xCOMET-lite: Bridging the Gap Between Efficiency and Quality in Learned MT Evaluation Metrics

Daniil Larionov[1] **Mikhail Seleznyov**[4,3] **Vasiliy Viskov**[3]
Alexander Panchenko[3,4] **Steffen Eger**[1,2]
[1] NLLG, University of Mannheim, [2] University of Technology Nuremberg, [3] Skoltech, [4] AIRI
daniil.larionov@uni-mannheim.de

Abstract

State-of-the-art trainable machine translation evaluation metrics like xCOMET achieve high correlation with human judgment but rely on large encoders (up to 10.7B parameters), making them computationally expensive and inaccessible to researchers with limited resources. To address this issue, we investigate whether the knowledge stored in these large encoders can be compressed while maintaining quality. We employ distillation, quantization, and pruning techniques to create efficient xCOMET alternatives and introduce a novel data collection pipeline for efficient black-box distillation. Our experiments show that, using quantization, xCOMET can be compressed up to three times with no quality degradation. Additionally, through distillation, we create an 278M-sized xCOMET-lite metric, which has only 2.6% of xCOMET-XXL parameters, but retains 92.1% of its quality. Besides, it surpasses strong small-scale metrics like COMET-22 and BLEURT-20 on the WMT22 metrics challenge dataset by 6.4%, despite using 50% fewer parameters. All code, dataset, and models are available online.

1 Introduction

Automatic evaluation metrics are crucial for reliably measuring the quality of responses from natural language generation (NLG) systems. Researchers and practitioners working on tasks such as machine translation (MT), summarization, poetry generation, etc., routinely use metrics to assess their systems' quality. Apart from directly assessing the systems, evaluation metrics have many other applications: **a)** filtering web-scale datasets (Peter et al., 2023); **b)** using metrics as reward functions for Reinforcement Learning (Xu et al., 2024); **c)** online re-ranking of outputs of multiple systems to choose the best response to return to the user (Fernandes et al., 2022).

With generative models' growing sizes and complexity, automatic evaluation metrics also evolve and become more computationally expensive. In the last few years, for MT evaluation, researchers have moved from traditional n-gram and character-based metrics, such as BLEU (Papineni et al., 2002) and chrF (Popović, 2015), to embedding-based metrics, such as BERTScore (Zhang et al., 2020) and MoverScore (Zhao et al., 2019), to learned metrics, which provide state-of-the-art correlation with human judgment. According to Freitag et al. (2023), the best-performing metrics for MT evaluation are xCOMET (Guerreiro et al., 2023), MetricX (Juraska et al., 2023), and GEMBA-MQM (Kocmi and Federmann, 2023). All those metric models have a large number of parameters: xCOMET and MetricX have 10.7B-13B parameters, while GEMBA-MQM relies on the Large Language Model (LLM) GPT4 (OpenAI, 2023), for which the number of parameters is unknown but speculated to be around 1.7T[1].

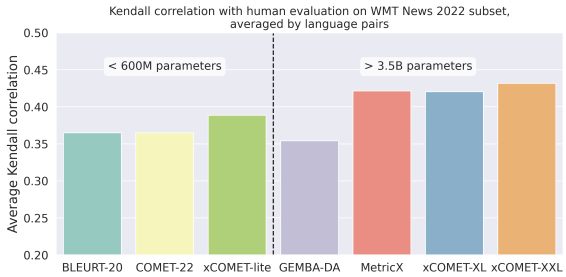

Figure 1: xCOMET can be distilled into a small model, which will be 6-7 percentage points better than SOTA models with comparable parameter count.

The lack of efficient alternatives to these models creates a disparity in access among researchers. Under-resourced labs, students, startups, and hobbyists without access to top-tier accelerators (with more than 22GB VRAM) or financial resources for paid APIs cannot employ those metrics. Those with access to such resources may also experience prolonged iteration time due to the computation

[1] https://twitter.com/soumithchintala/status/1671267150101721090

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 21934–21949
November 12-16, 2024

needed for those models. This is especially noticeable in the case of repeated evaluations during the hyperparameter optimization or processing of large-scale datasets. For instance, running the xCOMET-XXL model to filter a crawled dataset of 10^7 examples would take 142.2 hours on a capable consumer-grade GPU, requiring 42.6 kWh of electricity and emitting around 15.6 kg CO_2-eq.[2] Thus, developing alternative efficient metrics is now more vital than ever.

In this paper, we explore various techniques to develop an efficient alternative to the state-of-the-art xCOMET metric for evaluating MT quality. Our approach focuses on three main methods: knowledge distillation, quantization, and pruning. Knowledge distillation is a method of creating capable small deep fitted models by training them on the outputs of the larger model. We apply knowledge distillation (Hinton et al., 2015), training a smaller version of the xCOMET model on large amounts of data, using labels created by the original xCOMET-XXL model. Quantization reduces the precision of deep learning model parameters and activations from 32/16 bits into 8, 4, 3, and 2 bits, occupying less memory and allowing for faster computations. Pruning involves the removal of less significant parts of the model, either specific parameters, blocks of parameters, or entire layers. We apply layer pruning together with subsequent fine-tuning, which allows for accelerated inference throughput and helps mitigate potential accuracy loss. By exploring distillation, quantization, and pruning, as well as their combinations, we aim to create an efficient alternative to xCOMET that maintains a high level of quality while substantially reducing hardware requirements.

Our main contributions are as follows: **a)** we conduct a comprehensive study of different compression methods (knowledge distillation, quantization, and pruning) and their interactions for the state-of-the-art MT evaluation metric xCOMET. To the best of our knowledge, this is the first work to systematically investigate the effectiveness and trade-offs of these techniques when applied to a large-scale, complex metric like xCOMET; **b)** we introduce a novel data collection pipeline for preparing large-scale, high-quality datasets for black-box distillation of xCOMET. We collect 14M examples with translation hypotheses of varying quality paired with high-quality reference translations. This enables the distilled model to effectively transfer the evaluation capabilities of the teacher model, xCOMET-XXL; **c)** through our distillation method, we develop xCOMET-lite, a lightweight yet highly effective MT evaluation metric. xCOMET-lite achieves state-of-the-art quality among metrics with < 600M parameters, surpassing the previous best model, COMET-22, while being substantially smaller; **d)** we explore the use of quantization for compressing xCOMET and demonstrate that 3-bit quantization can effectively reduce hardware requirements for 3B and 11B model versions without compromising quality; **e)** we investigate the effectiveness of pruning for compressing xCOMET and show that while pruning up to 25% of the model layers can improve inference speed and memory consumption with only a marginal impact on quality, removing more layers leads to substantial quality degradation. **f)** We conduct a novel study of the interactions between compression methods, revealing that distillation combines well with quantization but is incompatible with pruning in our experiments.

2 Related Work

Recent work has explored improving the transparency and capabilities of MT evaluation metrics. Juraska et al. (2023) introduced MetricX. This learned regression-based metric achieves state-of-the-art correlations with human judgments through multi-stage fine-tuning on direct assessment data, consolidated MQM scores, and small-scale synthetic corpora, which is used to boost robustness. It is based on the mT5-XXL encoder-decoder model with 11B parameters. Kocmi and Federmann (2023) proposed GEMBA-MQM, which leverages the GPT-4 language model with a few-shot prompting approach to identify translation error spans and categories.

This enables detailed error analysis, though reliance on the computationally expensive proprietary GPT-4 LLM poses challenges for academic research. Guerreiro et al. (2023) developed xCOMET, a learned metric based on the XLM-RoBERTa-XL/XXL encoder that bridges sentence-level quality prediction with fine-grained error span detection. By training on direct assessment and MQM data, xCOMET achieves top quality on sentence-level, system-level, and error span prediction tasks while providing interpretability through

[2]Assumptions: GPU power draw of 350W, 0.05s per example on average and 0.368 kg CO_2-eq/kWh US power grid carbon intensity taken as reference.

its predicted error spans.

Previously, researchers have also explored techniques for creating more **efficient** MT evaluation metrics while preserving their correlation with human judgments. Kamal Eddine et al. (2022) proposed FrugalScore, which learns lightweight versions of metrics like BERTScore and MoverScore using knowledge distillation. Their distilled metrics perform similarly to the originals while being much faster and having orders of magnitude fewer parameters. Rei et al. (2022b) introduced COMETINHO, a more compact and faster version of the COMET metric. They optimize the COMET code using caching and length batching and further compress the model using pruning and knowledge distillation on synthetic data. The resulting model is 80% smaller and over 2 times faster than the original while maintaining competitive quality.

3 Methods

We explore three compression techniques to develop an efficient alternative to xCOMET for evaluating MT quality: quantization, pruning, and distillation. These methods aim to reduce the computational requirements and improve the inference speed of xCOMET while maintaining a high level of quality.

Quantization Quantization is a highly effective compression method with two main approaches: quantization-aware training (QAT) and post-training quantization (PTQ) (Nagel et al., 2021). QAT offers better prediction quality but requires costly training, making PTQ more popular. PTQ is further divided into data-free and data-aware methods, where the latter relies on calibration to estimate the data distribution parameters for higher prediction quality. Another distinction is weight-only quantization and weight & activation quantization, with the second approach having slightly lower prediction quality but potential for faster computations using efficient 8- or 4-bit CUDA kernels.

In a nutshell, the quantization process comes down to finding bias and scale for each floating point value $x \in [\alpha, \beta]$ to convert it to a n-bit integer $x_q \in [\alpha_q, \beta_q]$:

$$x_q = \left\lfloor \frac{1}{\sigma} x + x_0 \right\rceil, \sigma = \frac{\beta - \alpha}{\beta_q - \alpha_q}, x_0 = \left\lfloor \frac{\beta\alpha_q - \alpha\beta_q}{\beta - \alpha} \right\rceil$$

Dynamic quantization (Gholami et al., 2021) is a technique that generates the zero-point x_0 and scale σ parameters in real-time, thereby eliminating the need for calibration data. Due to the unknown distribution parameters, activations are maintained in floating-point format. The process of obtaining quantization parameters (α, β) and quantizing floating-point tensors to integer tensors is relatively straightforward, with the necessary statistics being computed during inference.

Among data-free quantization methods, LLM.int8() (Dettmers et al., 2022) and QLoRA (Dettmers et al., 2023) stand out as the most prominent. (i) LLM.int8() quantizes model weights to 8-bit precision using the absmax quantization technique. This method also dynamically quantizes activations to enable efficient matrix multiplications primarily in *int8*, with certain calculations performed in *fp16* for precision. (ii) QLoRA uses a more advanced double quantization approach. It utilizes the *nf4* data type for storage, minimizing memory demands, while computation is conducted in higher precision types (*fp16, bf16*), dequantizing weights on a per-layer basis.

GPTQ (Frantar et al., 2023) is an example of weight-only quantization methods. It performs layer-by-layer quantization, minimizing the squared error relative to the full precision layer output:

$$\arg\min_{\widehat{W}} \|WX - \widehat{W}X\|_F^2$$

Here, W are the full precision weights, X denotes the layer input corresponding to a small set of m data points running through the network, $\widehat{W}$ represents a matrix of quantized weights, and $\|\cdot\|_F$ is the Frobenius norm.

Pruning Pruning is the removal of the least significant parts of the neural network. It can be divided into structured and unstructured. The latter proves helpful on a CPU but is rarely practical on a GPU, since GPUs are heavily optimized for dense matrix multiplication. Structured pruning can take many forms, from enforcing 2:4 sparsity patterns (in each contiguous block of four values, two values must be zero) to pruning channels or entire blocks of the networks.

Inspired by recent works on layer pruning in LLMs (Gromov et al., 2024; Men et al., 2024) which remove 25-50% of layers with moderate quality drop, we test its applicability for inducing efficient metrics. Specifically, we adopt a simple pruning technique, described in Sec. 4.4 of Gromov et al. (2024): in an L-layer model, we drop layers

$(L-n)$ to $(L-1)$. This heuristic is based on the observations that pruning deeper layers should affect the model less, as fewer layers rely on changes made by this layer, but also that the ultimate layer is especially important as it "decodes" the hidden states for the last fragment of the network, and cannot be removed. To mitigate the quality drop incurred by layer removal, we apply parameter-efficient fine-tuning. Concretely, we fine-tune all biases in linear layers, LayerNorm affine parameters, layerwise attention weights, and the regression and tagging heads of xCOMET. This is akin to the BitFit (Zaken et al., 2022) sparse-fine-tuning approach, and has the benefit of adding no parameters and being extremely simple to implement.

We also evaluate magnitude pruning and Wanda pruning (Sun et al., 2024). In magnitude pruning, the importance of each weight S_{ij} is directly estimated by its magnitude $|W_{ij}|$. Wanda pruning refines this approach by weighting each $|W_{ij}|$ by the average L2 norm of its corresponding input features, $\frac{1}{N}\sum_{j=1}^{N}\|x_j\|_2$, aiming to provide a more informed measure of importance. In both methods, the weights with the lowest importance scores are pruned according to the specified sparsity pattern (unstructured, 2:4 or 4:8).

Distillation In distillation, we distinguish between white-box and black-box methods. White-box distillation, detailed in Li and Jin (2022) and Gu et al. (2023), necessitates access to the teacher model internal states, including logits and, possibly, attention maps. This method requires substantial memory and computational resources, as both teacher and student models must be loaded simultaneously, which can be impractical for very large teacher models.

Conversely, black-box distillation, as explored in Jiang et al. (2023); Wu et al. (2024); Fu et al. (2023), only requires the teacher model outputs, making it more scalable and feasible for large models or restricted access scenarios. Despite using less information from the teacher, black-box distillation effectively produces high-quality models with reduced computational demands.

For our study, we chose black-box distillation using xCOMET-XXL. This choice allows us to use a very large teacher model, xCOMET-XXL, without encountering the hardware limitations that would arise from white-box distillation. The approach involves using the teacher model to generate pseudo-labels for a large dataset of text triplets. Specifically, the teacher model assigns segment-level quality scores, $q \in [0, 1]$, and token-level error span annotations, $k_j \in \{critical, major, minor, no\text{-}error\}$, for each token in the machine translations, based on MQM annotation guidelines (Freitag et al., 2021a). We simplify the training approach proposed in the original xCOMET paper, adopting a single-phase training method that efficiently trains the student model using these pseudo-labels with both segment-level and word-level supervision.

Our approach resembles the recently proposed *Distilling step-by-step* method (Hsieh et al., 2023). Both methods utilize black-box distillation without access to the teacher model's internal states. Furthermore, both approaches train the student model on an additional supervision signal beyond the single task-specific label/score. In the case of *Distilling step-by-step*, it is LLM-produced rationales, while in our case, it is error span annotations produced by xCOMET-XXL.

4 Experiments

We compare quantization, pruning, and distillation for compressing xCOMET. We compare it to both released versions, -XL and -XXL. As we focus on computational efficiency, we measure the model **(i)** inference speed, **(ii)** resource requirements (in terms of GPU memory, vRAM), and **(iii)** metric prediction quality, expressed in Kendall-τ correlation with human judgment.

4.1 Evaluation

WMT MQM Human Evaluation dataset. This dataset contains all MQM human annotations from previous WMT Metrics shared tasks (Freitag et al., 2022, 2021b) and from Freitag et al. (2021a). It contains over 150k examples for three translation directions (Chinese-English, English-German, English-Russian), five domains (news, TED talks, conversational, social, e-commerce), and three years (2020, 2021, 2022). Following xCOMET (Guerreiro et al., 2023), we use the news 2022 subset (over 16k samples) for evaluation and the rest of the data for training.

Eval4NLP. We additionally use MT data from the Eval4NLP shared task (Leiter et al., 2023). There are three translation directions: English-Spanish, English-German, and English-Chinese, over 4400 examples in total. No reference translation is provided, which allows to test xCOMET in a reference-free regime.

Metric quality evaluation. We use the Kendall correlation to evaluate the quality of the compared metrics. See Appendix B for a definition. Each experiment that involves model training is conducted 3 times with different random seeds to account for any fluctuations. We report correlation values obtained by averaging across 3 runs.

Efficiency evaluation. To evaluate the computational efficiency of compressed models, we measure inference speed in samples per second (samples/s). For a given language pair, we divide the amount of examples by the total time needed to inference the model on the set. Due to the GPU execution and memory models, some operations, such as matrix multiplication, take the same time to execute regardless of the amount of data supplied. Thus, using the largest possible batch size that fits into the accelerator memory is most efficient. To select the optimal batch size, we start with batch size 1 and increase it by a factor of 2 until we reach the memory limit on the given GPU. We test model throughput on RTX 3090 and A100 to explore performance on consumer- and production-level GPUs. Additionally, we provide peak vRAM usage for each model on a fixed batch size of 8.

4.2 Setup

Quantization. We use the GPTQ (Frantar et al., 2023) quantization algorithm and quantize xCOMET to 8, 4, 3, and 2 bits per parameter. We keep default hyperparameters, except using a small subsample of the WikiText2 (Merity et al., 2017) dataset for calibration. In addition to that, we experiment with data-free quantization methods: LLM.int8() – 8 bit and QLoRA – 4 bit. We use the implementation from the *bitsandbytes* python library. Initial experiments indicated that models worked faster with their 4-bit quantization implementation if weights were converted to mixed precision beforehand.This observation was also true for 8-bit quantization, but in this case the quality drop became substantial. Thus, we report LLM.int8() without any uncompressed model transformations, and QLoRA with half-precision model weight conversion.

Pruning. Following the approach described in the §3, we apply layer pruning to the underlying encoder model of xCOMET. We remove the underlying layers from $L - n$ to $L - 1$, with n being 4, 8, 12, 16 or 20 layers. We also patch the layerwise attention component of the xCOMET model to reflect changes in the model structure. Subsequently, after pruning, we perform parameter-efficient fine-tuning on the training part of the WMT22 MQM dataset. Fine-tuning is performed for 1 epoch, using AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of $1e-4$, effective batch size of 128, and cosine learning rate warmup for 10% of the duration of training.

With Wanda pruning we try 2:4 and 4:8 patterns, to explore setups which can realistically provide speedups on GPU. We use 256 calibration samples from WikiText2[3], and do not finetune pruned model, as the original method does not require it. We also run simple magnitude pruning with 2:4 and 4:8 sparsity patterns.

Constructing dataset for distillation. To create a dataset for model compression through distillation, we collected a large number of examples for evaluating MT systems. The collection process involved three main stages.

First, we sampled 500k examples of high-quality parallel texts (source texts and their translations) from the NLLB dataset (Costa-jussà et al., 2022) for each of the following language pairs: Russian-English, German-English, and Chinese-English. As the NLLB dataset is automatically collected at scale using a bi-text mining model, some translations may be of subpar quality. To address this issue, we applied the xCOMET-XXL model in reference-free mode to filter out examples with low quality scores, which are more likely to be incorrect translations. The filtering threshold was set to the 95th percentile of scores for each language pair, resulting in a threshold of 1.0 (on a 0 to 1 scale) for Russian-English and German-English, and 0.85 for Chinese-English.

In the second stage, we generated translation hypotheses for the filtered examples using various MT models with different sizes, architectures, and release dates to ensure high variability in translation quality, following the approach of Rei et al. (2022b). Additionally, we applied synthetic corruption algorithms to generate hypotheses by corrupting reference translations, as suggested by Moosa et al. (2024). The complete list of models and algorithms used can be found in Appendix A.

Finally, in the third stage, we used the xCOMET-XXL model in reference-based mode to generate

[3]In (Sun et al., 2024) authors use 128 calibration samples from C4, but we couldn't reproduce the code related to sampling examples from C4.

labels for the collected dataset, including sentence-level scores and error spans. After deduplication and inverting language pairs, our final dataset consists of 14M examples, each containing a source text, reference translation, hypotheses, segment-level quality score and annotated error spans.

Distillation. We use mDeBERTa v3 (He et al., 2023) as a student. It has 278 M parameters — 13 times fewer than xCOMET-XL, 39 times fewer than xCOMET-XXL, and 2 times fewer than COMET-22 — one of the top performers in WMT22 Metrics Shared Task. This model was chosen as it shows superior quality on multilingual language understanding tasks such as XNLI (Conneau et al., 2018), compared to alternatives of similar size: InfoXLM (Chi et al., 2021) and XLM-RoBERTa (Conneau et al., 2020). We trained for 1 epoch, with learning rate of $2e-5$ for scoring head and $1e-5$ for encoder. We set the batch size to 64. Scoring head was configured with two hidden fully connected layers with sizes 3072 and 1024. We compare the prediction quality of the distilled model with original models xCOMET-XL/XXL, as well as with best-performing models of similar size: BLEURT-20 (Sellam et al., 2020) with 579 M parameters and COMET-22 (Rei et al., 2022a) with 581 M parameters.

4.3 Results

We present the results of our experiments on quantization, pruning, and distillation. Tables 1 and 3 show the effects of these techniques on xCOMET-XL and xCOMET-XXL models. Table 1 focuses on the trade-offs between model quality and memory consumption for pruning and quantization, and Table 3 presents the relationship between model quality and throughput for the same techniques. Separately, we present prediction quality for our distilled model in Table 2 and compare it to several baseline metrics of similar size.

Quantization. Quantization proves highly effective in reducing **memory consumption** while maintaining quality. For xCOMET-XL, GPTQ 8-bit achieves nearly identical quality to the baseline, with an average Kendall correlation of 0.420, while reducing peak memory usage by 33%. GPTQ 3-bit provides the largest memory reduction of 54% at the cost of a 0.013 decrease in correlation. Notably, xCOMET-XXL sees no quality degradation with GPTQ 8-bit and 3-bit, despite memory reductions of 38% and 64%, respectively. LLM.int8() and QLoRa are suboptimal in terms of quality / peak memory consumption tradeoff, dominated by GPTQ 8-bit and GPTQ 3-bit respectvely.

However, as we see in Table 3, GPTQ slows models down, most likely due to usage of non-optimized CUDA kernels, while QLoRa maintains the thoughput on par with non-compressed model.

Pruning. Layer pruning substantially improves **throughput**, particularly for xCOMET-XL. As we can see in Table 3, pruning 16 layers provides 67% speedup compared to the uncompressed model on an RTX 3090. However, the quality drop is larger compared to quantization methods.

Interestingly, magnitude pruning slightly outperforms Wanda pruning, though the latter uses more involved weight importance estimation. Moreover, magnitude pruning performs on par with removing 8 layers, despite keeping only 50% of non-zero weights. Due to some inefficiencies in official implementation, Wanda pruning and magnitude pruning get OOM error on RTX 3090 on some of the datasets; however, we expect they would show speedups similar to ones on A100.

Distillation. Distilling xCOMET-XXL into the much smaller xCOMET-lite model is a highly effective compression strategy. As we demonstrate in Table 2, despite having only 2.6% of the parameters (278M vs. 10.7B), the distilled model achieves an average Kendall correlation of 0.388, surpassing BLEURT-20 & COMET-22. On English-Russian translation, it even surpasses xCOMET-XL. The effectiveness of using our large-scale distillation dataset is further highlighted by the 10-point lower correlation achieved by a model trained on a smaller human-annotated dataset.

The distilled xCOMET-lite model offers unparalleled **speed and memory** efficiency, processing up to 153.8 samples/s on an RTX 3090, 15.2 times faster than the original model (7.8-10.1), as we demonstrate in Table 3. The distilled model has a peak memory consumption of just 1.79 GB, 12.5 times smaller than the original model (22.39 GB).

Additional experiments on reference-free evaluation (Appendix F) demonstrate that our distilled model remains competitive with the xCOMET models, achieving an average Kendall correlation of 0.363, just slightly lower than xCOMET-XXL (0.385) and xCOMET-XL (0.378).

Extended Results. In Appendix E, Figure 2, we present detailed results covering all evaluated

Model	Compression method	Average Kendall correlation	Peak memory consumption (GB) mean (max)
XL	None	0.421	7.76 (8.17)
XL	GPTQ 8 bit	0.420	5.20 (5.60)
XL	GPTQ 3 bit	0.408	3.54 (3.84)
XL	LLM.int8()	0.416	7.50 (8.32)
XL	QLoRA 4 bit	0.405	3.75 (4.16)
XL	Prune 8 layers	0.389	6.34 (6.66)
XL	Prune 16 layers	0.365	4.90 (5.14)
XL	Magnitude pruning 4:8	0.390	*7.77 (8.18)
XL	Wanda pruning 4:8	0.389	*8.09 (8.25)
XXL	None	0.433	22.27 (22.39)
XXL	GPTQ 8 bit	0.433	13.81 (14.66)
XXL	GPTQ 3 bit	0.435	7.99 (8.85)
XXL	LLM.int8()	0.428	17.86 (19.59)
XXL	QLoRA 4 bit	0.429	9.09 (9.94)
XXL	Prune 8 layers	0.417	19.39 (20.09)
XXL	Prune 16 layers	0.398	15.91 (16.48)
XXL	Magnitude pruning 4:8	0.418	*22.82 (23.65)
XXL	Wanda pruning 4:8	0.408	*22.88 (23.65)
XXL	Distilled (xCOMET-lite)	0.388	1.59 (1.79)

Table 1: An overview table with quality / peak memory consumption tradeoff for various representative compression methods in setting **with** reference translations. Average Kendall correlation and mean/max memory consumption is computed over three language pairs. Underlined values indicate compression methods with best prediction quality. XL stands for xCOMET-XL, XXL stands for xCOMET-XXL. For Wanda pruning, VRAM consumption is reported using the official method implementation, which stores pruned weights as zeros in original precision. However, potentially 4:8 pruning could deliver almost x2 memory usage reduction.

Metric	zh-en	en-ru	en-de	Avg.	# parameters
xCOMET-XL	0.399	0.414	0.448	0.420	3.5 B
xCOMET-XXL	0.390	0.435	0.470	0.432	10.7 B
BLEURT-20	0.336	0.380	0.379	0.365	579 M
COMET-22	0.335	0.369	0.391	0.361	581 M
COMETINHO	0.262	0.330	0.342	0.311	117 M
xCOMET-lite (WMT22 data only)	0.280	0.320	0.295	0.298	278 M
xCOMET-lite	0.360	0.422	0.384	0.388	278 M

Table 2: Distillation results on WMT MQM News 2022 subset. The numbers are Kendall correlation with human judgement. We compare against BLEURT-20 and COMET-22, which were strong contenders in WMT22 Metrics Shared Task. Additionally, we compare against a baseline of our model trained on smaller human-annotated dataset WMT22. For reference, there are also scores for large xCOMET models.

configurations of pruning and quantization. Notably, 3-bit GPTQ compression maintains prediction quality, contrary to observations in Dettmers and Zettlemoyer (2023), where 4 bits are Pareto-optimal. This suggests that encoder models may be less susceptible to the "outlier features" mentioned in Dettmers et al. (2022). Layer pruning shows promising results for xCOMET-XXL on 4 out of 6 translation directions, with up to 25% of layers pruned with minimal impact on quality, especially in the reference-free setting.

4.4 Interaction Analysis

To further understand the limits of compression of learned metrics for MT evaluation, we explore interactions between compression methods.

We can apply pruning to our distilled model xCOMET-lite to further shrink its size. Given that the encoder now only has 12 layers instead of 48, we evaluate 3 configurations, pruning 2, 4, or 6 layers from the model. In those experiments, we use the same hyperparameters as in §4.2. We notice a fatal drop in correlation with human judgment by at least 30% across configurations, to an average score of 0.2645. Please see Table 4 in Appendix C for the full results.

We can also apply quantization to the distilled model. Unfortunately, due to architectural details, GPTQ quantization is incompatible with the mDeBERTa architecture. Instead, we apply LLM.int8() and QLoRA quantization (8-bit and 4-bit, respectively). When comparing the 8-bit quantized xCOMET-lite model to the non-quantized one, we observe only a marginal drop in correlation with human judgment. The 8-bit model achieves an average score of 0.369 across language pairs with references, compared to original xCOMET-lite 0.388. For pairs without references, the 8-bit model scores 0.354, while xCOMET-lite achieves

Model	Compression method	Average Kendall correlation	Samples per second RTX 3090 (min / median / max)	Samples per second A100 (min / median / max)
XL	None	0.421	23.1 / 30.5 / 30.9	46.3 / 59.5 / 61.8
XL	GPTQ 8 bit	0.420	10.8 / 13.7 / 13.9	29.8 / 38.5 / 40.6
XL	GPTQ 3 bit	0.408	9.9 / 12.4 / 12.6	29.6 / 39.4 / 40.7
XL	LLM.int8()	0.416	20.9 / 28.1 / 28.5	29.8 / 38.5 / 40.6
XL	QLoRA 4 bit	0.405	22.1 / 28.8 / 29.4	44.8 / 62.9 / 63.4
XL	Prune 8 layers	0.389	29.3 / 38.3 / 39.1	59.8 / 72.7 / 78.5
XL	Prune 16 layers	0.365	38.6 / 50.3 / 51.6	72.0 / 91.6 / 96.6
XL	Wanda 4:8	0.389	25.2 / 32.8 / 34.2	56.0 / 72.1 / 75.5
XL	Magnitude pruning 4:8	0.390	25.4 / 33.4 / 33.8	53.3 / 71.7 / 72.1
XXL	None	0.433	7.8 / 10.0 / 10.1	17.5 / 22.5 / 23.3
XXL	GPTQ 8 bit	0.433	2.6 / 3.0 / 3.0	9.3 / 11.7 / 11.9
XXL	GPTQ 3 bit	0.435	2.7 / 3.2 / 3.2	9.0 / 11.2 / 11.4
XXL	LLM.int8()	0.428	9.7 / 12.4 / 12.4	13.3 / 19.0 / 19.8
XXL	QLoRA 4 bit	0.429	7.3 / 9.4 / 9.5	17.2 / 22.3 / 23.3
XXL	Prune 8 layers	0.417	9.4 / 12.2 / 12.3	21.3 / 26.8 / 27.6
XXL	Prune 16 layers	0.398	15.2 / 15.3 / 15.5	26.2 / 33.3 / 34.3
XXL	Wanda pruning 4:8	0.408	OOM	23.5 / 29.5 / 30.5
XXL	Magnitude pruning 4:8	0.418	OOM	23.0 / 29.4 / 29.6
XXL	Distilled (xCOMET-lite)	0.388	121.4 / 146.1 / 153.8	150.5 / 180.2 / 190.0

Table 3: Speed results for various methods in settings **with reference**. Importantly, here the memory consumption is higher than in Table 1, as we aim for maximal throughput on a given GPU. Average Kendall correlation is computed over three language pairs. Samples per second are reported for both 3090 and A100 GPUs. XL stands for xCOMET-XL, XXL stands for xCOMET-XXL. OOM means Out Of Memory error.

0.363. Notably, the model quantized into 4-bit mode yields a slightly higher correlation for pairs with references, namely 0.379. Furthermore, quantization substantially reduces memory usage. The 8-bit quantization decreases the peak memory consumption of the distilled model by 17% from 1.8 GB to 1.5 GB, while the 4-bit quantization further reduces it to 1.4 GB. These results demonstrate that quantization is a viable option for further compressing the distilled model without substantial quality degradation. See Table 5 in Appendix D for full results.

5 Discussion

The compression methods applied to xCOMET-XL and xCOMET-XXL models demonstrate the potential for reducing memory consumption and increasing processing speed while maintaining competitive prediction quality. Quantization methods, particularly GPTQ 8-bit and 3-bit, achieve substantial memory savings without compromising the models quality. Quantization can also be combined with distillation with little-to-no quality reduction.

Pruning methods, while capable of reducing memory consumption and increasing throughput, result in a more noticeable decrease in correlation compared to quantization. Our results align with the findings in Rei et al. (2022b), which conclude that up to 5 out of 24 layers of encoder model can be removed without noticeable quality degradation of the metric. At the same time, the layer pruning works slightly worse than in other tasks (Gromov et al., 2024; Men et al., 2024), where up to 50% of layers could be removed for large models. Pruning appears incompatible with our distilled model, due to a substantial drop in metric quality. Magnitude pruning with 4:8 sparsity pattern shows promising results with respect to quality / speedup trade-off. Moreover, it potentially offers almost 50% reduction in peak memory consumption (and e.g. `torch` library will likely support structured spasity formats quite soon).

The distillation of xCOMET-XXL into the smaller mDeBERTa-based model, xCOMET-lite, is a highly effective approach for improving computational efficiency while maintaining competitive metric quality. Our distillation method, based on collecting large-scale diverse dataset, proves successful for distilling the xCOMET metric and is easily scalable to additional translation directions.

When considering speed, the distilled xCOMET-lite outperforms other compression methods, processing a substantially higher number of samples per second on both consumer-grade RTX 3090 and HPC-grade A100 GPUs. Pruning is the next best performer, allowing for up to 1.3-1.5 times speedup while maintaining competitive metric quality.

6 Conclusion

In the rapidly evolving field of MT evaluation, the current top-performing metrics, such as MetricX, xCOMET, and GEMBA-MQM, are all based on

extremely large underlying models. These models, including mT5 with 13B parameters, XLM-RoBERTa-XXL with 11B parameters, and the closed-source GPT-4 with an estimated 1.7T parameters, pushed the boundaries of performance but come with substantial computational costs and hardware requirements.

Our research aims to address these challenges by comparing three commonly used compression methods — quantization, pruning, and knowledge distillation — in compressing the xCOMET model. We have demonstrated that these methods can effectively reduce memory consumption and increase processing speed while maintaining competitive performance, making them viable options for deploying large state-of-the-art learned metric for MT evaluation in a resource-constrained environments. In particular, our distilled model xCOMET-lite achieves competitive prediction quality with a substantially smaller model size, offering a solution for researchers and practitioners with no access to top-tier hardware.

Based on our findings, we recommend the following: for the highest quality with a reduced VRAM requirements, opt for 8-bit or 3-bit quantization with GPTQ. For improved speed without substantial quality penalty, test 4-bit quantization with QLoRA, try structured magnitude pruning (2:4, 4:8) or prune up to 25% of the model layers.For massive speedup and low hardware requirements, consider the distilled model xCOMET-lite or its quantized version, accepting a slight compromise on quality. The choice of compression method ultimately depends on the hardware, amount of data, and acceptable quality loss.

Acknowledgments

The NLLG group gratefully acknowledges support from the Federal Ministry of Education and Research (BMBF) via the research grant "Metrics4NLG" and the German Research Foundation (DFG) via the Heisenberg Grant EG 375/5-1.

The NLLG group acknowledges support by the state of Baden-Württemberg through bwHPC.

7 Limitations

While our research provides valuable insights into the compression of large language models for machine translation evaluation, it is important to acknowledge the limitations of our work.

- Our study focuses solely on machine translation evaluation and does not consider other tasks, such as summarization evaluation. To the best of our knowledge, all currently existing summarization evaluation metrics are regression-only and do not offer error span prediction. Therefore, it is unclear if the results would be different for this task. Future research could explore the applicability of these compression methods to a broader range of natural language processing tasks.
- Our measure of a metric quality, Kendall-τ correlation with human judgments, is known to incorrectly reward metrics for predicting ties (Deutsch et al., 2023).
- Although our research has potential implications for low-resource machine translation, we did not conduct experiments on low-resource language pairs. We plan to address this limitation when releasing the subsequent versions of our models to the public.
- Our distillation approach still requires the availability of the original teacher model. Training such a model is expensive in terms of both computational resources and the cost of human annotation for the training data.

References

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3576–3588, Online. Association for Computational Linguistics.

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

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *CoRR*, abs/2208.07339.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tim Dettmers and Luke Zettlemoyer. 2023. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 7750–7774. PMLR.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12914–12929, Singapore. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021a. 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, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. 2023. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pages 578–628, Singapore. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André 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 (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021b. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*, pages 733–774, Online. Association for Computational Linguistics.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. 2023. Specializing smaller language models towards multi-step reasoning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10421–10430. PMLR.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. A survey of quantization methods for efficient neural network inference. *CoRR*, abs/2103.13630.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. 2024. The unreasonable ineffectiveness of the deeper layers. *CoRR*, abs/2403.17887.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Nuno Miguel Guerreiro, Ricardo Rei, Daan van Stigt, Luísa Coheur, Pierre Colombo, and André F. T. Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *CoRR*, abs/2310.10482.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference*

on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023. OpenReview.net.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of proprietary large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3134–3154, Singapore. Association for Computational Linguistics.

Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh, and Markus Freitag. 2023. MetricX-23: The Google submission to the WMT 2023 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 756–767, Singapore. Association for Computational Linguistics.

Moussa Kamal Eddine, Guokan Shang, Antoine Tixier, and Michalis Vazirgiannis. 2022. FrugalScore: Learning cheaper, lighter and faster evaluation metrics for automatic text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1305–1318, Dublin, Ireland. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, pages 768–775, Singapore. Association for Computational Linguistics.

Christoph Leiter, Juri Opitz, Daniel Deutsch, Yang Gao, Rotem Dror, and Steffen Eger. 2023. The Eval4NLP 2023 shared task on prompting large language models as explainable metrics. In *Proceedings of the 4th Workshop on Evaluation and Comparison of NLP Systems*, pages 117–138, Bali, Indonesia. Association for Computational Linguistics.

Lujun Li and Zhe Jin. 2022. Shadow knowledge distillation: Bridging offline and online knowledge transfer. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *CoRR*, abs/2403.03853.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ibraheem Muhammad Moosa, Rui Zhang, and Wenpeng Yin. 2024. Mt-ranker: Reference-free machine translation evaluation by inter-system ranking. *CoRR*, abs/2401.17099.

Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. 2021. A white paper on neural network quantization. *CoRR*, abs/2106.08295.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Jan-Thorsten Peter, David Vilar, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, and Markus Freitag. 2023. There's no data like better data: Using QE metrics for MT data filtering. In *Proceedings of the Eighth Conference on Machine Translation*, pages 561–577, Singapore. 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.

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, Ana C Farinha, José G.C. de Souza, Pedro G. Ramos, André F.T. Martins, Luisa Coheur, and Alon Lavie. 2022b. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, pages 61–70, Ghent, Belgium. European Association for Machine Translation.

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.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Aji. 2024. LaMini-LM: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 944–964, St. Julian's, Malta. Association for Computational Linguistics.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. *CoRR*, abs/2401.08417.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1–9. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China. Association for Computational Linguistics.

A Models and Algorithms used for Data Collection

- OPUS-MT (Tiedemann and Thottingal, 2020) monodirectional models: *en-ru*, *ru-en*, *en-zh*, *zh-en*, *en-de*, *de-en*.
- OPUS-MT models for multiple languages: *mul-en* and *en-mul*.
- NLLB models (Costa-jussà et al., 2022), versions: Distilled 600M and 1.3B, Non-Distilled 1.3B and 3.3B.
- Word Drop: it was used to create translation hypotheses by randomly dropping 15% of the words from reference translation.
- Word Replacement with MLM: similarly we applied XLM-RoBERTa-Large for masked language modelling task to replace 15% of the words.
- Backtranslation: we applied NLLB-1.3B model to translate references into a proxy language and back. As a proxy languages we used French and Japanese.
- Backtranslation + MLM: consists of applying MLM to the results of backtranslation.

B Kendall Correlation

Kendall-τ correlation is defined as follows: let $(x_1, y_1), \ldots, (x_n, y_n)$ be observations of random variables X and Y such that all values of x_i and y_i are unique. A pair of observations (x_i, y_i) and (x_j, y_j) is said to be *concordant* if either $x_i < x_j;\ y_i < y_j$ or $x_i > x_j;\ y_i > y_j$, otherwise this pair is *discordant*. The Kendall correlation coefficient τ is

$$\tau = \frac{n_c - n_d}{C_n^2} = \frac{C_n^2 - n_d - n_d}{C_n^2}$$
$$= 1 - \frac{2n_d}{C_n^2} = 1 - \frac{4 \cdot n_d}{n(n-1)}$$

where n is the total amount of observations, n_c is the amount of concordant pairs, and n_d is the amount of discordant pairs. Kendall correlation coefficient is more robust to outliers than Pearson correlation and better captures non-linear dependencies. In our case, X is the ground truth MQM score, and Y is the score estimated by the neural metric.

C Interaction Analysis of Distillation and Pruning

# pruned layers	Avg. correlation with ref.	Avg. correlation without ref.
2	0.240	0.209
4	0.264	0.202
6	0.201	0.181

Table 4: Results of evaluation of xCOMET-lite distilled from xCOMET-XXL with applied pruning. *Avg. correlation* represents Kendall correlation averaged across 3 language pairs.

D Interaction Analysis of Distillation and Quantization

Method	# bits	Avg. correlation with ref.	Avg. correlation without ref.	Peak Mem. Cons. (GB)
LLM.int8()	8	0.369	0.355	1.2 (1.5)
QLoRA	4	0.379	0.345	1.1 (1.4)

Table 5: Results of evaluation of xCOMET-lite distilled from xCOMET-XXL with applied quantization. *Avg. correlation* represents Kendall correlation averaged across 3 language pairs.

E Detailed results on compression and quantization

See Figure 2.

F Results on Eval4NLP dataset

In addition to WMT Shared Metric dataset, we perform evaluations on Eval4NLP dataset, in setting without reference translation. The results are shown on Figure 3 and Tables 6, 7. All conclusions are stable with respect to another dataset.

G Varying seed for layer pruning

To check the robustness of finetuning procedure in layer pruning technique, we run the same pipeline with three seeds. The standard deviations are presented in Table 8.

H Additional Details

In this section we discuss some additional details concerning our research.

H.1 Risks

While our work demonstrates the potential of distillation, quantization, and pruning techniques in creating an efficient alternative to xCOMET, there are some risks to consider:

- The use of distilled models like xCOMET-lite, as well as over-pruned models, in high-stakes applications, such as filtering datasets or evaluating machine translation systems in sensitive domains (e.g., healthcare, legal), may lead to suboptimal decisions due to the slightly lower accuracy compared to the full xCOMET model. One must exercise discretion when considering acceptable loss of quality.
- Our work primarily focuses on high-resource languages, and the performance of the compressed models on low-resource languages

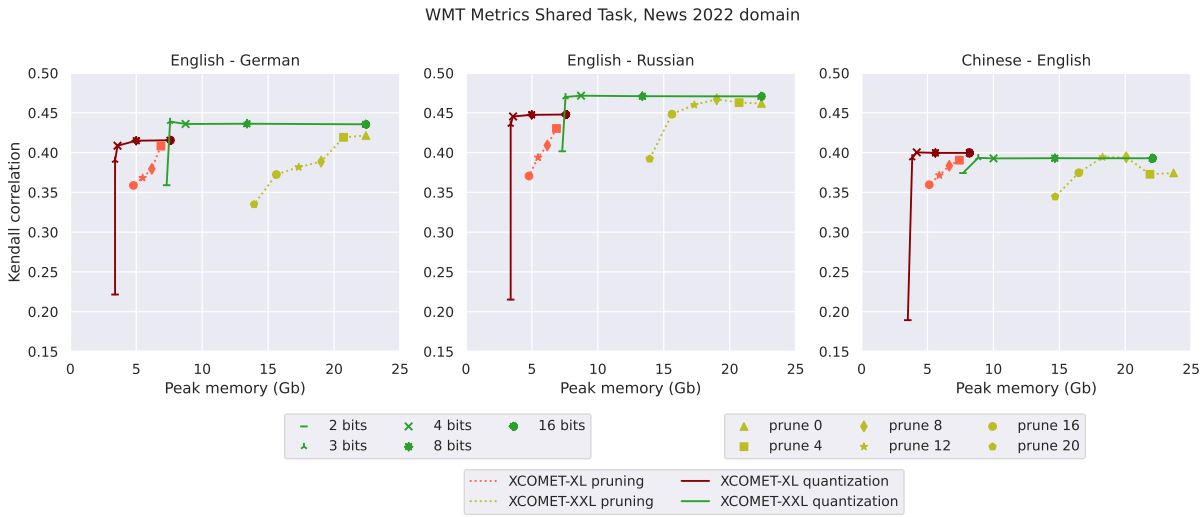

Figure 2: Results on WMT MQM Human Evaluation dataset. In this setting xCOMET has access to reference translation.

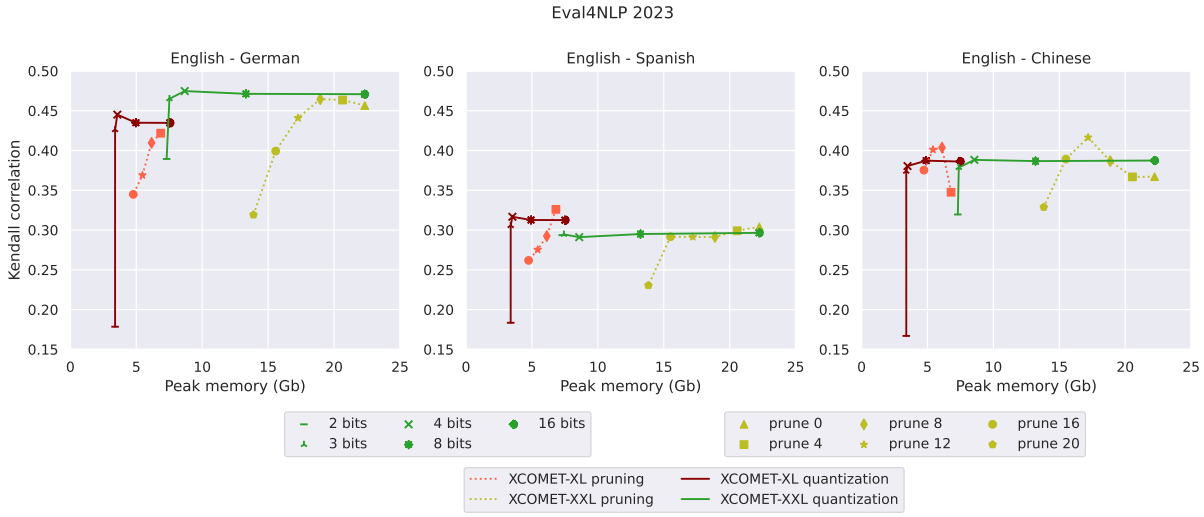

Figure 3: Results on Eval4NLP dataset. This is reference-free setting, also known as Quality Estimation (QE).

Model	Compression method	Average Kendall correlation	Peak memory consumption (Gb)
XL	None	0.378	7.51 (7.54)
XL	GPTQ 8 bit	0.379	4.94 (4.97)
XL	GPTQ 3 bit	0.372	3.39 (3.39)
XL	LLM.int8()	0.384	6.98 (7.06)
XL	QLoRA 4 bit	0.373	3.50 (3.53)
XL	Prune 8 layers	0.373	6.13 (6.16)
XL	Prune 16 layers	0.359	4.75 (4.77)
XL	Magnitude pruning 4:8	0.362	7.51 (7.55)
XL	Wanda pruning 4:8	0.342	8.01 (8.01)
XXL	None	0.385	22.24 (22.30)
XXL	GPTQ 8 bit	0.385	13.25 (13.32)
XXL	GPTQ 3 bit	0.378	7.44 (7.51)
XXL	LLM.int8()	0.383	16.78 (16.94)
XXL	QLoRA 4 bit	0.373	9.09 (9.94)
XXL	Prune 8 layers	0.381	18.91 (18.97)
XXL	Prune 16 layers	0.360	15.53 (15.57)
XXL	Magnitude pruning 4:8	0.340	22.25 (22.31)
XXL	Wanda pruning 4:8	0.340	22.50 (22.50)
XXL	Distilled (xCOMET-lite)	0.363	1.4 (1.4)

Table 6: An overview table with some representative results for various compression methods in setting **without** reference translations. Average is computed over three language pairs for Kendall correlation. For peak memory the mean and maximum values are computed, and the maximum is reported in parentheses. XL stands for xCOMET-XL, XXL – xCOMET-XXL.

Model	Compression method	Average Kendall correlation	Samples per second RTX 3090 (min / median / max)	Samples per second A100 (min / median / max)
XL	None	0.378	55.1 / 67.0 / 70.5	76.2 / 98.9 / 111.8
XL	GPTQ 8 bit	0.379	30.8 / 35.1 / 35.9	53.0 / 69.0 / 72.9
XL	GPTQ 3 bit	0.372	28.7 / 33.3 / 33.7	57.3 / 71.3 / 74.0
XL	LLM.int8()	0.384	50.9 / 59.7 / 63.8	64.1 / 87.2 / 88.1
XL	QLoRA 4 bit	0.373	55.0 / 66.2 / 68.7	93.0 / 123.2 / 135.5
XL	Prune 8 layers	0.373	70.5 / 85.6 / 87.7	94.5 / 119.5 / 131.2
XL	Prune 16 layers	0.359	82.9 / 108.6 / 110.2	110.4 / 128.3 / 149.2
XXL	None	0.385	22.1 / 24.2 / 25.2	35.4 / 48.3 / 48.6
XXL	GPTQ 8 bit	0.385	8.1 / 8.5 / 8.5	23.7 / 28.9 / 29.6
XXL	GPTQ 3 bit	0.378	8.6 / 9.2 / 9.3	20.9 / 23.9 / 29.1
XXL	LLM.int8()	0.383	27.8 / 30.8 / 32.1	38.3 / 48.2 / 48.9
XXL	QLoRA 4 bit	0.373	21.8 / 25.2 / 25.5	42.6 / 51.9 / 57.4
XXL	Prune 8 layers	0.381	25.4 / 28.4 / 29.8	42.6 / 56.7 / 60.2
XXL	Prune 16 layers	0.360	30.0 / 34.8 / 36.3	50.8 / 64.4 / 68.1
XXL	Distilled (xCOMET-lite)	0.363	312.1 / 352.0 / 358.0	229.0 / 232.2 / 241.9

Table 7: Speed results for different methods in setting **without reference**. Importantly, here the memory consumption is higher than in Table 6, as we aim for maximal throughput on a given GPU. Average and std are computed over three language pairs for Kendall correlation. Samples per second are reported for both 3090 and A100 GPUs. XL stands for xCOMET-XL, XXL – xCOMET-XXL.

Model	Compression method	Chinese - English	English - Russian	English - German	Peak memory consumption (GB)
XL	None	0.399	0.448	0.415	7.76 (8.17)
XL	Prune 8 layers	0.387 ± 0.005	0.414 ± 0.006	0.381 ± 0.004	6.34 (6.66)
XL	Prune 16 layers	0.362 ± 0.002	0.369 ± 0.006	0.359 ± 0.009	4.90 (5.14)
XXL	None	0.390	0.470	0.435	22.27 (22.39)
XXL	Prune 8 layers	0.398 ± 0.000	0.435 ± 0.000	0.385 ± 0.000	19.39 (20.09)
XXL	Prune 16 layers	0.372 ± 0.001	0.445 ± 0.004	0.352 ± 0.006	15.91 (16.48)

Table 8: Robustness of layer pruning approach to random seed, setting **with** reference translations. For peak memory consumption, the mean and maximum values are computed, and the maximum is reported in parentheses. XL stands for xCOMET-XL, XXL – xCOMET-XXL.

remains unexplored. The lack of training data and the potential differences in linguistic characteristics may lead to suboptimal performance when applying these models to evaluate translations in low-resource language pairs. This could result in inaccurate quality assessments and hinder the development of reliable machine translation systems for these languages.

- The availability of highly efficient evaluation metrics like xCOMET-lite may prompt researchers and practitioners to conduct large-scale experiments, such as web-scale dataset filtration or extensive hyperparameter optimization. While these experiments can lead to valuable insights and improvements in machine translation systems, they may also consume substantial amounts of computational resources and power. This increased energy consumption could contribute to environmental concerns and raise questions about the sustainability of such practices.

H.2 Artifacts

The main artifact that we use in our research is a set of two pre-trained metrics for MT evaluation: xCOMET-XL and xCOMET-XXL, released by (Guerreiro et al., 2023). Those models are released under *cc-by-nc-sa-4.0* license. Our use of these models complies with the license and is consistent with usage permissions.

We plan to release two of our own artifacts: the distilled model xCOMET-lite and the dataset that was used to train it. Both of those artifacts will also be released under *cc-by-nc-sa-4.0* according to the "share-alike" requirement of this license, as derivatives of the original xCOMET models.

H.3 PII in the dataset

According to the dataset card of the NLLB dataset [4], the data may contain personally identifiable information (PII). Identifying and anonymizng such information is outside of the scope of this work. We plan to address it in future, before releasing dataset to the public.

H.4 Used packages

In our experiments we use the following key software libraries:

- PyTorch: v2.0.1
- Transformers: v4.41.2
- BitsAndBytes: v0.41.1
- AutoGPTQ: v0.7.0
- Optimum: v1.11.0
- SciPy: v1.11.1
- Unbabel COMET: v2.0.2

[4] https://huggingface.co/datasets/allenai/nllb