Papago's Submission to the WMT21 Quality Estimation Shared Task

Seunghyun S. Lim
Papago, Naver Corp.

Hantae Kim
Papago, Naver Corp.

Hyunjoong Kim
Papago, Naver Corp.

{shaun.lim, hantae.kim, soy.lovit}@navercorp.com

Abstract

This paper describes Papago's submission to the WMT 2021 Quality Estimation `Task 1: Sentence-level Direct Assessment`. Our multilingual Quality Estimation system explores the combination of Pretrained Language Models and Multitask Learning architectures. We propose an iterative training pipeline based on pretraining with large amounts of in-domain synthetic data and finetuning with gold (labeled) data. We then compress our system via knowledge distillation in order to reduce parameters yet maintain strong performance. Our submitted multilingual systems perform competitively in multilingual and all 11 individual language pair settings including zero-shot.

1 Introduction

Quality Estimation (QE) evaluates the quality of machine translated output without human reference translation (Blatz et al., 2004). QE has a variety of applications in the Machine Translation (MT) pipeline and is particularly useful in industry settings by informing translation quality to end-users. High performance in sentence-level QE (Specia et al., 2020) is achieved by building a model on top of Pretrained Language Model (PLM); XLM-RoBERTa-large (Conneau et al., 2020) performs particularly well as shown in previous WMT sentence-level QE Shared Task. However, such PLMs contain extremely large number of parameters. This year's task is different from the previous years' task as submitted systems are ranked based on both model size[1] and model performance[2]. For concurrent work Gajbhiye et al. (2021) applies knowledge distillation (Hinton et al., 2015) from a PLM-based QE architecture to a much lighter BiRNN-based architecture, reducing memory requirements. Data scarcity is another issue relevant to QE tasks where there are often limited amount of gold training data. Previous WMT systems incorporate data augmentation techniques and show improvements in model performance when training with additional sources of data (Baek et al., 2020; Ranasinghe et al., 2020a).

Our system builds a model on top of PLM and trains with Multi-task Learning (MTL) (Caruana, 1997). Similar to Hoang et al. (2018); Zhang et al. (2018) where back-translation is iteratively applied to the same monolingual corpus to successively generate higher quality synthetic training data in the context of Neural Machine Translation (NMT), our proposed approach consists of an iterative knowledge transfer procedure which aims to repeatedly produce better quality pseudo labels for large amounts of synthetic training data. During the final stage of our training pipeline, knowledge distillation is applied from teacher to student model in order to reduce model size while maintaining competitive performance. We participate in WMT 2021 Quality Estimation (Specia et al., 2021) `Task 1` for multilingual and all individual language pair settings. Our system is a single multilingual sentence-level QE model that performs very strongly in both multilingual and individual language pair settings.

2 Data

In this year's task, participants are provided with 7K train set (`Train`), 1K development set (`Dev`), and 1K test set (`Test20`) for 7 language pairs: high-resource English-German (En-De) and English-Chinese (En-Zh), medium-resource Romanian-English (Ro-En) and Estonian-English (Et-En), and low-resource Sinhalese-English (Si-En) and Nepalese-English (Ne-En), as well as Russian-English (Ru-En). The source side sentences of language pairs excluding Ru-En are col-

[1] Disk space without compression and number of parameters.

[2] Pearson's correlation coefficient, root mean square error (RMSE), mean absolute error (MAE).

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 935–940
November 10–11, 2021. ©2021 Association for Computational Linguistics

lected from Wikipedia data; the source side sentences of Ru-En is collected from a combination of Wikipedia articles and Reddit articles. Target side sentences are collected by translating source side sentences using NMT models and each sentence pair is annotated by at least three professional translators with a score between 0-100 according to the perceived translation quality. Systems are required to inference z-standardized direct assessment (DA) scores for 1K blind test set for each language pair. This year's task also include zero-shot scenario for 4 new language pairs: English-Czech (En-Cs), English-Japanese (En-Ja), Pashto-English (Ps-En), and Khmer-English (Km-En). As additional resource, participants are also provided with parallel data used to train NMT models (except for Ru-En and zero-shot language pairs) and NMT models used to generate target side sentences of the dataset.

3 Approach

Figure 1 summarizes our approach. Below we describe relevant components to our sentence-level QE model.

3.1 Base Model Architecture

Our QE model stacks feed-forward layers on top of feature vector extracted from Pretrained Language Models (PLM). Our choices of PLM are XLM-RoBERTa-base ($L = 12$) and XLM-RoBERTa-large ($L = 24$). Given source sentence src^X in language X and target sentence tgt^Y in language Y, the concatenation of src^X and tgt^Y are fed as input to the PLM and feature vector CLS_{cat} is produced by taking the concatenation of [CLS] representations from all layers of the PLM; our feature vector is based on using [CLS] token representation due to its superior performance over other pooling strategies (Ranasinghe et al., 2020b; Fomicheva et al., 2020). QE model f predicts direct assessment scores as follows:

$$\begin{aligned} f(src^X, tgt^Y) = W_{score} \cdot &LeakyReLU(\\ &W_2 \cdot LeakyReLU(\\ &W_1 \cdot CLS_{cat} + b1) + b2) \end{aligned} \tag{1}$$

where $W_{score} \in \mathbb{R}^{1\times512}$, $W_2 \in \mathbb{R}^{512\times2048}$, $b_2 \in \mathbb{R}$, $W_1 \in \mathbb{R}^{2048\times N}$, $b_1 \in \mathbb{R}$, and N is XLM-RoBERTa's hidden dimension size (1024) times number of layers (L).

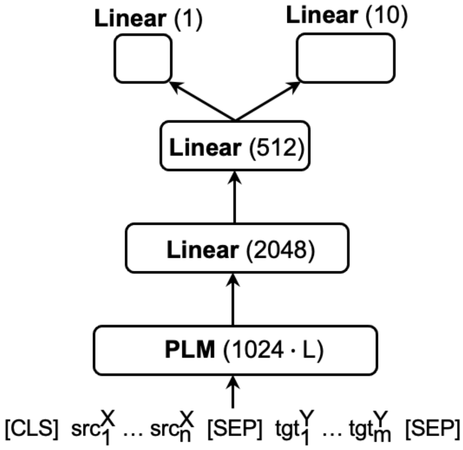
Linear (1)
Linear (10)
Linear (512)
Linear (2048)
PLM (1024 · L)
[CLS] src$_1^X$ ... src$_n^X$ [SEP] tgt$_1^Y$ ... tgt$_m^Y$ [SEP]

Figure 2: The network architecture for Multi-task Learning (§3.2) with XLM-RoBERTa as PLM. Concatenation of source and target sentences (with special tokens) are tokenized and fed as input to the PLM. Numbers in parenthesis denote the output dimension size of each network block.

3.2 Multi-task Learning (MTL)

We train our QE model in multi-task fashion by adding a classification objective to the base model architecture (§3.1). As shown in Figure 2, a classification layer W_{class}, where $W_{class} \in \mathbb{R}^{10\times512}$, is stacked next to W_{score} in equation (1). Given the n_{th} train set sample's z-standardized DA score $score_n$, we scale $score_n$ by applying min-max normalization and assign bin (class) labels to each sample. For our experiments, the number of bins is set to 10. Note that min-max scaling is applied to each language pair data set in order to account for different scales of $score_n$ per data set. The model is trained with a combined loss of mean squared error and cross entropy loss as shown in equation (2), with λ set to 0.6. Our intuition is that QE is inherently a complex task even for humans such that human-labeled DA scores may contain noise. We expect that training with an auxiliary classification loss, where bin labels are less susceptible to noise, can make training more robust and produce a model that is more generalizable.

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{mse} + (1 - \lambda) \cdot \mathcal{L}_{ce} \tag{2}$$

3.3 Data Augmentation

We create large amounts of synthetic direct assessment samples for 7 language pairs (non zero-shot) using parallel data and NMT models which both are provided as additional resource. For data augmentation, we utilize source side sentences from parallel data. We sub-sample from parallel data

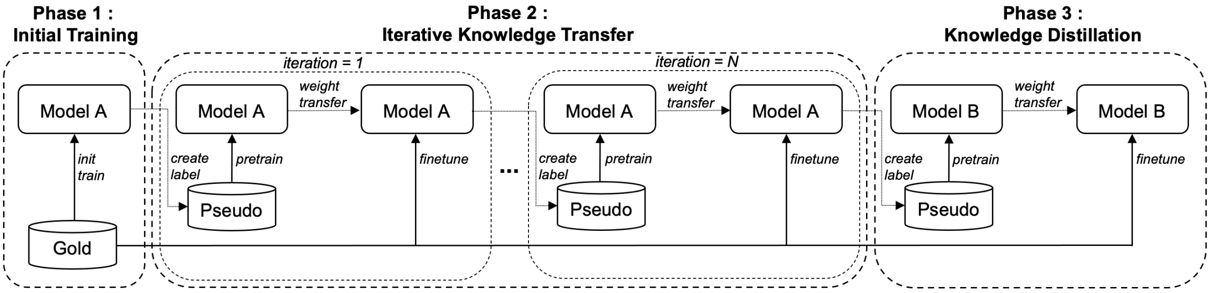

Figure 1: Pipeline of our proposed approach. `Gold` refers to `Train` set provided by task organizers (§2). `Pseudo` refers to synthetic sentence pairs generated as described in §3.3, while labels for `Pseudo` are created as described in §3.4. `Phase 2` and `Phase 3` each refer to §3.4 and §3.5 respectively. `Model A` and `Model B` each refer to large and small models in terms of model size. In our experiments, the architecture for `Model A` is a base model architecture (§3.1) with MTL (§3.2) using XLM-RoBERTa-large as PLM, which we denote as $\textbf{Base}_{\textbf{large}}$ **+ MTL**; `Model B` instead uses XLM-RoBERTa-base as its PLM and we denote it as $\textbf{Base}_{\textbf{small}}$ **+ MTL**.

Language	# parallel data	# sampled
En-De	19,298,476	400,000
En-Zh	15,178,232	400,000
Ro-En	3,901,626	400,000
Et-En	879,922	400,000
Ne-En	498,272	73,207
Si-En	646,781	400,000
Ru-En	12,061,155	400,000

Table 1: Number of parallel data provided in WMT2021 Task 1 and number of synthetic sentence pairs sampled as augmented data. For Ru-En, we collect parallel data from the Commoncrawl dataset.

for each language pair such that the distribution of sampled source sentences follows the distribution of source side sentences of gold data (§2) in terms of sentence length; this is to reduce the discrepancy between actual data and synthetic data. We then forward-translate source side sentences to target using provided NMT models to collect approximately 2.4M pseudo sentence pair data which are used as additional training resource. Table 1 shows the total amount of parallel data provided and the amount of synthetic sentence pairs generated. We describe how pseudo labels for synthetic data are created in the next section (§3.4).

3.4 Iterative Knowledge Transfer (IKT)

Given a QE model that is initially trained only on gold data (refer to `Phase 1` in Figure 1), iterative knowledge transfer aims to produce higher quality training signals or pseudo labels for synthetic data by iteratively performing pretraining and finetuning as shown in `Phase 2` in Figure 1. For pretraining, the model is always initialized with random weights (PLM weights are loaded from HuggingFace[3]) and is trained using synthetic direct assessment sentence pair data collected from §3.3. Pseudo labels for synthetic data in the current iteration are created with score predictions from model trained in the prior phase or iterative step. The aim of pretraining is to expose our model to large amounts of in-domain synthetic training data with sub-optimal labels. Similar to Sellam et al. (2020), the key aspect of the pretraining technique is to "warm up" the model before finetuning on gold data. At the start of finetuning, the model is initialized with parameter weights from the pretrain stage and is trained only with gold data. Because psuedo labels for synthetic data are newly generated for each iterative step in `Phase 2`, we expect the quality of "warm up" during pretraining to increase in each successive iteration. We stop the iterative process when the model's Pearson's correlation performance does not improve on `Test20`; we empirically find that performance does not improve after the second iteration.

3.5 Knowledge Distillation (KD)

`Phase 3` in Figure 1 demonstrates knowledge distillation from a large to smaller model. Akin to `Phase 2` (§3.4), a 2 stage pretrain-to-finetune training procedure is conducted and pseudo labels for synthetic data is generated using a teacher model which is the model produced from the last iteration of `Phase 2`. As our results will show, the compressed model performs on par with our baseline large model with approximately less than half the number of model parameters.

[3] https://huggingface.co/

Model	Data	En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En	Avg	# params
$\text{Base}_{\text{small}}$ (§3.1)	Dev	0.447	0.475	0.841	0.722	0.715	0.631	0.685	0.645	297M
	Test20	0.428	0.437	0.843	0.742	0.706	0.611	0.720	0.641	
+ MTL (§3.2)	Dev	0.452	0.496	0.847	0.737	0.730	0.639	0.683	0.655	297M
	Test20	0.473	0.449	0.854	0.740	0.729	0.625	0.732	0.657	
$\text{Base}_{\text{large}}$	Dev	0.488	0.496	0.891	0.788	0.794	0.703	0.715	0.696	611M
	Test20	0.481	0.473	0.882	0.803	0.762	0.664	0.764	0.690	
+ MTL	Dev	0.530	0.489	0.901	0.796	0.788	0.706	0.737	0.707	611M
	Test20	0.563	0.486	0.892	0.812	0.795	0.667	0.786	0.715	
+ $\text{IKT}_{\text{iter=1}}$ (§3.4)	Dev	0.550	0.527	0.906	**0.809**	0.798	**0.716**	**0.751**	0.722	611M
	Test20	0.543	**0.502**	**0.903**	0.814	**0.806**	0.676	0.791	0.719	
+ $\text{IKT}_{\text{iter=2}}$	Dev	**0.576**	**0.535**	**0.910**	0.807	**0.801**	0.714	0.742	**0.726**	611M
	Test20	**0.583**	0.497	0.901	**0.817**	0.792	**0.678**	**0.803**	**0.724**	
+ KD (§3.5)	Dev	0.523	0.522	0.880	0.773	0.758	0.680	0.712	0.692	297M
	Test20	0.544	0.488	0.883	0.770	0.764	0.662	0.756	0.695	

Table 2: Pearson's correlation with human judgments on the `Dev` and `Test20` set. Model names starting with + sign indicates approaches that are cumulative.

4 Settings

For all training phases and experiments, we train our model in data parallelism on multiple NVIDIA Tesla V100 GPUs for 3 epochs with batch size of 8 and is optimized with Adam (Kingma and Ba, 2015) with a learning rate of $7e^{-6}$. Dropout (Srivastava et al., 2014) with 0.15 is applied to activation function outputs in equation 1. Each model variant is trained 3 times with different random seeds, and for each model variant the best performing system in terms of Pearson's correlation coefficient is reported.

All models trained within the scope of this paper are multilingual QE models. We concatenate the `Train` set of each individual language pair to create a single multilingual train set for training. We apply the same for `Dev` and `Test20` set such that validation, model selection and evaluation can be performed at a multilingual level.

5 Results

In this section, we present results of our architectures described in §3. Pearson's correlation coefficient between predictions and gold standard scores is the main evaluation metric to measure performance; this year's task also considers model size to rank systems. Table 2 shows the Pearson's correlation with human judgments on the development and test set (`Dev` and `Test20`). Each row in Table 2 corresponds to model variants derived from certain phases of the training pipeline as described in Figure 1. We first observe that incorporating **MTL** (§3.2) improves over both our small baseline model $\textbf{Base}_{\textbf{small}}$ and large baseline model $\textbf{Base}_{\textbf{large}}$ with respect to all language pair settings. We observe further improvements in performance using Iterative Knowledge Transfer (§3.4) where the average performance of second iterative model $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT}_{\textbf{iter=2}}$ is better than the first iterative model. Comparing $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT}_{\textbf{iter=2}}$ to our large baseline model $\textbf{Base}_{\textbf{large}}$, the average performance gain is 3.4 percentage point but gain with respect to individual language pairs varies, with 10.2 percentage point increase for En-De being the greatest.

Our final compressed model $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT}_{\textbf{iter=2}}\textbf{+KD}$ not only outperforms $\textbf{Base}_{\textbf{small}}\textbf{+MTL}$ in all language pairs but also outperforms our large baseline model $\textbf{Base}_{\textbf{large}}$ in 4 out of 7 language pair settings with less than half the number of model parameters.

Table 3 compares performance between the organizer's baseline model and two of our submitted systems. We submit two systems: $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT}$ and $\textbf{Base}_{\textbf{large}}\textbf{+MTL+IKT+KD}$. Systems can be evaluated on two ranking schemes: `R1` indicates overall ranking[4] which considers both model performance and size, while `R2`[5] ranks systems based only on model performance. As shown

[4] Overall ranking is computed by taking the average of individual ranks of the following metrics: Pearson's correlation coefficient, root mean square error, mean absolute error, disk space without compression and number of parameters.

[5] Ranking scheme based on Pearson's correlation coefficient

	+ IKT (§3.4)		+ KD (§3.5)		Organizer's
	Pearson	R2	Pearson	R1	Pearson
Multi	0.6577	4th	0.6132	3rd	0.5411
En-De	0.5677	3rd	0.5511	**2nd**	0.4025
En-Zh	0.5668	4th	0.5534	3rd	0.5248
Ro-En	0.9008	**2nd**	0.8786	3rd	0.8175
Et-En	0.7941	4th	0.7588	3rd	0.6601
Ne-En	0.8530	4th	0.8233	3rd	0.7376
Si-En	0.5947	3rd	0.5819	**1st**	0.5127
Ru-En	0.7927	**2nd**	0.7436	4th	0.6766
En-Cs	0.5722	4th	0.4969	5th	0.3518
En-Ja	0.3315	4th	0.2755	5th	0.2301
Ps-En	0.6368	**2nd**	0.5816	3rd	0.4760
Km-En	0.6616	**2nd**	0.6251	6th	0.5623
# params	611M		297M		281M
Disk space	2,503MB		1,249MB		1,142MB

Table 3: Submission results on the `Test21` blind set. `+IKT` refers to model from either row 5 or 6 of Table 2; `+KD` refers to model from row 7 of Table 2.

	Supervised							Zero-shot			
	En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En	En-Cs	En-Ja	Ps-En	Km-En
Δ Pearson	-0.016	-0.013	-0.022	-0.035	-0.029	-0.012	-0.049	-0.075	-0.056	-0.055	-0.036
Δ RMSE	+0.007	+0.019	+0.034	+0.039	+0.040	+0.022	+0.043	+0.017	+0.011	+0.032	+0.064
Δ MAE	+0.007	+0.020	+0.034	+0.040	+0.040	+0.023	+0.043	+0.017	+0.012	+0.033	+0.064
% Δ Pearson	-2.8	-2.3	-2.4	-4.4	-3.4	-2.1	-6.1	-13.1	-16.9	-8.6	-5.5
% Δ RMSE	+1.2	+3.0	+8.6	+7.6	+7.6	+2.9	+7.5	+2.2	+1.2	+4.3	+7.3
% Δ MAE	+1.2	+3.2	+8.6	+ 7.8	+7.6	+3.0	+7.5	+2.2	+1.4	+4.4	+7.8

Table 4: Changes in performance on the `Test21` blind set when transitioning from `+IKT` (before compression) to `+KD` (after compression). `Supervised` indicates 7 language pairs that are provided in `Train`, `Dev` and `Test20`; `Zero-shot` indicates 4 zero-shot language pairs that are only evaluated with `Test21` blind set. Δ `metric` (row 1 to 3) measures the change in performance; % Δ `metric` (row 4 to 6) measures the percentage change.

in Table 3, when ranking systems based purely on performance (`R2`), **Base$_{\text{large}}$+MTL+IKT** performs strongly. However, when systems are ranked based on both performance and size (`R1`), our compressed model **Base$_{\text{large}}$+MTL+IKT+KD** ranks very competitively. Moreover, our compressed model outperforms the organizer's baseline in all language pair settings with a great margin using approximately 5.7% more parameters.

We observe in Table 3 that our compressed model is relatively less competitive under zero-shot than in supervised settings when ranked based on `R1`. As demonstrated in Table 4, model compression causes performance degradation in all language pairs with respect to all three performance metrics. In particular, the amount of degradation in terms of Pearson's correlation coefficient is greater under zero-shot than in supervised settings. Interestingly, this trend does not apply to other performance metrics (RMSE, MAE) where the amount of degradation under zero-shot and supervised settings is not significantly different. This indicates that model compression degrades the strength of correlation particularly more under zero-shot than in supervised settings, while degradation in performance measured by magnitude of error is not significantly different between two settings.

6 Conclusions

In this paper, we describe our submission to the WMT 2021 Quality Estimation `Task 1: Sentence-level Direct Assessment.` We introduce a QE model architecture trained with multi-task objective and show improvements in performance. We show that iterative knowledge transfer techniques applied in QE tasks can further improve model's performance and demonstrate that knowledge distillation is effective for building a competitive lighter-weight QE model, making it more suitable for practical use. Although our submitted systems show strong performance in general, we observe that our compressed model becomes relatively less competitive under zero-shot settings. Further analysis of this phenomenon and improvements on zero-shot are challenges that we need to overcome in future work.

References

Yujin Baek, Zae Myung Kim, Jihyung Moon, Hyunjoong Kim, and Eunjeong Park. 2020. PATQUEST: Papago translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 991–998, Online. Association for Computational Linguistics.

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, Geneva, Switzerland. COLING.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard 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, Online. Association for Computational Linguistics.

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.

Amit Gajbhiye, Marina Fomicheva, Fernando Alva-Manchego, Frédéric Blain, Abiola Obamuyide, Nikolaos Aletras, and Lucia Specia. 2021. Knowledge distillation for quality estimation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5091–5099, Online. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

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*.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020a. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020b. Transquest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020. Findings of the WMT 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, pages 743–764, Online. Association for Computational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the wmt 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.