Classification-based Quality Estimation: Small and Efficient Models for Real-world Applications

Shuo Sun,[1] Ahmed El-Kishky,[2] Vishrav Chaudhary,[3]
James Cross, [3] Francisco Guzmán,[3] Lucia Specia[4]
[1]Johns Hopkins University, [2]Twitter Cortex, [3]Facebook AI, [4]Imperial College London
[1]ssun32@jhu.edu, [2]aelkishky@twitter.com
[3]{vishrav,jcross,fguzman}@fb.com, [4]l.specia@imperial.ac.uk

Abstract

Sentence-level Quality Estimation (QE) of machine translation is traditionally formulated as a regression task, and the performance of QE models is typically measured by Pearson correlation with human labels. Recent QE models have achieved previously-unseen levels of correlation with human judgments, but they rely on large multilingual contextualized language models that are computationally expensive and thus infeasible for many real-world applications. In this work, we evaluate several model compression techniques for QE and find that, despite their popularity in other NLP tasks, they lead to poor performance in this regression setting. We observe that a full model parameterization is required to achieve SoTA results in a regression task. However, we argue that the level of expressiveness of a model in a continuous range is unnecessary given the downstream applications of QE, and show that reframing QE as a classification problem and evaluating QE models using classification metrics would better reflect their actual performance in real-world applications.

1 Introduction

Quality Estimation (QE) (Specia et al., 2020) is the task of predicting the quality of an automatically translated sentence at test time, without the need to rely on reference translations. Formally, given a source sentence, s and a translated sentence, t, the goal of QE is to learn a regression model, m such that $m(s,t)$ returns a score that represents the quality of the translated sentence.

There are many important applications of quality estimation, for example, translation companies use QE systems to identify mistranslated sentences, which would reduce post-editing costs and efforts, while online platforms use QE systems as filters to hide poorly translated sentences from end-users. Additionally, with the proliferation of mined parallel data obtained from web-crawls as a source of NMT training data (El-Kishky et al., 2020b,a), QE has become an important tool in performing quality control on translations from models trained on noisy training data.

The performance of a QE system is usually measured by the correlation between predicted QE and human-annotated QE scores. However, the predictions of QE models are primarily used to make binary decisions (Zhou et al., 2020): only translations above a certain QE threshold would be given to a human for post-edition in a translation company, or shown to the user in an online platform. Therefore, Pearson correlation might not be the best metric to evaluate the actual performance of the QE models in real-world use cases.

In recent iterations of the QE shared task at the Conference on Machine Translation (WMT) (Fonseca et al., 2019; Specia et al., 2020), the top-performing QE systems have been built on large multilingual contextualized language models that were pre-trained on huge amounts of multilingual text data. Further, these QE models are multilingual and work well in zero-shot scenarios (Sun et al., 2020). This characteristic makes them very appealing for real-life scenarios because it removes the need to train one bilingual model for every pair of languages.

However, these neural QE models contain millions of parameters and as such their memory and disk footprints are very large. Moreover, at inference time they are often more computationally expensive than the upstream neural machine translation (NMT) models, making them unsuitable for deployment in applications with low inference latency requirements or on devices with disk or memory constraints. In this paper we explore applying compression techniques to these large QE models to yield more practical, lower-latency, models while retaining state-of-the-art (SoTA) performance.

Our **main contributions and findings** are:

1. We conduct a thorough study on the efficiency

of SoTA neural QE models.

2. We shed light on the performance of recent compression techniques on a multilingual regression task and show that these techniques are inadequate for regression.

3. We empirically show that regression has a lower level of compression effectiveness than classification, on publicly available multilingual QE datasets.

4. We argue that the level of expressiveness of a regression model in a continuous range is unnecessary given the downstream applications of QE, and evaluating QE models using classification metrics would better reflect their actual performance in real-world applications.

5. We find that multilingual QE models are not as effective as bilingual QE models on both regression and binary classification, for models with higher degrees of compression.

2 Related Work

Early work on QE built models on manually crafted features extracted from the source and translated sentences, or confidence features directly from machine translation (MT) systems (Specia et al., 2009). In contrast, SoTA models are usually trained in an end-to-end manner using neural networks (Specia et al., 2018; Fonseca et al., 2019; Specia et al., 2020; Tuan et al., 2021), without the additional step of feature extraction.

The proliferation of many-to-many NMT (Fan et al., 2021; Ko et al., 2021) has motivated similar multilingual QE models. These multilingual QE models have exploited large pre-trained contextualized multilingual language models to achieve a previously-unseen level of correlation with human judgments in recent iterations of the WMT QE shared task. For example, the top-performing QE model at WMT 2019 (Kepler et al., 2019) is a neural predictor-estimator model based on multilingual BERT (Devlin et al., 2019), while the best QE models at WMT 2020 (Ranasinghe et al., 2020a; Fomicheva et al., 2020b) are regression models built on XLM-R (Conneau et al., 2019). Sun et al. (2020) find that these models generalize well across languages and training a single multilingual QE model is more effective than training a bilingual model for every language direction. Unfortunately, these models are computationally infeasible for real-world applications.

Several approaches have been proposed to address the latency issues of these large contextualized language models. Most of these work are based on **knowledge distillation** (Sanh et al., 2019; Jiao et al., 2019; Aguilar et al., 2020; Tang et al., 2019; Sun et al., 2019) where large contextualized language models (teacher) are used to supervise the training of smaller student models (Hinton et al., 2015), or **pruning**, which discards redundant model components. Some examples are model weights (Gordon et al., 2020), tokens (Goyal et al., 2020a), encoder layers (Sajjad et al., 2020) and attention heads (Michel et al., 2019).

Existing work has looked at model compression for other multilingual tasks: Tsai et al. (2019) obtained 27x speedup on multilingual sequence labeling without any significant performance degradation, while Mukherjee and Awadallah (2020) obtain 51x speedups on NER tasks while retaining 95% performance. In our experiments with QE we do not observe the same levels of compression effectiveness, suggesting that the QE is a much harder task for model compression.

A call to reframe QE as a classification problem was made by Zhou et al. (2020), based on the perspective that classification is more suitable for real-world use cases and binary classes are easier to interpret than the predicted QE scores. Our work suggests the same direction, but now from the perspective of modeling, where we empirically find that the level of expressiveness of a regression-based QE model is unnecessary and evaluating QE models using classification metrics would better reflect their actual performance in real-world applications.

3 Background and hypothesis

Current state of the art QE systems (Fomicheva et al., 2020b; Ranasinghe et al., 2020a; Sun et al., 2020). are built on XLM-R (Conneau et al., 2019), a contextualized language model pre-trained on more than 2 terabytes of filtered CommonCrawl data (Wenzek et al., 2020). As seen in Figure 1, the model concatenates a pair of source and translated sentences with a separator token in between and appends a special **CLS** token to the beginning of the concatenated string. It then converts the pre-processed string into a sequence of embedding vectors using a pre-trained embedding lookup

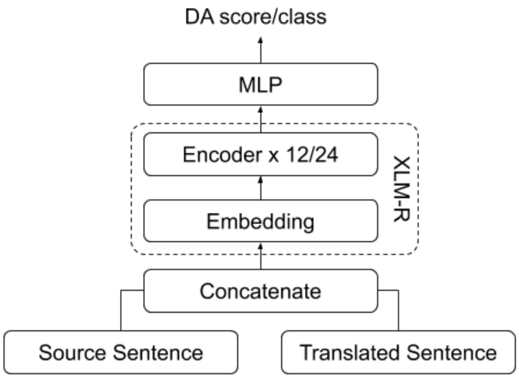


Figure 1: XLM-R based neural QE model. The base version of XLM-R has 12 encoder layers while the large version of XLM-R has 24 encoder layers.

table. The embedding vectors are then encoded using the multi-head self-attention mechanism as described in Vaswani et al. (2017). This step is repeated 12 times for the *base* version of XLM-R and 24 times for the *large* version of XLM-R. Finally, the neural QE model converts the final encoding of the **CLS** token into a QE score via a multilayer perceptron (MLP) layer.

In this work, we follow the neural architecture in Sun et al. (2020), and experiment with both **bilingual models (BL)**, where each model is trained on data from only one language direction, and **multilingual models (ML)** that are trained on the concatenated data from all language pairs available in the dataset. We choose mean squared error and binary cross-entropy as the objective functions for the regression and binary classification tasks respectively. We use AdamW (Loshchilov and Hutter, 2017) for parameter tuning and use a common learning rate of $1e^{-6}$ for all experiments.

3.1 Efficiency of neural QE models

To gain insights on the practicality of deploying the aforementioned baseline QE models in real-world situations, we gathered benchmark results on a server with 40 physical cores and 512GB of RAM. We measure the average time required to compute DA score for every sentence pair in the test sets across all seven language directions. We use a batch size of 1 and run the QE models on CPU. We highlight some of the findings in Table 1.

Memory XLM-R *base* has around 280 million parameters, and 69% of the parameters are in the embedding layer. XLM-R *large* has around 563 million parameters, which is more than 2 times the number of parameters in XLM-R *base*, and 54% of the parameters are in the encoder layers. Given that these models take up around 1-2 GB of memories on disk, they might be unsuitable for devices with small RAM capacity.

	base		***large***	
Module	**#P**	**L(ms)**	**#P**	**L(ms)**
Embedding	192M	0.7	257M	0.9
Encoder	7.1M	10.0	12.6M	15.3
MLP	2.4M	0.6	4.2M	0.8
Total	280M	122	563M	370

Table 1: Number of parameters (#P) and latency measured in milliseconds (L) for different components of the base and large version of XLM-R. For the encoder, we show the average statistics for each layer. Note that the *base* version has 12 encoder layers, while the large version has 24.

Latency Most of the computations take place in the encoder layers, where the latency of each encoder layer is 15.3 milliseconds for XLM-R *large* and 10 milliseconds for XLM-R *base*. Although the embedding layer contains a significant number of parameters, it requires much fewer computations than the encoder layers since its embedding matrices are only used as lookup tables.

Given the benchmark results in Table 1, it is clear that recent QE models based on XLM-R *large* are computationally expensive and memory intensive. At an average inference time of 370 milliseconds per sentence pair, these QE models can be even slower than the upstream MT systems, making them infeasible in real-world applications that require real-time response. As more than 98% of inference time is spent in the encoder layers of the neural QE models, we will explore model compression techniques that could reduce the number of parameters and computations in those layers.

4 Model compression techniques

Given the vast amount of work in the field of model compression, we explore three broad techniques and examine whether they could be successfully applied to compressing QE models.

4.1 Pruning

Pruning techniques are inspired by observations that large pre-trained contextualized language models might be over-parameterized for downstream tasks and most model parameters can be removed without significantly hurting model performance (Kovaleva et al., 2019).

Layer pruning Sajjad et al. (2020) demonstrated that it is possible to drop encoder layers from pre-trained contextualized language models while maintaining up to 98% of their original performance. We apply the *top-layer strategy* to XLM-R by dropping the top N encoders layers and then fine-tune the reduced neural architecture on QE datasets. We experiment with different values for N from {3, 6, 9, 12, 15, 18, 21, 23}.

Token pruning Goyal et al. (2020b) observed that token vectors start carrying similar information as they pass through the encoder layers of contextualized language models. The authors propose a method that progressively removes redundant word vectors by only keeping the top K vectors at each encoder layer based on an attention scoring mechanism. To determine the optimal value of K for each encoder layer, they add a soft extraction layer with learnable parameters in the range [0, 1] that represents the degree of usefulness for every token vector. L1 regularizers are used to optimize the weights of the parameters in the soft extraction layer. Following the original implementation, we tune a hyper-parameter that controls the tradeoff between the loss of the original tasks and the regularizers.

4.2 Knowledge distillation

Recent knowledge distillation (KD) methods (Jiao et al., 2019; Mao et al., 2020; Sanh et al., 2019) use larger BERT models (teacher) to supervise the training of smaller BERT models (student), typically with the help of the same raw text data that was used by the teacher models. Given that XLM-R was trained on more than 2 terabytes of multilingual text data, it would be computationally difficult to adapt the KD techniques to XLM-R. Instead, we experiment with a simplified KD setup inspired by Xu et al. (2020).

Module replacement We explore whether it is effective to compress N encoder layers into a single encoder layer. As seen in Figure 2, we use the top N layers of a fine-tuned QE model to supervise the training of one encoder layer in a smaller QE model. For the student QE model, we randomly initialize the target encoder layer and copy all other parameters from the teacher QE model. During training, we freeze all parameters except for the ones in the target encoder layer. The loss function is defined as the sum of 1) the mean squared error between the output of the final teacher encoder and the output of the target student encoder, and 2) the original objective function. We use sentence pairs in the MLQE dataset to train the student model and experiment with the following values for N: {2, 6, 12, 18, 23, 24}.

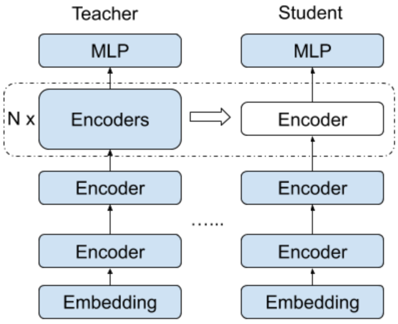


Figure 2: Module replacement: replacing N encoder layers with a single encoder layer.

5 Experimental settings

We report results on the MLQE-PE dataset using Pearson correlation for regression and F1 for classification.

5.1 QE dataset

MLQE-PE is the publicly released multilingual QE dataset used for the WMT 2020 shared task on QE (Fomicheva et al., 2020a). This dataset was built using source sentences extracted from Wikipedia and Reddit, translated to and from English with SoTA bilingual NMT systems that were trained on publicly available parallel corpora. It contains seven language pairs: the high-resource English-German (En-De), English-Chinese (En-Zh), and Russian-English (Ru-En); the medium-resource Romanian–English (Ro-En) and Estonian–English (Et-En); and the low-resource Sinhala–English (Si-En) and Nepali–English (Ne-En). Each pair of sentences was manually annotated for quality using a 0–100 direct assessment (DA) scheme as shown in table 2. A z-normalized version of these scores is used directly for **regression**.

As previously mentioned, since the most common use case of the QE is to make binary decisions based on predicted QE scores (Zhou et al., 2020), i.e, to determine whether a translated sentence is

DA score	Meaning
0-10	incorrect translation
11-29	translation with few correct keywords, but the overall meaning is different from the source
30-50	a translation with major mistakes
51-69	a translation which is understandable and conveys the overall meaning of the source but contains typos or grammatical errors
70-90	a translation that closely preserves the semantics of the source sentence
90-100	a perfect translation

Table 2: Annotation scheme used to build the MLQE-PE dataset

adequate based on a certain threshold on QE score, we also experiment with models that are directly optimized for **binary classification**. To that end, we modify the MLQE-PE dataset to predict the *acceptability* of a translation by assigning the label *not acceptable* to sentence pairs with DA scores less than some threshold values and the label *acceptable* for the remainder of the translations. The notion of acceptability is thus based on the guidelines provided for the MLQE-PE annotations in Table 2. Here, we require that a translation is understandable but not necessarily perfect, and experiment with two thresholds of ≥ 51 and ≥ 70 to signify acceptability.[1]

5.2 Evaluation metrics

Following the standard practice used by the QE research community, we measure the performance of a QE model by calculating the *Pearson Correlation coefficient* of its predicted DA scores and the actual human-annotated DA scores on a test set. Formally, let $x_i = m(s_i, t_i)$ be the DA score for a sentence pair (s_i, t_i) predicted by a QE model, m, and y_i be the actual human-annotated DA score. Then the performance of m on a test set $T = \{(s_1, t_1, y_1), (s_2, t_2, y_2), \dots (s_N, t_N, y_N)\}$ is defined as:

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}} \quad (1)$$

r is in the range $[-1, 1]$ and a better model would have a r closer to 1.

For binary classification, we use the F1 score defined as:

$$F1 = \frac{2 \times (P \times R)}{(P + R)} \quad (2)$$

where P is the precision and R is the recall. For head-to-head comparison, we also evaluate regression models using F1 by converting the predicted QE scores into binary classes using the same threshold described in previous subsection.

6 Results

The baseline results on the MLQE dataset[2] used at WMT 2020 are shown in Table 3. We keep the same train-dev-test splits as the WMT 2020 shared task[3] for all experiments. The results of our regression models are comparable to the results reported in recent work (Fomicheva et al., 2020b; Ranasinghe et al., 2020b). In general, QE models based on XLM-R *large* outperform models based on XLM-R *base*, showing that a higher number of parameters benefits the QE task. This is especially true for the regression tasks, where on average, the large models outperform the base models by 11% and 56.8% for bilingual and multilingual models respectively. However, the same levels of performance degradation are not observed on the classification tasks: On average, the large QE models only perform 3.7%/5.6% and 1.2%/7.1% better than the base QE models for bilingual and multilingual settings respectively at different thresholds. This shows that classification performance depends less on the number of model parameters and therefore we could potentially observe better compression results and more accurate model performance in real-world application if we evaluate QE with classification metrics. In the remaining of this section we explore different compression techniques on both regression and classification to test this hypothesis.

[1] We acknowledge that this threshold may be application-dependent. In other cases, where a higher level of quality is desired (e.g. for *knowledge dissemination*), a ≥ 90 threshold might be more appropriate.

[2] https://github.com/sheffieldnlp/mlqe-pe

[3] http://www.statmt.org/wmt20/quality-estimation-task.html

Model	Type	Regression corr. (ρ)		Classification F1 (≥ 51 / ≥ 70)	
		Base	**Large**	**Base**	**Large**
BERGAMOT -LATTE	BL ML	- -	0.67 0.69	- -	- -
TransQuest	BL ML	- -	0.69 0.67	- -	- -
This work	BL ML	0.63 0.44	0.70 0.69	0.82/0.72 0.83/0.70	0.85/0.76 0.84/0.75

Table 3: Baseline results of bilingual (BL) and multilingual (ML) models on the MLQE-PE dataset for both regression and binary classification at different thresholds. Results are averaged over 5 different runs and 7 language directions. Our results are comparable to the results of BERGAMOT-LATTE (Fomicheva et al., 2020b) and TransQuest (Ranasinghe et al., 2020b), the top-performing systems at WMT 2020 QE shared task.

6.1 Compression techniques results

We apply each compression technique to bilingual QE models using XLM-R *large*, and average the results over 5 different runs for every language direction. We then compute the average speedup of every compressed model to its original model. The performance drop against speedup plots[4] of regression models and classification models are shown in Figure 3 and Figure 4 respectively.

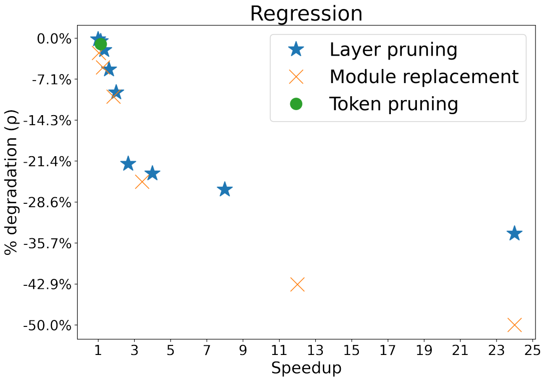


Figure 3: % degradation (ρ) vs. speedup for regression models.

Based on the results of our experiments, we find that layer pruning outperforms module replacement, especially at higher speedup. For example, at around 24x speedup, layer pruning outperforms module replacement by 31.8% and 2.9% for regession and classification respectively. For token pruning, we find that it does not offer any significant benefit over the other 2 compression techniques. Further, the token-pruned models that we tuned to the best of our efforts are conservative and prefer model accuracy over speedup: 14.4% and 26.4% faster than the original QE models with $< 1\%$ performance degradation for regression and classification respectively. In the remainder of this paper, we will focus on running model compression experiments with the layer pruning strategy.

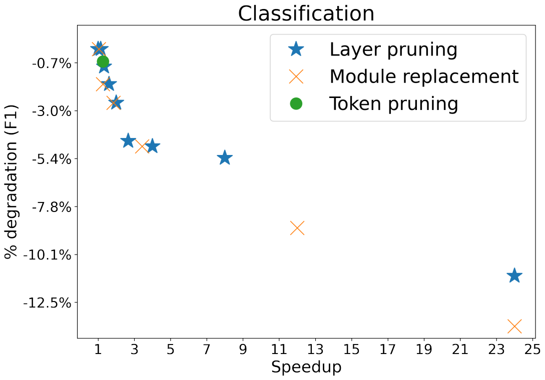


Figure 4: % degradation ($F1$) vs. speedup for classification models (≥ 51).

6.2 Compression techniques are inadequate when evaluating QE as a regression task

As seen in Figure 3, the compressed regression-based QE models have a performance degradation of more than 9% at just 2x speedup. The performance degradation worsens to 23% at 4x speedup and 34% at 24x speedup. These results are significantly worse than the numbers reported on other NLP tasks: Jiao et al. (2019) reported a 9.4x speedup with 96.8% performance retention on GLUE benchmark (Wang et al., 2018) , Mukherjee and Awadallah (2020) reported 51x speedup with 95% performance retention on a multilingual NER task with 41 languages and (Wang et al., 2020) reported 2x speedup with more than 99% performance retention on SQUAD 2.0 (Rajpurkar et al., 2018).

Our results suggest that QE datasets might not suffer from the same degree of *overparameterization problem* Kovaleva et al. (2019) observed in other NLP datasets. We hypothesize that the performance of QE *depends heavily on the large number of parameters in XLM-R*. This is further supported by the results in Table 3, where the XLM-R *large* models, with around twice the number of parameters in XLM-R base models, outperform the latter by more than 11%.

[4] averaged over 7 language directions

6.3 Better compression results when evaluating QE with classification metric

	En-De	En-Zh	Ru-En	Ro-En
Regr.	-8.0%	-13.7%	-8.8%	-4.5%
Class. ($\geq$ 51)	0.0%	0.0%	-1.1%	-1.1%
Class. ($\geq$ 70)	0.0%	0.0%	-2.8%	-3.9%
	Et-En	**Si-En**	**Ne-En**	**Average**
Regr.	-14.1%	-9.2%	-10.3%	-9.8%
Class. ($\geq$ 51)	-6.0%	-1.4%	-12.7%	-3.2%
Class. ($\geq$ 70)	-12.1%	-0.1%	-9.1%	-4.0%

Table 4: Performance drops of *base* QE models with respect to *large* QE models for regression (top) and binary classification at different thresholds.

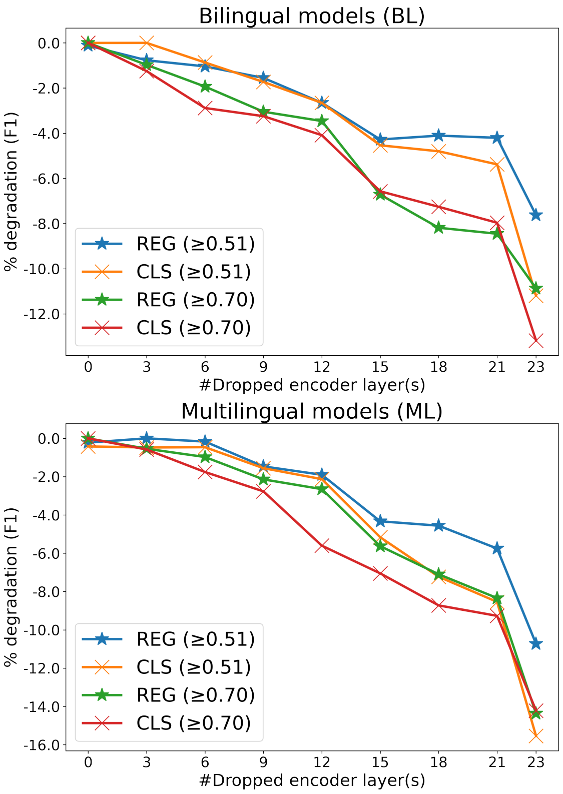
In Table 4, we compute the relative percentage of performance drops when using XLM-R *base* instead of XLM-R *large*. We observe that the base QE models perform significantly worse than the large QE models on regression tasks, with an average performance drop of 9.8% over 7 language directions. In contrast, the average performance drops of 3.2% and 4.0% for binary classification at different thresholds are significantly lower, with less than 1.5% drop on majority of 7 language directions. Figure 4 also shows that it is possible to retain 94.6% performance with 8x speedup and 88.8% with 24x speedup.

Comparing these to the results in the previous subsection, we observe significantly less model degradation when evaluating the QE models with classification metric instead of regression metric.

6.4 Regression or binary classification?

In practice, the predicted QE scores from regression models are primarily used to make binary decisions based on predetermined thresholds. To test whether it would be better to directly optimize QE models for binary classification, we convert the predicted DA scores from regression models into binary classes using the same threshold as the one in Section 5.1 and then compute F1 scores.

As shown in Figure 5, the F1 performances of regression models are comparable to the classification models at lower compression settings. However, the regression models start to outperform the classification models when we further compress the models by dropping more encoder layers. Our results show that in our case of using thresholds of $\geq$ 51 and $\geq$ 70 with layer pruning as our model compression strategy, optimizing QE as a regression task and then converting the predicted DA scores into binary classification labels seems better than directly optimizing QE as a binary classification task.


Bilingual models (BL)

Multilingual models (ML)

Figure 5: Comparison between regression models (REG) and classification models (CLS) for different layer pruning configurations. All models are based on XLM-R *large*. Top shows the results for bilingual models and bottom shows the results for multilingual models.

6.5 Pearson correlation is misleading

Looking at the results in Figures 3, 4 and 5, it is apparent that the drastic drops in Pearson correlation at higher compression settings do not translate to equivalent degrees of performance degradation in terms of F1. For example, at approximately 24x speedup, the bilingual regression models suffer an average performance degradation of 33.9% in Pearson correlation, which is significantly higher than the 7.5% performance degradation observed in binary classification.

The explanation lies in the fact that Pearson correlation penalizes predictions that do not follow the same linear trend as the gold DA scores. However, getting the linear trend right is not useful for binary classification, i.e., a predicted DA score is correct as long as it falls on the right side of the decision boundary, regardless of the degree of closeness to

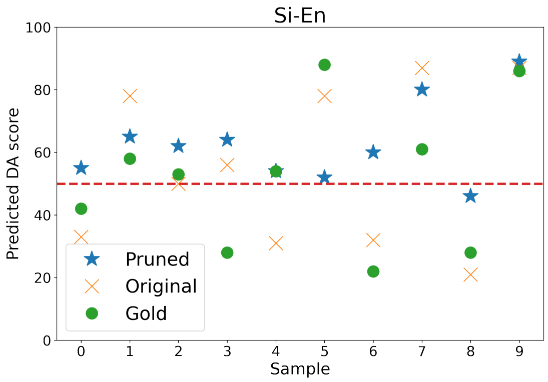


Figure 6: Predicted DA scores from a pruned and an original QE model, as well as the gold labels for 10 random samples from Sinhala-English test set. Pearson correlation is 0.47 and 0.64, and F1 is 0.72 and 0.73 for the pruned and original models respectively. The red dashed line is the decision boundary.

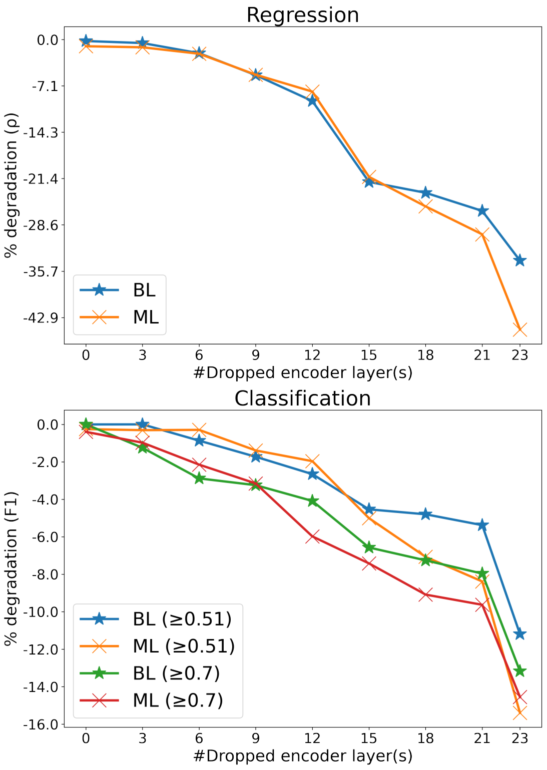


Figure 7: Comparison of multilingual models (ML) against bilingual models (BL) for different layer pruning configurations.

the gold DA score. This phenomenon is illustrated in Figure 6 where the predicted scores of a QE model based on XLM-R *large* are generally closer to the gold DA scores than the predicted scores of a layer pruned QE model, which explains their 36.2% difference in Pearson correlation. However, the two models obtain comparable F1 scores because their predicted DA scores generally fall on the same sides of the decision boundary.

Our results suggest that larger QE models are required to make more accurate QE score predictions that have a higher Pearson correlation with human-annotated QE scores. However, the higher accuracy and more optimal ordering of the predicted QE scores do not necessarily contribute to higher accuracy when making binary judgments. We believe that Pearson correlation is a misleading evaluation metric that deviates from the use cases in real-world settings. Chasing higher Pearson correlation could lead us down the path of building larger models that are computationally infeasible, yet having better exact DA predictions is not necessarily useful for better binary classification.

Based on these results, we recommend that *classification metrics are used to evaluate the effectiveness of compressed QE models.*

6.6 Does model compression affect the performance of multilingual QE?

We plot the averaged results of bilingual and multilingual models for different configurations of layer pruning in Figure 7. We observe that at lower compression settings with less than or equals to 12 dropped layers, the results of the multilingual models are generally comparable to the results of the bilingual models and this corroborates the finding that multilingual QE model generalize well across languages (Sun et al., 2020). However, at higher compression settings with more than 12 dropped layers, the multilingual models start to lose their effectiveness, underperforming the bilingual models significantly. We hypothesize that the multilingual neural models no longer have enough model capacities for 7 different language directions beyond a certain degree of model compression.

7 Conclusions

This paper presents a thorough study on the efficiency of SoTA neural QE models and explores whether recent compression techniques can be successfully applied to reduce the size and improve the latency of these models. Our experimental results show that recent compression techniques are inadequate for regression as we observe significant performance degradation on the QE task with little improvement in model efficiency. We argue that the level of expressiveness of a QE model in a continuous range is unnecessary since the outputs of the QE model are usually used to make binary decisions. Our results show that it is more appropriate to reframe the QE task as a classification problem, and evaluating QE models using classification metrics would better reflect their actual performance in real-world applications. This enables us to achieve SoTA performance with tiny and efficient models.

Our experimental results suggest that compressing large neural QE models for the QE regression task remains a challenging problem, especially in the case of multilingual models, where they start showing higher degrees of performance degradation than their bilingual counterparts.

References

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. 2020. Knowledge distillation from internal representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7350–7357.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco 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. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020a. A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969.

Ahmed El-Kishky, Philipp Koehn, and Holger Schwenk. 2020b. Searching the web for cross-lingual parallel data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2417–2420.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020a. Mlqe-pe: A multilingual quality estimation and post-editing dataset.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020b. 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.

Erick Fonseca, Lisa Yankovskaya, André FT 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 (Volume 3: Shared Task Papers, Day 2)*, pages 1–10.

Mitchell Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing bert: Studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 143–155.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020a. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pages 3690–3699. PMLR.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020b. PoWER-BERT: Accelerating BERT inference via progressive word-vector elimination. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699, Virtual. PMLR.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Fabio Kepler, Jonay Trénous, Marcos 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.

Wei-Jen Ko, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Naman Goyal, Francisco Guzmán, Pascale Fung, Philipp Koehn, and Mona Diab. 2021. Adapting high-resource nmt models to translate low-resource related languages without parallel data. *arXiv preprint arXiv:2105.15071*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yihuan Mao, Yujing Wang, Chufan Wu, Chen Zhang, Yang Wang, Yaming Yang, Quanlu Zhang, Yunhai Tong, and Jing Bai. 2020. Ladabert: Lightweight adaptation of bert through hybrid model compression. *arXiv preprint arXiv:2004.04124*.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *arXiv preprint arXiv:1905.10650*.

Subhabrata Mukherjee and Ahmed Hassan Awadallah. 2020. Xtremedistil: Multi-stage distillation for massive multilingual models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2221–2234.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020a. Transquest at wmt2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*.

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*, pages 5070–5081.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. Poor man's bert: Smaller and faster transformer models. *arXiv preprint arXiv:2004.03844*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

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, Varvara Logacheva, Ramón Astudillo, and André FT Martins. 2018. Findings of the wmt 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 689–709.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *13th Annual Conference of the European Association for Machine Translation*, pages 399–410.

Shuo Sun, Marina Fomicheva, Frédéric Blain, Vishrav Chaudhary, Ahmed El-Kishky, Adithya Renduchintala, Francisco Guzmán, and Lucia Specia. 2020. An exploratory study on multilingual quality estimation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 366–377, Suzhou, China. Association for Computational Linguistics.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4314–4323.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical BERT models for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3632–3636, Hong Kong, China. Association for Computational Linguistics.

Yi-Lin Tuan, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Francisco Guzmán, and Lucia Specia. 2021. Quality estimation without human-labeled data. *arXiv preprint arXiv:2102.04020*.

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.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4003–4012.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. Bert-of-theseus: Compressing bert by progressive module replacing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7859–7869.

Junpei Zhou, Ciprian Chelba, Yuezhang, and Li. 2020. Practical perspectives on quality estimation for machine translation.