Transn's Submission for CCMT 2023 Quality Estimation Task

Zeyu Yan(✉), Wenbo Zhang, Qiaobo Deng, Hongbao Mao, Jie Cai, and Zhengyu He

Transn IOL Technology Co., Ltd., Wuhan 430070, Hubei, China
yzyalbert@gmail.com

Abstract. Machine translation quality estimation is a kind of technique to rate and choose the best translation from several translations of text in source language, which is suitable for the application trend of machine translation in international communication. CCMT 2023 Quality Estimate Task focuses on predicting sentence level score for each sentence pair in English to/from Chinese. This paper describes our methods for predicting Human Translation Edit Rate (HTER) score of sentence pairs in both directions. Various source-translation interactive structures are explored to enhance the representations of source and translation to make a more accurate prediction. On account of the well-known powerful ability of pretrained language models, the combinations between varied pretrained language models and interactive structures are also searched to obtain better model performance. Meanwhile, some pretraining methods and the ensemble method are applied to boost the single model performance on Dev and Test data. Our method gets competitive results in both directions with these augmentations.

Keywords: Quality Estimation · Pretrained Language Model · Text Matching

1 Introduction

Machine translation has been widely used nowadays, which requires a quality estimation system to ensure its appropriate usage. However, traditional estimation methods depend on pure human judgment, which is inefficient and resource-intensive. It is important to accomplish automatic translation quality estimation with high efficiency and low cost. Machine translation Quality Estimation (QE) is an automatic evaluation method for choosing the best translation from several machine-translated sentences (*mt*) candidates of one source sentence (*src*) without reference (*ref*).

QE can be realized at sentence level or word level. Sentence level QE aims at predicting a quality score of *mt* sentence and word level QE aims at predicting an OK/BAD label for each word in *mt* to indicate whether this word is translated properly or improperly. The quality estimation task of CCMT 2023 focuses on

Y. Feng and C. Feng (Eds.): CCMT 2023, CCIS 1922, pp. 1–12, 2023.
https://doi.org/10.1007/978-981-99-7894-6_1

English to/from Chinese language directions and predicts Human Translation Edit Rate (HTER) [7,16] score for each *mt* sentence, where HTER measures the number of editing that needs to perform to change *mt* into *ref*. Our method utilizes different pretrained language models (PLMs) to encode sentence pairs and predict HTER score. We also explore new pretraining approaches for quality estimation and the deep interaction between words in *src* and *mt*, which shows significant improvements on this task. In addition, ensemble methods boost model performance not surprisingly.

2 Related Work

Quality Estimation algorithms before deep learning usually extract various features from word, POS tags, syntax, length, and other binary features presenting different aspects in *src* or *mt*. Then a machine learning model uses these features to make predictions. Many machine learning tools are designed for this purpose like QuEst [1] and QuEst++ [2]. Such procedures can be abstracted as the predictor-estimator framework [11].

After the broad usage of neural networks and Transformers [10], deep learning models play the role of feature extractor or even score estimator. DeepQuest [3] and OpenKiwi [4] provide tools for multilevel quality estimation by adopting neural networks. In recent years, TransQuest [5] and COMET [6] have shown significant improvements in quality estimation tasks, by using big PLM as feature extractor and then predicting sentence-level scores or word-level labels. These models encode *src* and its *mt* into high dimensional embedding vectors through multilingual PLMs, then input another neural network to get quality predictions. Diverse multilingual PLMs can perform as sentence encoder such as mBERT [8], XLM-RoBERTa [21], InfoXLM [20], mDeBERTa [19], RemBERT [22] and so on.

3 Feature-Enhanced Estimator for Sentence-Level QE

3.1 Model Architecture

Quality estimation task involves measuring the editing number in *mt* word by word, which expects the meaning of each word to be translated precisely and unambiguously. PLMs like BERT [8] and RoBERTa [9] have revealed marvelous capability of representation and feature extraction in natural language processing. Consequently, our model designs several feature interactive structures between *src* and *mt* after being encoded by multilingual PLM encoders. Both *src* and *mt* are concatenated and input into PLM to get their last hidden state representations (s_1, s_2, ..., s_m) and (t_1, t_2, ..., t_n), where m and n are word numbers of *src* and *mt* as shown in Eq. 1 and Eq. 2. Afterward, three kinds of feature interactive modules are completed on top of PLMs as illustrated in Fig. 1.

$$outputs = PLM\_encoder([src; mt]) \tag{1}$$

$$[s_1, s_2, ..., s_m], [t_1, t_2, ..., t_n] = split\_src\_mt(outputs) \tag{2}$$

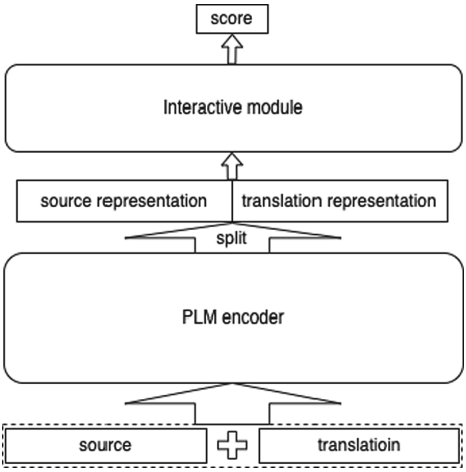

Fig. 1. Model architecture with interactive module

Simple Interactive Module (SIM). Following the settings of CometKiwi [12], we use the mean pooling to generate the vector representations of *src* and *mt*. Then the element-wise product and subtraction results between *src* and *mt* representations are concatenated together with the vector representations of *src* and *mt*. A MLP module predicts HTER score based on these representations as follows:

$$s_{mean} = mean\_pooling([s_1, s_2, ..., s_m]) \tag{3}$$

$$t_{mean} = mean\_pooling([t_1, t_2, ..., t_m]) \tag{4}$$

$$hter = MLP([s_{mean}; t_{mean}; s_{mean} \odot t_{mean}; |s_{mean} - t_{mean}|]) \tag{5}$$

RNN-Based Interactive Module (RIM). Recurrent neural network (RNN) is a popular model in the natural language processing area, which can capture intra-sentence dependency in a text sequence. Therefore, we use the Bidirectional LSTM [15] (BiLSTM) layer to encode the word-level output hidden states of the encoders for further reinforcing the feature interactions and predicting HTER scores between *src* and *mt* as follows:

$$lstm\_output = BiLSTM([s_1, s_2, ..., s_m; t_1, t_2, ..., t_m]) \tag{6}$$

$$hter = MLP(mean\_pooling(lstm\_output)) \tag{7}$$

Multilevel Interactive Module (MIM). Inspired by ESIM [14] and RE2 [13], the cross attention between *src* and *mt* reflects the similarity between words in different languages. Considering that different layers in an encoder catch different levels of information of *src* and *mt*, we determine to combine these two kinds of

features to strengthen the representations of *src* and *mt*. Specifically, a weighted sum of layer-wise hidden states of *src* or *mt*

$$s^l = mean\_pooling([s_1^l, s_2^l, ..., s_m^l]), for\, each\, layer\, l \tag{8}$$

$$t^l = mean\_pooling([t_1^l, t_2^l, ..., t_n^l]), for\, each\, layer\, l \tag{9}$$

$$s_{mix} = \sum_{l=1}^{L} w_s^l \cdot s^l,\ where \sum_{l=1}^{L} w_s^l = 1 \tag{10}$$

$$t_{mix} = \sum_{l=1}^{L} w_t^l \cdot t^l, where \sum_{l=1}^{L} w_t^l = 1 \tag{11}$$

with a total layer number L (Eq. 8–Eq. 11) is concatenated with its cross attention layer output (Eq. 12–Eq. 16) to transform into a rich representation through MLP layer.

$$e_{ij} = s_i^T t_j \tag{12}$$

$$s_i^{ca} = \sum_{j=1}^{n} \frac{exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})} t_j,\ \forall i \in [1, 2, ..., m] \tag{13}$$

$$t_j^{ca} = \sum_{i=1}^{m} \frac{exp(e_{ij})}{\sum_{k=1}^{m} exp(e_{kj})} s_i, \forall j \in [1, 2, ..., n] \tag{14}$$

$$s_{ca} = mean\_pooling([s_1^{ca}, s_2^{ca}, ..., s_m^{ca}]) \tag{15}$$

$$t_{ca} = mean\_pooling([t_1^{ca}, t_2^{ca}, ..., t_n^{ca}]) \tag{16}$$

Features of *src* and *mt* are fused separately (Eq. 17, Eq. 18) for further combination.

$$s_{comb} = MLP([s_{ca}; s_{mix}; |s_{ca} \odot s_{mix}|; s_{ca} - s_{mix}]) \tag{17}$$

$$t_{comb} = MLP([t_{ca}; t_{mix}; |t_{ca} \odot t_{mix}|; t_{ca} - t_{mix}]) \tag{18}$$

Then HTER score is computed by another MLP layer.

$$hter = MLP([s_{comb}; t_{comb}]) \tag{19}$$

This module is illustrated in Fig. 3.

Loss Selection. We choose the Mean Squared Error (MSE) loss for finetuning models. Besides, the square root of the original HTER score is set as label when using MSE loss since the distribution of HTER score in training data is dense in lower score range (see Fig. 2).

$$loss = MSE(hter_{pred}, \sqrt{hter_{true}}) \tag{20}$$

Taking the square root of HTER score can increase the divergence among different scores and make model predict easier.

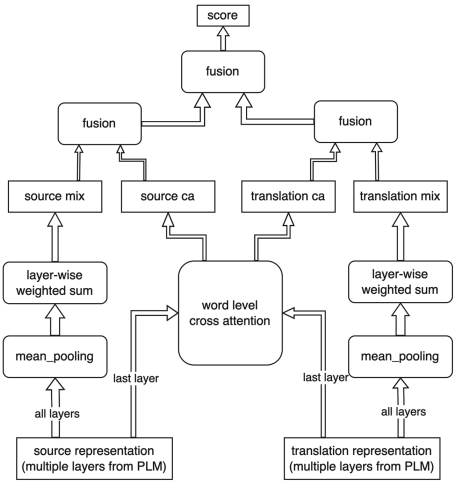

Fig. 2. The ranking of HTER scores on En-Zh training data in ascending order

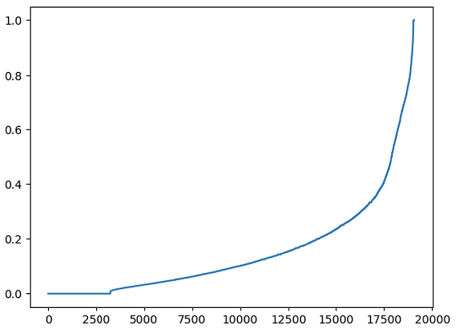

Fig. 3. The structure of Multilevel Interactive Module

3.2 Pretraining Corpus Generation

Transformer models often need plenty of training data for supervised learning so we need to generate more data for pretraining quality estimation models from bilingual parallel corpus. Motivated by [17], to generate *mt* data, we use several open-source neural machine translation models, including mbart50-m2m [28], Helsinki NLP Opus-en-zh [29], Helsinki NLP Opus-zh-en [30], M2M100 [31] and NLLB [32], which are provided by huggingface.com.

Some simple rules like length restriction and removal of special characters together with LaBSE [18] model are used to filter semantically unrelated sentence pairs from the original parallel corpus and we sample part of the filtered parallel corpus due to the computing power limit. For each sentence pair in sampled parallel corpus, we translate both sentences from one language to another, from which we can get two pairs of *src* and *mt*. Then we compute the HTER score between *mt* and *ref* by using sacrebleu [27] and filter out those sentence pairs with HTER score greater than 1.

After these steps, we generate nearly three million of (*src*, *mt*, *ref*, *hter score*) tuples as pretraining dataset before finetuning on the quality estimation dataset. These data are utilized to do two separate pretraining tasks on encoders. The first task *mlm* is mask language modeling on pretraining dataset, identically in BERT-like models [8,9,21]. Both *src* and *ref* are concatenated and masked partial words randomly as input into encoder, and encoder predicts what the masked tokens should be. Another pretraining task *hter* is predicting HTER scores on pretraining dataset based on concatenated *src* and *mt* with MSE loss.

3.3 Model Ensemble

Since several models are finetuned to predict HTER scores, we need to do model filtering and ensemble for better results developed on Dev set. For model filtering, we select the models with a higher Pearson correlation coefficient. Then we integrate the results from different models by two means. The first way is simply averaging models' scores to get the final prediction score for each *mt*

$$score = \frac{1}{model\_num} \sum_{i=1}^{model\_num} score_i \tag{21}$$

and the second method is to calculate the weighted sum of predictions of one sample, where the weights are the performance rank of each model on Dev set for each language pair.

$$total = \sum_{i=1}^{model\_rank} i \tag{22}$$

$$score = \sum_{i=1}^{model\_rank} \frac{i}{total} score_i \tag{23}$$

4 Experiments

4.1 Datasets

We use the data from CCMT 2023 Machine Translation Quality Estimation tasks for model finetuning while restoring the tokenized sentences to the original forms by removing spaces. The bilingual parallel dataset for CCMT 2023 Chinese-English translation task is filtered and used for pretraining as described in Sect. 3.2. The QE dataset statistics are shown in Table 1.

Table 1. QE data statistics of CCMT Quality Estimation Task

language pair	Train	Dev
En-Zh	19060	1528
Zh-En	13983	1412

4.2 Training and Evaluation

Training and model Python codes are completed with PyTorch [24] 1.13 and transformers [25] 4.26.1. All models are trained on NVIDIA GeForce RTX 3090 24G for both pretraining on pretraining corpus and finetuning on quality estimation data. Models are finetuned by AdamW [23] optimizer with learning rate of 1e−5, max sequence length of 128, batch size of 16, and 3 epochs. The model checkpoint with the best Pearson correlation coefficient calculated by SciPy [26] on Dev set is selected for Test set. Chinese-to-English and English-to-Chinese directions are trained and evaluated separately. We also evaluate finetuned results with and without pretraining.

4.3 Results and Analysis

The following Tables 2 and Table 3 show the Pearson correlation coefficient results of different encoders with different interactive modules on Dev set. These models are pretrained on *hter* task at first. When the interactive module fuses more diverse features from encoder, the Pearson correlation coefficient grows even with different encoders with respect to models with encoder only.

However, the pretraining approaches described in Sect. 3.2 have positive or negative effects on different encoders shown in Table 4 when using encoders and multilevel interactive module. The *hter* improves the performance of all models which reveals that the amount of training data is the key to superior quality estimation. The *mlm* boosts the performance of most models except InfoXLM and RemBert. A probable explanation for this is that InfoXLM is pretrained in a contrastive learning way [20] while mask language modeling is harmful to the original capability. And RemBert has a similar reason for this phenomenon [22].

Table 2. Results on En-Zh Dev set of interactive modules in Sect. 3.1

Language pair	En-Zh			
interactive module	no module	SIM	RIM	MIM
XLM-RoBERTa-Large	0.3526	0.3160	0.3315	0.3626
InfoXLM-Large	0.4067	0.4454	0.4588	0.4603
mBERT	0.2633	0.3545	0.3369	0.3345
RemBert	0.3100	0.3632	0.3676	0.3309
mDeBERTa-v3-base	0.3548	0.4002	0.4005	0.4025

Table 3. Results on Zh-En Dev set of interactive modules in Sect. 3.1

Language pair	Zh-En			
interactive module	no module	SIM	RIM	MIM
XLM-RoBERTa-Large	0.4855	0.4462	0.4994	0.4486
InfoXLM-large	0.4641	0.5299	0.5252	0.4763
mBERT	0.4235	0.4505	0.4369	0.4557
RemBert	0.4573	0.5178	0.5193	0.5046
mDeBERTa-v3-base	0.4872	0.4920	0.5064	0.4918

Table 4. Results on Dev set of pretraining methods

Language pair	En-Zh			Zh-En		
pretraining method	mlm	hter	mlm+hter	mlm	hter	mlm+hter
XLM-RoBERTa-Large	0.4147	0.3664	0.4563	0.5199	0.4834	0.5538
InfoXLM-Large	0.1061	0.4564	0.1526	0.0462	0.5168	0.1416
mBERT	0.3016	0.3379	0.3738	0.4825	0.4603	0.4952
RemBert	0.1536	0.3804	0.3646	0.2695	0.4961	0.4902
mDeBERTa-v3-base	0.3825	0.3962	0.4120	0.4681	0.4894	0.4827

In addition, not all models benefit from the selection of loss defined in Eq. 20. Table 5 gives the comparison between two loss functions when using different encoders and multilevel interactive module. The same model on different language pairs shows opposite effects which suggests that the loss function must be carefully designed.

Table 5. Results on Dev set of loss functions

Language pair	En-Zh		Zh-En	
loss type	MSE	MSE w/ sqrt	MSE	MSE w/ sqrt
XLM-RoBERTa-Large	0.3664	0.3943	0.4834	0.4540
InfoXLM-large	0.4564	0.4546	0.5168	0.5070
mBERT	0.3379	0.3025	0.4603	0.4754
RemBert	0.3804	0.3402	0.4961	0.5193
mDeBERTa-v3-base	0.3962	0.3766	0.4894	0.4978

4.4 Model Ensemble

According to the experiments of single model, we do a grid search on combinations of different models with high Pearson correlation coefficient on Dev set. Table 6 compares the results by using two different ensemble methods as described in Sect. 3.3.

Table 6. Results on Dev and online Test of ensembles

Language pair	En-Zh			Zh-En		
dataset	Dev	Online Test	Offline Test	Dev	Online Test	Offline Test
average	0.4712	0.5059	–	0.5743	0.5307	–
rank-weighted sum	0.4747	0.5120	0.3668	0.5690	0.5357	0.4687

We can see that allocating distinct weights to different models can perform better which implies that more adjustments on weights may surpass the current results. And our results are competitive even on offline Test set.

5 Conclusion

This paper describes our method for CCMT 2023 Quality Estimation Task on both English-to-Chinese and Chinese-to-English directions. With the help of PLMs and specially designed pretraining tasks, we can get better representations of *src* text and its *mt* text. The application of pretraining on generated HTER data helps model predict more accurate scores while *mlm* pretraining harms some

PLMs' ability to make better predictions. Both pretraining on HTER data and *mlm* way can further improve model performance in most cases. Since the prediction of HTER score requires taking account of the editing on word level, interactions between source words and translation words need to be modeled deeply to reflect the change of consistency semantically and grammatically. Experiment results show that our models can generate scores of higher Pearson correlation coefficient with true HTER scores when making deeper and multiple levels of representations interactive between *src* and its *mt*. When combining different levels of interactive modules on different language pairs, different PLMs show better or worse results which suggests that it is hard to design a universal module for various language pairs but using interactive modules always boosts the performance. We will leave it as future work. Moreover, the change of loss function increases Pearson correlation coefficient significantly on Dev set. Also, the ensemble method based on the rankings of models makes the prediction scores more relevant which indicates that it has much potential to explore the best combination of weights. Our models achieve competitive results in both English-to-Chinese and Chinese-to-English directions.

Acknowledgement. The participants would like to express heartfelt thanks to the committee and the organizers of the CCMT Quality Estimation Task. We would also like to show our gratitude to the reviewers for their invaluable suggestions. This work is supported by Transn IOL Technology Co., Ltd.

References

1. Specia, L., Shah, K., De Souza, J.G., Cohn, T.: QuEst-A translation quality estimation framework. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 79–84, August 2013
2. Specia, L., Paetzold, G., Scarton, C.: Multi-level translation quality prediction with quest++. In: Proceedings of ACL-IJCNLP 2015 System Demonstrations, pp. 115–120, July 2015
3. Ive, J., Blain, F., Specia, L.: DeepQuest: a framework for neural-based quality estimation. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 3146–3157, August 2018
4. Kepler, F., Trénous, J., Treviso, M., Vera, M., Martins, A.F.T.: OpenKiwi: an open source framework for quality estimation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 117–122. Association for Computational Linguistics, Florence, Italy (2019)
5. Ranasinghe, T., Orasan, C., Mitkov, R: TransQuest: translation quality estimation with cross-lingual transformers. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 5070–5081. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020)
6. Rei, R., Stewart, C., Farinha, A.C., Lavie, A.: COMET: a neural framework for MT evaluation. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 2685–2702. Association for Computational Linguistics (2020)

7. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, pp. 223–231 (2006)
8. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: 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, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
9. Liu, Y., et al.: RoBERTa: A Robustly Optimized BERT Pretraining Approach (2019)
10. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
11. Kim, H., Jung, H.Y., Kwon, H., Lee, J.H., Na, S.H.: Predictor-estimator: neural quality estimation based on target word prediction for machine translation. ACM Trans. Asian Low-Resource Lang. Inf. Process. (TALLIP) **17**(1), 1–22 (2017)
12. Rei, R., et al.: CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In: Proceedings of the Seventh Conference on Machine Translation (WMT), pp. 634–645. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid) (2022)
13. Yang, R., Zhang, J., Gao, X., Ji, F., Chen, H.: Simple and effective text matching with richer alignment features. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4699–4709). Association for Computational Linguistics, Florence, Italy (2019)
14. Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for natural language inference. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1657–1668. Association for Computational Linguistics, Vancouver, Canada (2017)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
16. Specia, L., Farzindar, A.: Estimating machine translation post-editing effort with HTER. In: Proceedings of the Second Joint EM+/CNGL Workshop: Bringing MT to the User: Research on Integrating MT in the Translation Industry, pp. 33–43 (2010)
17. Tuan, Y.-L., El-Kishky, A., Renduchintala, A., Chaudhary, V., Guzmán, F., Specia, L.: Quality estimation without human-labeled data. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 619–625. Association for Computational Linguistics (2021)
18. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT sentence embedding. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 878–891. Association for Computational Linguistics, Dublin, Ireland (2022)
19. He, P., Gao, J., Chen, W.: DeBERTaV3: improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In: The Eleventh International Conference on Learning Representations (2022)

20. Chi, Z., et al.: InfoXLM: an information-theoretic framework for cross-lingual language model pre-training. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3576–3588. Association for Computational Linguistics (2021)
21. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451. Association for Computational Linguistics (2020)
22. Chung, H.W., Fevry, T., Tsai, H., Johnson, M., Ruder, S.: Rethinking embedding coupling in pre-trained language models. In: International Conference on Learning Representations (2020)
23. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2018)
24. Pytorch Homepage. https://pytorch.org/. Accessed 11 Sept 2023
25. Huggingface-transformers Homepage. huggingface.co/docs/transformers/index. Accessed 11 Sept 2023
26. SciPy Homepage. https://scipy.org/. Accessed 11 Sept 2023
27. Sacrebleu Homepage. https://github.com/mjpost/sacrebleu. Accessed 11 Sept 2023
28. mBART50 mt Model Page. https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt. Accessed 11 Sept 2023
29. Helsinki-NLP's en-zh mt Model Page. https://huggingface.co/Helsinki-NLP/opus-mt-en-zh. Accessed 11 Sept 2023
30. Helsinki-NLP's zh-en mt Model Page. https://huggingface.co/Helsinki-NLP/opus-mt-zh-en. Accessed 11 Sept 2023
31. M2M100 Model Page. https://huggingface.co/facebook/m2m100_418M. Accessed 11 Sept 2023
32. NLLB-200 Model Page. https://huggingface.co/facebook/nllb-200-distilled-600M. Accessed 11 Sept 2023