# MetricX-23: The Google Submission to the WMT 2023 Metrics Shared Task

**Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh,** and **Markus Freitag**
Google
{jjuraska,marafin,dandeutsch,adisid,mahdim,freitag}@google.com

## Abstract

This report details the MetricX-23 submission to the WMT23 Metrics Shared Task and provides an overview of the experiments that informed which metrics were submitted. Our 3 submissions—each with a quality estimation (or reference-free) version—are all learned regression-based metrics that vary in the data used for training and which pretrained language model was used for initialization. We report results related to understanding (1) which supervised training data to use, (2) the impact of how the training labels are normalized, (3) the amount of synthetic training data to use, (4) how metric performance is related to model size, and (5) the effect of initializing the metrics with different pretrained language models. The most successful training recipe for MetricX employs two-stage fine-tuning on DA and MQM ratings, and includes synthetic training data. Finally, one important takeaway from our extensive experiments is that optimizing for both segment- and system-level performance at the same time is a challenging task.[1]

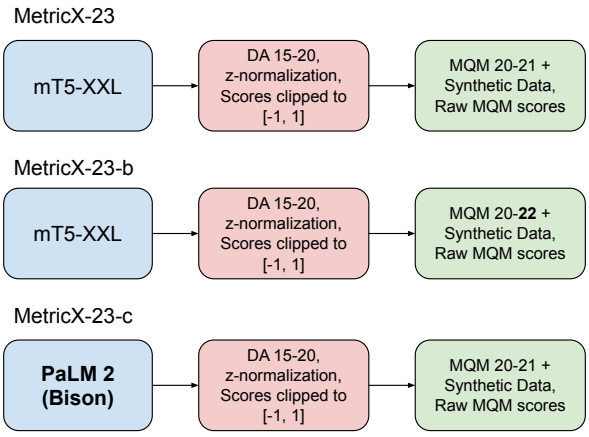



Figure 1: A high-level summary of the 3 different submissions to the WMT'23 Metrics Shared Task. MetricX-23-b and MetricX-23-c differ from the primary submission in that the "b" version is finetuned on MQM 2022 data in addition to 2020 and 2021, and the "c" version uses PaLM 2 as its pretrained language model (differences in bold). Each of the submission also includes a QE variant that follows the same training procedure.

## 1 Introduction

Automatic evaluation metrics are critical to the development of machine translation (MT) systems. They are the most frequently used method for comparing two MT systems and deciding which generates higher quality translations. Each year, the Conference on Machine Translation (WMT) runs a Metrics Shared Task to benchmark the quality of state-of-the-art evaluation metrics (Freitag et al., 2022). Meta-evaluating metrics by measuring how well they correlate to human ratings of translation quality is critical for understanding the extent to which automatic evaluations of MT systems are trustworthy.

This report details the MetricX-23 submission to the Metrics Shared Task. MetricX is a learned regression-based metric that is trained to predict a floating point score that represents the quality of a candidate translation. This year, we made 3 different submissions to the shared task that vary in the training data that is used for finetuning and which pretrained language model is used for initialization. Our primary submission, denoted MetricX-23, is based on the mT5 encoder-decoder language model (Xue et al., 2021), which is further finetuned on direct assessment (DA) ratings, MQM data (Lommel et al., 2014; Freitag et al., 2021), and synthetic data. Our contrasting submission, MetricX-23-b, includes additional MQM data, and MetricX-23-c finetunes the PaLM 2 language model (Anil et al., 2023) instead of mT5. Each of the 3 submissions has a reference-based and quality estimation (QE, or reference-free) version.

Figure 1 contains a high-level overview of the training recipe that we used for our submissions. In

[1]Our code and mT5-based models can be found at https://github.com/google-research/metricx.



*Proceedings of the Eighth Conference on Machine Translation (WMT)*, pages 756–767
December 6–7, 2023. 

order to arrive at the metrics that were ultimately submitted to the shared task, we ran various experiments that are detailed in this report. The key takeaways from those experiments include:

1. Training on $z$-normalized DA scores instead of the raw scores tends to be a trade-off between segment- and system-level performance;
2. Training on raw MQM ratings is better than $z$-normalized ratings;
3. Training on DA data followed by MQM data yields a better metric than on either type of data individually;
4. Synthetic data is necessary for the metric to learn to score the reference against itself higher than a machine translation against the reference;
5. Metric performance improves significantly as the size of the pretrained language model increases.

## 2 Metric Descriptions

The MetricX-23 metrics that were submitted to the Metrics Shared Task are all learned regression-based metrics that are trained to predict a floating point number that represents the quality of a given translation.

The input to the reference-based metrics is the candidate translation (hypothesis) and reference segments—each with a corresponding prefix ("candidate:" and "reference:", respectively)—concatenated together. The combined input is encoded by the model, and then the metric uses the encoding to predict a score. This stands in contrast to COMET-style metrics in which the hypothesis and reference are encoded separately, then combined in order to predict a score (Rei et al., 2020). The QE variants use the source segment instead of the reference, with the prefix changed to "source:".

We use two different network architectures for different versions of the metric. The choice of architecture depends on which pretrained language model is used to initialize the model.

The first architecture is based on the encoder-decoder mT5 language model (Xue et al., 2021). The input is encoded by the encoder, then the output logit from an arbitrary token in the vocabulary distribution from the first step of decoding is selected to represent the score for the hypothesis and trained accordingly.[2] In practice, we found that this method for using the pretrained weights for both the encoder and decoder worked better than using a regression head on top of the encoder and discarding the decoder.

The second architecture is the prefix language model based on Transformer (Vaswani et al., 2017) used by the PaLM 2 model (Anil et al., 2023). We augment the architecture by adding a feedforward regression layer on top of the input encoding. The output from the feedforward layer is trained to predict the translation quality score.

Both types of model are trained with a mean squared error (MSE) loss function. Further implementation details related to checkpoint selection, optimization, etc., can be found in §3.3. Information related to training data, label normalization, etc., can be found in §4.

## 3 Experimental Setup

### 3.1 Training and Evaluation Data

The two data sources that are primarily used to train and meta-evaluate MT metrics are the direct assessment (DA) data and Multi-dimensional Quality Metrics (MQM; Lommel et al., 2014; Freitag et al., 2021) that have been collected by WMT over the years, and both of which are publicly available. We use the DA data for training and the MQM data for both training and evaluation.

The DA judgments come from non-expert annotators that score the quality of a translation on a scale from 0 to 100. Often, the scores are $z$-normalized per rater in order to better compare across raters, since each rater may have a different rating strategy despite using the same scale. We experiment with using different subsets of the DA data from 2015 to 2021, as well as using the raw rating or $z$-normalized rating as the ground-truth quality score.

In contrast, the MQM ratings are done by professional raters. Each rater marks specific spans of text within a translation that contain an error, and label that error with a severity level and category. Each error receives a weight based on its severity and category. A segment's MQM score is the sum of the error weights in the segment. Our experiments use the MQM data collected by WMT

[2]The specific token to use can be chosen arbitrarily from the vocabulary, but the same token is then used throughout the training and inference. In our case, we opted for one of the `<extra_id_**>` tokens reserved in mT5's vocabulary.

from 2020 to 2022. The 2022 ratings are our primary evaluation dataset, and all correlations that we report are calculated on this dataset.

Our experiments additionally leverage a metrics challenge set, DEMETR (Karpinska et al., 2022), for metric meta-evaluation. DEMETR is a collection of paired translations that probe a metric's ability to correctly model different phenomena. The pairs of translations differ by some linguistic phenomena, with one translation assumed to be higher-quality than the other, and the metric is evaluated on how often it correctly ranks the two translations. We use DEMETR to evaluate how frequently the reference translations evaluated against itself receives a higher score than a machine translation evaluated against the same reference.

### 3.2 Meta-Evaluation

In our experiments, we calculate the metrics' agreements with human judgments of translation quality using four different correlations. At the system-level, we use both Pearson's $r$ and pairwise accuracy (Kocmi et al., 2021). System-level Pearson's $r$ captures how strong the linear relationship is between the metric and human scores for MT systems. Pairwise accuracy evaluates a metric's ranking of MT systems by calculating the proportion of all possible pairs of MT systems that are ranked the same by the metric and human scores.

At the segment-level, we use the no-grouping Pearson's $r$ and the group-by-item pairwise accuracy with tie calibration as described by Deutsch et al. (2023).[3] The no-grouping Pearson's $r$ quantifies the linear relationship between the metric and human scores across all possible translations from every system and document. The group-by-item pairwise accuracy calculates the proportion of all possible pairs of translations for the same input segment that are ranked the same or predicted to be a tie by the metric and human, then averages the accuracies over all possible input segments. Since regression-based metrics rarely predict ties and the segment-level pairwise accuracy rewards correct tie predictions, Deutsch et al. (2023) uses a procedure called tie calibration that automatically introduces ties into metric scores by introducing an $\epsilon$ such that any two translations with a difference in metric score less than $\epsilon$ are considered to be tied.

[3]We chose this pairwise accuracy over Kendall's $\tau$, which has typically been used in WMT Metrics evaluation, for its superior handling of ties in metric scores.

### 3.3 Implementation Details

Our metrics are implemented with TensorFlow (Abadi et al., 2015) and the T5X library (Roberts et al., 2022). Each training run uses 64 TPUs and trains for a maximum of 10K steps with a batch size of 512 on the DA data, or 3K steps with a batch size of 256 on the much smaller MQM dataset. Adafactor is used for optimization (Shazeer and Stern, 2018). Checkpoint selection is done by selecting the model that has the highest segment-level pairwise accuracy after tie calibration on the en-de and zh-en language pairs.

We are publicly releasing our mT5-based submissions, converted from TensorFlow to PyTorch (Paszke et al., 2019) checkpoints, along with corresponding code to use them to predict translation quality scores.

## 4 Experimental Results

We made three different submissions to the shared task, each with a reference-based and QE variant:

1. **MetricX-23(-QE):** An mT5-XXL model that was finetuned on a combination of DA data from 2015–2020, MQM data from 2020–2021, and synthetic data.
2. **MetricX-23(-QE)-b:** The same as MetricX-23(-QE) except we additionally included MQM data from 2022.
3. **MetricX-23(-QE)-c:** The same as MetricX-23(-QE) except it is a finetuned PaLM-2 Bison model.

An overview of these submissions is shown in Figure 1. In the rest of this section, we describe the experimental results that led us to these submissions.

The experiments in the process of determining the best training recipe were performed with mT5-XL (3.7B parameters), but our final submissions then use the XXL variant with 13B parameters. All results are reported as the mean of 3 independent runs, along with the standard deviation across the runs, unless stated otherwise.

### 4.1 Training Data

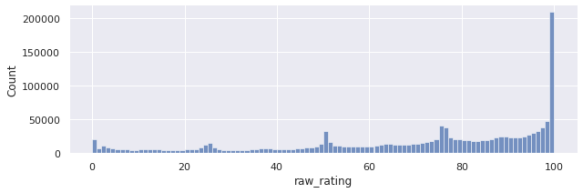
We start by determining which of the data available from previous WMT Metrics Shared Tasks is useful for training our metric, and whether it is beneficial to perform any transformations of the ratings before using them for training.

#### 4.1.1 DA Ratings

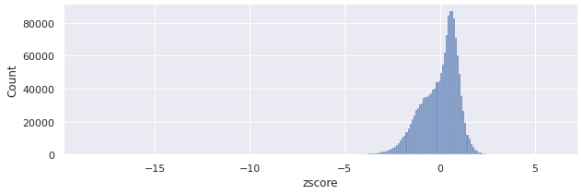
DA ratings have been used for scoring candidate translations in the shared task since 2015. To obtain DA ratings, human annotators were asked to provide an integer score on a scale of 0 to 100 for a translation produced by an MT system, given a reference translation produced by an expert translator.[4] There are over 2M raw DA ratings available from the years between 2015 and 2021, spanning 40 different language pairs. The total number of ratings drops to ca. 1M when ratings for the same segment (from different raters) are aggregated, which we do in order to avoid providing the model with different signals for the same translation.

Since the DA ratings come from hundreds of different raters, the WMT Metrics Shared Task organizers typically $z$-normalize them per rater before using them as the ground truth for metric evaluation. This is to make the ratings more comparable across different raters, considering some of them can be very strict, others lenient, and some can use the whole rating scale, while others just a narrow range of it. Hence a DA rating of, say, 50 can end up being used for translations of widely varying quality. The normalization ensures that the mean of each rater's ratings is 0 and the standard deviation is 1. These official normalized DA ratings, which we refer to as $z$-scores, are available along with the *raw* DA ratings in the data from the shared tasks.

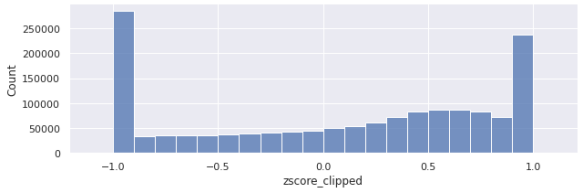
**Score Normalization.** In our first experiment, we compare the performance of our metric finetuned on the raw DA ratings and on the $z$-scores on different subsets of the DA data. As the first two rows of Table 1 show, using $z$-scores results in an overall weaker performance, but a drastic improvement on segment-level Pearson's $r$ (42.93 vs. 38.83 for zh-en). Thus, picking between using raw ratings or $z$-scores for training the metric comes down to the preference between high *system*-level or high *segment*-level performance. Given the models' performance on the system-level metrics is already relatively high (between 80 and 99), we choose to use $z$-normalized DA ratings over their raw counterparts for our submissions. Nevertheless, as we show in Section 4.2, adding synthetic data during finetuning can restore some of the system-level performance. Moreover, our later experiments in Section 4.3 demonstrate that further finetuning on MQM ratings is more effective using a model first finetuned on DA z-scores than raw DA ratings.




(a) Raw DA ratings.




(b) DA $z$-scores.




(c) Clipped DA $z$-scores.

Figure 2: WMT15-20 DA rating distributions after $z$-normalization and after clipping to $[-1.0, 1.0]$, compared to the raw rating distribution.

**Score Clipping.** Normalized and raw ratings follow very different distributions, as depicted in Figure 2. The raw rating distribution is relatively flat across the whole range with a large spike at the maximum value, i.e., 100. In contrast, the $z$-score distribution ranges roughly from $-17$ to 5, with the majority of the mass between $-1$ and 1, a sharp peak around 0.65, and a long tail on the negative side. In order to prevent the model from putting too much weight on the outliers during training, and not learning to differentiate well between translations scored around zero, we propose clipping the scores to be in the $[-1.0, 1.0]$ range.[5] This creates a spike on the right end, similar to that observed on the raw rating distribution, but also a spike on the left end, similar in magnitude to the other spike (see Figure 2c). Finetuning a model on the clipped $z$-scores results in segment-level performance gains compared to the unmodi-

[4] Technically, some of the translation data is "target-original", meaning that the reference (target) is the text originally to be translated, and the source is the translation. This is the case for some language pairs in the DA data from earlier years.

[5] MSE loss magnifies errors in predictions greater than 1 and shrinks errors smaller than 1 relative to the absolute difference.

| | SEG pairwise acc. | | SEG Pearson | | SYS pairwise acc. | | SYS Pearson | |
|---|---|---|---|---|---|---|---|---|
| | en-de | zh-en | en-de | zh-en | en-de | zh-en | en-de | zh-en |
| **15-20 raw** | **59.98** ±0.03 | **51.63** ±0.15 | 43.51 ±0.77 | 38.83 ±0.44 | **83.33** ±1.28 | **89.38** ±1.68 | **90.79** ±1.39 | **98.51** ±0.35 |
| **15-20 $z$** | 59.57 ±0.20 | 51.26 ±0.40 | 47.08 ±0.23 | 42.93 ±0.13 | 81.62 ±1.48 | 85.35 ±1.68 | 85.60 ±1.00 | 98.02 ±0.20 |
| **15-20 $z$ clipped** | 59.77 ±0.21 | 51.45 ±0.15 | 46.89 ±0.32 | **45.05** ±0.24 | 79.49 ±0.00 | 86.08 ±1.68 | 83.24 ±0.97 | 97.28 ±0.11 |
| **15-21 $z$ clipped** | 59.70 ±0.10 | 50.76 ±0.30 | **47.76** ±1.02 | 43.26 ±0.70 | 80.34 ±1.96 | 85.35 ±1.68 | 85.21 ±1.67 | 97.31 ±0.82 |

Table 1: Performance of MetricX that is initialized with mT5-XL, finetuned on DA ratings in different ways. "15-20" indicates that ratings from the years 2015 through 2020 were used, "$z$" indicates $z$-normalized scores, and "clipped" denotes experiments with the scores clipped to the $[-1.0, 1.0]$ range. Note that the data from 2015 is a small set of 2,500 $z$-normalized ratings only, so there is technically no data from 2015 in the "15-20 raw" setting.

fied $z$-scores, though further sacrifices some of the system-level performance (compare rows 2 and 3 in Table 1). Compared to finetuning on raw ratings (row 1 vs. row 3), there is up to 16% increase on the segment level (Pearson on zh-en), at the cost of an up to 9% drop in system-level performance (Pearson on en-de).

**Data Selection.** The DA ratings we used in our experiments thus far were from 2015 to 2020. Saving the WMT22 data for the validation set, we have the option of adding the WMT21 DA ratings to the training set. Doing this leads to moderate gains in system-level en-de performance, yet an equal, if not bigger, performance drop in zh-en across all metrics (compare the last two rows in Table 1). We also tried excluding earlier years of DA ratings, such as 2015–2017 or 2015–2018, since up until 2018 a half of the translations in the data were target-original (Barrault et al., 2019), and all DA annotations were reference-based (Ma et al., 2019), as opposed to source-based, such as is the case with a good part of the DA annotations from 2019 onward, and all of the MQM ratings. We hypothesized that the older data might thus be providing some low-quality signals to the model during training, negatively affecting the performance. Nevertheless, the model seems to prefer additional training data, even if of a mixed quality, as we consistently observed a slight drop in performance after excluding the earlier years, especially on system level. Therefore, the rest of the experiments uses DA data from 2015 to 2020.

#### 4.1.2 MQM Ratings

MQM ratings have been collected in the context of the WMT Metrics Shared Task only since 2020. Due to the MQM annotation being significantly more labor-intensive, there is significantly less data collected per year than using the DA methodology. In fact, MQM ratings are only available for three language pairs, namely en-de, zh-en and en-ru. Since we reserve the ratings from 2022 for validation, we are left with only two years worth of MQM data to use for training, which amounts to approximately 114K ratings. For both training and evaluation we negate the MQM scores, changing thus the range to $[-25, 0]$, so that the score corresponding to no errors in the translation would be the highest value, as opposed to the lowest value, in the range. In the following paragraphs, we discuss our experiments with finetuning a model on MQM ratings only, in order to see if the model learns anything different than what it learns from the DA ratings.

**Data Selection.** We start this set of experiments with finetuning our model on the combination of '20 and '21 MQM ratings, and confirming that there is an added benefit to it over finetuning on just one of the years. As demonstrated by the first two rows in Table 2, finetuning on the '20 and '21 data individually leads to very different performance across the set of metrics. Using just the '20 MQM data by itself, our model achieves better segment-level performance than using all of the DA data, however, it is the opposite case on system level. As for training on '21 data only, the performance is significantly worse overall despite a slightly better segment-wise pairwise accuracy on en-de. Although this may suggest that the '21 MQM data is of a lower quality, it may also simply be the consequence of the '21 data having only a little over a third of the number

| | SEG pairwise acc. | | SEG Pearson | | SYS pairwise acc. | | SYS Pearson | |
|---|---|---|---|---|---|---|---|---|
| | en-de | zh-en | en-de | zh-en | en-de | zh-en | en-de | zh-en |
| **20 raw** | 59.91 ±0.24 | 52.52 ±0.13 | **48.03** ±1.56 | 54.43 ±0.58 | **77.35** ±2.67 | **84.98** ±0.63 | **75.30** ±0.71 | 85.46 ±1.76 |
| **21 raw** | 60.43 ±0.12 | 51.92 ±0.04 | 42.21 ±1.69 | 54.07 ±1.00 | 69.66 ±1.96 | 82.42 ±1.10 | 56.62 ±4.67 | 85.92 ±0.40 |
| **20-21 raw** | **60.77** ±0.17 | **52.72** ±0.24 | 47.59 ±1.82 | **55.41** ±0.43 | 73.93 ±0.74 | 83.52 ±1.10 | 72.70 ±0.15 | 85.68 ±0.70 |
| **20-21 $z$** | 59.74 ±0.33 | 51.98 ±0.82 | 45.84 ±0.99 | 54.34 ±0.75 | 74.79 ±0.74 | 82.78 ±2.29 | 70.79 ±2.79 | **86.14** ±1.06 |

Table 2: Performance of MetricX initialized with mT5-XL on the WMT22 MQM dataset, finetuned on different subsets of MQM ratings. "$z$" indicates $z$-normalized scores.

of ratings in the '20 data.[6] Finetuning on both the '20 and the '21 data combined, however, outperforms both of the individual years on segment-level metrics, while it lands somewhere in between according to system-level metrics (see row 3 in the table). Hence, we use the MQM ratings from both years in all of our subsequent experiments.

**Score Normalization.** We observed on the DA data that $z$-normalization has certain benefits, so we experiment with it even on the MQM data. We perform the $z$-normalization ourselves in the same way the shared task organizers normally do for the DA ratings, and compare a model finetuned on these $z$-scores to one finetuned on the raw MQM ratings. It is clear from the comparison of rows 3 and 4 in Table 2 that $z$-normalization drags the performance down, especially on the segment level.[7] One possible explanation could be that the normalization has a negative effect here because of the raters having annotated different sets of documents each, and the set of raters being very small at the same time. It could also be that $z$-normalization is actually not a very practical transformation of training labels for this task, yet it helps in case of DA ratings, which are of a much lower quality.[8] The metrics are evaluated against raw MQM ratings, so $z$-normalization during training could negatively impact its Pearson correlation at test time. At any rate, based on this result, we opt for the raw MQM ratings when finetuning our models henceforth.

[6] In the '20 MQM data, candidate translations have multiple ratings, so we also experimented with averaging them, but that, somewhat surprisingly, resulted in a consistently lower performance across the board.

[7] We verified that this is the case both with and without aggregating the ratings of the same translations by multiple raters in the '20 data.

[8] For instance, $z$-normalization discounts the ratings of raters who gave the same score, e.g., 100, to most of the translations they rated.

### 4.2 Adding Synthetic Data

Using the DEMETR challenge set, we discovered that training MetricX on either the DA or the MQM dataset does not teach it to reliably score a translation that exactly matches the reference higher than or equal to a machine translation, which should be a basic sanity check for an evaluation metric. In order to fix this behavior, we create simple synthetic examples where the reference is copied as the candidate translation and the label is set to the maximum score. Depending on the training set these synthetic examples are used along with, the labels may need to be rescaled to ensure they correspond to the maximum score, e.g., 100 when used with raw DA ratings or 0 with MQM ratings. Since $z$-scores do not have a maximum value, per se, there is no straightforward way of incorporating this synthetic data into such a training set. Clipped $z$-scores make this trivial though, which is another argument for training MetricX on clipped DA $z$-scores instead of the full range of $z$-scores. We use all of the references across all language pairs in the DA data between 2015 and 2022 to construct this synthetic dataset, which amounts to a little over 180K unique examples.

We also prepare a second synthetic dataset with the opposite type of examples, that is, ones that have no candidate translation and therefore a label corresponding to the minimum score (i.e., 0 for DA, and $-25$ for MQM). It is created using the same data as the other synthetic dataset, only instead of copying the reference, we set all of the translations to an empty string, resulting in the same number of synthetic examples.

Next, we perform experiments to determine what the minimum ratio of synthetic examples to regular training examples is with which a high accuracy on

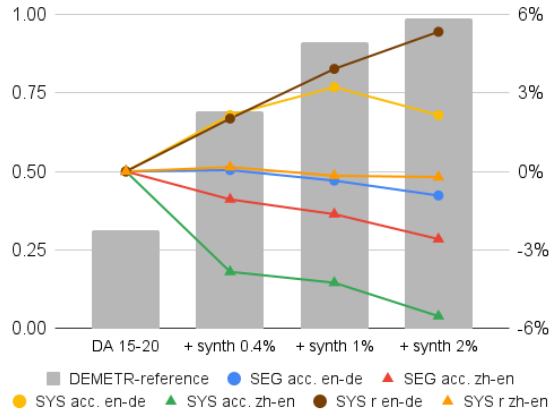



Figure 3: Effects of adding different proportions of synthetic data to the DA 15-20 training set (using clipped $z$-scores). The gray bars indicate the DEMETR-reference score (with the scale on the left y-axis), while the lines show the relative increase or decrease of the correlation metrics w.r.t. using no synthetic data (with the scale on the right y-axis). "Acc." stands for pairwise accuracy, and "r" denotes Pearson's $r$. Segment-level Pearson's $r$ followed similar trends to segment-level accuracy, and was therefore omitted for better readability.

the DEMETR-reference metric can be achieved.[9] Figure 3 illustrates the effects of adding the synthetic data to the "DA 15-20 $z$ clipped" training set in the following proportions: $0.4\%$, $1\%$, and $2\%$ of the DA examples.[10] As we can see by looking at the gray bars, the synthetic data has the desired effect of bringing the accuracy of correctly scoring a reference higher than a machine translation from a mere $31\%$ to almost $100\%$. However, this comes at a cost. With the increasing amount of synthetic data, the segment-level performance slowly degrades for both language pairs (see the blue and red lines in the plot). On the system level side, we see mixed results: a big improvement for en-de (see the yellow and brown lines) and a comparable drop in performance for zh-en, though only according to pairwise accuracy (see the orange and green lines). As such, we find adding $1\%$ of synthetic examples to the DA data a good compromise between achieving relatively high DEMETR scores and maintaining good performance on the segment- and system-level correlation metrics. For the actual scores, we refer the reader to row 1 in Table 3.

We carried out a similar study with the MQM 20-21 training data, landing on $2\%$ of synthetic examples as the best proportion.[11] Note that, with a model finetuned on raw MQM ratings, the accuracy on DEMETR was relatively high to begin with, typically between $80\%$ and $90\%$. The $2\%$ of synthetic data brought it up to nearly $99\%$ though. Interestingly, training on the combination of MQM and synthetic data did not have a negative impact on the correlation metrics, as was the case with DA data. On the contrary, the performance received a consistent boost across all metrics, with an up to $1\%$ increase in system-level scores and $5\%$ in segment-level scores (see row 2 in Table 3).

### 4.3 Two-Stage Finetuning

In the previous two sections, we identified the best subset and format of the DA/MQM ratings, and we found the right balance between the DA/MQM examples and the synthetic examples for training a MetricX model. Here, we take it one step further and perform two-stage finetuning experiments, wherein we first finetune the model on the DA 15-20 training set, and then further finetune it on MQM 20-21 data with a smaller learning rate. The reason for finetuning on the two datasets in this order is 3-fold: (1) MQM is substantially smaller and has a limited language coverage, (2) MQM is a higher-quality dataset, and (3) the metric's performance is ultimately evaluated on MQM ratings (whether it be our validation set, or the shared task's test set).

Of all the four combinations of raw and $z$-normalized DA and MQM ratings, we found, somewhat surprisingly, that using DA $z$-scores (aggregated) in the first stage followed by raw MQM ratings in the second stage leads to the best results. Intuitively, using raw ratings in both stages, or $z$-scores in both stages, should provide a smoother learning process for the model, as it does not need to relearn the label scales.[12] Nevertheless, it appears that the neural model is not negatively affected by the shift, and instead it prefers learning from the DA and MQM data in their respective most effective format.

Before diving into the two-stage experiment results, let us recap that, so far, MetricX achieved the best segment-level performance when finetuned on the MQM 20-21 dataset, and the best system-level performance on the DA 15-20 dataset (in fact, substantially better than on the MQM dataset). The top two rows in Table 3 show the performance of models finetuned on these datasets individually,

[9] When included in the training data, both of the synthetic datasets are added in the same proportion.

[10] $1\%$ corresponds to ca. 10K synthetic examples.

[11] Since the MQM dataset is much smaller, $2\%$ here corresponds to only ca. 2K synthetic examples.

[12] In the experiments with raw ratings in both stages, we rescaled the DA ratings to the $[-25, 0]$ range, so as to match the MQM scale used then in the second stage.

| | SEG pairwise acc. | | SEG Pearson | | SYS pairwise acc. | | SYS Pearson | | DEMETR |
|---|---|---|---|---|---|---|---|---|---|
| | en-de | zh-en | en-de | zh-en | en-de | zh-en | en-de | zh-en | |
| **$DA_{syn}$** | 59.56 | 50.61 | 46.75 | 43.44 | **82.05** | 82.42 | **86.50** | **97.12** | 91.03 |
| **$MQM_{syn}$** | 60.96 | 52.93 | 49.98 | 56.12 | 73.08 | 83.88 | 73.49 | 85.72 | **98.57** |
| **DA $\rightarrow$ MQM** | 61.50 | **54.09** | 51.59 | 58.53 | 73.93 | **86.45** | 75.46 | 88.93 | 46.20 |
| **DA $\rightarrow$ $MQM_{syn}$** | **61.61** | 54.05 | 51.55 | 58.81 | 75.21 | 85.71 | 76.65 | 89.89 | 91.93 |
| **$DA_{syn}$ $\rightarrow$ MQM** | 61.46 | 53.70 | 52.20 | 57.97 | 73.93 | 85.71 | 75.63 | 88.39 | 82.03 |
| **$DA_{syn}$ $\rightarrow$ $MQM_{syn}$** | 61.35 | 53.61 | 52.30 | 57.57 | 75.64 | 86.08 | 77.18 | 89.35 | 97.53 |
| **$DA^{21}_{syn}$ $\rightarrow$ MQM** | 61.49 | 53.67 | 51.36 | 58.07 | 74.79 | 84.62 | 76.30 | 89.26 | 92.30 |
| **$DA^{21}_{syn}$ $\rightarrow$ $MQM_{syn}$** | 61.45 | 53.93 | **53.62** | **58.88** | 75.21 | 85.71 | 78.15 | 89.85 | 98.17 |

Table 3: Performance of MetricX initialized with mT5-XL, finetuned first on DA 15-20 clipped $z$-scores, and subsequently on MQM 20-21 raw ratings, with synthetic data added at different stages. $DA_{syn}$ denotes the DA dataset with $1\%$ of synthetic examples, and $MQM_{syn}$ is the MQM dataset with $2\%$ of synthetic examples. For comparison, the first two rows show the performance of models finetuned on $DA_{syn}$ and $MQM_{syn}$ individually. The last two rows correspond to models finetuned in the first stage on the DA dataset with the '21 ratings added but with all into-English language pairs excluded. Standard deviations are omitted in this table for better readability.

with synthetic data included, serving thus as baselines for the following experiments. Now, the third row shows the scores for a model finetuned in the two-stage fashion without any synthetic data. Comparing these results with those of a model finetuned on the MQM dataset only (see row 3 in Table 2), we see a dramatic improvement in performance across the board. For example, the segment-wise accuracy for zh-en increases from $52.72$ to $54.09$, and Pearson's $r$ for en-de from $47.59$ to $52.30$. In system-level metrics we see similar gains, such as the accuracy going up from $83.52$ to $86.45$, and Pearson's $r$ from $72.70$ to $75.46$ for zh-en. Similarly, rows 2 and 4 in Table 3 can be compared to see a similar difference, only this time for models trained with synthetic data too. This demonstrates a clear benefit of training our MetricX model on both the DA and the MQM data over training it on either of them individually. That being said, the system-level performance still lags significantly behind models finetuned on DA data only, so there appears to be a trade-off between segment- and system-level performance.

Next, we examine whether there is a difference between including synthetic data in the first stage or the second stage of finetuning. Rows 4 and 5 in Table 3 correspond to these two experiments. The scores show that using synthetic examples in the second stage not only ensures a higher DEMETR accuracy ($91.93$ vs. $82.03$), but also higher correlations with the human scores according to virtually all of the other metrics. Moreover, compared to not using synthetic data at all (see row 3 in the table), combining it with the MQM data in the second stage does not generally sacrifice the overall performance, not to mention it almost doubles the DEMETR accuracy.

Finally, we also tried including synthetic data in both finetuning stages, but not until after the shared task's submission window has passed, hence, we did not use this method in our final MetricX version. The gains over using the synthetic data in the second stage only are rather inconsistent, nevertheless the DEMETR score further increases to $97.53$ (see row 6).

### 4.4 Additional Experiments

Although in Section 4.1 we concluded that adding DA ratings from WMT21 to the training set dragged the performance down, we noted that that was the case for the zh-en language pair only. Revisiting this experiment after the submission, we found that excluding the into-English DA data from WMT21 and including the synthetic data as described in Section 4.2 is, in fact, a potentially better training set than the same with the WMT21 ratings omitted altogether. As we can see in Table 3, the model performs consistently better across most of the metrics when finetuned with the WMT21 out-of-English language pairs included (compare rows 7–8 with rows 5–6). So, while this seems to be the best training recipe, it is not the one we followed for the shared task submissions. Instead, we used DA 15-20 (without synthetic data) in the first stage, corresponding to row 4 in the table.

### 4.5 Scaling Analysis

Having arrived at our final MetricX training recipe using mT5-XL (3.7B parameters) as the pretrained model, we now briefly compare its performance

| | SEG pairwise acc. | | SEG Pearson | | SYS pairwise acc. | | SYS Pearson | |
|---|---|---|---|---|---|---|---|---|
| | en-de | zh-en | en-de | zh-en | en-de | zh-en | en-de | zh-en |
| **mT5-L** | 59.63 ±0.34 | 53.98 ±0.06 | 50.82 ±2.16 | 55.91 ±0.53 | 76.07 ±0.74 | **87.55** ±0.63 | 79.92 ±1.20 | **93.63** ±2.48 |
| **mT5-XL** | 61.61 ±0.12 | 54.05 ±0.29 | 51.55 ±1.06 | 58.81 ±0.12 | 75.21 ±0.74 | 85.71 ±1.10 | 76.65 ±1.44 | 89.89 ±0.23 |
| **mT5-XXL** | **61.92** ±0.19 | **54.76** ±0.13 | **54.57** ±0.36 | **60.06** ±0.50 | **82.48** ±0.74 | 86.81 ±1.10 | **84.49** ±0.30 | 90.85 ±0.31 |

Table 4: Performance of MetricX with different variants of mT5 as the initialization model, trained using our final recipe, which involves first finetuning on DA 15-20 ratings and then on MQM 20-21 ratings with synthetic data.

with two other variants of mT5: one smaller (mT5-Large with 1.2B parameters) and one larger (mT5-XXL with 13B parameters). From Table 4 it is clear that MetricX can benefit from a bigger pretrained model for initialization, as mT5-XXL has a good margin on mT5-XL across all metrics. The XXL variant thus becomes our choice for all of our mT5-based submissions to the shared task.

Interestingly, mT5-Large outperforms the two bigger variants in system-level metrics on the zh-en language pair, and that not by a negligible margin. Combined with the results in the earlier sections and our observation that the system-level performance of MetricX is typically highest right at the beginning of training, and quickly declines as the model gradually improves on segment-level metrics, it appears it may be challenging to come up with a single MT evaluation metric that excels at both the segment and the system level. This phenomenon can also be observed among several of the top metrics in the WMT22 Metrics Shared Task (Freitag et al., 2022), as well as in recent large language model-based approaches to automatic MT evaluation (Kocmi and Federmann, 2023).

### 4.6 Submission Summary

Throughout the whole of Section 4 thus far, we were reporting results averaged across three independent runs, so as to more reliably develop the best training recipe for the reference-based version of MetricX. Here, we present the performance of our individual final submissions to the WMT23 Metrics Shared Task, described at the beginning of this section, including our QE (or reference-free) metric submissions. All of our submissions follow the same recipe—i.e., are first finetuned on the DA 15-20 aggregated and clipped $z$-scores, and then further finetuned on MQM 20-21 ratings combined with synthetic examples—but differ in (1) the pretrained model used for initialization, (2) whether they use reference or source segments in the input (the latter being used for the QE submissions), and (3) whether the second-stage training set includes the '22 MQM ratings or not. For the QE variants, we followed the same training recipe as the reference-based version; we did not do a significant amount of analysis into whether the design choices we made for the reference-based metric were also the correct decisions for the QE version. For the models that do use the '22 data for finetuning (which we otherwise use as the validation set), we do not report any scores. For these two submissions, we picked the model checkpoint based on the equivalent training runs without the '22 ratings.

Our remaining four submissions have their scores summarized in Table 5. Between the two reference-based variants (rows 1–2), the mT5-based MetricX is dominant in segment-level scores, whereas the PaLM-2-based one has a strong lead on the system level. The story is similar for the two QE variants (rows 3–4), only the segment-level score differences are less pronounced. Overall, on the WMT22 MQM validation set, the QE variants are not very far behind their reference-based counterparts in performance.

## 5 Related Work

For many years, lexical-based metrics like BLEU (Papineni et al., 2002) and ChrF (Popović, 2015) were the standard method for automatically evaluating MT output. However, as it was demonstrated that learned evaluation metrics correlate to human ratings significantly higher than lexical-based metrics (Freitag et al., 2022), the vast majority of recent research on MT evaluation has used learned metrics.

Learned MT metrics, such as BLEURT (Sellam et al., 2020; Pu et al., 2021) and COMET

| MetricX variant | Pretrained model | SEG p. acc. | | SEG Pearson | | SYS p. acc. | | SYS Pearson | |
|---|---|---|---|---|---|---|---|---|---|
| | | en-de | zh-en | en-de | zh-en | en-de | zh-en | en-de | zh-en |
| 23 | mT5-XXL | **62.09** | **54.84** | **54.21** | **60.06** | **82.05** | 86.81 | 84.19 | 91.20 |
| 23-c | PaLM-2-Bison | 61.56 | 54.17 | 51.62 | 52.24 | 76.92 | **92.31** | 91.41 | **98.61** |
| 23-QE | mT5-XXL | 60.64 | 54.04 | 49.78 | 56.41 | 78.21 | 87.91 | 81.29 | 91.32 |
| 23-QE-c | PaLM-2-Bison | 60.23 | 53.96 | 49.28 | 52.48 | **82.05** | 91.21 | **93.64** | 96.90 |

Table 5: Meta-evaluation scores of our four MetricX submissions that did not include the '22 MQM data in the training set. Note that these scores correspond to single runs, as opposed to all the previous results that were averaged across 3 runs.

(Rei et al., 2020, 2022a), differ largely in their network architecture and the specific tasks they are trained to do. Our metric is most closely related to BLEURT. Like BLEURT, the MetricX network architecture creates a joint encoding of both the hypothesis and reference translations together, in contrast to COMET-style metrics that encode them independently. Our network is trained only to do either reference-based or reference-free (QE) judgments of sentence-level translation quality, whereas some metrics like UniTE (Wan et al., 2022) are trained to both tasks at the same time or CometKiwi (Rei et al., 2022b), which learns to predict both word-level quality scores in addition to an overall sentence-level score. Other learned metrics, such as MaTESe (Perrella et al., 2022), take an alternative approach to regression-based metrics and derive a sentence-level quality score by identifying error spans in translations, like is done in the human evaluations of MQM.

More recent approaches to MT evaluation leverage large language models (LLMs) to do zero-shot scoring by either directly predicting scalar quality scores or phrase-level error tagging (Kocmi and Federmann, 2023; Fernandes et al., 2023). These approaches typically leverage models that are orders of magnitude larger than metrics that are trained specifically for MT evaluation.

In comparison to the MetricX submission to the WMT22 Metrics Shared Task, this year's submission shares the same architecture, but we performed a significantly larger number of experiments to arrive at the final models, which are detailed in this report. We also explore how metric performance changes as a function of the number of parameters, experiment with initializing with different pretrained language models, and include a QE submission.

## 6 Conclusion

In this report, we presented in detail our approach to training MetricX-23, a regression-based MT evaluation metric. We submitted six versions of MetricX-23 to the WMT23 Metrics Shared Task, including both reference-based and QE variants. Some of our findings are that (1) training on direct assessment (DA) ratings and subsequently on MQM ratings leads to a significantly better performance than training on either of the two datasets alone, (2) $z$-normalization of DA ratings helps achieve better segment-level performance, but is not useful for high-quality MQM ratings, and (3) adding a small amount of synthetic data to the training set, targeting a challenge set, can also boost the metric's overall performance.

Throughout our experiments, we observed an undesirable tension between segment- and system-level performance, making it challenging to improve our metric in both aspects at the same time. Nevertheless, increasing the size of the model used to pre-initialize the metric appears to be one reliable way to increase the overall performance, at least to a certain point. Future work may benefit from a better understanding of the trade-off between segment- and system-level performance, and whether it would be better to focus on separate metrics for these two types of MT evaluation.

## References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan,

Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. PaLM 2 Technical Report.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties Matter: Modifying Kendall's Tau for Modern Metric Meta-Evaluation.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André F. T. Martins, Graham Neubig, Ankush Garg, Jonathan H. Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, Errors, and Context: A Large-Scale Study of Human Evaluation for Machine Translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chikiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and AndrÃ© F. T. Martins. 2022. Results of WMT22 Metrics Shared Task: Stop Using BLEU - Neural Metrics Are Better and More Robust. In *Proceedings of the Seventh Conference on Machine Translation*, pages 46–68, Abu Dhabi. Association for Computational Linguistics.

Marzena Karpinska, Nishant Raj, Katherine Thai, Yixiao Song, Ankita Gupta, and Mohit Iyyer. 2022. DEMETR: Diagnosing Evaluation Metrics for Translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9540–9561, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To Ship or Not to Ship: An Extensive Evaluation of Automatic Metrics for Machine Translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional Quality Metrics (MQM): A Framework for Declaring and Describing Translation Quality Metrics. *Tradumàtica*, (12):0455–463.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

Stefano Perrella, Lorenzo Proietti, Alessandro ScirÃ¨, NiccolÃ² Campolungo, and Roberto Navigli. 2022. Matese: Machine translation evaluation as a sequence tagging problem. In *Proceedings of the Seventh Conference on Machine Translation*, pages 569–577, Abu Dhabi. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Amy Pu, Hyung Won Chung, Ankur P Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of EMNLP*.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022a. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). 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.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, JosÃ© G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and AndrÃ© F. T. Martins. 2022b. Cometkiwi: Ist-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation*, pages 634–645, Abu Dhabi. Association for Computational Linguistics.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. 2022. Scaling Up Models and Data with `t5x` and `seqio`. *arXiv preprint arXiv:2203.17189*.

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.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, pages 4596–4604. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yu Wan, Keqin Bao, Dayiheng Liu, Baosong Yang, Derek F. Wong, Lidia S. Chao, Wenqiang Lei, and Jun Xie. 2022. Alibaba-translate china's submission for wmt2022 metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation*, pages 586–592, Abu Dhabi. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.