Quality Estimation Using Dual Encoders with Transfer Learning

Dam Heo[*] **WonKee Lee**[*] **Baikjin Jung**[*] **Jong-Hyeok Lee**[*,†]
[*]Departmet of Computer Science and Engineering,
[†]Graduate School of Artificial Intelligence,
Pohang University of Science and Technology (POSTECH), Republic of Korea
{dammy, wklee, bjjung, jhlee}@postech.ac.kr

Abstract

This paper describes POSTECH's quality estimation systems submitted to Task 2 of the WMT 2021 quality estimation shared task: Word and Sentence-Level Post-editing Effort. We aim to improve the stability of recently proposed quality estimation models, which usually have a single encoder based on the self-attention mechanism to simultaneously process both of the two input data: a source sequence and its machine translation; considering that such models are not propped up by pre-trained language models' monolingual word representations, which are generally accepted as reliable representations for various natural language processing tasks. Therefore, our model first uses two pre-trained monolingual encoders and then exchanges their output information through two additional cross attention networks. According to the official leaderboard, our systems outperform the baseline systems in terms of the Matthews correlation coefficient for machine translations' word-level quality estimation and in terms of the Pearson's correlation coefficient for sentence-level quality estimation by 0.4126 and 0.5497 respectively.

1 Introduction

Quality estimation (QE) is the task of estimating the quality of given machine translations without regard to their reference translations (Blatz et al., 2004; Specia et al., 2009). As reference translations are generally unavailable in real life, QE should help to treat output texts of machine translation (MT) systems. QE can be categorized into several subtasks, and this round of the WMT QE task has three subtasks, yet we focus on Task 2: Word and Sentence-Level Post-editing Effort. In Task 2, while sentence-level QE aims to predict the Human-Targeted Translation Edit Rate (HTER, Snover et al. 2006), which measures the edit distance between an MT output (*mt*) and its human post-edited text (*pe*), word-level QE aims to predict OK–BAD tags for three sequences of tokens: the sequence of words in a source text (*src*) depending on whether they are correctly translated referring to *mt*; the sequence of words in *mt* depending on their correctness; and `<GAP>` tokens, which each represent the gap between two adjacent words, depending on the existence of any missing words (Specia et al., 2020).

As other recent QE models do, our method also applies transfer learning, considering that pre-trained language models (LM) have been successfully applied to various natural language processing (NLP) tasks including QE; many previous studies (Fomicheva et al., 2020; Hu et al., 2020; Wu et al., 2020; Lee, 2020; Moura et al., 2020; Nakamachi et al., 2020; Rubino, 2020) that apply pre-trained LMs to QE have adopted multilingual or cross-lingual LMs such as multilingual-BERT (Pires et al., 2019), XLM (Conneau and Lample, 2019), and XLM-R (Conneau et al., 2020) to process the two input data *src* and *mt*. Such cross-lingual LMs have a single Transformer (Vaswani et al., 2017) encoder using only the self-attention mechanism to create vector representations of the input data and predict the labels. However, it appears possible to further improve the stability of those models, considering that they are not propped up by pre-trained LMs' monolingual word representations, which are generally accepted as reliable representations for various NLP tasks.

With this background, we propose a QE model that has two separate pre-trained encoders that each produce monolingual representations of *src* and *mt*, respectively. On top of each encoder, we add a cross attention network for the learning of the cross-lingual context between *src* and *mt*; these networks will produce two sets of cross-lingual representations for QE. We conduct simple experiments to compare the performance of our systems and ensembles of them with that of the baseline systems and that of other submitted systems for

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 920–927
November 10–11, 2021. ©2021 Association for Computational Linguistics

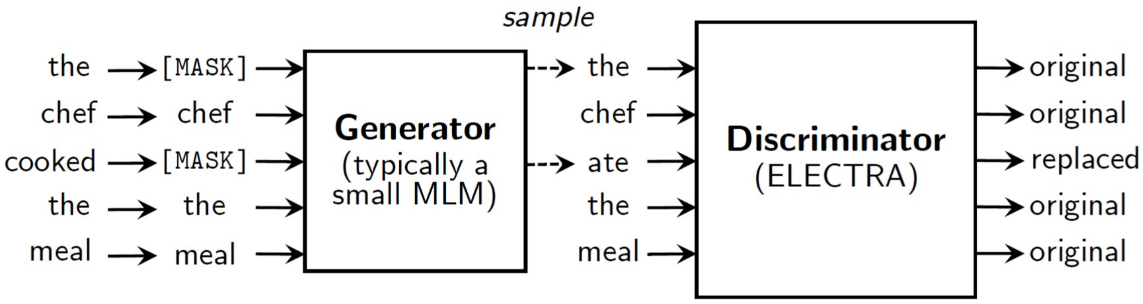

Figure 1: A diagram depicting the training task of ELECTRA (Clark et al., 2020)

Task 2. Experimental results imply that although our systems do not always outperform the baseline systems, they do in terms of the Matthews correlation coefficient (MCC) for *mt*'s word-level QE and in terms of the Pearson's correlation coefficient (PCC) for sentence-level QE by 0.4126 and 0.5497 respectively.

2 Related Work

Because our model does not confine its monolingual encoders to specific pre-trained LMs, all pre-trained LMs can be considered relevant. Among them, most of the recently proposed pre-trained LMs are denoising autoencoders, of which the pre-training task is usually to select about 15% of tokens in unlabeled input sequences and apply the attention mechanism to those tokens (Yang et al., 2019) or is to mask certain tokens (Devlin et al., 2019) and then restore them. However, in our experiments, our systems use ELECTRA (Clark et al., 2020). ELECTRA introduces "replaced token detection" as an additional pre-training task and let the language model learn to distinguish between real input tokens and specious but artificially generated tokens. In detail, when the generator network predicts the tokens in the masked positions, some of the predicted tokens are corrupted, and then this output sequence is fed into a Transformer-based discriminator network, which predicts whether each token in the fed sequence is the same as the original one or is a replaced one (Figure 1). We suppose that this process and QE are similar to each other in that both of them predict the soundness of the given tokens, so ELECTRA would be one of the most appropriate pre-trained LMs for our QE model's monolingual encoders, especially for Task 2.

3 Model Description

Our model uses two ELECTRAs: one ELECTRA[1] that is pre-trained with English corpora and the other ELECTRA[2] pre-trained with German corpora. Figure 2 depicts the overall structure of our model.

3.1 Dual Monolingual Encoders

Our model has dual encoders: a pre-trained English ELECTRA processing *src* and a pre-trained German ELECTRA processing *mt*. These encoders will produce reliable monolingual representations of *src* and *mt* respectively to provide these refined representations to the upper cross attention networks.

Because unlike other pre-trained QE models that have a single Transformer encoder being fed with the concatenation of *src* and *mt*, our model lets the two different encoders process the two input data respectively, we exclude the segment embeddings, which are used to distinguish one language from another, and assign different positional embeddings to each input data. In addition, for sentence-level QE, *mt*'s special token `<CLS>` is used to predict the HTER.

3.2 Cross Attention Networks

We attach a cross attention network to each pre-trained encoder; it learns the cross-lingual context information by using the encoders' refined monolingual representations of the two input data. Although the structure of a cross attention network is identical to that of the encoders, the cross attention networks are not pre-trained, so we train them after the random initialization of their parameters. We

[1] https://huggingface.co/google/electra-base-discriminator

[2] https://huggingface.co/german-nlp-group/electra-base-german-uncased

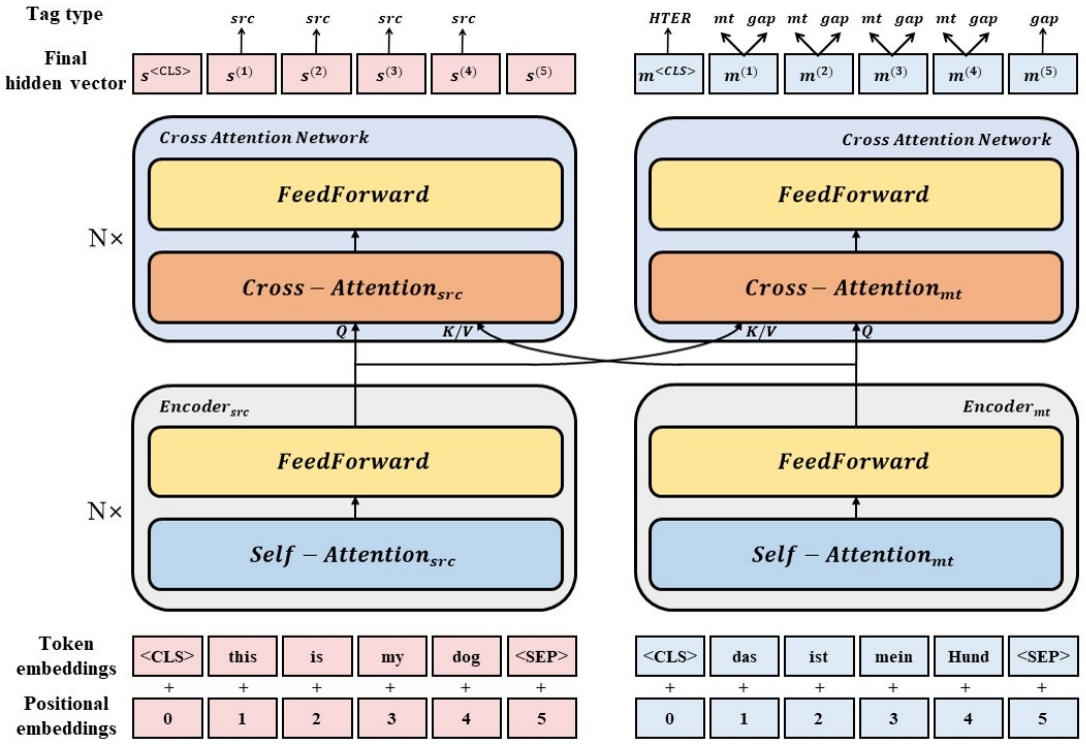

Figure 2: The overall structure of our model

find that applying transfer learning to the cross attention networks is not available due to the absence of pre-trained language models that are pre-trained to perform cross attention on cross-lingual input data by using one side as a query vector and the other side as both a key vector and a value vector just as the Transformer decoder performs multi-head attention on the output of the Transformer encoder.

3.2.1 Sentence-Level QE

To predict the HTER for sentence-level QE we employ the final hidden vector $m^{\texttt{<CLS>}}$ of the *mt*-side cross attention network, which is the final representation of the <CLS> token, as the representation of the *mt* sequence as a whole. After this representation passes through double linear layers with the GELU (Hendrycks and Gimpel, 2016) activation function, the HTER of the given *mt* sentence is estimated as follows.

$$\begin{aligned} \mathbf{l} &= \mathbf{W}_h m^{\texttt{<CLS>}} + \mathbf{b}_0 \\ \hat{y}_{\text{HTER}} &= \mathbf{w}_h^T \text{GELU}(\mathbf{l}) + b_1 \end{aligned} \quad (1)$$

We have trainable parameters $\mathbf{W}_h \in \mathbb{R}^{H \times H}$, $\mathbf{w}_h \in \mathbb{R}^H$, $\mathbf{b}_0 \in \mathbb{R}^H$, and $b_1 \in \mathbb{R}$; H denotes hidden vectors' dimension.

We use the mean squared error of this estimator, that is, the difference between the estimated HTER $\hat{y}_{\text{HTER}}$ and the ground truth HTER value y_{HTER}, as the training loss

$$\mathcal{L}_{\text{HTER}} = \text{MSE}(\hat{y}_{\text{HTER}}, y_{\text{HTER}}). \quad (2)$$

3.2.2 Word-Level QE

***src*-Side Prediction** We use the final hidden vector $s^{(i)}$ $(i \in \{1, ..., |S|\}$, where $|S|$ is the number of tokens in the tokenized *src* sequence) of the *src*-side cross attention network corresponding to each token in *src* to predict OK or BAD in the token's position. After each of these representations passes through a linear layer, the word-level probability of the corresponding token being OK or BAD is predicted with a sigmoid activation function:

$$P_s^{(i)} = \text{sigmoid}(\mathbf{w}_s^T s^{(i)}), \quad (3)$$

where $\mathbf{w}_s \in \mathbb{R}^H$ is a trainable parameter.

We use the binary cross-entropy loss function; we also introduce an extra hyperparameter k_s to prevent our model from being overfitted because the statistics of the ratio between the number of OK tags and that of BAD tags in our training data (Table 1) can misguide the model to have the tendency

		# of words	OK	BAD
Artificial data	*src*	54.6M	34.3M (62.81%)	20.3M (37.19%)
	mt.word	50.5M	29.7M (58.82%)	20.8M (41.18%)
	mt.gap	53.5M	50.6M (94.53%)	2.9M (5.47%)
WMT 21 train	*src*	115K	84K (73.05%)	31K (26.95%)
	mt.word	112K	81K (71.85%)	32K (28.15%)
	mt.gap	119K	114K (95.41%)	5K (4.59%)
WMT 21 dev	*src*	16K	12K (74.21%)	4K (25.79%)
	mt.word	16K	12K (72.49%)	4K (17.51%)
	mt.gap	17K	16K (95.83%)	0.7K (4.17%)

Table 1: Statistics of QE datasets used in our experiments.

of outputting OK even when it should output BAD. The *src*-side loss is as follows:

$$\mathcal{L}_{src} = \frac{1}{|S|}\sum_{i=1}^{|S|}\Big\{k_s y_s^{(i)} log P_s^{(i)} + (1 - y_s^{(i)}) log(1 - P_s^{(i)})\Big\}, \quad (4)$$

where $y_s^{(i)}$ is a ground-truth OK–BAD tag.

***mt*-Side Prediction** We use the final hidden vector $m^{(i)}$ ($i \in \{1, ..., |M|\}$, where $|M|$ is the number of tokens in the tokenized *mt* sequence) of the *mt*-side cross attention network corresponding to each token in *mt* to predict OK or BAD in the token's position. We estimate the probabilities of the word tokens

$$P_m^{(i)} = \text{sigmoid}(\mathbf{w}_m^T m^{(i)}), \quad (5)$$

where $\mathbf{w}_m \in \mathbb{R}^H$ is a trainable parameter.

We also use the final hidden vector $m^{(j)}$ ($j \in \{1, ..., |M|+1\}$ including the vector in the position of the last `<SEP>` token to predict OK or BAD for the last `<GAP>` token. We estimate the probabilities of the `<GAP>` tokens

$$P_g^{(j)} = \text{sigmoid}(\mathbf{w}_g^T m^{(j)}), \quad (6)$$

where $\mathbf{w}_g \in \mathbb{R}^H$ is a trainable parameter.

The *mt*-side prediction loss equals the sum of the losses for word tokens and `<GAP>` tokens:

$$\mathcal{L}_{mt} = \mathcal{L}_m + \mathcal{L}_g, \quad (7)$$

where

$$\mathcal{L}_m = \frac{1}{|M|}\sum_{i=1}^{|M|}\Big\{k_m y_m^{(i)} log P_m^{(i)} + (1 - y_m^{(i)}) log(1 - P_m^{(i)})\Big\}, \quad (8)$$

and

$$\mathcal{L}_g = \frac{1}{(|M|+1)}\sum_{j=1}^{|M|+1}\Big\{k_g y_g^{(j)} log P_g^{(j)} + (1 - y_g^{(j)}) log(1 - P_g^{(j)})\Big\}, \quad (9)$$

$y_m^{(i)}$ and $y_g^{(j)}$ being ground-truth OK–BAD tags for a word token and a `<GAP>` token respectively and hyperparameters k_m and k_g being introduced for the same reason why we introduce k_s.

Finally, we define the word-level loss and the overall QE loss of our model as follows.

$$\mathcal{L}_{\text{word}} = \mathcal{L}_{src} + \mathcal{L}_{mt} \quad (10)$$

$$\mathcal{L}_{\text{QE}} = \mathcal{L}_{\text{word}} + \mathcal{L}_{\text{HTER}} \quad (11)$$

4 Experiments

4.1 Datasets

In our experiments, we used the eSCAPE (Negri et al., 2018) dataset, which is a collection of data triplets each of which is composed of *src*, *mt*, and *pe*; we used this dataset to make artificial QE training data. In this process, to make our artificial data have a similar statistics as those of WMT 2021's official training data, we filtered eSCAPE triplets according to various criteria, such as the sequence lengths of *src* and *mt*, the sequence length ratio

Encoder	PCC↑	*src*-side MCC↑	*mt*-side	
			Words MCC↑	`<GAP>`s MCC↑
BERT	0.4832	0.3092	0.3684	0.03617
ELECTRA	0.5109	0.3100	0.4104	0.1401

Table 2: Single Dual-Encoder model's performance respect to pre-trained model applied to its encoders for the WMT 2020 English-German QE task2.

Systems	PCC↑	*src*-side MCC↑	*mt*-side	
			Words MCC↑	`<GAP>`s MCC↑
Baseline	0.5285	0.3220	0.3696	0.1157
Single	0.5038	0.3200	0.4126	0.1096
Top4-ens	0.5458	0.3165	0.4296	0.1096
Top6-ens	0.5497	0.3186	0.4271	0.1225

Table 3: Our systems' performance for the WMT 2021 English–German QE Task 2. Single is a single system modelled on our proposed model. Top4-ens and Top6-ens are the ensembles of the top four and the top six single systems respectively in terms of their performance on the validation dataset.

between *src* and *mt*, and TER. Then, we created a tuple of labels (T_{src}, $\mathrm{T}_{mt}^{\mathrm{word}}$, $\mathrm{T}_{mt}^{\mathrm{gap}}$, the TER (Snover et al., 2006)) for each triplet [3]. Finally, we tokenized and truncated both of the artifical data and the WMT 2021 official data by using a pre-trained tokenizer based on WordPiece (Wu et al., 2016).

4.2 QE Pre-training

After obtaining about three million of artificial triplets, we made the final QE pre-training data by joining the artificial training data and the official human-labeled data together; especially, we augmented the quantity of the latter by replication to allow our systems to learn from both kinds of training data relatively more evenly during the QE pre-training. Our systems learn to predict all kinds of labels jointly (L_{QE}, Eqn. 11) considering the close correlation among the subtasks in Task 2. We used 1,000 triplets in the WMT 2021's official development dataset as validation data.

4.3 Fine-Tuning

We used only the WMT 2021 human-labeled data for fine-tuning. In contrast with the QE pre-training, we fine-tuned our systems to each subtask: the prediction of the sentence-level task (L_{HTER}, Eqn. 2) and the word-level task (L_{word}, Eqn. 10) Considering the overproportion of OK tags in our training data (Table 1), we set a large k_s, k_m, and k_g (§ 3.2.2) in our experiments.

4.4 Ensemble Learning

Besides single fine-tuned systems, we also made ensembles of our best fine-tuned systems, each of which has a different random seed from that of the others. In detail, after fine-tuning several single systems with different random seeds, for each seed, we picked out the top two systems, each of which is different from the other in certain variable training conditions such as how its cross attention networks have been randomly initialized in that instance, in terms of their performance on our validation dataset. Finally, we averaged the weights of the systems element-wisely for better generalization and made the ensembles.

4.5 Hyperparameters

We used ELECTRA-base (Clark et al., 2020)s as pre-trained monolingual LMs for our dual monolingual encoders[4]. In the QE pre-training, we used `get_schedule_with_warmup`[5] as our learning rate scheduler with 3,000 warm-up steps. We used the AdamW (Loshchilov and Hutter, 2018) optimizer that has a weight decay with λ=0.5, β_1=0.9, β_2=0.999, and ϵ=1e-8, together with gradient clipping. Setting a batch size of 64 for both the QE pre-training and fine-tuning, we set a learning rate of $1e{-}5$ and a tuple of ($k_s = 1$, $k_m = 1$, $k_g = 3$) for the QE pre-training and a learning rate of $5e{-}5$ and a tuple of ($k_s = 2$, $k_m = 2$, $k_g = 4$) for the

[3] https://github.com/deep-spin/qe-corpus-builder

[4] Our encoders are available at https://huggingface.co/models

[5] https://huggingface.co/transformers/main_classes/optimizer_schedules.html

Systems	PCC↑	RMSE↑	MAE↑	Disk Footprint (GB)↓	Model Params↓
HW-TSC	0.6531	0.1513	0.1079	2.0898	560.9M
IST-Unbabel	0.6173	0.1715	0.1163	2.1373	569.4M
ACBU-NMT	0.5773	0.1743	0.1154	2.0894	560.1M
POSETCH (Ours)	0.5497	0.1741	0.1304	1.4540	390.2M
Baseline	0.5285	0.1828	0.1291	1.0640	281.3M
ENSBRT	0.5199	0.1711	0.1287	1.2700	502M

Table 4: The reported sentence-level QE performance of the systems submitted to the WMT 2021 English–German QE Task 2 according to the official leaderboard.

Systems	*mt*-side		*src*-side MCC	Disk Footprint (GB)↓	Model Params↓
	Words MCC	<GAP>s MCC			
JHU-Microsoft	0.5231	0.2559	-	6.3918	484.4M
HW-TSC	0.5095	0.2997	0.4499	2.0898	560.9M
IST-Unbabel	0.4661	0.1833	0.4042	2.1373	569.4M
ACBU-NMT	0.4368	-	0.3915	2.0894	560.1M
POSETCH (Ours)	0.4126	0.1096	0.3200	1.4540	390.2M
Baseline	0.3696	0.1157	0.3220	1.0640	281.3M

Table 5: The reported word-level QE performance of the systems submitted to the WMT 2021 English–German QE Task 2 according to the official leaderboard. A hyphen indicates that no corresponding score exists.

fine-tuning, respectively. We validated the performance of our systems on our validation set every 5,000 steps during the QE pre-training and every 200 steps during the fine-tuning, respectively; we applied early stopping with a patience of 30.

4.6 Results

In comparison with our single system, our ensembles report an improved PCC, *mt*-side words MCC, and *mt*-side <GAP>s MCC of about 0.5497, 0.4296, and 0.1225 respectively (Table 2). Compared to other systems submitted to the WMT 2021 English–German QE Task 2, our systems outperform the baseline systems in terms of the sentence-level PCC (Table 3) and the *mt*-side words MCC (Table 4). Our systems are inferior to the baseline systems in terms of the *src*-side MCC and the *mt*-side <GAP>s MCC by a narrow margin (Table 4). However, because our systems have a smaller number of parameters than other submitted systems, we expect that it is possible to improve the performance of our systems by adopting larger pre-trained LMs such as ELECTRA-large (Clark et al., 2020).

5 Conclusion

We model our systems submitted to Task 2 of the WMT 2021 QE shared task on our proposed model, which uses dual pre-trained monolingual encoders and two additional cross attention networks to process the two input data *src* and *mt* more effectively considering that the latest Transformer-based QE models are not propped up by pre-trained monolingual word representations. We expect that the cross attention networks enable the two pre-trained monolingual encoders to exchange cross-lingual information without losing their stability and to learn the subtasks of Task 2 jointly and also separately. Experimental results partially supports this expectation: according to the official leaderboard, our systems outperform the baseline systems in terms of the *mt*-side words MCC and the sentence-level PCC by 0.4126 and 0.5497 respectively, although they do not in terms of the *src*-side MCC and the *mt*-side <GAP>s MCC. Neverhteless, it appears possible to improve the performance of our systems by adopting larger pre-trained LMs, and thus, our future work will explore such aspects and other related new methods.

Acknowledgements

This work was carried out as part of the HPC Support Project supported by the Ministry of Science and ICT (MSIT) and the National IT Industry Promotion Agency (NIPA), and was funded by the Institute of Information & Communications Technology Planning & Evaluation (IITP) supported by the Korean government (MSIT): Grant No. 2019-0-01906, Graduate School of Artificial Intelligence

(POSTECH).

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Coling 2004: Proceedings of the 20th international conference on computational linguistics*, pages 315–321.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32:7059–7069.

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

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020. Bergamot-latte submissions for the wmt20 quality estimation shared task. Association for Computational Linguistics.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Chi Hu, Hui Liu, Kai Feng, Chen Xu, Nuo Xu, Zefan Zhou, Shiqin Yan, Yingfeng Luo, Chenglong Wang, Xia Meng, et al. 2020. The niutrans system for the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1018–1023.

Dongjun Lee. 2020. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1024–1028.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Joao Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André FT Martins. 2020. Ist-unbabel participation in the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036.

Akifumi Nakamachi, Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. Tmuou submission for wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1037–1041.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. Escape: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Raphael Rubino. 2020. Nict kyoto submission for the wmt'20 quality estimation task: Intermediate training for domain and task adaptation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1042–1048.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. 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*, pages 223–231.

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. Association for Computational Linguistics.

Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. 2009. Estimating the sentence-level quality of machine translation systems. In *EAMT*, volume 9, pages 28–35.

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.

Haijiang Wu, Zixuan Wang, Qingsong Ma, Xinjie Wen, Ruichen Wang, Xiaoli Wang, Yulin Zhang, Zhipeng Yao, and Siyao Peng. 2020. Tencent submission for wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1062–1067.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.