HW-TSC's Participation at WMT 2021 Quality Estimation Shared Task

Yimeng Chen[1*], **Chang Su**[1*], **Yingtao Zhang**[1*], **Yuxia Wang** [1], **Xiang Geng** [2],
Hao Yang [1], Shimin Tao [1], Jiaxin Guo [1], Minghan Wang [1], Min Zhang [1], Yujia Liu [1], Shujian Huang [2]
[1] Huawei Translation Services Center, Beijing, China
[2] Nanjing University, Nanjing, China
{chenyimeng,suchang8,zhangyintao9,wangyuxia5,yanghao30,taoshimin,
guojiaxin1,wangminghan,zhangmin186,liuyujia13}@huawei.com
{gx@smail, huangsj@}nju.edu.cn

Abstract

This paper presents our work in WMT 2021 Quality Estimation (QE) Shared Task. We participated in all of the three sub-tasks, including Sentence-Level Direct Assessment (DA) task, Word and Sentence-Level Post-editing Effort task and Critical Error Detection task, in all language pairs. Our systems employ the framework of Predictor-Estimator, concretely with a pre-trained XLM-Roberta as Predictor and task-specific classifier or regressor as Estimator. For all tasks, we improve our systems by incorporating post-edit sentence or additional high-quality translation sentence in the way of multitask learning or encoding it with predictors directly. Moreover, in zero-shot setting, our data augmentation strategy based on Monte-Carlo Dropout brings up significant improvement on DA sub-task. Notably, our submissions achieve remarkable results over all tasks.

1 Introduction

Quality Estimation (QE) focuses on estimating the quality of machine translation (MT) system output when no ground truth reference is available (Specia et al., 2018). QE covers wide range of tasks including word-level, sentence-level and document-level. It has wide range of applications in MT quality check and post-editing effort estimation.

In WMT2021 Quality Estimation shared task[1], there are three sub tasks — Sentence-Level Direct Assessment task, Word and Sentence-Level Post-editing Effort task and Critical Error Detection task. Each sub task involves several language pairs. Our team participated in all the above three tasks over all language pairs. We summarized our main contributions as follow:

- We employ Predictor-Estimator architecture (Kim et al., 2017b; Kim and Lee, 2016) which is a two-stage model consisting of a word prediction model trained from large-scale parallel corpora, and a estimation model trained from quality-annotated QE data. Different from the original Predictor-Estimator model in (Kim et al., 2017a), we use pre-trained XLM-Roberta large as predictor instead of RNN-based model to achieve better QE features, and use task-specific classifier or regressor as quality estimator.

- We extend PE assisted QE (PEAQE) (Kepler et al., 2019; Wang et al., 2020) by integrating real PE or addtional high-quality translation in the way of multitask learning or directly encoding it with predictor.

- We explore data augmentation method based on Monte Carlo (MC) dropout (Gal and Ghahramani, 2016) to enhance the performance of zero-shot language pairs in Direct Assessment(DA) task.

Our methods achieve impressive performance on both word and sentence level tasks. Specifically, we peak the top-1 on sentence-level DA over English-German and English-Japanese pairs. For word and sentence-level post-editing effort task, our submissions of the majority language pairs obtain the best Pearson's correlation or Matthews correlation coefficient. We also win the first place in critical error detection task in English-Chinese and English-Japanese.

We will describe the tasks, datasets, and our methods for DA task, post-editing task, and critical error detection task in section 2, section 3, and section 4 respectively. Section 5 presents details of our experimental setup and results, with a brief discussion and conclusion in the end.

[*] Indicates equal contribution.

[1] http://www.statmt.org/wmt21/quality-estimation-task.html

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 890–896
November 10–11, 2021. ©2021 Association for Computational Linguistics

2 Sentence-Level Direct Assessment Task

2.1 Task Description

The sentence-level Direct Assessment task focuses on estimating sentence-level translation quality scores which are annotated with Direct Assessment (DA) scores by professional translators. The original DA scores are in scale of 0-100. The scores are then standardised using the z-score by rater. The goal is to estimate a z-standardised DA score for each translation sentence.

Sentence-level DA task is evaluated by Pearson's correlation between the predicted score and the gold human annotated z-standardised DA score. The system is assessed from two aspects: single language pair and multilingual track which takes all languages into account, including zero-shot pairs, calculating the averaged Pearson correlation overall.

2.2 Dataset

For each language, 7000, 1000 and 1000 sentence pairs are provided officially as training, development and test20 set before releasing another 1000 for the real blind test21, including high-resource English-German (En-De) and English-Chinese (En-Zh), medium-resource Romanian-English (Ro-En) and Estonian-english (Et-En), low-resource Sinhalese-English (Si-En) and Nepalese-English (Ne-En), as well as Russian-English (Ru-En). Besides, 4 language pairs — English-Czech (En-Cz), English-Japanese (En-Ja), Pashto-English (Ps-En) and Khmer-English (Km-En), are only offered blind test (1000), without training data.

2.3 Implemented Systems

The systems for DA employ Predictor-Estimator architecture. Following previous sota works (Fomicheva et al., 2020; Moura et al., 2020; Rei et al., 2020), we use a pre-trained XLM-Roberta (XLM-R)(Conneau et al., 2019) model as a predictor due to its impressive performance on cross-lingual downstream tasks.

Practically, we concatenate source(SRC) and target(MT) sentences in the format `[CLS] SRC [EOS] [SEP] MT [EOS]` following XLM-R usage, and take the embedding of pooled output of [CLS] token as features of a sentence pair. For Estimator, we simply stack two-layer FFN, taking the [CLS] feature generated above as the input to predict sentence-level DA scores.

2.3.1 PE Assisted Sentence-Level DA Prediction

Inspired by the Pseudo-PE techniques (Kepler et al., 2019; Wang et al., 2020), we take full use of post-editing sentences provided in Post-editing Effort task through multitask learning. The model jointly learns to score (SRC, MT) pair in a regression task, and distinguish between translations and post-edited sentences — which is the better translation in a classification task. In inference stage, the model only conducts regression task to predict DA score, as post-editing sentences are not available for blind test set.

The regression task applies loss function as:

$$\mathcal{L}_{reg} = (\phi(E_{s,t}) - Y_{human})^2 \qquad (1)$$

where $E_{s,t}$ is the embedding of sentence pair (source, mt), ϕ is the regressor taking them as input, through a two-layer FFN to compute DA score, and Y_{human} is the Z-normalized DA score annotated by human.

The classification task forces the model to capture more expressive cross-lingual sentence representation which is paramount for DA score. In implementation, we get the model to learn which is the pair with better translation between embedding of concatenated source and target $E_{s,t}$ and embedding of concatenated source and PE $E_{s,p}$. We splice two vectors in random order and apply two stacked FFN layers to compute classification result, in which 0 means the former pair is the better (i.e. the former contains PE), 1 means the former is the worse and 2 means translation and post-edit are exactly the same. Equation (2) gives the loss function for the classification task, where M is the number of classes ($M = 3$), Y is the binary indicator (0, 1, 2) if class label c is the correct classification for observations, P is the model predicted probabilities that the observation is of classes.

$$\mathcal{L}_{cls} = -\sum_{c=0}^{M-1} Y_c \log(P_c) \qquad (2)$$

2.3.2 Data Augmentation for Zero-shot Languages

Instead of directly applying the multilingual DA model trained on other 7 language pairs to zero-shot languages, we exploit a data augmentation strategy based on MC dropout to improve the performance. Specifically, we compute the expectation and variance for the set of estimated DA scores of zero-shot languages obtained by performing N (N=30) stochastic forward passes through the well-trained but dropout-perturbed QE model. In order to control the uncertainty introduced by the disturbance, we only retain dropout in estimator and last two layers in XLM-R. We take variance as an indicator to detect observations with less uncertainty and use expectation as DA score label. Then, we mix the generated zero-shot DA data with randomly selected non-zero-shot training set to fine-tune the model. Experiments show that our data augmentation is effective to improve the performance, achieving better Pearson correlation.

3 Word and Sentence-Level Post-editing Effort Task

3.1 Task description

Word-Level QE estimates the translation quality by producing a sequence of tags for source and target. For target sentences(MT), each token is tagged as either OK or BAD, each gap between two words is tagged as BAD if one or more missing words should have been there, and OK otherwise. So the number of total tags for each target sentence is $2N + 1$, where N is the number of tokens in the target sentence. For source sentences(SRC), tokens are tagged as OK if they were correctly translated, and BAD otherwise. The number of total tags for each source sentence is M, where M is the number of tokens in the source sentence. The evaluation metrics of the word-level task is the Matthews Correlation Coefficient (MCC).

Sentence-Level QE predicts the Human Translation Error Rate (HTER). HTER is the ratio between the number of edits (insertions / deletions / replacements) needed and the reference translation length. The evaluation metrics of the sentence-level task is Pearson's correlation metric.

3.2 Dataset

The dataset in these task provides the same source and translation as DA task, with an extra post-edit sentence for each observation and task-specific token-level and sentence-level labels. Besides, we generate addition-translation sentence (AMT) for each source sentence by using well-trained machine translation systems. The motivation here is to add an additional criterion which is in the same language as the provided translation sentence. We suppose that to detect the difference between two sentence in the same language is a simpler task for model. There are some important label properties to highlight:

- The number of BAD tags and OK tags is imbalanced, especially for GAP tags.
- AMT's BLEU score is significantly lower than MT taking post-edits as reference. Its average HTER is higher than MT. It indicates that the generated AMT is less closer to post-edits than MT.

3.3 Method

The systems for QE shared task2 also employ Predictor-Estimator architecture(Kim et al., 2017b).

Predictor. Similar to Task1, we use pre-trained XLM-Roberta (XLM-R) model as predictor after fine-tuning it with mask language modeling task(Devlin et al., 2018) using the provided source and PE sentences. In order to improve the performance, refers to approach in (Wang et al., 2020), we concatenate SRC, MT, AMT sentences together in the format of `[BOS] SRC [EOS] [SEP] MT [EOS] [SEP] AMT [EOS]`.

We notate the predictor as f; SRC, MT and AMT text as X and Y and Z, corresponding features as H_x, H_y, H_z respectively:

$$H_x, H_y, H_z = f(X, Y, Z), \tag{3}$$

Estimator. We utilise 4 independent 2-layers FFN including binary three classification tasks to predict SRC word tags, MT/AMT word tags, MT/AMT gap tags respectively, and a regression task to predict HTER score of MT/AMT. All predictions are obtained by performing specific transformations ϕ. We define the predicted logits of SRC word, MT word, MT gap, AMT word, AMT gap as $\hat{V_{xw}}$, $\hat{V_{yw}}$, $\hat{V_{yg}}$, $\hat{V_{zw}}$, $\hat{V_{zg}}$; and HTER predicted score of MT and AMT as $\hat{V_{yh}}$, $\hat{V_{zh}}$. The estimator can be described as:

$$\begin{aligned}
\hat{V_{xw}} &= \phi_{xw}(H_x), \\
\hat{V_{yw}} &= \phi_w(H_y), \\
\hat{V_{zw}} &= \phi_w(H_z), \\
\hat{V_{yg}} &= \phi_g(f_{cat}(H_y, \hat{V_{yw}})), \\
\hat{V_{zg}} &= \phi_g(f_{cat}(H_z, \hat{V_{zw}})), \\
\hat{V_{yh}} &= \phi_h(f_{gap}(f_{cat}(H_y, \hat{V_{yw}}, \hat{V_{yg}}))), \\
\hat{V_{zh}} &= \phi_h(f_{gap}(f_{cat}(H_z, \hat{V_{zw}}, \hat{V_{zg}}))),
\end{aligned} \tag{4}$$

where f_{cat} is the concatenate method in the last dimension, f_{gap} is the global average pooling in the second dimension ignoring padding tokens in a batch just like (Lin et al., 2013) 3.2.

Loss. We prepend and append two special <*pad*> labels to the original word label sequence, append a special <*pad*> label to the original gap label sequence during training, but loss of the padded labels is not computed. For all classification tasks, to deal with the problem of imbalance between OK and BAD number, we use weighted cross entropy as the loss function, and the weight is calculated as $w_i = \frac{N}{\sum C_i}$, where w_i is the inverse of the proportion of the instance with class C_i. For sentence-level HTER score loss, we use mean squared error (MSE) as the loss function. We define the tags of SRC word, MT word, MT gap, AMT word, AMT gap as V_{xw}, V_{yw}, V_{yg}, V_{zw}, V_{zg}; and HTER score of MT and AMT as V_{yh}, V_{zh}.

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

$$\begin{aligned} loss = & \sum_{\tau\in\{xw,yw,yg,zw,zg\}} \lambda_\tau logP(V_\tau|X,Y,Z)+ \\ & \sum_{\tau\in\{hy,hz\}} \lambda_\tau \sqrt{\sum(V_\tau-\hat{V}_\tau)^2}, \end{aligned} \quad (5)$$

where xw, yw, yg, zw, zg represents for classification tasks, hy, hz represents for regression tasks, λ is the weight of loss for a specific task. The multi-task framework can improve the overall performance.

4 Critical Error Detection

4.1 Task Description

This is a new QE task focusing on predicting sentence-level binary scores indicating whether or not a translation contains (at least one) critical error. The key point is to identify whether the translation will lead to misleading or more serious consequences, e.g. the translation involves critical mistranslation, hallucination or critical content deletion. Only binary prediction (whether or not any critical error contained) is required. The evaluation metrics of this task is also the MCC.

4.2 Dataset

The dataset contains 4 languages which are English-German, English-Chinese, English-Czech, English-Japenese. 7000 training, 1000 validation, and 1000 blind test sentence pairs are available for each language. Ground truth label has two classes, NOT means no catastrophic error, and ERR means at least one catastrophic error in the translation. It is noticed that the number of NOT and ERR tag is imbalanced.

Language	Baseline	+Multitask	+Ensemble
En-De	0.490	0.552	0.547
En-Zh	0.494	0.502	0.519
Ro-En	0.886	0.897	0.902
Et-En	0.798	0.805	0.814
Ne-En	0.776	0.789	0.801
Si-En	0.648	0.677	0.675
Ru-En	0.761	0.787	0.787
Average	**0.693**	**0.716**	**0.721**

Table 1: Pearson correlation between prediction of our system and human DA judgement of non-zero-shot language pairs on test20 set.

Language	Baseline	+AugData	+All
En-De	0.481	/	0.584
En-Zh	0.523	/	0.583
Ro-En	0.878	/	0.901
Et-En	0.775	/	0.808
Ne-En	0.810	/	0.858
Si-En	0.564	/	0.581
Ru-En	0.753	/	0.787
En-Cz	0.546	0.557	0.573
En-Ja	0.297	0.349	0.364
Ps-En	0.592	0.622	0.622
Km-En	0.661	0.653	0.659
Multilingual	**0.621**	/	**0.665**

Table 2: Pearson correlation between prediction of our system and human DA judgement on test21 set.

4.3 Methods

Similar as the above two tasks, our baseline system takes pre-trained XLM-R as predictor, stacked FFN layers as binary classifier. We also experimented with replacing XLM-R by mBART (Liu et al., 2020) and replacing FFN layers with TextCNN, Bi-LSTM and other types of network.

Based on the intuition that the semantic difference between two monolingual sentences are easier to distinguish than that of two cross-lingual sentences, we propose to incorporate a “good” MT of the source sentence into *(src. mt)* pair during training, so that the auxiliary information provided by the “good” MT can help the model to directly compare *mt* with MT+*src*, instead of only depending on cross-lingual *src*. With consideration of expensive overhead of manual translation, we assume that au-

Score	Method	En-Zh	En-DE	Ru-En	Ro-En	Et-En	Si-En	Ne-En
Pearsonr	**baseline(dev)**	0.3013	0.4910	0.4475	0.5381	0.5997	0.6062	0.5899
	+AMT(dev)	0.3481	0.6003	0.5387	0.8479	0.7832	0.8031	0.6902
	+Ensemble(dev)	0.3772	0.6678	0.5704	0.8914	0.8249	0.8573	0.7849
	All(test)	0.3681	0.6531	0.5615	0.8623	0.8094	0.8690	0.7976
SRCW	**baseline(dev)**	0.1991	0.3019	0.2904	0.4132	0.4173	0.3899	0.4027
	+AMT(dev)	0.2895	0.4378	0.3991	0.6027	0.5204	0.5780	0.5109
	+Ensemble(dev)	0.3128	0.4502	0.4277	0.6374	0.5396	0.6033	0.5576
	All(test)	0.3098	0.4499	0.4258	0.6140	0.5490	0.6159	0.5450
MTW	**baseline(dev)**	0.1354	0.3988	0.3500	0.4980	0.4533	0.5393	0.4418
	+AMT(dev)	0.3346	0.4907	0.4331	0.6642	0.6006	0.7446	0.6721
	+Ensemble(dev)	0.3726	0.5149	0.4479	0.6807	0.6177	0.8102	0.7007
	All(test)	0.3536	0.5095	0.4507	0.6664	0.6058	0.8469	0.6741
MTG	**baseline(dev)**	0.0998	0.1987	0.2249	0.2856	0.2017	0.2844	0.3129
	+AMT(dev)	0.1799	0.3101	0.3481	0.4379	0.3119	0.5023	0.4001
	+Ensemble(dev)	0.1822	0.3158	0.3725	0.4531	0.3280	0.5573	0.4490
	All(test)	0.1719	0.2997	0.3877	0.4457	0.3115	0.6392	0.4027

Score	Method	En-Cs	En-Jp	Ps-En	Km-En	Multilingual
Pearsonr	**baseline(test)**	0.2910	0.0999	0.3722	0.3571	0.5002
	+Ensemble(test)	0.4750	0.2620	0.5343	0.4750	0.6314
SRCW	**baseline(test)**	0.1981	0.1523	0.2344	0.3183	—-
	+Ensemble(test)	0.3128	0.2166	0.3044	0.4101	—-
MTW	**baseline(test)**	0.2107	0.1372	0.2789	0.3077	—-
	+Ensemble(test)	0.3801	0.2581	0.4497	0.6364	—-
MTG	**baseline(test)**	0.1149	0.0901	0.1342	0.2691	—-
	+Ensemble(test)	0.2126	0.1523	0.2602	0.4190	—-

Table 3: Pearsonr correlation, MCC of words in SRC, MCC of words in MT and MCC of gaps in MT between prediction of our system and labels. SRCW is SRC words MCC, MTW is MT words MCC, MTG is MT gaps MCC, Test20 set is used as training set. Results of test set are from official leaderboard.

tomatic machine translation (AMT) of top commercial machine translation tools can also be competent at this work. Practically, we apply Baidu Fanyi [2] and Google Translate [3] API, obtaining two corresponding AMTs given a source sentence. Then we concatenate it with source and original machine translation in the format of `[CLS] SRC [EOS] [SEP] MT [EOS] [SEP] AMT [EOS]`, followed by encoding the concatenated triplet to the predictor.

Voting-Based Ensemble. Finally, we ensemble several models and take their majority voting as prediction results.

5 Experimental Results

5.1 Task1: Sentence-level Direct Assessment

Experimental Settings Our system is implemented with hugging face transformers package. The pre-trained xlm-roberta-large model which has approximately 550M parameters is taken as predictor. We train the predictor and the estimator together on the multilingual QE DA dataset using Adam(Kingma and Ba, 2015) as optimizer with constant learning rate of $1e^{-6}$ and training batch size of 16. The model is trained on a Nvidia Tesla V100 GPU.

Results Table 1 shows the results on test20 set. Our baseline is the system described in section 2.3. +multitask method is introduced in section 2.3.1. To achieve more competitive scores while also maintain a relatively small number of parameters, we ensemble our result with MC dropout approach, that is to run N (N=50) pass forwards with dropout and take the expectation of the N predictions as final answers. Table 2 presents the experimental results on blind test21 set. The baseline here is the same as Table 1 baseline. +AugData is the approach mentioned in section 2.3.2. +All is our final submitted result that integrates multi-task, data augmentation and ensemble.

[2] `https://fanyi.baidu.com/`

[3] `https://translate.google.com/`

Dataset	Pre-trained Model	Classification Layer	AMT	En-Zh	En-De	En-Cs	En-Ja
Dev	baseline	FFN	/	0.1873	0.4008	0.3974	0.2193
	MBart			0.2317	0.3940	0.4112	0.2148
	XLMR-Large			**0.2989**	**0.4846**	**0.4537**	**0.2744**
	XLMR-Large	TextCNN	/	0.1820	0.2008	0.2139	0.1429
		Bi-LSTM		**0.2350**	**0.4279**	**0.4132**	0.1981
		RCNN		0.2045	0.3850	0.3463	**0.2523**
	XLMR-Large	FFN	BaiduTrans	**0.3474**	0.4623	0.4372	**0.2948**
			GoogleTrans	0.2515	**0.4732**	**0.4551**	0.2724
	Ensemble			**0.3962**	**0.5104**	**0.4854**	**0.3542**
Test	Ensemble			**0.3533**	**0.4899**	**0.4482**	**0.3184**

Table 4: MCC of all language pairs over development(dev) set and test set.

5.2 Task2: Word and Sentence-Level Post-editing Effort Task

Settings: The batch size in training stage is 8. We use Adam as optimizer with learning rate of $2e^{-5}$. Each estimator FFN layer has a 0.1 dropout. Loss weight are: ($\lambda_{yh} = 2$, $\lambda_{zh} = 2$, $\lambda_{xw} = 4$, $\lambda_{yw} = 1$, $\lambda_{yg} = 1$, $\lambda_{zw} = 1$, $\lambda_{zg} = 1$) / 12. Our model params is 560,944,640, disk footprint(in bytes, without compression) is 2,243,954,093.

Results Table 3 shows the results on dev and test21 set. Our baseline is the QE system without AMT data. +AMT method is the QE system with AMT data. In the experiments, we generate 3 different kinds of AMT data with the machine translation system trained for the WMT2021 Machine Translation of News Shared Task, Baidu Fanyi [4] and Google Translate[5]. For each kind of AMT, we run N (N=10) pass forward with dropout=0.1 using the a unified model trained with all AMT together. The expectations of 3N predictions of score and token labels is taken as the final answers.

5.3 Task3: Critical Error Detection

Table 4 shows the results of our system on development and blind test set. Experiments show that the best results obtained when applying XLMR-Large and FFN layer on development set. The involvement of AMT also brings significant improvement over all language pairs. For ensemble settings, we ensemble multiple models with different pre-trained models and classification layers using voting-based method as introduced in section 4.3.

In order to solve the problem of label imbalance, we also investigate different label weights when computing cross-entropy loss. Due to the large gap between the number of NOT and ERR labels in the dataset, the weights(NOT:ERR) are clipped as 1:6, 1:4, 1:5, 1:15 for enzh, ende, encs, enja. Meanwhile, to better fit the data in the test set and avoid over-fitting, we utilise dropout with rate of 0.1 and weight decay of $1e^{-5}$.

6 Conclusion

We present our work on WMT 2021 QE shared task in this paper. For all the three tasks to estimate sentence-level DA, token and sentence-level post-edit effort and sentence-level critical error, we employ predictor-estimator framework as our baseline. To further boost performance, we investigate the usage of additional high-quality translations. For task1, we mainly focus on introducing post-edits with multi-task learning. Also, the effect of data augmentation method based on MC dropout is studied here to improve the result of zero-shot pairs. For task 2 and 3, we generate high-quality translations for each observation using multiple well-trained machine translation systems. By directly concatenating AMT with the original source and target sentence then encoding it with pre-trained predictor, we achieved remarkable results over all language pairs and tasks. In future, we will continue to invest time and effort on studying the effect of involving additional translations into QE tasks, for example, how the additional translation quality will affect QE performance, what the better ways are to incorporate additional translations in.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco

[4] https://fanyi.baidu.com/

[5] https://translate.google.com/

Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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*.

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. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1010–1017, Online. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

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, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017a. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 17:1–22.

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. 2017b. 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*.

Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

João Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André F. T. Martins. 2020. IST-unbabel participation in the WMT20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1029–1036, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

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

Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen, and Liangyou Li. 2020. HW-TSC's participation at WMT 2020 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1056–1061, Online. Association for Computational Linguistics.