HW-TSC's Participation at WMT 2020 Quality Estimation Shared Task

Minghan Wang[1], **Hao Yang**[1], **Hengchao Shang**[1], **Daimeng Wei**[1], **Jiaxin Guo**[1], **Lizhi Lei**[1], **Ying Qin**[1], **Shimin Tao**[1], **Shiliang Sun**[3], **Yimeng Chen**[1], **Liangyou Li**[2]
[1]Huawei Translation Services Center, Beijing, China
[2]Huawei Noah's Ark Lab, Hong Kong, China
[3]East China Normal University, Shanghai, China
{wangminghan,yanghao30,shanghengchao,weidaimeng,guojiaxin1,
leilizhi,qinying,taoshimin,chenyimeng,liliangyou}@huawei.com
slsun@cs.ecnu.edu.cn

Abstract

This paper presents our work in the WMT 2020 Word and Sentence-Level Post-editing Effort Quality Estimation (QE) Shared Task. Our system follows standard Predictor-Estimator architecture, with a pre-trained Transformer as the Predictor, and specific classifiers and regressors as Estimators. We integrate Bottleneck Adapter Layers in the Predictor to improve the transfer learning efficiency and prevent from over-fitting. At the same time, we jointly train the word- and sentence-level tasks with a unified model with multitask learning. Pseudo-PE assisted QE (PEAQE) is proposed, resulting in significant improvements on the performance. Our submissions achieve competitive result in word/sentence-level sub-tasks for both of En-De/Zh language pairs.

1 Introduction

Quality Estimation (QE) assesses the translation quality of machine translation (MT) system output when ground truth reference is not available (Specia et al., 2013, 2018). For the word-level QE task, participants are required to tag tokens and gaps of the translation output from an unknown MT system with OK and BAD, as well as tokens in the source with the same tags. The result is measured by Matthews Correlation Coefficient (MCC). For the sentence-level task, participants are required to predict the Human Translation Error Rate (HTER) scores of the machine translation outputs and the result is evaluated in terms of the Pearson's correlation coefficient.

Our team participates in some of the sub-tasks in two language pairs (En-De and En-Zh) (Specia et al., 2020). With regard to the En-De track, at word-level, our model achieves the MCC score of 0.5828 on the target side, and 0.5234 on the source side; at sentence-level, our model ranks the first place with a Pearson R score of 0.7583. With regard to the En-Zh track, we only submit the target side of word-level sub-task, and achieves a MCC score of 0.5872.

Our system is implemented with fairseq (Ott et al., 2019) (for En-De track) and THUMT (Zhang et al., 2017) (for En-Zh track). We extend the original Transformer (Vaswani et al., 2017) model to fit the QE task, and leverage transfer learning to fine-tune the model with the task specific dataset (Dai and Le, 2015; Howard and Ruder, 2018; Radford et al., 2018). The contributions of our work are as follows:

- We follow the Predictor-Estimator (Kim and Lee, 2016; Kim et al., 2017; Wang et al., 2018; Li et al., 2018; Kepler et al., 2019) architecture and build a unified QE model based on the standard Transformer MT model.
- Bottleneck Adapter Layers (Houlsby et al., 2019; Yang et al., 2020) are integrated into the model for efficient transfer learning.
- We propose the Pseudo-PE assisted QE (PEAQE) method which effectively improve the performance.

The architecture of our model is shown in Figure 1.

2 Task Description

A more detailed description of the word- and sentence-level QE tasks is given in this section.

2.1 Word-Level

Word-level QE estimates the translation quality by producing a sequence of tags for both source and target. For target sentences, both tokens and gaps are required to be tagged with OK or BAD, while for source sentences, only tokens are tagged

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1056–1061

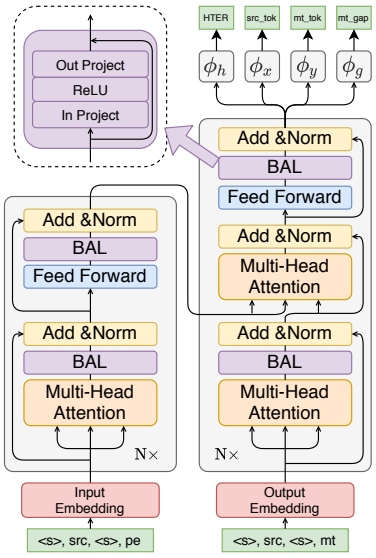

Figure 1: This figure shows the architecture of our model, where SRC and Pseudo-PE are concatenated as the encoder input, a copy of SRC and MT are concatenated as the decoder input. The output feature are passed through four linear layers to make prediction for four tasks.

with OK or BAD. This is usually modelled as a sequential labelling problem. The tag of token indicates whether the word is correctly translated or not, while the tag of gap indicates whether one or more words are missing here. The number of total tags for each MT sentence is $2N + 1$, where N is the number of tokens in the sentence.

The evaluation metrics of the word-level task is the Matthews Correlation Coefficient (MCC), an appropriate measurement for unbalanced labels. MCC is defined as follows:

$$S = \frac{TP + FN}{N}$$
$$P = \frac{TP + FP}{N}$$
$$MCC = \frac{\frac{TP}{N} - SP}{\sqrt{SP(1-S)(1-P)}}, \quad (1)$$

where TP, TN, FP and FN represent for true positives, true negatives, false positives and false positives respectively; and N is the number of instance (Fonseca et al., 2019).

2.2 Sentence-Level

The sentence-level QE predicts the Human Translation Error Rate (HTER) (Specia et al., 2018) of given translation outputs. HTER is an edit-distance measure, calculating the ratio between the number of edits (insertions/deletions/replacements) required and the reference translation length, namely HTER = (Insertions + Delations + Replacement) / Reference Words. In the QE task, where references are not available, HTER is roughly an estimation. As HTER is a real value ranging from 0 to 1, it can be modeled as a regression task. The evaluation metrics of the sentence-level task is the Pearson correlation coefficient.

Attributes	En-De	En-Zh
# Instance	7,000	7,000
# SRC Token	11,4980	115,585
# MT Token	112,342	120,015
% MT Token BAD	28.15	54.33
% MT Gap BAD	4.60	8.04
% SRC Token BAD	26.95	53.60
BLEU (MT, PE)	49.40	30.40
μ(HTER)	0.3181	0.6280
σ(HTER)	0.2017	0.2040

Table 1: The statistics of the training set for both language pairs.

3 Dataset

The dataset contains 7,000 sentences for the training set, 1,000 for the dev and 1,000 for the test. Except from tags and HTER scores (labels), the dataset also provides post-edit (PE) text, as the reference for generating QE labels. Note that this data is also used in the Automatic Post Editing task in WMT 2020. Detailed statistics of the dataset is listed in Table 1, with some metrics of the source (SRC) and translation (MT). The proportion of BAD tags agains OK tags is imbalanced, especially for Gap tags.

Apart from the brief descriptive statistics listed in the table, our in-depth investigation on the provided dataset unveils some interesting findings:

- Different from the dataset in WMT 2019 QE task (Fonseca et al., 2019), which is sampled from IT domain, the dataset this year is collected from Wikipedia. Therefore, mixing data from previous years may not help to improve this year's performance.

- The BLEU score (Papineni et al., 2002) for 2020 dataset is significantly lower than that of 2019, indicating much more operations are required to edit the translation outputs into the references. As a result, the distribution

of labels for 2020 dataset is changed as well when comparing with that of last year.

Unlike a standard translation task, where various data augmentation techniques, such as back-translation (Sennrich et al., 2016), are available, QE task can hardly be improved with data augmentation, as it requires unbiased and high-quality PEs to generate tags and HTER scores. Meanwhile, the change of dataset domain makes it impossible to enlarge the dataset by incorporating the dataset of last year. Facing this challenging task, we propose the PEAQE method, which will be further explained in details in the following section.

4 Model

4.1 Unified QE Model

Our model follows the Predictor-Estimator (Kim et al., 2017; Kepler et al., 2019) architecture. The Predictor is considered as a feature extractor, which can be a pre-trained language model (LM) or a translation model. In our implementation, we use the standard Transformer without the causal mask as the Predictor, which is pre-trained with dataset in news translation task of WMT 2019 En-De and WMT 2020 En-Zh. The Estimator can be task specific classifiers which map the extracted features into the target space, and can be modelled by several fully connected layers. We use a unified QE model to solve both word- and sentence-level tasks by building three classifiers and a regressor to make prediction on SRC tag, MT token tag, MT gap tag and HTER score, respectively.

We define the encoder and decoder of the Transformer as functions f and g; SRC and MT text as X and Y; tags of SRC, MT token and MT gap as V_x, V_y, V_g; and HTER score as V_h. The representation $\mathbf{H}_X$ and $\mathbf{H}_Y$ are obtained by passing the X and Y into the encoder and decoder respectively:

$$\mathbf{H}_X = f(X) \tag{2}$$

$$\mathbf{H}_Y = g(Y, \mathbf{H}_X). \tag{3}$$

For a translation model, we usually append and prepend the special token $\langle s \rangle$ to the SRC and TGT text. Here we follow the same rule and thereby the lengths of SRC and MT embeddings are $M+1$ and $N+1$ respectively. Meanwhile, we append and prepend a special label $\langle pad \rangle$ to the original label sequence during training, but loss of the padded label is not computed. All predictions are obtained by performing specific transformations $\phi_\cdot$ on the hidden stats:

$$\hat{V}_x = \phi_x(\mathbf{H}_X) \tag{4}$$

$$\hat{V}_y = \phi_y(\mathbf{H}_Y) \tag{5}$$

$$\hat{V}_g = \phi_g(\mathbf{H}_Y) \tag{6}$$

$$\hat{V}_h = \phi_h(h_{Y,0}). \tag{7}$$

Note that the regressor ϕ_h only takes the embedding of the MT's first token to make predictions, similar to the usage of [CLS] in BERT (Devlin et al., 2018).

For all classification tasks, we use weighted cross entropy as the loss function, and the weight is calculated as follows: $w_i = \frac{\sum |C_i|}{|C_i|}$, which is the inverse of the proportion of the instance with class C_i. We use weighted cross entropy because labels are highly imbalanced, and the loss should be manipulated with the weight. For sentence-level QE, we use mean squared error (MSE) as the loss function, which is quite intuitive.

The model is trained under the multi-task learning framework by summing up the loss of all subtasks with specific weights:

$$\begin{aligned}\mathcal{L} =& \lambda_h \sqrt{(\hat{V}_h - V_h)^2} \\ &- \sum_{\tau \in \{x,y,g\}} \lambda_\tau \log P(V_\tau | X, Y),\end{aligned} \tag{8}$$

where $\{x, y, g\}$ represents for classification tasks and h represents for regression task, and λ is the weight of loss for a specific task. Although the multi-task setting could improve the overall performance, the evaluation is performed separately, it means we can train models that are optimized for the specific task, which can be achieved by giving larger weight to the loss of that task.

4.2 Bottleneck Adapter Layer

As mentioned in the previous section, the provided training set is relatively small, make the model to be easily over-fitted if all weights are updated. Therefore, we decide to integrate the Bottleneck Adapter Layers (BAL) (Houlsby et al., 2019) while keeping parameters of original Transformer fixed (Yang et al., 2020).

BAL can be easily implemented with two fully-connected layers with a non-linear activation, and is embedded into the Transformer with residual connections after the self-attention layer and the FFN layer, respectively.

In the experiment, we find that the bottle with a "thick" neck ("like FFN layers in the Transformer with higher dimension in the middle part") could further improve the performance without seriously sacrificing training efficiency. More specifically, we tested three neck sizes, i.e. thin, same and thick. The thin and same neck basically reaches 97%-99% of performance compared with training the full Transformer without using BAL, which yields the same result with (Houlsby et al., 2019). By increasing the parameter size of BALs, we find that the performance also increases linearly, finally reaching the pick of 104% of the baseline performance with the neck to have 2 $\times$ hidden size.

4.3 Pseudo-PE Assisted QE

QE tags can be generated with post-edits (PEs) or reference (REF) of MT. In this dataset, PE is provided, and QE tags are generated accordingly, if PE can be directly used to assist the model learning QE tags, the training efficiency will be dramatically increased. Inspired by the Pseudo-PE technique proposed in the (Kepler et al., 2019), we hope to fully leverage PE, for example, integrating them as part of the network input. However, for the test set, where PEs are not available, we must find an alternative approach. So, we made following assumption:

$$\delta(\text{MT}, \text{REF}) \approx \delta(\text{MT}, \text{PE}) + \delta(\text{PE}, \text{REF}), \quad (9)$$

where δ is any distance measurement function. In the equation, PE is regarded as an intermediate node between MT and REF. Under such assumption, if we could find any translation that is better than MT, although not as good as PE, the translation can also be used as a substitute of PE, denoted as PE'. we call this method as Pseudo-PE assisted QE (PEAQE). Finding PE' is relatively easy when we could access unconstrained resources. Using an APE system or a robust online translation system to produce better translation outputs are two feasible approaches. After comparing the BLEU scores of the training set between many online translation services and an APE system trained by us, we decide to use Google Translate outputs as the Pseudo-PE. The BLEU score for official MTs and Google MTs in the dev set are 50.9/ 67.8 for En-De, and 22.62/41.77 for En-Zh, indicating that Google MT outputs, with a high quality, could be used as Pseudo-PEs in the testing phase.

To leverage PEs, we simply concatenate them with the SRCs and encoded them via an encoder. We find that using the features of SRC text from the encoder could not produce acceptable predictions. Therefore, we decide to concatenate SRCs with MTs again on the decoder side, and use the decoder to extract features for both of them. More formally:

$$\mathbf{H}_{[X;Z]} = f([X; Z]) \quad (10)$$

$$\mathbf{H}_{[X;Y]} = g([X; Y], \mathbf{H}_{[X;Z]}), \quad (11)$$

where Z represents for official PEs (training) or Pseudo-PEs (testing). Finally, the hidden state $\mathbf{H}_{[X;Y]}$ is sliced with the max length of X, and recover back to $\mathbf{H}_X$ and $\mathbf{H}_Y$, which are used as in the original model. Official PE and Pseudo-PE can be used respectively during training and testing to assist the model to make better prediction.

5 Experiment

Our experiments of all sub-tasks for En-De and part of sub-tasks for En-Zh trak are performed on the WMT 2020 dataset. The model without Pseudo-PE assistance is considered as the baseline.

5.1 Experimental Settings

Our models are implemented with fairseq (Ott et al., 2019) and THUMT (Zhang et al., 2017). The fairseq version mainly deals with En-De tasks thanks to the pre-trained models trained in WMT 2019 news translation task. The En-Zh pre-trained model is implemented with THUMT and is trained in WMT 2020 news translation task by our team. For the En-De model, input and output embeddings are shared, therefore SRC and TGT text can be conveniently concatenated. For the En-Zh model, vocabulary is not shared, when creating the input sequence, we firstly pass tokens of English (SRC) and Chinese (MT and PE) with specific word embedding layer respectively, and than, concatenate the hidden states of them accordingly. The number of parameters of the En-De and En-Zh models are 270M and 353M, respectively. The batch size used for training is 32. We use Adam (Kingma and Ba, 2015) to optimize parameters with learning rate of 1e-4 without any scheduler. Note that when dealing with labels of sub-tokens, for each token, we only assign the first sub-token with the label and subsequent sub-tokens are assigned with the dummy pad labels, which keeps the distribution of labels unchanged. Our QE models are trained on a Nvidia Tesla V100 GPU, and converge within 5 epochs.

Lang	Model	MCC-MT	MCC-SRC	Pearson-R
En-De (Dev)	Baseline	44.50	32.46	55.26
	+ PEAQE	60.05	45.31	71.69
	+ Ensemble (14)	64.70	51.17	73.33
En-De (Test)	+ Ensemble (14)	58.28	52.34	75.83
En-Zh (Dev)	Baseline	43.06	-	-
	+ PEAQE	57.90	-	-
	+ Ensemble (5)	59.28	-	-
En-Zh (Test)	+ Ensemble (5)	58.72	-	-

Table 2: The experimental results of our model, where the baseline model is introduced in section 4.1. The evaluation results of the test set are from the official leader-board.

5.2 Experimental Results

Table 2 shows the experimental results on the dev and test sets. The performance of the baseline model is relatively poor. By leveraging PEAQE, the model achieves much better performance, demonstrating that integrating PE directly into the QE model could effectively assist the prediction. With PEs, the model can receive stronger supervision signal and is actually learning the procedure done by the tagging script, making the entire learning process easier. However, we clearly understand that the performance of PEAQE strongly depends on the quality of Pesudo-PEs, which becomes another problem that should be solved in the future.

Here is another interesting finding during our experiment. Initially, we also performed experiments with mBERT (Devlin et al., 2018) and XLM (Conneau and Lample, 2019) but not producing desirable results. The reason might be size of the dataset. We find that performing transfer learning with pre-trained NMT model on the limited size QE dataset is more effective than other pre-trained multilingual LMs. We consider that NMT models are naturally fit for MT related tasks because of the learned prior between bilingual text, which might not be captured by multilingual LMs where text in different languages are trained independently.

6 Conclusion

We present our works for WMT 2020 QE shared task. The experimental results demonstrate that performing transfer learning with a pre-trained NMT model on the QE task is effective. Compared to only using SRC and MT text, we propose PEAQE which could significantly improve the performance of the model. But generating reliable Pseudo-PEs that are compatible with QE tasks remains a problem that would be investigated in our future works.

References

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7057–7067.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Erick R. Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 3: Shared Task Papers, Day 2*, pages 1–10.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. *arXiv preprint arXiv:1902.00751*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Fabio Kepler, Jonay Trénous, Marcos V. Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019. Unbabel's participation in the WMT19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation, WMT*

2019, Florence, Italy, August 1-2, 2019 - Volume 3: Shared Task Papers, Day 2, pages 78–84.

Hyun Kim and Jong-Hyeok Lee. 2016. Recurrent neural network based translation quality estimation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 787–792.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568.

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

Maoxi Li, Qingyu Xiang, Zhiming Chen, and Mingwen Wang. 2018. A unified neural network for quality estimation of machine translation. *IEICE TRANSACTIONS on Information and Systems*, 101(9):2417–2421.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

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

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. *Quality Estimation for Machine Translation*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Lucia Specia, Kashif Shah, José GC De Souza, and Trevor Cohn. 2013. QuEst-A translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84.

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

Jiayi Wang, Kai Fan, Bo Li, Fengming Zhou, Boxing Chen, Yangbin Shi, and Luo Si. 2018. Alibaba submission for WMT18 quality estimation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 809–815.

Hao Yang, Minghan Wang, Ning Xie, Ying Qin, and Yao Deng. 2020. Efficient transfer learning for quality estimation with bottleneck adapter layer. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, EAMT 2020, Lisbon, Portugal, 3 - 5 November, 2020*, pages 29–34.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huan-Bo Luan, and Yang Liu. 2017. THUMT: an open source toolkit for neural machine translation. *CoRR*, abs/1706.06415.