The AISP-SJTU Translation System for WMT 2022

Guangfeng Liu[1] **Qinpei Zhu**[1] **Xingyu Chen**[2] **Renjie Feng**[1] **Jianxin Ren**[1] **Renshou Wu**[1]
Qingliang Miao[1] **Rui Wang**[2] **Kai Yu**[1,2]
[1]AI Speech Co., Ltd., Suzhou, China
[2]Shanghai Jiao Tong University, Shanghai, China

Abstract

This paper describes AISP-SJTU's participation in WMT 2022 shared general MT task. In this shared task, we participated in four translation directions: English→Chinese, Chinese→English, English→Japanese and Japanese→English. Our systems are based on the Transformer architecture with several novel and effective variants, including network depth and internal structure. In our experiments, we employ data filtering, large-scale back-translation, knowledge distillation, forward-translation, iterative in-domain knowledge finetune and model ensemble. The constrained systems achieve 48.8, 29.7, 39.3 and 22.0 case-sensitive BLEU scores on EN→ZH, ZH→EN, EN→JA and JA→EN, respectively.

1 Introduction

We participate in the WMT 2022 shared general MT task, including English↔Chinese(EN↔ZH) and English↔Japanese(EN↔JA). All of our systems are built with constrained data sets.

For model architectures, we exploit several Transformer variants including transformer-DLCL (Wang et al., 2019), transformer-ODE (Bei Li, 2021), transformer-RPR (Shaw et al., 2018), transformer-Coda (Zheng et al., 2021).

In this year's translation tasks, we mainly employ data filtering (Zhou et al., 2021; Zeng et al., 2021), large-scale back-translation (Sennrich et al., 2015; Lample et al., 2017), knowledge distillation, forward-translation, in-domain knowledge finetune and model ensemble to improve the final model's performance.

For the synthetic data generation, we first exploit large-scale back-translation (Sennrich et al., 2015) method to leverage the target-side monolingual data and the knowledge distillation (Kim and Rush, 2016) to leverage the source-side of bilingual data. To use the source-side monolingual data, we explore forward-translation by ensemble models to get general domain synthetic data.Furthermore, several data augmentation methods are applied to improve the model robustness, including different token-level noise and different sampling methods.

We mainly use three training strategies in the training phase, including the warmup strategy (He et al., 2016) to adjust the learning rate in training, different sampling methods (Holtzman et al., 2019) and the Graduated Label Smoothing (Wang et al., 2020).

In the fine-tuning stage, the test set is clustered into seven categories, and then use the TFIDF-Ngram algorithm (Ramos et al., 2003) to search for similar bilingual and monolingual data in all data according to these seven domains. The monolingual data is then generated using forward translation to generate pseudo-data, and finally fine-tuned together with the searched bilingual data.

We pay more attention to the differences between different models in this year. We compute Self-BLEU (Zhu et al., 2018) from the translations of the models on the valid set to quantify the diversity among different models. To be precise, we use the translation of one model as the hypothesis and the translations of other models as references to calculate an average BLEU score. A lower Self-BLEU means this model is more different from other models.

For ensemble method in every category, the self-BLEU scores of the models are calculated to represent their differences from other models, and according to the self-BLEU scores of the model, the distribution weight when they perform ensemble is calculated through the Softmax-Temperature (Zhu et al., 2018; Cheng et al., 2017). Now seven domain ensemble models are obtained, then use the model for each domain to predict the test set of the corresponding domain separately.

This paper is structured as follows: Sec. 2 describes the novel model architectures. We introduce

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 310 - 317
December 7-8, 2022 ©2022 Association for Computational Linguistics

our system and training strategy in detail in Sec. 3. Experimental settings and results are shown in Sec. 4. We conduct analytical experiments in Sec. 5. Finally, we conclude our work in Sec. 6.

2 Model Architectures

2.1 Model Configurations

As the number of model parameters increases, the model's performance is better, so deeper and wider architectures are used in our system. However, the training of the deep model is unstable, and the loss is not easy to converge. Recent studies (Liu et al., 2020a; Huang et al., 2020) show that the unstable training problem of Post-Norm Transformer can be mitigated by modifying initialization of the network and the successfully converged Post-Norm models generally outperform Pre-Norm counterparts. We adopt the Admin initialization method (Liu et al., 2020b) in our training flows to stabilize the training of deep Post-Norm Transformer. Our experiments have shown that the Post-Norm model has a good diversity compared to the Pre-Norm model and slightly outperform the Pre-Norm model.

In our experiments, we use multiple model configurations with 24/30-layer encoders to build deeper models, and the decoder layers are all 6, and the hidden layer size of all models is 4096. Note that all model configurations above apply to the following variant models.

In addition, We use Transformer-ODE as the baseline model.

2.2 Transformer-RPR

According to the research of (Shaw et al., 2018), adding relative position representation to the self-attention mechanism is used to characterize the distance relationship of elements in the sequence, which can further improve the performance of the machine translation performance. So we incorporate relative position representation (RPR) into the self-attention mechanism on both the Transformer encoder and decoder side. Preliminary experiments demonstrate that only relative key information is enough, and we set the relative window size to 8.

2.3 Transformer-Coda

At the heart of the Transformer architecture is the Multi-Head Attention (MHA) mechanism which models pairwise interactions between the elements of the sequence. Despite its massive success, the current framework ignores interactions among different heads, leading to the problem that many of the heads are redundant in practice, which underutilizes the capacity of the model. To improve parameter efficiency, according to the research of (Zheng et al., 2021), we adopt cascaded head-colliding attention (CODA) which explicitly models the interactions between attention heads through a hierarchical variational distribution.

2.4 Transformer-DLCL

From the perspective of improving the residual network structure, we introduce the DLCL(Dynamic Linear Combination of Layers) method to solve the problem of gradient disappearance or explosion in deep model training. According to the research of (Wang et al., 2019), this DLCL method can effectively improve the performance of deep models.

2.5 Transformer-ODE

According to the research of (Bei Li, 2021), residual networks are an Euler discretization of solutions to Ordinary Differential Equations (ODE), and a residual block of layers in Transformer can be described as a higher-order solution to ODE. Inspired by this work, we adopt ODE to relieve the problem of gradient disappearance or explosion in deep model training.

3 System Overview

3.1 Data Filtering

For ZH-EN and JA-EN language pairs, the filtering rules are as follows:

* Filter out sentences which are longer than 120 words or contain a long word with over 40 characters.

* The word ratio between the source and the target sentence must not exceed 1:3 or 3:1.

* Filter out the sentences that have invalid Unicode characters or HTML tags.

* Filter out the duplicated sentence pairs.

* The number of punctuation difference between the source and the target sentence must not exceed 5.

* The number of digit difference between the source and the target sentence must not exceed 3.

* Filter out sentence pairs in which English sentence has Chinese or Japanese characters.

Besides these rules, several models are trained with constrained corpus for filtering corpus:

* Filter the bilingual corpus with semantic matching models.

* Filter the bilingual corpus with word align models (Dyer et al., 2013) .

* Filter out incomplete English sentences by a discriminative model.

* Filter out incomplete Japanese sentences by a discriminative model.

* Filter out classical Chinese and ancient poetry sentences by a discriminative model.

The monolingual corpus is also filtered with the above rules and models which are suitable for monolingual data. All the above rules and models are applied to synthetic parallel corpus as well.

3.2 Data Augmentation

In the field of NLP text classification, (Wei and Zou, 2019) proposed EDA technology, which can further improve the performance of the model. Inspired by this work, we introduce three operations of synonym replacement, random swap, and random deletion to generate new data. Here we call it **Aug**. Specifically, we choose 15% of sentence pairs to add noise and keep the remaining 85% of sentence pairs unchanged. For a chosen pair, we keep the target sentence unchanged, and perform the following three operations on the source sentence:

* 30% probability of synonym replacement.

* 50% probability of random swap.

* 20% probability of random deletion.

3.3 General Domain Synthetic Data Generation

In this section, we describe our methods for constructing general domain synthetic data. The general domain synthetic data is generated via large-scale back-translation, forward-translation and knowledge distillation to enhance the models' performance for all domains. In the following sections, we elaborate the above techniques in detail.

3.3.1 Back-Translation

Back-translation is the most commonly used data augmentation technique to make good use of the target side monolingual data in NMT (Hoang et al., 2018). Previous work (Edunov et al., 2018) has shown that Different generation strategies have different effects on the quality of generated pseudo-data. After these efforts, we employ the following three generation strategies.

* Sampling Top-K: At each time step, the model generates the probability that each word in the dictionary is likely to be the next word, which we randomly draw from a sample of k = 10 most likely candidates in this distribution. Afterwards, words are generated at the next time step based on the previously selected words.

* Sampling Top-P: Top-P Sampling (Nucleus sampling) is to preset a probability limit p-value, and then arrange all possible words from high to low according to the probability, and select words in turn. Stop when the cumulative probability of a word is greater than or equal to the p-value, and then sample from the already selected words to generate the next word. In our experiments, p is set to 0.9.

* Beam Search: Generate target translation by beam search with beam size 5.

Besides, we also use Tagged Back-Translation (Caswell et al., 2019) in En→Zh, Zh→En, En→Ja and Ja→En.

3.3.2 Forward-Translation

Forward translation refers to the generation of pseudo-data using source-side monolingual data (Sennrich et al., 2015). We use the ensemble model to generate high-quality forward translation data, which can greatly improve the robustness and performance of the model. Forward translation provides steady improvements on all four tracks we competed.

3.3.3 Knowledge Distillation

Knowledge Distillation (KD) has been proven to be a powerful technique for NMT (Kim and Rush, 2016; Wang et al., 2020) to transfer knowledge from the teacher model to student model. Specifically, we use an integrated teacher model to generate target-side pseudo-data from the source side

Domains	Zh	EN	JA
CLIENT	345	364	0
conversational	0	0	502
ecommerce	518	515	453
medicals	277	1454	0
news	505	1910	505
social	503	0	0
t1	0	279	191
t3	0	14343	305
voa	0	19	0

Table 1: The distribution of the blind test sentences in different domains.

of bilingual data. Likewise, Knowledge Distillation has steadily improved on all four tracks we participated in.

3.4 In-domain Finetune

Domain adaption (Luong and Manning, 2015) plays an important role in improving the translation performance. Different from the single domain (news) in previous years, the blind test of this year has shifted to a multi-domain. Firstly, we extract the domain information of every sentence from the "doc" tag in the XML files. The distribution of the blind test sentences in different domains is shown in Table 1. Secondly, we build 1-gram, 2-gram, 3-gram, 4-gram vocab for every domain and adopt the TF-IDF algorithm to extract fine-tuning data for each domain from the whole training set. Thirdly, we finetune the models for each domain using the corresponding domain data, 90% of which is used for training and 10% for validation. Finally, the models of each domain are ensembled and generate translation results of the test sentence in the corresponding domain.

3.5 Softmax-T Self-BLEU based Ensemble

After we get numerous fine-tuned models, we need to integrate them for better results. We improve on the traditional Self-BLEU method (Zhu et al., 2018). First, we calculate the Self-BLEU score of each model in each domain, and then obtain the weight score assigned to each model in each domain through the Softmax-Temperature (Zhu et al., 2018; Cheng et al., 2017). Finally, we use the models of the respective domains to integrate according to the assigned weight scores to generate data for the respective domains.

4 Experiments and Results

4.1 Settings

All our models are implemented based on fairseq 1.0.0. All the models are carried out on 8 NVIDIA V100 GPUs, each of which has 32 GB memory. We use the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$. We use an initial learning rate of 0.001 and use a warm-up strategy during the training phase. We use warm-up step = 4000. The max token is set to 3500 tokens per GPU and we set the “update-freq” parameter in Fairseq to 8. The value of the parameter Dropout is set to 0.3, and the value of Relu-Dropout is set to 0.1. We use the officially required sacreBleu to calculate all our models.

4.2 Dataset

The statistics of all training data is shown in Table 2. For each language pair, the bilingual data is the combination of all parallel data released by WMT22. For monolingual data, we select data from News Crawl, Common Crawl and Extended Common Crawl, and the amount of data after processing is shown in Table 2.

For generating pseudo-data, we use all source monolingual to generate forward translation data and all target monolingual to generate back-translation data. Finally we use the source side of bilingual data to generate knowledge distillation data. We use the methods described in Sec. 3.1 to filter bilingual and monolingual data.

4.3 Pre-processing and Post-processing

Before model training, we pre-process the training data uniformly and customize the processing according to the requirements of each model. Chinese sentences are segmented by Jieba [1], and English, we use Moses [2] for segmentation, and Japanese, we use Mecab [3]. Punctuation normalization is applied in Chinese, English and Japanese data. Truecasing is also applied for all the languages. For all the languages, we use byte pair encoding (BPE) with 40K operations to do subword segmentation (Sennrich et al., 2016).

For the post-processing, we apply de-tokenizing and de-trucaseing on the translation results with the scripts provided in Moses. And we use punctuation normalization for the Chinese and Japanese translations.

[1] https://github.com/fxsjy/jieba

[2] http://www.statmt.org/moses/

[3] https://github.com/taku910/mecab

Data	**En→Zh**	**Zh→En**	**En→Ja**	**Ja→En**
Bilingual Data	25M	25M	26M	26M
Source Mono Data	15M	15M	10M	10M
Target Mono Data	45M	45M	20M	20M

Table 2: Statistics of all training data

System	**En→Zh**	**Zh→En**	**En→Ja**	**Ja→En**
Baseline	45.0	31.0	38.5	23.2
+Back Translation	46.5	33.6	39.5	27.9
+Knowledge Distillation	47.0	34.7	-	-
+Forward Translation	47.4	35.0	40.2	28.2
+$OurIndomainFinetune$	47.5	-	-	**28.9**
+$NormalEnsemble$	48.0	35.7	40.3	28.3
+$OurEnsemble$	**48.1**	**35.9**	**40.4**	28.4

Table 3: Case-sensitive BLEU scores(%) on the four directions *newstest2020*. $OurEnsemble$ method outperform the $NormalEnsemble$. $OurIndomainFinetune$ prove to be effective through validation in the news domain. The final submitted system is a $OurEnsemble$ of all models which are finetuned in each domain using $OurIndomainFinetune$.

BASELINE-MODEL	**En→Zh**	**Zh→En**	**En→Ja**	**Ja→En**
Transformer	44.0	30.5	37.7	22.3
Transformer#	44.3	30.7	37.9	22.6
Transformer-RPR#	44.6	31.0	38.0	22.8
Transformer-Coda#	45.2	31.1	38.1	22.8
Transformer-DLCL#	45.3	31.3	38.2	23.0
Transformer-ODE#	45.0	31.0	38.5	23.2

Table 4: Case-sensitive BLEU scores (%) on the four translation directions *newstest2020* for different architecture in the **baseline** stage. The model with # indicates that the initialized strategy is ADMIN.

BASELINE-MODEL	**En→Zh**	**Zh→En**	**En→Ja**	**Ja→En**
Transformer#	44.3	30.7	37.9	22.6
Transformer-SourceAug#	44.8	31.1	38.2	23.0
Transformer-TargetAug#	44.6	30.6	37.9	22.5
Transformer-BothAug#	44.2	30.5	37.7	22.4

Table 5: Case-sensitive BLEU scores (%) on the four translation directions *newstest2020* for different **data augmentation methods** in the **baseline** stage.

4.4 English→Chinese

The results of En→Zh on *newstest2020* are shown in Table 3. For the En→Zh task, there is a significant improvement in the valid set after adopting our data filtering method. Our baseline score is 45.0. After applying large-scale Back-Translation, we obtain +1.5 BLEU score on the baseline. We further gain +0.5 BLEU score after applying knowledge distillation and +0.4 BLEU from forward-translation.

In preliminary experiments, we select all models distilled from knowledge as our ensemble combinations obtaining +0.6 BLEU score. On top of that, We tried various combinations but couldn't get better results. After using our proposed ensemble strategy, the BLEU score continue to improve by 0.1, which saves a lot of manpower to select models.

4.5 Chinese→English

The Zh→En task follows the same training procedure as En→Zh. As shown in Table 3, we can observe that Back-Translation can improve 2.6 BLEU from baseline. After this, knowledge distillation brings a big improvement, which can increase the BLEU scores from 33.6 to 34.7. Forward translation further boosts the BLEU score to 35.0. Likewise, our ensemble strategy saves a lot of manpower while delivering a small BLEU boost, from 35.7 to 35.9.

4.6 English→Japanese

The results of En→Ja on *newstest2020* are shown in Table 3. The bilingual training data is 31M in total, and we filter it down to 26M sentence pairs through the filtering rules and models described earlier. Because *newstest2020* has detailed results of each step as a reference, we regard the *newstest2021* as the valid set and the *newstest2020* as the test set during training. The 26 million bilingual training data brings the baseline model to 38.5 BLEU score on *newstest2020*.

For the back translation, our training data consists of three parts: 1) 26 million bilingual target data, 2) Japanese monolingual data, 3) Bilingual augmented data. In addition to the 26 million bilingual target sentences, we sample 20 million Japanese monolingual data from the combination of News Crawl and Common Crawl. Then we used the JA-EN ensemble model to generate the hypotheses as the pseudo data set via the Top-k, Top-p and beam search strategy. We randomly extract 2 million from the bilingual data, and add noise to the source sentences as described in Sec 3.2. We improve BLEU by 1.0 with the synthetic back translation training data.

And then, we merge knowledge distillation and forward translation together. We extract 26 million bilingual source sentences and 10 million source monolingual data, and generate pseudo data using the ensemble model of the back translation models. We also use 2 million noised data like used in back translation. We improve the BLEU score from 39.5 to 40.2.

In the ensemble stage, we observe that both of the normal ensemble and our ensemble strategy have only a very slight improvement.

4.7 Japanese→English

The Ja→En task follows the same training procedure as En→Ja. From Table 3, we can observe that back translation can improve the BLEU score from 23.2 to 27.9. The knowledge distillation and forward translation further improve 0.3 BLEU score. In this task, we verify the effectiveness of our in-domain fine tuning method in the News domain. It is worth mentioning that out in-domain fine tuning method brings 0.7 BLEU after forward-translation. For the comparability of the experiment, we still ensemble models which are on the base of forward-translation. We observe that both ensemble methods make results worse.

5 Analysis

To verify the effectiveness of our approach, we conduct analytical experiments on model variants, data augmentation methods, and ensemble strategies in this section.

5.1 Effects of Model Architecture

We conduct several experiments to validate the effectiveness of Transformer (Vaswani et al., 2017) variants we used in the baseline stage and list results in Table 4. Here we take the En→Zh and En→Ja models as examples to conduct the experiments. The results in the Zh→En direction are similar to En→Zh, and the results for the Ja→En direction are similar to En→Ja.

As shown in Table 4, Transformer-DLCL achieves the best performance in En→Zh direction, and Transformer-ODE achieves the best performance in En→Ja direction. For Admin (Liu et al., 2020b) initialization, Transformer#'s BLEU is 0.2 higher than Transformer in En→Zh and En→Ja directions, so this verifies the effectiveness of Admin initialization in deep models.

5.2 Effects of Data Augmentation

For data augmentation, we conduct several experiments based on the Transformer# baseline model in four directions. Specifically, we adopt three methods detailed in Section 3.2:

* **SourceAug** Aug on the source text of the sentence pair.

* **TargetAug** Aug on the target text of the sentence pair.

* **BothAug** Aug on the source text and target text of the sentence pair.

The experimental results are shown in Table 5. Taking En→Zh direction as an example, the SourceAug achieves a BLEU score of 44.8, TargetAug achieves a BLEU score of 44.6, and BothAug achieves 44.2. Results in other directions show the same trend. Therefore, we operate on the source text of sentence pairs in the data augmentation process.

6 Conclusion

This paper summarizes the results of the shared general MT task in the WMT 2022 produced by the AISP-SJTU team. In this shared task, we participated in four translation directions: English→Chinese, Chinese→English, English→Japanese and Japanese→English. We investigate various novel Transformer based architectures to build MT systems. Our systems are also built on several popular data augmentation methods such as back-translation, knowledge distillation, forward-translation and in-domain finetune. In the future, we hope to explore more efficient model architectures and data augmentation techniques in MT systems. We hope that our practice can facilitate research work and industrial applications.

References

Tao Zhou Shuhan Zhou Xin Zeng Tong Xiao Jingbo Zhu Bei Li, Quan Du. 2021. Ode transformer: An ordinary differential equation-inspired model for neural machine translation. *arXiv preprint arXiv:2104.02308*.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. *arXiv preprint arXiv:1906.06442*.

Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2017. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *NAACL*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Vu Cong Duy 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*, pages 18–24.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. 2020. Improving transformer optimization through better initialization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4475–4483. PMLR.

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, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. 2020a. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. 2020b. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*.

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*.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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. *arXiv preprint arXiv:1906.01787*.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. *arXiv preprint arXiv:2005.00963*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational 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. *arXiv preprint arXiv:2108.02401*.

Lin Zheng, Zhiyong Wu, and Lingpeng Kong. 2021. Cascaded head-colliding attention. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 536–549, Online. Association for Computational Linguistics.

Shuhan Zhou, Tao Zhou, Binghao Wei, Yingfeng Luo, Yongyu Mu, Zefan Zhou, Chenglong Wang, Xuanjun Zhou, Chuanhao Lv, Yi Jing, et al. 2021. The niutrans machine translation systems for wmt21. *arXiv preprint arXiv:2109.10485*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.