective than beam search with the same volume of monolingual sentences, resulting in 2.5/3.7 BLEU improvement on *newstest2018* for EN→DE and DE→EN respectively. `Transformer-DLCL` with 25 encoder layers and 4096 filters obtained +2.5/1.7 BLEU improvement. Iterative Knowledge Distillation and 8 models combination yielded another +0.8/1.4 BLEU points. Unfortunately, we failed to identify any significant improvement from reranking in terms of validation BLEU scores. Perhaps the features we used were not strong enough to score the n-best properly. It's worthy noting that we re-normalized the quotes in German for the additional 1.8 BLEU improvement on EN→DE.

	EN-DE		DE-EN	
System	18test	19test	18test	19test
Base	41.4	38.3	40.8	42.3
+Paracrawl	43.2	39.5	42.7	44.7
+Beam	44.0	39.7	46.2	45.0
+Sampling	45.7	40.7	46.4	45.5
DLCL25filter4096	48.2	42.7	48.1	47.0
+EKD	48.6	44.2	47.0	47.6
+Ensemble	49.4	45.5	48.4	48.3

Table 2: Results for EN↔DE on official WMT test set

4.4 English ↔ Russian

For EN↔RU, we used the following resource provided by WMT, including News Commentary-v14, ParaCrawl-v3, CommonCrawl and Yandex Corpus. The parallel corpus we used was comprised of 7.66M sentences after removing the bad

case mentioned in Section 3.1. We experimented different BPE code size, ranging from $30,000$ to $80,000$, inspired by the morphology richness of Russian. Considering the efficiency and performance, we finally chose $50,000$ for both directions. We used the same data selection strategy as in EN↔DE and retained only 16M monolingual data from NewsCrawl2015-2018[7]. The selected sentences were then divided into two parts evenly. We generated the pseudo corpus from the first part with beam search sized 4 and trained our NMT models with this corpus together with the parallel ones. The other 8M data were back-translated by random sampling and used to fine-tune each model.

Our final submissions consisted of four Deep Transformer models strengthened by Knowledge Distillation, including DLCL25, DLCL30, DLCL25RPR and DLCL30RPR for EN→RU. The reverse direction contained DLCL25, DLCL25RPR with 4096 filters, DLCL30RPR and DLCL30Filter with 4096 filters. The overall results of our system were reported in Table 3. We observed the same phenomenon as in EN→ZH, where back-translation could yield better results on *newstest2018* but inferior ones on *newstest2019*.

	EN-RU		RU-EN	
System	18test	19test	18test	19test
Base	29.0	27.8	30.9	38.2
+Beam	30.4	28.9	33.0	37.8
+Sampling	32.2	28.3	33.6	37.5
DLCL25RPR	33.4	29.8	34.9	38.9
+EKD	34.1	33.1	35.9	39.5
+Ensemble	35.1	33.8	36.5	40.0
+Reranking	35.5	34.0	36.7	40.0

Table 3: Results for EN↔RU on official WMT test set

4.5 English ↔ Kazakh

This section described our EN↔KK submissions, where we ranked first in both directions. This task was different from the above three language pairs, whose bilingual data, including News Commentary-v14 and English-Kazakh crawled corpus, contained only $97,000$ sentences after filtering. It was not possible to train a large NMT model, with only 2.6/10.1 BLEU on *newsdev2019* as shown in Table 4. We used Russian as the pivotal language to construct the additional EN↔KK bilingual corpus from the crawled RU↔KK corpus as well as the RU↔EN one provided by WMT organizers, resulting in 3.78M high-quality bilingual data[8].

For back-translation, we generated the pseudo corpus via random samplimng from 2M monolingual data selected by Xenc in the collection of Common Crawl, News Commentary, News crawl and Wiki dumps. This pseudo corpus was extremely effective for our system.

For KK→EN system, we adopted the same training procedure, except that we chose 4M English monolingual sentences from News crawl 2015-2018 instead, which consisted of 2M in-domain sentences selected by Xenc and 2M randomly sampled. The detailed experiment results could be seen in Table 4.

	EN-KK		KK-EN	
System	19dev	19test	19dev	19test
Big	2.6	1.9	10.1	11.5
+Pivot	14.9	7.8	23.4	19.8
+Sampling	19.7	10.3	26.2	28.8
DLCL25	20.5	10.7	26.3	29.0
+RPR	-	-	26.6	30.1
+Ensemble	21.3	11.1	26.8	30.5

Table 4: Results for EN↔KK on official WMT test set

4.6 English ↔ Lithuanian

For EN ↔ LT submissions, we used all parallel data available as following: Europarl-v9, ParaCrawl-v3 and Rapid corpus of EU press releases. Through data filtering mentioned in Section3-1, 1.93M bilingual corpus were remained. Lithuanian monolingual resources containing Common Crawl, Europarl, News crawl and Wiki dumps were back-translated to strengthen the EN→LT translation quality by sampling approach. Similarly, News Crawl from 2015 to 2018 were used for reverse direction pair. We adopted the same performance improvement pipelines mentioned above, including various deep self-attentional architectures, greedy based ensemble and knowledge distillation teacher, except for feature reranking. We showed the detailed experiment results in Table 5.

[7]All monolingual data from NewsCrawl2015-2018 were selected for both directions

[8]The training data we used included the pseudo corpus as well as the provided parallel corpus

System	EN-LT 19dev	EN-LT 19test	LT-EN 19dev	LT-EN 19test
Base	18.3	11.5	27.1	29.2
+Pseudo	24.8	13.8	32.2	30.2
DLCL25	25.1	14.0	33.2	31.5
+EKD	26.1	15.0	34.6	33.8
+Ensemble	26.7	15.2	35.1	34.3

Table 5: Results for EN↔LT on official WMT test set

4.7 Gujarati → English

Our GU→EN system was based on Bible Corpus, crawled corpus, OPUS and wikipedia, resulting in totally 0.5M sentence pairs. Additionally, 1.5M HindEnCorp corpus were converted to GU→EN bilingual corpus in terms of the alphabet mapping between Gujarati and Hindi languages. Due to the grammar divergence in two languages, we built a baseline model by bilingual data to score the corpus and removed the bad cases which the scores were inferior to the threshold predefined. Preliminary experiments have shown that data filtering was extremely crucial, for noisy signals in training data did harm to our translation quality. Only 0.98 bilingual pairs were remained after strict data clean, including parallel corpus provided by WMT and pivot pairs originated from HindEnrop corpus.

We used the same approach to select pseudo corpus with KK→EN task, while different generation approach were applied. Our pseudo corpus consisted of two parts: 2M pseudo data by beam search within $(1.2, 10)$ for alpha and beam size respectively and another 1M through randomly sampling. From Table 6 we found that the data quantity was the key factor to enhance the translation quality in this task, and deep DLCL25RPR took full advantage of deep encoder layers to extract more expressive representations.

System	GU-EN 19dev	GU-EN 19test
Base	3.1	3.0
+Pivot	16.3	12.5
+Beam	30.7	19.7
+Sampling	32.5	21.3
DLCL25RPR	34.2	22.8
+EKD	34.9	23.8
+Ensemble	35.5	24.6
+Reranking	36.1	24.9

Table 6: Results for EN→GU on official WMT test set

4.8 German ↔ Czech

This section we show our unsupervised submission on DE↔CS, Table 7 presents the BLEU scores on *newstest2013* and *newstest2019*. We removed the sentences which duplicated, exceptional length ratio. As a result, we used 24.38M Czech monolingual data and 24.36M German monolingual data for each direction respectively from News Crawl2007-2018. All texts were segmented with scripts provides by Moses, and $60,000$ BPE merge operations were applied to unsupervised NMT systems.

We used the Transformer architecture as described in Lample and Conneau (2019) that we revised the `Transformer-Big` with 8 attention heads, learned positional embedding and GELU activation functions. From Table 7 we observed that through several techniques, the unsupervised SMT has gained significantly improvement on *newstest2013* and *newstest2019*. Moreover, leveraging the pseudo corpus generated by unsupervised-SMT system can bring furthermore enhancement though the unsupervised SMT is inferior to NMT system. We both experimented the training strategies mentioned in Section 3.6, and the iterative training method was more efficient. We only fused two single models in decoding procedure and there is no significant improvement on both valid and test sets. Note that, we fixed the quotes in both directions.

System	DE-CS 13test	DE-CS 19test	CS-DE 13test	CS-DE 19test
SMT Base	9.3	7.9	10.5	9.1
+weight-tune	10.0	8.2	11.2	9.5
+sub-word	11.0	9.2	12.4	10.7
+iterative-BT	13.3	11.7	14.7	12.7
NMT Base	17.8	15.8	18.8	16.2
+warmup	20.0	17.4	20.6	17.8
+iteration	20.1	17.6	21.0	18.0
Ensemble	20.3	17.6	21.2	18.1
+fix quotes*	-	18.9	-	17.7

Table 7: Unsupervised results for DE↔CS on official WMT test set, note that the *newstest2019* contains 1997 sentence pairs for both directions

5 Conclusion

This paper described all 13 submissions of NiuTrans systems in WMT19 news shared translation tasks including both supervised and unsupervised

sub tracks, showing that we could adopt an universal training strategies to gain promising achievement. We built our final submissions considering two mainstreams:

- Neural architecture improvement by employing several deep self-attentional based models.
- Taking full advantage of both additional source and target monolingual data by knowledge distillation and back-translation, respectively.

In addition, a greed-based ensemble algorithm was helpful to search a robust combination of models, and we adopted hypothesis combination strategy for more diverse re-ranking. Our systems performed strongly among all the constrained submissions: we ranked first in EN→KK, KK→EN and GU→EN respectively, and almost in the Top-3 of the remained language pairs.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. *arXiv preprint arXiv:1902.01313*.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033, Brussels, Belgium. Association for Computational Linguistics.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, Changfeng Zhu, and Boxing Chen. 2018. Alibaba's neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 372–380, Belgium, Brussels. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Marcin Junczys-Dowmunt. 2018. Microsoft's submission to the wmt2018 news translation task: How i learned to stop worrying and love the data. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 429–434, Belgium, Brussels. Association for Computational Linguistics.

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

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv: Computation and Language*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *arXiv preprint arXiv:1804.07755*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, (100):73–82.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*.

Felix Stahlberg, Adria de Gispert, and Bill Byrne. 2018. The university of cambridge's machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 508–516, Belgium, Brussels. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. 2018. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018. The niutrans machine translation system for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 532–538, Belgium, Brussels. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, and Jingbo Zhu. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Italy, Florence. 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 *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pages 19–24, Stroudsburg, PA, USA. Association for Computational Linguistics.