Adding Chocolate to MINT: Mitigating Metric Interference in Machine Translation

José Pombal[1,2,3], **Nuno M. Guerreiro**[1,2,3,4], **Ricardo Rei**[1], **André F. T. Martins**[1,2,3,5]
[1]Unbabel, Portugal, [2]Instituto de Telecomunicações, Portugal,
[3]Instituto Superior Técnico, Universidade de Lisboa, Portugal,
[4]MICS, CentraleSupélec, Université Paris-Saclay, France, [5]ELLIS Unit Lisbon, Portugal
jose.pombal@tecnico.ulisboa.pt

Abstract

As automatic metrics become increasingly stronger and widely adopted, the risk of unintentionally ''gaming the metric'' during model development rises. This issue is caused by metric interference (MINT), *i.e.*, the use of the same or related metrics for both model tuning and evaluation. MINT can misguide practitioners into being overoptimistic about the performance of their systems: As system outputs become a function of the interfering metric, their estimated quality loses correlation with human judgments. In this work, we analyze two common cases of MINT in machine translation-related tasks: Filtering of training data, and decoding with quality signals. Importantly, we find that MINT strongly distorts instance-level metric scores, even when metrics are not directly optimized for—questioning the common strategy of leveraging a different, yet related metric for evaluation that is not used for tuning. To address this problem, we propose MINTADJUST, a method for more reliable evaluation under MINT. On the WMT24 MT shared task test set, MINTADJUST ranks translations and systems more accurately than state-of-the-art-metrics across a majority of language pairs, especially for high-quality systems. Furthermore, MINTADJUST outperforms AUTORANK, the ensembling method used by the organizers.[1]

1 Introduction

Automatic evaluation metrics are evolving beyond their traditional role in model selection, now serving as tools for enhancing the performance of large language models (LLMs) across their entire development pipeline. Applications include filtering high-quality training data, or serving as proxies for human preferences in alignment and decoding (Xu et al., 2024b; Zhu et al., 2024; Wu et al., 2024). In the field of machine translation (MT), several works have leveraged such methods to claim state-of-the-art results (Fernandes et al., 2022; Xu et al., 2024b; Rei et al., 2024). In these cases, the same or similar metrics to the one used for optimization are also used downstream for evaluation—we call this phenomenon **metric interference** (MINT).

An unintended consequence of MINT is that it may jeopardize evaluation. For example, an interfering metric can lose much of its correlation with human judgements when ranking systems (Fernandes et al., 2022), or make the optimized model generate absurd outputs that the metric scores highly (Yan et al., 2023). At worst, authors disregard MINT and use the same metric for optimization and evaluation. At best, as recommended by Kocmi et al. (2024b), they acknowledge it and perform evaluation with another metric (Fernandes et al., 2022; Rei et al., 2024; Kocmi et al., 2024a). The latter solution, however, may not work well if metrics correlate strongly among each other, as they can have similar biases. The effects of MINT are not well understood, and existing mitigation methods lack effectiveness.

In the first part of our work, we investigate the implications of MINT for evaluation with a series of metrics on two common MINT cases in MT: 1) data filtering, where a metric is used to select high-quality data from a larger pool to more efficiently train a translation model; 2) MBR, a widely used decoding strategy that uses a metric to select the best translation from a pool of candidates. We find that the effects of MINT are stronger for MBR than for data filtering. Specifically, the instance-level scores of metrics are heavily distorted, to the point where large improvements in an interfering metric become consistently related to deterioration in other metrics. Importantly, similar distortions hold when comparing an interfering metric with human scores, explaining

[1]We will release a codebase for replicating the results in this work upon publication.

Transactions of the Association for Computational Linguistics, vol. 13, pp. 1319–1339, 2025. https://doi.org/10.1162/TACL.a.37
Action Editor: Hermann Ney. Submission batch: 2/2025; Revision batch: 4/2025; Published 10/2025.

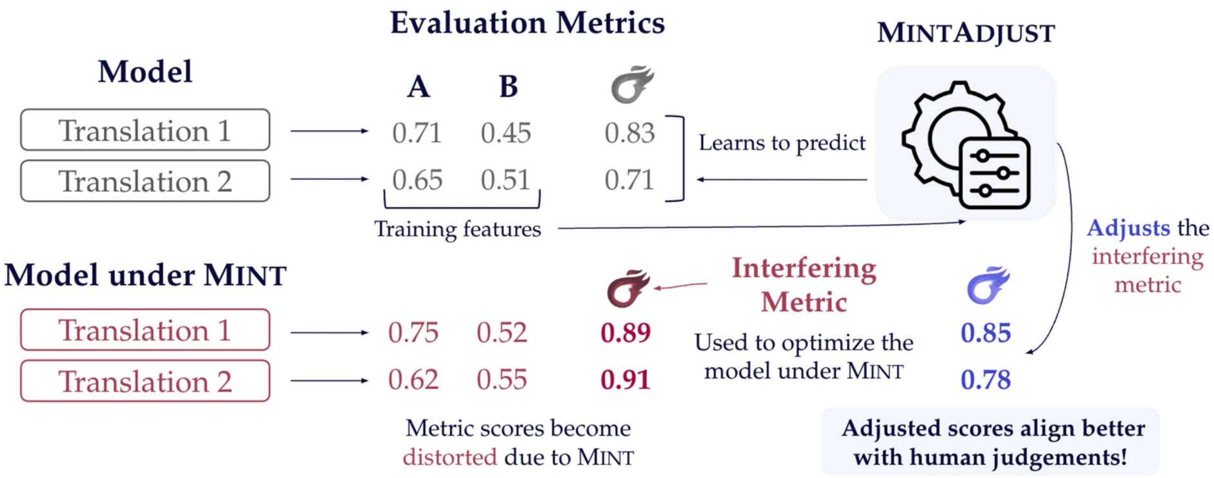

Figure 1: MINTADJUST learns to predict the instance-level scores of an interfering metric, given the scores of other metrics measured in translations produced by models that are not under MINT (in gray). At test time, for each instance from the model under MINT, MINTADJUST receives the scores of other metrics, and produces a new interfering metric score that correlates better with human judgments.

the system-level findings of Fernandes et al. (2022). Furthermore, the scores of metrics that correlate with the interfering metric will also be overestimated. This brings into question the common practice of using a single related metric for evaluation.

Building on our findings, we propose a method for correcting an interfering metric's scores by learning its distribution without MINT from a large set of other metrics, as opposed to relying on a single one: MINTADJUST (summarized in Figure 1). Using WMT24 human-judged translations, we show that, under MINT, MINTADJUST ranks systems and individual translations more accurately than the interfering metric and other state-of-the-art metrics on a majority of language pairs. Its performance is particularly strong when comparing high-quality MT systems. Our correction method generalizes across translation models and metrics—*i.e.*, it can be learned from a given set of models and metrics and applied to a different model.

Through our fine-grained analysis and the proposed methods, we lay the groundwork for better evaluation practices in settings where automatic metrics are used for both evaluation and model optimization. Our work has implications not only for MT but for evaluation in general.

2 Metric Interference

When used for evaluation, most automatic metrics $\mathcal{M}$, may be described as functions that produce a judgement $\hat{q}$ given an input x, a model's output h, and a reference output r:[2]

$$\hat{q} = \mathcal{M}(x, h(x), r). \quad (1)$$

MINT occurs when $\mathcal{M}$—the **interfering metric**—influences a model's output, directly or indirectly, through its judgements. Thus, during evaluation, $\mathcal{M}$ becomes a function of itself:

$$\hat{q} = \mathcal{M}(x, h(x; \mathcal{M}'), r) \quad (2)$$

$\mathcal{M}'$ and $\mathcal{M}$ can be the same or spuriously correlated metrics—we will show later that MINT also affects the latter. **Direct interference** occurs when a metric is used to choose among model outputs at test time; for example, in MBR the best candidate out of a pool of model generations is chosen by a metric. **Indirect interference** happens when the metric influences the model's outputs by influencing the model's weights, θ, through either data or tuning: $\hat{q} = \mathcal{M}(x, h(x; \theta(\mathcal{M})), r)$. Training data filtering is a case of indirect interference, whereby the metric influences the model's weights by changing its training distribution. Preference optimization is an example of indirect interference through tuning. We posit the negative effects of MINT on evaluation are stronger for direct interference.

To the best of our knowledge, MINT has never been formally defined but its effects have been

[2] In translation, some metrics only use x and h (*e.g.*, quality estimation), or h and r (*e.g.*, lexical metrics)

studied empirically to some extent. For example, LLM judges tend to favor outputs from their own model family (Panickssery et al., 2024; Verga et al., 2024). In MT, this is also the case with LLMs (Agrawal et al., 2024) and more traditional neural metrics: models optimized for BLEURT can produce absurd translations that score highly (Yan et al., 2023), while metric-based decoding strategies (e.g., MBR) reduce metric correlation with human judgments at the system level (Fernandes et al., 2022).

A typical solution to MINT is using a different metric, $\mathcal{M}'$, for evaluation. The validity of this approach hinges on the often omitted assumption that $\mathcal{M}$ and $\mathcal{M}'$ do not have spurious correlations beyond their correlations with human judgments. We posit that this does not hold in MT, since most metrics are trained on similar data and architectures and so may share similar behaviours, which can be beneficial or detrimental for evaluation.

3 Experimental Setting

3.1 Overview

We analyze four language pairs (LPs) of two scripts where high-quality test data is available: English↔German and English↔Chinese. We use the WMT23 test set and a diverse suite of reference-based, reference-free, neural, and lexical metrics for evaluation: COMET-22 (Rei et al., 2022a), COMETKIWI (Rei et al., 2022b), xCOMET-XL-REF and QE (Guerreiro et al., 2023), METRICX-XL-REF and QE (Juraska et al., 2023), BLEURT (Sellam et al., 2020), CHRF (Popović, 2015), and BLEU (Papineni et al., 2002). We also present additional findings using LLM Judges PROMETHEUS 2 7B (Kim et al., 2024) and INSTRUCTSCORE (Xu et al., 2023).

3.2 Training Data Filtering

Quality Estimation (Specia et al., 2010, QE) is the task of assessing the quality of a translation without a reference. Using QE metrics to filter high-quality training data from a larger pool—as opposed to training models on the entire pool or a random sample—is an efficient method for building stronger models for translation (Peter et al., 2023; Alves et al., 2024). However, using a metric for data filtering might lead to bias in evaluation. For example, a model trained on COMETKIWI-filtered data may produce translations that COMETKIWI and COMET score higher than those of a model trained on METRICX-filtered data only because it has learned properties that the former metrics prefer, but that the latter metric and humans do not.

To investigate whether this is the case, we fine-tune GEMMA-2B (Gemma Team, 2024) models[3] on 80,000 examples from a pool of 10 million examples[4] sourced from OPUS (Tiedemann, 2012), sampled in 4 different ways: 1) random sampling (baseline); 2) top-k COMETKIWI (CKiwi) filtering; 3) top-k METRICX-XL-QE (MX) filtering; and 4) top-k xCOMET-XL-QE (xC) filtering.[5] Table 7 reveals minimal overlap in training sets, indicating that different metrics are likely selecting translations based on distinct preferred aspects.

3.3 Minimum Bayes Risk Decoding

Decoding strategies informed by quality metrics such as minimum Bayes risk (MBR) decoding consistently outperform alternatives like greedy decoding or nucleus sampling, according to humans (Fernandes et al., 2022; Freitag et al., 2022a; Nowakowski et al., 2022; Farinhas et al., 2023). Fernandes et al. (2022) show the effectiveness of MBR for LLM-based translation but warn against using the same metric for evaluation due to MINT. We aim to expand this kind of analysis to more metrics and to the instance-level to provide more extensive insight. Our experiments leverage TOWER-v2-7B, a state-of-the-art LLM for machine translation. We generate pools of 20 candidates through ϵ-sampling (Freitag et al., 2023) with $\epsilon = 0.02$, and perform MBR with four metrics: COMET-22, BLEURT, CHRF, and BLEU. Greedy decoding serves as the baseline. We replicate the analysis with 50 candidates and TOWER-v2-70B to assess the impact of model and pool sizes and reach similar conclusions (see Tables 14 and 15).

[3] We use GEMMA-2B because it is a small (practical for finetuning), strong LLM that is not tailored for translation, so the effects of data filtering are more likely to be visible.

[4] 80k instances is a typical dataset size for fine-tuning LLMs for translation (Xu et al., 2024a; Alves et al., 2024), and there are diminishing marginal returns in performance to increasing it. Sampling from a much larger pool is realistic and it allows us to measure the effects of filtering with different metrics more easily (the larger the pool, the more likely metrics are to select different instances).

[5] We provide full details on the metrics and checkpoints used in Appendix Section A.

Filtering	Lexical		Neural						
	CHRF↑	BLEU↑	COMET↑	BLEURT↑	COMETKIWI↑	xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Random	24.36	21.70	76.78	59.71	67.79	70.19	69.47	3.78	3.93
COMETKIWI	**26.57**	**26.16**	**79.28**	**62.73**	$\boxed{\textbf{72.99}}$	74.35	73.23	2.86	2.59
xCOMET	26.13	24.16	77.51	59.05	68.54	**75.27**	$\boxed{\textbf{74.75}}$	2.97	3.30
METRICX	24.04	23.07	78.74	61.87	72.08	74.85	73.81	**2.66**	$\boxed{\textbf{2.44}}$

Table 1: System-level performance on the WMT23 English→Chinese test set across several metrics of GEMMA-2-2B with different training data filtering strategies. Cells where the interference and evaluation metrics are the same have boxes around values. Cells are green if they represent an improvement with respect to random filtering, red otherwise. Values in **bold** are the highest of each evlauation metric.

Decoding	Lexical		Neural						
	CHRF↑	BLEU↑	COMET↑	BLEURT↑	COMETKIWI↑	xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Greedy	43.23	**44.46**	87.22	72.89	81.23	87.27	84.31	1.40	1.34
MBR									
CHRF$_L$	$\boxed{\textbf{43.45}}$	43.97	87.15	72.78	81.16	86.76	84.14	1.42	1.32
BLEU$_L$	42.99	$\boxed{44.34}$	87.17	72.79	81.08	86.82	84.02	1.43	1.35
COMET$_N$	39.75	40.21	$\boxed{\textbf{88.38}}$	72.55	**81.95**	**87.95**	**85.58**	1.30	**1.17**
BLEURT$_N$	38.91	39.14	87.06	$\boxed{\textbf{74.33}}$	81.65	87.61	85.41	**1.29**	1.18

Table 2: System-level performance on the WMT23 English→Chinese test set across several metrics of TOWER-v2-7B with different decoding strategies. Cells where the interference and evaluation metrics are the same have boxes around values. Cells are green if they represent an improvement with respect to greedy, red otherwise. Values in **bold** are the highest of each evlauation metric. Subscripts indicate metric family: L means lexical, N is Neural.

4 Analysis

We focus on English→Chinese, where the phenomena we describe are more pronounced. Similar conclusions hold for other LPs (see Tables 8 to 13).

4.1 Training Data Filtering

Metrics prefer systems trained on data hat they selected. Table 1 shows system-level evaluations for data filtering. Systems optimized for a certain metric score higher on that metric, which is consistent with findings from past literature for MBR (Fernandes et al., 2022). Reference-based versions of QE metrics also score higher on these cases (e.g., xCOMET-REF is highest when filtering with xCOMET-QE), likely because they were trained with similar architectures and data. This is a warning against evaluating with a single metric even in cases of indirect interference.

There are few disagreements in metric preferences. While the interfering metric seems to be biased in favor of itself and metrics of the same family, most metrics agree on improvements over the baseline. Notably, both neural and lexical metrics usually improve with data filtering—contrary to past findings on MBR (Fernandes et al., 2022)—with COMETKIWI filtering leading to improvements across the board on all LPs (see Tables 8 to 10). While this reinforces that models trained on filtered data outperform those trained on random samples, MINT makes it hard to understand which filtering metric is the best.

4.2 Minimum Bayes Risk Decoding I: System-Level Analysis

Interfering metrics prefer the systems they were used on and there are disagreements. Table 2 shows system-level evaluations on

Decoding	CHRF↑	COMET↑	Ptheus↑	IScore↑
Greedy	**49.80**	81.58	3.54	−6.07
COMET	48.87	**82.32**	3.58	−5.83
Ptheus	46.99	80.78	**4.06**	−5.99
IScore	47.24	81.00	3.66	**−4.86**

Table 3: Variant of Table 2 for Chinese→English with two LLM Judges: PROMETHEUS 2 7B (Ptheus) and INSTRUCTSCORE (IScore). Values in **bold** are the highest of each evlauation metric.

English→Chinese for MBR. Again, systems optimized for a certain metric score higher on that metric, with considerable differences in scores (e.g., >1 COMET point).[6] However, unlike data filtering, there are significant disagreements among metrics. With respect to greedy decoding, lexical metrics always deteriorate when performing MBR with neural metrics—as reported by Fernandes et al. (2022). Neural metrics can also disagree, like BLEURT and COMET, but this is an exception: every pair of neural metrics agree on all other LPs (see Tables 11 to 13). Conversely, performing MBR with lexical metrics rarely leads to improvements on any metric. In this case, it is both hard to choose the best MBR metric, and whether improvements over greedy decoding are *actually real*.

LLM judges also suffer from MINT. Table 3 shows additional results using LLM Judges PROMETHEUS 2 and INSTRUCTSCORE for evaluation and MBR on Chinese→English.[7] Both judges behave similarly to neural metrics in that they improve according to themselves and cause degradation on all MT-specific metrics.

4.3 Minimum Bayes Risk Decoding II: Instance-Level Analysis

We performed an instance-level analysis to understand the root cause of the system-level findings in Section 4.2. We focused on comparing differences in metric scores between pairs of models with disparate performances. One pair is TOWER-v2-70B and TOWER-v2-7B with greedy decoding, which represents a metric-agnostic improvement. The other is TOWER-v2-7B with COMET MBR and greedy decoding, an improvement driven by MINT. Comparing the two pairs allows us to isolate the effect of MINT on metric score deltas to better understand metric disagreements.

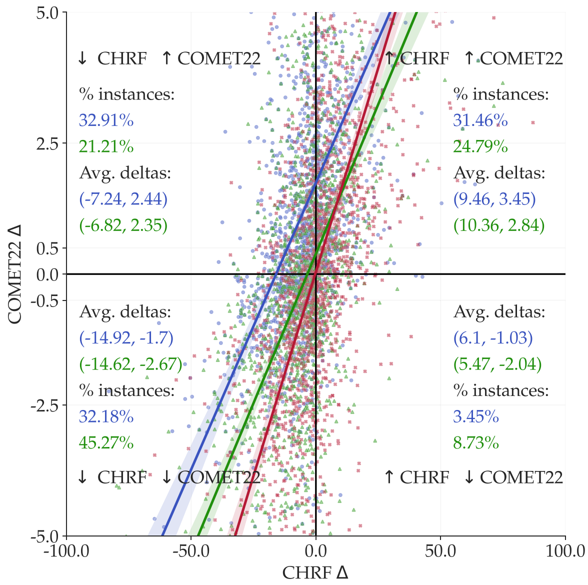

Figure 2: Difference (Δ) in COMET and CHRF between models of three sets: (TOWER-v2-70B greedy, TOWER-v2-7B greedy); (TOWER-v2-7B + COMET MBR, TOWER-v2-7B greedy); (TOWER-v2-7B + BLEURT MBR, TOWER-v2-7B greedy). We include a linear regression for each set of points. Metric interference causes distortions in score relationships between the interfering metric and correlated metrics.

System-level metric disagreement is *actually* driven by a minority of instances. Figure 2 shows the difference in CHRF and COMET between the aforementioned model pairs (in red and blue, respecitvely) for English→Chinese. When comparing models where MINT is not an issue, we see that most instances (>70%) lie in quadrants where both metrics agree on the direction of change. When comparing models under MINT, a mass of instances shifts towards the disagreement quadrant that favors the interfering metric (COMET). The shift is so pronounced that a delta of 0 points in CHRF corresponds to a non-zero change in COMET,[8] highlighting a stark distortion

[6] According to Kocmi et al. (2024b) this would mean that humans would prefer the winning system >90% of the time; that percentage is likely overestimated due to MINT, though.

[7] Out of the WMT23 LPs, INSTRUCTSCORE only supports Chinese→English and English→German.

[8] The change is almost 2 points on average, which, according to Kocmi et al. (2024b), would correspond to humans preferring translations from the system 95% of the time in a setting without MINT. Surely, given the observed score distortions, we cannot make such interpretations under MINT.

in the scores of the interfering metric. That said, this only occurs to roughly 33% of instances; for most instances (around 63%), metrics still agree on the direction of change. Similar phenomena occur among other metrics (we omit these plots for the sake of brevity).

Metric interference score distortions affect correlated metrics. A common strategy to mitigate MINT is to use different metrics for interference and evaluation. Figure 2 shows a third model pair in green: TOWER-v2-7B with BLEURT MBR and greedy decoding. The impact of BLEURT MBR in distorting metric scores is similar to that of COMET MBR, though less pronounced. This is likely because BLEURT is highly correlated with COMET,[9] and it highlights the inadequacy of using correlated metrics for evaluation under MINT: They may be less biased than the interfering metric but they will still be biased.

Instance-level metric distortions extend to human evaluation. Figure 3 shows that the same score distortions we observed among automatic metrics hold between an interfering metric and human scores. We use WMT24 English→Chinese data because it has both human evaluations and systems that are comparable to our initial setup. We consider system sets instead of pairs for a larger sample of points. Our representative unbiased system (previously TOWER-v2-70B greedy) becomes GPT-4, the best model in this LP according to human evaluation. The biased system is still TOWER-v2-70B with COMET MBR. For each translation of these systems, we obtain deltas from four other systems that we are sure did not leverage COMET (yielding four points of each color per translation): IOL-Research, IKUN, IKUN-C, and Llama-3-70B.

4.4 Final Remarks

We analyzed the impact of MINT on evaluation with several metrics at system- and instance-levels. Our system-level findings corroborate those of existing literature, and our instance-level analysis adds an important layer of understanding: MINT causes non-trivial score distortions between the interfering metric and other metrics and human scores. Importantly, these distortions extend to correlated metrics. These findings lead to two conclusions: 1) instance-level information is crucial for understanding MINT; 2) performing evaluation with metrics that are spuriously correlated with the interfering metric is potentially just as misleading as using the interfering metric itself, bringing common practices into question.

[9]It is generally known that neural metrics correlate strongly due to similar training data and architectures (Kocmi et al., 2024b; Kovacs et al., 2024). We include an instance-level correlation matrix in Table 16 to further support this; COMET-BLEURT correlation is 0.75. Furthermore, on system level evaluation, while the two metrics disagree when using the other for MBR on English→Chinese, they agree on all other LPs (see Tables 11 to 13. If anything, the result on English→Chinese shows how system-level information is insufficient to fully understand MINT.

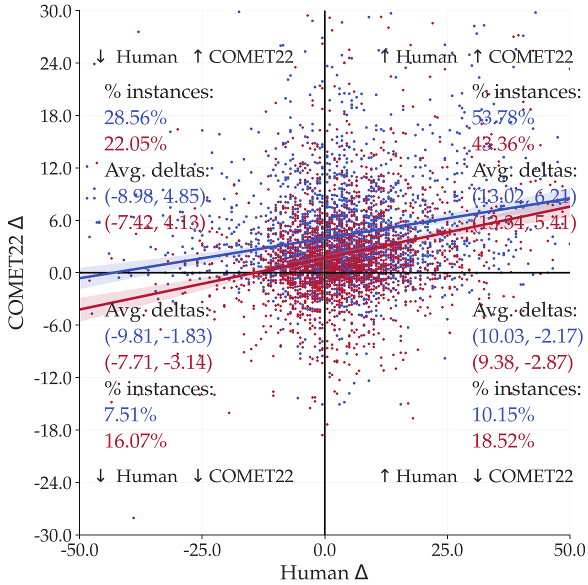

Figure 3: Difference (Δ) in COMET and Human scores on WMT24 English→Chinese of two sets of systems (MINT affects the blue set).

5 Improving Evaluation Under MINT

5.1 MINTADJUST

Leveraging the insights of Section 4, we propose MINTADJUST, a method for correcting the instance scores of the interfering metric during evaluation (see Figure 1). MINTADJUST does not require human judgments, instead leveraging the scores of other metrics. For a test set $\mathcal{D}$ containing (source, translation, reference) triplets $\langle x, h, r \rangle$, MINTADJUST learns to predict the interfering metric, $\hat{q}_{\text{int}}$, using a set of other, less biased metrics, $\{\hat{q}_{\text{safe}}\}$, as features. MINTADJUST is trained on translations from one or more models that are not subject to MINT, and predicts the $\hat{q}_{\text{int}}$ scores of a model affected by MINT, producing $\hat{q}_{\text{adjusted}}$. The framework is agnostic to the regression method.

MintAdjust is inspired by the analysis performed in Section 4: we found that Mint distorts the relationships of instance-level scores among metrics; thus, we hypothesize that a less biased version of the interfering metric can be learned from the instance-level scores of other metrics on models where Mint is not a concern. It should be noted that Mint distorts all spuriously correlated metrics, not just the interfering one, so the inputs of MintAdjust at inference time will be biased. However, the distortion is less pronounced, and by leveraging a large set of metrics we posit the biases of each one may be diluted. Next, we detail the experimental setup for evaluating MintAdjust.

5.2 Experimental Setup

Overview. Meta-evaluating (i.e., evaluating an evaluation method) MintAdjust requires: 1) translations of one or more systems that did not leverage metrics; 2) translations of one or more systems that used metrics (Mint); 3) knowledge of the interfering metric; 4) human judgments for the translations. The most recent edition of the WMT24 MT shared task (Kocmi et al., 2024a) contains all this information for state-of-the-art MT systems on 11 language pairs.[10] Furthermore, one relevant consequence of Mint in this shared task was Tower-v2-MBR—which used Comet for MBR—ranking first on all language pairs according to automatic metrics but not to humans.

Meta-Evaluation Metrics. The goal of Mint mitigation methods is to rank the model affected by Mint more accurately. Thus, for system pairs that contain Tower-v2, we report **soft pairwise accuracy (SPA)** (Thompson et al., 2024)—the system-level meta-metric used in the latest WMT metrics shared task (Freitag et al., 2024):

$$SPA = \binom{N}{2}^{-1} \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} 1 - \left| p_{ij}^{h} - p_{ij}^{m} \right| \quad (3)$$

N is the number of systems, p_{ij}^{h} the p-value for the hypothesis that humans prefer system i over j, and p_{ij}^{m} is the p-value for the hypothesis that a metric prefers system i over j. SPA judges metrics both on whether they rank systems like humans and on whether their confidence in the rankings is similar.

Since instance-level meta-evaluation is also important to discriminate between metrics (Freitag et al., 2022b), we also report an **instance-level version of pairwise accuracy (Ins-PA)**:

$$\text{Ins-PA} = \frac{\sum_{d \in \mathcal{D}} \mathbb{I}\left[\mathcal{M}(d_+) > \mathcal{M}(d_-)\right]}{|\mathcal{D}|} \quad (4)$$

where $\mathcal{D}$ is the set, across all system pairs, of all unique tuples, d, containing a source, its human-favored translation, its human-disfavored translation, and a reference ($\langle x, h_+, h_-, r \rangle$); d_+ corresponds to $\langle x, h_+, r \rangle$, d_- corresponds to $\langle x, h_-, r \rangle$, and $\mathcal{M}$ is a metric's judgement. If the human score for d_+ and d_- is the same, that instance is scored 1 only if $\mathcal{M}(d_+) = \mathcal{M}(d_-)$.[11]

Further Meta-Evaluation Details. When aggregating the aforementioned metrics across LPs, we report their macro-average, and, for Table 4, a normalized Borda count (Colombo et al., 2022) by averaging the ranks of each metric over all LPs (lower is better). This metric is better at representing consistency across LPs than an average, which is more sensitive to outliers. Furthermore, since metrics are known to perform differently when comparing low- and high-quality systems, we also report results on a subset of high-quality systems: namely, Tower-v2 pairs with GPT-4, Claude-Sonnet-3.5, and Gemini-1.5-Pro, when available.[12]

Metric Baselines. As mentioned before, the usual practice under Mint is to either keep using the interfering metric for evaluation, or to use a single other metric. Thus, our baselines are Comet and representative state-of-the-art MT metrics of several families: chrF (lexical), CometKiwi (reference-less), xComet-XL, and MetricX-XL (neural).

Ensemble Baselines. We report two baselines that involve ensembling metric scores: AutoRank

[10] The language pairs are English-German, -Icelandic, -Czech, -Spanish, -Ukrainian, -Russian, -Japanese, -Chinese, -Hindi, Czech-Ukrainian, and Japanese-Chinese. The amount of evaluated systems per language pair ranges from 11 to 19.

[11] As such, while Ins-PA considers ties, it does not calibrate a threshold for ties, as the meta-metric used in the metrics shared task (Deutsch et al., 2023). If anything, this will penalize MintAdjust and favor xComet or MetricX, which produce more ties for perfect translations. We computed results when disregarding ties and reached similar conclusions.

[12] Gemini-1.5-Pro is not available for Icelandic.

Metrics	System-level Average ↑ All	System-level Average ↑ High-Q	System-level Borda Count ↓ All	System-level Borda Count ↓ High-Q	Segment-level Average ↑ All	Segment-level Average ↑ High-Q	Segment-level Borda Count ↓ All	Segment-level Borda Count ↓ High-Q
Baselines								
Metrics								
chrF$_{L,REF}$	0.5958	0.4556	5.4545	5.4545	0.4548	0.4328	6.3636	5.0000
CometKiwi$_{N,QE}$	0.8219	0.6476	3.0909	2.6364	0.4716	0.4345	5.3636	5.1818
xComet$_{N,REF}$	0.8199	0.6476	3.2727	2.6364	0.4968	0.4510	3.1818	3.3636
MetricX$_{N,REF}$	0.8219	0.6522	2.5455	2.3636	0.4868	0.4385	4.4545	4.4545
Comet$_{N,REF}$*	0.8277	0.6775	3.0000	2.1818	0.4955	0.4458	3.0000	3.4545
Ensembles								
AutoRank$_E$	0.8199	0.6476	3.2727	2.6364	–	–	–	–
MetaMetrics-MT$_E$	0.8080	0.6506	4.1818	2.4545	0.4434	0.4115	7.4545	6.5455
This work								
AutoRank-Ins$_E$	0.8190	0.6757	3.0000	2.0909	**0.4978**	0.4509	**2.3636**	2.9091
MintAdjust$_E$	**0.8278**	**0.6846**	**2.4545**	**1.5455**	0.4956	**0.4551**	3.2727	**2.3636**

Table 4: System-level SPA and instance-level PA for baseline metrics, AutoRank-Ins, and MintAdjust on the WMT24 test set for two sets of system pairs: all pairs that contain Tower-v2-MBR (All), and a subset of high-quality system pairs containing Tower-v2-MBR (High-Q). Comet (with the *) is the interference (and adjusted) metric. **Bold** values are the best overall. Subscripts indicate metric family: L means lexical, N is Neural, QE is reference-less, REF is reference-based, and E is ensemble.

(Kocmi et al., 2024a) and MetaMetrics-MT (Anugraha et al., 2024). The former was used to obtain the preliminary rankings of the WMT24 MT shared task, and the latter is a state-of-the-art metric ensembling method, according to the results of the WMT24 shared task. AutoRank works by, for each LP, linearly scaling (between 1 and the number of systems) a set of system-level metric averages and then computing the average among the metrics to obtain a single value per system. In the shared task, MetricX-XL and CometKiwi-XL were used; for the sake of comparability with our setup, we ensemble the baseline metrics stated in the previous paragraph, except for Comet. MetaMetrics-MT, on the other hand, like most other metric, learns to predict translation scores of humans. However, instead of leveraging vectorial representations of text as learning features, it uses the scores of other metrics, therefore functioning as a learned ensemble. The authors use Gaussian Processes to this end, but MetaMetrics-MT is agnostic to the regressor class. For the sake of comparability, we use the same regression model as MintAdjust (the Random Forest described in the ‘‘Training details’’ paragraph of this section), and the same metrics for ensembling as AutoRank. Crucially, MetaMetrics-MT requires an annotated training set, while MintAdjust does not. We use all the human assessments available for all language pairs of WMT23 for training.[13]

AutoRank-Ins Baseline. As an additional baseline, we propose a instance-level version of AutoRank. Given an ensemble of metrics, for each LP, AutoRank-Ins is obtained by linearly scaling the score of each metric between 1 and the total number of instances (across all systems), and then averaging this over all the metrics in the ensemble. AutoRank-Ins yields instance-level scores that can be averaged to obtain a system-level scores.

[13]This includes 9 language pairs: Czech→Ukrainian, English→{Czech, German, Japanese, Chinese}, and {German, Hebrew, Japanese, Chinese}→English. Training a single regression model on all LPs (the alternative suggested by Anugraha et al. (2024) for whenever a test LP is not seen in training) yielded better results than training LP-specific models.

Metrics	de	es	cs	ru	uk	is	ja	zh	hi	cs→uk	ja→zh
Baselines											
chrF$_{L,REF}$	0.4338	0.5248	0.3069	0.4963	0.4010	**0.8091**	**0.7655**	0.5427	0.5790	**0.9113**	**0.7835**
CometKiwi$_{N,QE}$	0.8912	0.6940	0.9983	0.9161	0.8266	0.7781	0.5697	0.9017	0.9161	0.8033	0.7457
xComet$_{N,REF}$	0.8912	0.6940	0.9983	0.9161	0.8266	0.7781	0.5477	0.9017	0.9161	0.8033	0.7457
MetricX$_{N,REF}$	0.8921	0.6940	0.9983	0.9161	0.8266	0.7781	0.5477	0.9017	0.9232	0.8132	0.7497
Comet$_{N,REF}$*	0.8912	0.6941	0.9983	0.9161	0.8138	0.7781	0.5477	0.9017	0.9162	0.8999	0.7472
AutoRank$_{E}$	0.8912	0.6963	0.9983	0.9161	0.7156	0.7781	0.5489	0.9017	0.9227	0.8936	0.7465
MetaMetrics-MT$_{E}$	0.7514	0.6938	0.9983	0.9161	0.8266	0.7781	0.5477	**0.9085**	0.9159	0.8033	0.7485
This work											
AutoRank-Ins$_{E}$	0.8912	0.6940	0.9983	0.9161	**0.8266**	0.7781	0.5477	0.9017	0.9161	0.8033	0.7457
MintAdjust$_{E}$	**0.8937**	**0.7061**	**0.9983**	**0.9163**	0.7154	0.7781	0.6023	0.8995	**0.9346**	0.9032	0.7582

Table 5: System-level SPA results for all LPs (direction omitted for from-English LPs).

Training Details. We use MintAdjust to predict Comet scores for Tower-v2-MBR on all language pairs of the WMT24 test set. In the main results, for training, we use scores of baseline metrics from the previous paragraph, except for Comet, on Tower-v2 greedy WMT24 translations as features (we explore other training metric features and systems later). For each language pair, we train a regressor to predict Comet scores. This trained regressor is MintAdjust. At inference time, we query MintAdjust for every instance of every system. Our regressor is a random forest with 1000 trees, a maximum depth of 4, and the default hyperparameters of `scikit-learn` (Pedregosa et al., 2011), which we detail in Appendix B.[14]

5.3 Results & Discussion

MintAdjust ranks systems and instances more accurately than metrics and AutoRank. Table 4 shows our main system- and instance-level accuracy results.[15] At a system level, MintAdjust ranks systems more accurately on average and in terms of Borda count (meaning it is more consistent across LPs). The superiority of MintAdjust is more noticeable when ranking high-quality systems; this comparison is particularly important for determining the *de facto* state-of-the-art systems. Conclusions are similar for the instance-level evaluation, where AutoRank-Ins performs slightly better when considering all systems. Additionally, Table 5 shows that, out of 11 LPs, MintAdjust outperforms all baselines on 5 LPs, ranks second on 4, and third on 2, highlighting the method's generalizability across languages.

MintAdjust outperforms MetaMetrics-MT, which requires annotated data. Table 4 also shows that MintAdjust outperforms MetaMetrics-MT across the board. This is an encouraging result, considering that MetaMetrics-MT requires human annotations. Relying on human annotations can be especially problematic if the test set contains language pairs for which there is no pre-existing representative data, as is the case for Icelandic and Hindi, where MetaMetrics-MT performs particularly poorly (see Tables 18 and 17 in the Appendix). Even when data is available, it might not be representative, and building a training corpus still entails a series of design decisions. MintAdjust, on the other hand, does not require human annotations as it is meant to be used directly on the test set of interest. This finding hints at the benefits of learning to adjust a metric rather than ensembling metrics to learn human scores in cases of Mint.

Other metrics do not necessarily outperform the interfering metric. Remarkably, the interfering metric, Comet, performs similarly to alternative metrics like xComet and MetricX on both sets of system pairs. Thus, in this case, using other metrics in isolation for evaluation would not even achieve better results than evaluating with the interfering metric. Moreover, AutoRank, an ensembling baseline, does not seem

[14]We do not optimize hyperparameters, and opt for a relatively high number of estimators but low maximum depth to avoid overfitting.

[15]We present per-LP results in Table 5 and 19.

Method	Table 4	Ensemble Metrics: All other metrics	QE only	REF only	Neural only	Lexical only
AUTORANK						
AUTORANK	0.8199	0.8199	0.8199	0.8340	0.8199	0.5488
AUTORANK-Ins	0.8190	0.8334	0.8199	0.8457	0.8199	0.5605
MINTADJUST						
Training system(s)						
TOWER-v2 Greedy	0.8278	0.8356	0.8199	0.8512	0.8225	0.6074
CLAUDE-SONNET-3.5	0.8263	0.8338	0.8199	0.8615	0.8205	0.6014
All other models	**0.8347**	**0.8356**	**0.8199**	***0.8656***	**0.8236**	**0.6505**

Table 6: System-level soft pairwise accuracy for variations of AUTORANK and MINTADJUST on the WMT24 for all system pairs containing TOWER-v2. On the columns, we vary the metrics used in each method's ensemble; on the rows, we vary the systems considered for training MINTADJUST. The goal is to assess how sensitive the performance of methods are to changes in their ensembling / training procedures. Values in gray serve as baselines, taken from Table 4. Values in green and red represent the best and worst metric configurations, respectively. Values in **bold** are the best training system configuration for MINTADJUST, for each metric configuration. The value in *italic* is the best overall.

to perform particularly well either. Instead, our proposed instance-level variant of AUTORANK, AUTORANK-Ins, performs better, underscoring the importance of preserving instance-level information to maximize system-level performance.

MINTADJUST and AUTORANK can be built from a reduced set of metrics. In Section 4, we reported that metrics react differently to MINT according to their kind. For example, neural metrics usually agreed with neural interfering metrics, while lexical metrics did not. These disagreements are valuable information that should not be discarded; they might help making more accurate predictions as to whether a translation *really* improved in quality. With this in mind, we leveraged a diverse set of metrics in our main results for learning MINTADJUST and for computing AUTORANK. However, it could be the case that certain sets of metrics work better. We assess this in in Table 6, where we report the average system-level SPA of MINTADJUST variants and AUTORANK built with distinct features (excluding COMET): 1) the same as Table 4; 2) every metric in Table 1; 3) all QE metrics; 4) all reference-based metrics; 5) all neural metrics; 6) all lexical metrics. Variants built from lexical metrics perform visibly worse, while the other variants are more similar to the baseline. The signal from neural metrics seems important for learning MINTADJUST on COMET (a neural metric itself), even though they are also potentially biased at inference time.[16] Notably, using only reference-based metrics surpasses the baseline considerably, showing that there is some room for optimization for both AUTORANK and MINTADJUST.

MINTADJUST can be learned from a diverse set of models. We learned MINTADJUST from greedy-sampled translations of TOWER-v2 because it was the most immediate option: we adjust the COMET MBR scores with the scores of the corresponding greedy-decoding system. However, during evaluation we may not have access to translations with and without MINT for the same system; this was the case with TOWER-v2 for the organizers of the shared task. In the worst case, an unbiased equivalent to a biased system may not exist or be prohibitively hard to obtain (e.g., to a model pre-trained on filtered translation data). Thus, we assessed whether MINTADJUST could be learned from the translations of other systems in Table 6, which contains three versions of MINTADJUST, each trained on translations from distinct systems: 1) TOWER-v2 greedy; 2) CLAUDE-SONNET-3.5; 3) all systems available to the shared task organizers.

[16] Lexical metrics may also be biased *against* the interfering metric. Quantifying what part of the difference among metrics is *real* and what is bias is interesting for future work.

Results are similar across variants, with the variant learned from all models available performing the best. This finding is encouraging for three reasons: 1) generalizability: MINTADJUST can be strong even when learned from a single other system; 2) practicality: the non-MINT equivalent of the MINT system is not required to build MINTADJUST; 3) scalability: better results can be achieved by obtaining more translations from other systems.

6 Related Work

6.1 Metric Interference

The role of automatic metrics has expanded beyond evaluation to applications in model training (e.g., as learning objectives in minimum error rate training) (Och, 2003; Zaidan and Callison-Burch, 2009), in reward modeling for human-preference alignment (Shu et al., 2021; Xu et al., 2024b; Zhu et al., 2024), in training data filtering (Alves et al., 2024; Rei et al., 2024), and in quality-aware decoding (Fernandes et al., 2022; Wu et al., 2024), where a metric is used to choose the best candidate out of a pool of model generations. The aforementioned applications give rise to **Metric Interference (MINT)**—formalized in Section 2—where such metrics are used for both optimization and evaluation. MINT introduces additional evaluation challenges due to inherent metric flaws and biases. For example, LLM judges tend to favor outputs from their own model family (Panickssery et al., 2024; Verga et al., 2024); in machine translation, this is also the case according to Agrawal et al. (2024). It has been long known that traditional metrics suffer from similar issues. Callison-Burch et al. (2006) show how optimizing a translation system directly for BLEU need not lead to improvements in translation quality according to humans. This is because of inherent shortcomings of the metric, for example its insensitivity to synonyms. Similarly, even models optimized for stronger metrics like BLEURT can produce absurd translations that score highly (Yan et al., 2023). Furthermore, Fernandes et al. (2022) show that MBR and translation re-ranking with neural metrics lead to overestimated scores at a system level, reducing their correlation with human judgments and lexical metrics. Gisserot-Boukhlef et al. (2024) report similar findings with Contrastive Preference Optimization (Xu et al., 2024b), a method whereby models are trained to produce translations preferred by automatic metrics. Some of these works recommend using a different metric for evaluation than that used for optimization. We find such strategies are insufficient to mitigate MINT, and propose an alternative.

6.2 Interference Mitigation Strategies

For general-purpose evaluation of LLMs, Verga et al. (2024) propose using an ensemble of LLM judges to reduce intra-model bias (i.e., models preferring their own outputs). For MT, Kocmi et al. (2024b) study how metric improvements relate to the probability of humans preferring the improved translation. While they do not study the effect of MINT, they warn against its effects, and recommend not using interfering metrics for evaluation. Other existing works that deal with MINT more directly attempt to improve the underlying optimization rather than the evaluation process. Kamigaito et al. (2024) improve the reliability of MBR by averaging the scores yielded by the same metric trained from different initializations. Kovacs et al. (2024) show that MBR metrics prefer their own systems or those of correlated metrics. To address this, they perform MBR with an ensemble of metrics—as opposed to a single one—showing better correlation with human judgements. Instead of trying to improve MBR, the goal of our proposed method is to recover more accurate metric scores to make evaluation more reliable under cases of metric interference.

6.3 Broader Context on Metric Interference

Goodhart's law states that ‘‘when a measure becomes a target, it ceases to be a good measure’’. The notion that relying too much on a limited set of evaluation methods can lead to the gaming thereof has been acknowledged in machine learning (ML) literature for long (Hutchinson et al., 2022). And this over-reliance need not imply optimizing explicitly for automatic metrics, as in the use-cases our work focuses on. Simply using the same benchmarks and metrics repeatedly over time can deteriorate their value as proxies of human preference (Hutchinson et al., 2022). Indeed, through comprehensive surveys that span various ML subfields (e.g., computer vision, NLP, recommender systems, RL, graph processing) Zhang et al. (2019) and (Liao, 2021) find a consistent trend of overfitting to a small amount of benchmarks, and discuss how this reduces the capability of automatic evaluation to measure scientific progress. Furthermore, besides the aforementioned works on NLP and MT, similar patterns of metric gaming through overuse or explicit optimization have been found in applications of ranking (Yilmaz and Robertson, 2010) and computer vision (Carter et al., 2020; Cardelino et al., 2013; Jiang et al., 2019; Maier-Hein et al., 2022). In most cases, the solution that is either explicitly proposed or hinted at is the same: more diverse and representative benchmarking methodologies—be they test sets or metrics—should be developed to ensure the continued relevance and credibility of automatic evaluation.

7 Discussion on the Implications of Metric Interference

MINT poses a threat to the integrity of evaluation frameworks; shared tasks and leaderboards where participants may tune models to optimize correlated metrics are prominent examples. As discussed in Section 6, a common solution across various subfields of ML for this type of issue is to avoid relying on a small amount of evaluation methods. As an ensemble method that can leverage an arbitrary combination of MT metrics, MINTADJUST is aligned with this idea.

However, this may be a partial solution; other interventions could involve changing the design of evaluation protocols themselves, particularly in how optimization and evaluation metrics are selected and separated. For example, shared task organizers could partition metrics into disjoint groups: some available during model development, others held out for evaluation. Ideally, metrics chosen for optimization should be orthogonal to those used for evaluation, minimizing the risk of indirect overfitting. However, in practice, this separation is rarely clean, as many metrics are spuriously correlated: as shown in our analysis, optimization of one leads to measurable gains on others but not necessarily on human performance. This suggests that simply banning the use of evaluation metrics during training is insufficient unless their relationships can be robustly characterized and accounted for. Another option could be to foster the design of automatic metrics that are ‘‘aware’’ they are being gamed. An interesting first step in this direction could be to study how uncertainty relates to metric gaming, perhaps through analyzing how uncertainty-aware evaluation metrics (Glushkova et al., 2021) behave under metric interference.

These are open challenges, and further research requires the existence of more datasets where metric interference and human judgements are present. This is the primary resource required to assess the effects of MINT on correlations of automatic metrics with human judgements. In addressing metric interference explicitly, we hope this work encourages a broader rethinking of evaluation design towards setups that are robust to strategic optimization.

8 Conclusion

We analyzed two common cases of MINT (defined in Section 2) in MT—training data filtering and MBR—highlighting its negative effects on system- and instance-level evaluation with a myriad of metrics. We find that interfering metric instance-level scores are distorted under MINT, with large improvements becoming more often associated to deterioration in the scores of other metrics and humans. Importantly, such distortions hold for metrics that correlate with the interfering metric, bringing into question the common practice of using a single different metric for evaluation.

Based on these insights, we develop MINTADJUST, a method for correcting the interfering metric during evaluation. MINTADJUST outperforms the interfering metric, other state-of-the art metrics, and the ensembling method used in the WMT24 MT shared task at ranking systems and translations.

In the future, it would be interesting to explore more cases of MINT (e.g., preference optimization) on more metrics and tasks. That said, despite the importance of MINT, we note a lack of human evaluation data necessary to perform such studies and highlight the need to create such resources.

Acknowledgments

We thank Sweta Agrawal, Amin Farajian, and Patrick Fernandes for their constructive feedback on the paper. We acknowledge EuroHPC JU for awarding the project ID EHPC-AI-2024A01-085 access to MareNostrum 5 ACC. This work was supported by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by the project DECOLLAGE

(ERC-2022-CoG 101088763), by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (Center for Responsible AI), and by Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020.

References

Sweta Agrawal, António Farinhas, Ricardo Rei, and Andre Martins. 2024. Can automatic metrics assess high-quality translations? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14491–14502, Miami, Florida, USA. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.802

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

David Anugraha, Garry Kuwanto, Lucky Susanto, Derry Tanti Wijaya, and Genta Winata. 2024. MetaMetrics-MT: Tuning meta-metrics for machine translation via human preference calibration. In *Proceedings of the Ninth Conference on Machine Translation*, pages 459–469, Miami, Florida, USA. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.wmt-1.32

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of BLEU in machine translation research. In *Conference of the European Chapter of the Association for Computational Linguistics*.

Juan Cardelino, Vicent Caselles, Marcelo Bertalmío, and Gregory Randall. 2013. A contrario selection of optimal partitions for image segmentation. *SIAM Journal on Imaging Sciences*, 6:1274–1317. https://doi.org/10.1137/11086029X

Brandon Carter, Siddhartha Jain, Jonas W. Mueller, and David Gifford. 2020. Overinterpretation reveals image classification model pathologies. In *Neural Information Processing Systems*.

Pierre Colombo, Nathan Noiry, Ekhine Irurozki, and Stéphan Clémençon. 2022. What are the best systems? New perspectives on NLP benchmarking. *Advances in Neural Information Processing Systems*, 35:26915–26932.

Daniel Deutsch, George Foster, and Markus Freitag. 2023. Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration. In *The 2023 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2023.emnlp-main.798

António Farinhas, José de Souza, and Andre Martins. 2023. An empirical study of translation hypothesis ensembling with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11956–11970, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.733

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and André 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. https://doi.org/10.18653/v1/2022.naacl-main.100

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon sampling rocks: Investigating sampling strategies for minimum Bayes risk decoding for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9198–9209, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.617

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022a. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825. https://doi.org/10.1162/tacl_a_00491

Markus Freitag, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo

Rei, Brian Thompson, Frédéric Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? Results of the wmt24 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 47–81. https://doi.org/10.18653/v1/2024.wmt-1.2

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022b. 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.

Gemma Team. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hippolyte Gisserot-Boukhlef, Ricardo Rei, Emmanuel Malherbe, Céline Hudelot, Pierre Colombo, and Nuno M. Guerreiro. 2024. Is preference alignment always the best option to enhance LLM-based translation? An empirical analysis. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1373–1392. https://doi.org/10.18653/v1/2024.wmt-1.127

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-aware machine translation evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3920–3938. https://doi.org/10.18653/v1/2021.findings-emnlp.330

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *arXiv preprint arXiv:2310.10482*. https://doi.org/10.1162/tacl_a_00683

Ben Hutchinson, Negar Rostamzadeh, Christina Greer, Katherine Heller, and Vinodkumar Prabhakaran. 2022. Evaluation gaps in machine learning practice. *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. https://doi.org/10.1145/3531146.3533233

Ming Jiang, Qiuyuan Huang, Lei Zhang, Xin Wang, Pengchuan Zhang, Zhe Gan, Jana Diesner, and Jianfeng Gao. 2019. Tiger: Text-to-image grounding for image caption evaluation. In *Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/D19-1220

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. https://doi.org/10.18653/v1/2023.wmt-1.63

Hidetaka Kamigaito, Hiroyuki Deguchi, Yusuke Sakai, Katsuhiko Hayashi, and Taro Watanabe. 2024. Theoretical aspects of bias and diversity in minimum bayes risk decoding. *arXiv preprint arXiv:2410.15021*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondrej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinþór Steingrímsson, and Vilém Zouhar. 2024a. Findings of the wmt24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation, USA. Association for Computational Linguistics*. https://doi.org/10.18653/v1/2024.wmt-1.1

Tom Kocmi, Vilém Zouhar, Christian Federmann, and Matt Post. 2024b. Navigating the metrics maze: Reconciling score magnitudes and accuracies. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1999–2014, Bangkok, Thailand. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.110

Geza Kovacs, Daniel Deutsch, and Markus Freitag. 2024. Mitigating metric bias in minimum Bayes risk decoding. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1063–1094, Miami, Florida, USA. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.wmt-1.109

Thomas Liao. 2021. Are we learning yet? A meta review of evaluation failures across machine learning. In *NeurIPS Datasets and Benchmarks*.

Lena Maier-Hein, Annika Reinke, Evangelia Christodoulou, Ben Glocker, Patrick Godau, Fabian Isensee, Jens Kleesiek, Michal Kozubek, Mauricio Reyes, M. Riegler, Manuel Wiesenfarth, Michael Baumgartner, Matthias Eisenmann, Doreen Heckmann-Notzel, Ali Emre Kavur, Tim Radsch, Minu Dietlinde Tizabi, Laura Ación, Michela Antonelli, Tal Arbel, Spyridon Bakas, Peter Bankhead, Allison Benis, M. Jorge Cardoso, V. Cheplygina, Beth A. Cimini, Gary Stephen Collins, Keyvan Farahani, Bram van Ginneken, Daniel A. Hashimoto, Michael M. Hoffman, Merel Huisman, Pierre Jannin, Charles E. Kahn, Alexandros Karargyris, Alan Karthikesalingam, Hannes Kenngott, Annette Kopp-Schneider, Anna Kreshuk, Tahsin M. Kurç, Bennett A. Landman, Geert J. S. Litjens, Amin Madani, Klaus H. Maier-Hein, Anne L. Martel, Peter Mattson, Erik H. W. Meijering, Bjoern H. Menze, David Moher, Karel G. M. Moons, Henning Muller, Felix Nickel, Brennan Nichyporuk, Jens Petersen, Nasir M. Rajpoot, Nicola Rieke, J. Saez-Rodriguez, Clarisa S'anchez Guti'errez, Shravya Shetty, Maarten van Smeden, Carole Helene Sudre, Ronald M. Summers, Abdel Aziz Taha, Sotirios A. Tsaftaris, Ben Van Calster, Gaël Varoquaux, and Paul F. Jager. 2022. Metrics reloaded: Recommendations for image analysis validation. *Nature Methods*.

Artur Nowakowski, Gabriela Pałka, Kamil Guttmann, and Mikołaj Pokrywka. 2022. Adam Mickiewicz University at WMT 2022: NER-assisted and quality-aware neural machine translation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 326–334, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics*, pages 160–167. https://doi.org/10.3115/1075096.1075117

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. LLM evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*.

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. https://doi.org/10.3115/1073083.1073135

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

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. https://doi.org/10.18653/v1/2023.wmt-1.50

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. https://doi.org/10.18653/v1/W15-3049

Ricardo Rei, José Pombal, Nuno M. Guerreiro, João Alves, Pedro Henrique Martins, Patrick

Fernandes, Helena Wu, Tania Vaz, Duarte Alves, Amin Farajian, Sweta Agrawal, Antonio Farinhas, José G. C. De Souza, and André Martins. 2024. Tower v2: Unbabel-ist 2024 submission for the general MT shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 185–204. https://doi.org/10.18653/v1/2024.wmt-1.12

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, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C. Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022b. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892. https://doi.org/10.18653/v1/2020.acl-main.704

Raphael Shu, Kang Min Yoo, and Jung-Woo Ha. 2021. Reward optimization for neural machine translation with learned metrics. *ArXiv*, abs/2104.07541.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine Translation*, 24:39–50. https://doi.org/10.1007/s10590-010-9077-2

Brian Thompson, Nitika Mathur, Daniel Deutsch, and Huda Khayrallah. 2024. Improving statistical significance in human evaluation of automatic metrics via soft pairwise accuracy. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1222–1234. https://doi.org/10.18653/v1/2024.wmt-1.118

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *LREC*, volume 2012, pages 2214–2218. Citeseer.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating LLM generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Ian Wu, Patrick Fernandes, Amanda Bertsch, Seungone Kim, Sina Pakazad, and Graham Neubig. 2024. Better instruction-following through minimum bayes risk. *arXiv preprint arXiv:2410.02902*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024a. A paradigm shift in machine translation: Boosting translation performance of large language models. In *The Twelfth International Conference on Learning Representations*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024b. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *Forty-first International Conference on Machine Learning*.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Wang, and Lei Li. 2023. Instructscore: Towards explainable text generation evaluation with automatic feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5967–5994. https://doi.org/10.18653/v1/2023.emnlp-main.365

Yiming Yan, Tao Wang, Chengqi Zhao, Shujian Huang, Jiajun Chen, and Mingxuan Wang. 2023. Bleurt has universal translations: An analysis of automatic metrics by minimum risk training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5428–5443. https://doi.org/10.18653/v1/2023.acl-long.297

Emine Yilmaz and Stephen E. Robertson. 2010. On the choice of effectiveness measures for learning to rank. *Information Retrieval*, 13:271–290. https://doi.org/10.1007/s10791-009-9116-x

Omar Zaidan and Chris Callison-Burch. 2009. Feasibility of human-in-the-loop minimum error rate training. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 52–61. https://doi.org/10.3115/1699510.1699518

J. Zhang, Mark Harman, Lei Ma, and Yang Liu. 2019. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 48:1–36. https://doi.org/10.1007/s10994-018-05777-9

Dawei Zhu, Sony Trenous, Xiaoyu Shen, Dietrich Klakow, Bill Byrne, and Eva Hasler. 2024. A preference-driven paradigm for enhanced translation with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3385–3403. https://doi.org/10.18653/v1/2024.naacl-long.186

A Metric Details

We list below the metric implementations and checkpoints used in this paper:

- **chrF**: `sacrebleu==2.4.2`, default settings.
- **Bleu**: `sacrebleu==2.4.2`, default settings.
- **Comet**: Huggingface checkpoint: `Unbabel/wmt22-comet-da`.
- **CometKiwi**: Huggingface checkpoint: `Unbabel/wmt22-cometkiwi-da`.
- **xComet**: Huggingface checkpoint: `Unbabel/XCOMET-XL`
- **MetricX**: Huggingface checkpoint: `google/metricx-23-xl-v2p0`.
- **Bleurt**: Huggingface checkpoint: `lucadiliello/BLEURT-20`.
- **Prometheus 2**: Huggingface checkpoint: `prometheus-eval/prometheus-7b-v2.0`
- **InstructScore**: Huggingface checkpoint: `xu1998hz/InstructScore`.

B Random Forest Hyperparameters

We use the default hyperparameters of the scikit-learn implementation, except for the number of estimators and max depth. Namely: n_estimators=100, *, criterion=‘‘squared_error’’, max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=1.0, max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, ccp_alpha=0.0, max_samples=None, monotonic_cst=None

C Supplementary Results

Filtering	Random	CKiwi	xC	**MX**
Random	–	–	–	–
CKiwi	0.0042	–	–	–
xC	0.0040	0.0293	–	–
MX	0.0042	0.0307	0.0002	–

Table 7: Jaccard Index ($\frac{A \cap B}{A \cup B}$, where A and B are filtered training sets) among training instances selected by different metrics. 0 means no overlap.

Filtering	Lexical chrF↑	Bleu↑	Comet↑	Bleurt↑	CometKiwi↑	Neural xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Random	50.75	21.03	69.34	53.61	65.76	62.34	61.23	5.32	6.65
CometKiwi	53.25	24.17	73.61	57.83	70.63	70.47	69.72	4.11	4.90
xComet	39.61	16.17	65.46	53.97	59.18	64.91	63.00	5.29	10.06
MetricX	48.39	19.32	72.17	57.51	68.15	69.85	68.98	3.99	5.22

Table 8: Data filtering results table like Table 1 for English→German.

Filtering	Lexical chrF↑	Bleu↑	Comet↑	Bleurt↑	CometKiwi↑	Neural xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Random	55.69	28.86	79.17	64.71	73.66	77.39	80.22	4.60	5.94
CometKiwi	57.13	32.78	80.10	65.46	75.08	80.90	83.24	4.49	5.77
xComet	52.37	28.26	76.93	62.53	72.44	78.08	79.51	5.05	7.71
MetricX	55.50	31.36	77.85	63.55	73.19	77.25	78.63	5.56	7.34

Table 9: Data filtering results table like Table 1 for German→English.

Filtering	Lexical chrF↑	Bleu↑	Comet↑	Bleurt↑	CometKiwi↑	Neural xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Random	38.67	13.09	73.85	58.70	72.34	72.73	78.01	4.19	4.15
CometKiwi	40.12	14.69	76.30	61.11	75.27	78.47	84.02	3.23	2.87
xComet	37.36	13.27	74.05	58.96	72.39	76.97	82.23	3.58	3.43
MetricX	38.90	14.58	74.77	58.97	72.96	76.65	81.37	3.74	3.41

Table 10: Data filtering results table like Table 1 for Chinese→English.

Filtering	Lexical chrF↑	Bleu↑	Comet↑	Bleurt↑	CometKiwi↑	Neural xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Greedy	68.19	42.61	84.90	71.87	80.74	86.45	85.76	1.71	2.01
MBR									
chrF$_L$	68.22	41.57	84.92	71.85	80.78	86.15	85.36	1.67	1.97
Bleu$_L$	67.53	42.07	84.68	71.45	80.58	86.23	85.37	1.76	2.07
Comet$_N$	66.94	39.99	86.21	72.86	81.45	87.52	86.92	1.49	1.71
Bleurt$_N$	66.44	39.29	85.20	74.26	81.24	87.09	86.57	1.44	1.65

Table 11: MBR results table like Table 2 English→German.

Filtering	Lexical chrF↑	Bleu↑	Comet↑	Bleurt↑	CometKiwi↑	Neural xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Greedy	69.65	47.20	86.02	75.26	80.03	89.94	91.07	2.34	2.37
MBR									
chrF$_L$	69.67	46.32	86.00	75.13	80.07	89.90	91.07	2.33	2.32
Bleu$_L$	69.63	47.54	86.07	75.26	80.04	90.02	91.06	2.35	2.36
Comet$_N$	68.50	45.08	86.76	75.51	80.24	90.47	91.40	2.17	2.19
Bleurt$_N$	69.05	45.63	86.34	76.18	80.20	90.32	91.30	2.17	2.21

Table 12: MBR results table like Table 2 German→English.

Filtering	Lexical chrF↑	BLEU↑	COMET↑	BLEURT↑	CometKiwi↑	Neural xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Greedy	49.80	23.84	81.58	69.15	80.79	87.51	90.18	2.17	2.00
MBR									
chrF$_L$	$\boxed{50.05}$	23.25	81.63	69.17	80.85	87.17	89.98	2.18	2.03
BLEU$_L$	49.29	$\boxed{23.61}$	81.58	68.98	80.71	87.35	90.13	2.20	2.03
COMET$_N$	48.87	22.71	$\boxed{82.32}$	69.30	81.15	87.82	90.71	2.06	1.87
BLEURT$_N$	48.89	22.21	81.71	$\boxed{70.12}$	81.07	87.97	90.59	2.03	1.85

Table 13: MBR results table like Table 2 Chinese→English.

Filtering	Lexical chrF↑	BLEU↑	COMET↑	BLEURT↑	CometKiwi↑	Neural xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Greedy	43.23	44.46	87.22	72.89	81.23	87.27	84.31	1.40	1.34
MBR									
chrF$_L$	$\boxed{44.23}$	44.91	87.42	73.09	81.33	87.22	84.29	1.39	1.33
BLEU$_L$	44.02	$\boxed{45.49}$	87.43	73.03	81.18	87.18	84.23	1.40	1.36
COMET$_N$	40.35	40.93	$\boxed{88.72}$	72.88	82.19	88.36	85.96	1.26	1.15
BLEURT$_N$	39.25	39.62	87.28	$\boxed{74.93}$	81.80	88.10	85.64	1.25	1.15

Table 14: MBR results table like Table 2 for 50 candidates.

Filtering	Lexical chrF↑	BLEU↑	COMET↑	BLEURT↑	CometKiwi↑	Neural xC-REF↑	xC-QE↑	MX-REF↓	MX-QE↓
Greedy	43.66	45.03	87.32	72.90	81.22	87.73	84.59	1.36	1.33
MBR									
chrF$_L$	$\boxed{43.83}$	44.40	87.33	72.87	81.23	87.06	84.18	1.37	1.32
BLEU$_L$	43.17	$\boxed{44.88}$	87.21	72.65	81.04	87.04	84.13	1.40	1.37
COMET$_N$	40.05	40.71	$\boxed{88.53}$	72.91	82.05	88.38	86.05	1.26	1.17
BLEURT$_N$	39.06	39.63	87.21	$\boxed{74.59}$	81.66	88.09	85.77	1.24	1.18

Table 15: MBR results table like Table 2 for Tower-v2-70B.

Metric	C22	CK22	xCR	xCQ	MxR	MxQ	BT	chrF	BLEU
COMET (C22)		0.560	0.623	0.409	0.692	0.470	0.752	0.673	0.665
CometKiwi (CK22)	0.560		0.539	0.496	0.520	0.616	0.471	0.350	0.338
xCOMET-REF (xCR)	0.623	0.539		0.651	0.605	0.496	0.586	0.445	0.435
xCOMET-QE (xCQ)	0.409	0.496	0.651		0.411	0.468	0.359	0.242	0.228
MetricX-REF (MxR)	0.692	0.520	0.605	0.411		0.548	0.678	0.513	0.497
MetricX-QE (MxQ)	0.470	0.616	0.496	0.468	0.548		0.418	0.285	0.268
BLEURT (BT)	0.752	0.471	0.586	0.359	0.678	0.418		0.663	0.651
chrF	0.673	0.350	0.445	0.242	0.513	0.285	0.663		0.937
BLEU	0.665	0.338	0.435	0.228	0.497	0.268	0.651	0.937	

Table 16: Spearman correlation between evaluation metrics on the WMT23 English→Chinese test set. Values are computed by averaging the correlations among scores for every unique source.

Metrics	de	es	cs	ru	uk	is	ja	zh	hi	cs→uk	ja→zh
Baselines											
chrF$_{L,REF}$	0.3443	0.4163	0.0193	0.0357	0.2720	**0.9985**	**0.6993**	0.2883	0.4050	**0.7440**	**0.7883**
CometKiwi$_{N,QE}$	0.7543	0.5837	0.9960	0.9847	0.7317	0.5015	0.3013	0.7903	0.9243	0.3443	0.2117
xComet$_{N,REF}$	0.7543	0.5837	0.9960	0.9847	0.7317	0.5015	0.3013	0.7903	0.9243	0.3443	0.2117
MetricX$_{N,REF}$	0.7543	0.5837	0.9960	0.9847	0.7317	0.5015	0.3013	0.7903	0.9420	0.3773	0.2117
Comet$_{N,REF}$*	0.7543	0.5837	0.9960	0.9847	0.7317	0.5015	0.3013	0.7903	0.9243	0.6663	0.2183
AutoRank$_E$	0.7543	0.5837	0.9960	0.9847	0.7317	0.5015	0.3013	0.7903	0.9243	0.3443	0.2117
MetaMetrics-MT$_E$	0.7543	0.5837	0.9960	0.9847	0.7317	0.5015	0.3013	**0.8100**	0.9247	0.3443	0.2247
This work											
AutoRank-Ins$_E$	0.7543	0.5840	0.9960	0.9847	0.7317	0.5015	0.3047	0.7903	**0.9250**	0.6453	0.2153
MintAdjust$_E$	**0.7577**	**0.5847**	**0.9960**	**0.9847**	**0.7317**	0.5015	0.3160	0.7903	0.9243	0.6773	0.2660

Table 17: System-level SPA results (high-quality systems) for all LPs (direction omitted for from-English LPs).

Metrics	de	es	cs	ru	uk	is	ja	zh	hi	cs→uk	ja→zh
Baselines											
chrF$_{L,REF}$	0.3653	0.4704	0.4845	0.4466	0.3543	0.6071	0.5117	0.4860	0.4973	0.4286	0.3505
CometKiwi$_{N,QE}$	0.3771	0.4642	0.5338	0.4748	0.4016	0.5752	0.4951	**0.5634**	0.5254	0.4319	0.3449
xComet$_{N,REF}$	**0.4386**	0.4839	0.5667	0.5053	**0.4282**	0.6479	0.5133	0.5455	0.5250	0.4374	0.3731
MetricX$_{N,REF}$	0.4322	0.4931	0.5340	0.5010	0.4241	0.6394	0.5041	0.5339	0.4984	0.4172	0.3774
Comet$_{N,REF}$*	0.4016	0.4916	**0.5778**	**0.5129**	0.4027	0.6668	0.5130	0.5365	0.5194	0.4501	**0.3780**
MetaMetrics-MT$_E$	0.3504	0.4484	0.5122	0.4463	0.3954	0.5908	0.4209	0.4660	0.4512	0.4252	0.3709
This work											
AutoRank-Ins$_E$	0.3983	0.5030	0.5697	0.5059	0.4009	**0.6742**	**0.5195**	0.5481	**0.5294**	**0.4523**	0.3745
MintAdjust$_E$	0.4172	**0.5038**	0.5683	0.5039	0.4049	0.6565	0.5095	0.5400	0.5223	0.4514	0.3740

Table 18: Instance-level PA results for all LPs (direction omitted for from-English LPs).

Metrics	de	es	cs	ru	uk	is	ja	zh	hi	cs→uk	ja→zh
Baselines											
chrF$_{L,REF}$	0.3182	0.4666	0.4608	0.4450	0.3454	0.5561	**0.5219**	0.4824	0.4781	0.4058	0.2806
CometKiwi$_{N,QE}$	0.3128	0.4255	0.4813	0.4408	0.3691	0.5577	0.4781	**0.5608**	0.4950	0.4082	0.2506
xComet$_{N,REF}$	**0.3759**	0.4387	0.5064	0.4797	0.3913	0.5774	0.4924	0.5155	0.4971	0.4208	0.2656
MetricX$_{N,REF}$	0.3560	0.4602	0.4737	0.4860	**0.4034**	0.5600	0.4771	0.4924	0.4655	0.3975	0.2518
Comet$_{N,REF}$*	0.3347	0.4518	**0.5193**	**0.4866**	0.3802	0.5798	0.4760	0.4934	0.4860	0.4249	0.2710
MetaMetrics-MT$_E$	0.3299	0.4234	0.4760	0.4313	0.3749	0.5047	0.4055	0.4471	0.4271	0.4075	**0.2992**
This work											
AutoRank-Ins$_E$	0.3237	0.4608	0.5105	0.4824	0.3760	**0.5916**	0.5034	0.5182	0.4961	**0.4268**	0.2710
MintAdjust$_E$	0.3491	**0.4734**	0.5164	0.4750	0.3865	0.5695	**0.5008**	0.5271	**0.5008**	0.4265	0.2812

Table 19: Instance-level PA results (high-quality systems) for all LPs (direction omitted for from-English LPs).