A Measure of the System Dependence of Automated Metrics

Pius von Däniken and **Jan Milan Deriu** and **Mark Cieliebak**
Centre for Artificial Intelligence
ZHAW School of Engineering
{vode,deri,ciel}@zhaw.ch

Abstract

Automated metrics for Machine Translation have made significant progress, with the goal of replacing expensive and time-consuming human evaluations. These metrics are typically assessed by their correlation with human judgments, which captures the monotonic relationship between human and metric scores. However, we argue that it is equally important to ensure that metrics treat all systems fairly and consistently. In this paper, we introduce a method to evaluate this aspect.

1 Introduction

Recent years have seen significant advances in machine translation (MT), marked notably by the introduction of the transformer architecture (Vaswani et al., 2017). Current large-scale commercial systems such as GPT (Brown et al., 2020) continue this trend and show promising results (Kocmi et al., 2023; Hendy et al., 2023; Wu and Hu, 2023). A critical supplement to these advancements is thorough and reliable evaluation procedures, which are essential not only for measuring overall progress but also for effectively comparing different systems. While evaluation based on human raters is still considered the gold standard, it is expensive and time-intensive. Therefore, considerable efforts have been made to develop automated metrics for assessing translation quality. Notably, the WMT Metrics series of shared tasks are dedicated to this purpose (Freitag et al., 2023, 2022, 2021, i.a.). Automated metrics usually assign a scalar [1] quality rating to a candidate translation based on the source segment and a reference translation. A system-level rating is derived by averaging the segment ratings over a test set.

To measure a metric's usefulness, we usually measure two aspects: its correlation to human judgments on the segment-level (which checks if there is a monotonic function between metric ratings and human ratings) and whether the system-level ratings can reproduce the same ranking as human ratings (Kocmi et al., 2021; von Däniken et al., 2024). In this paper, we argue that this evaluation of metrics is insufficient, as it ignores a central requirement, namely, that it should treat all systems under evaluation equally. As stated more colloquially, a measuring stick should not change length depending on the measured object. However, this is exactly what we observe in current metrics.

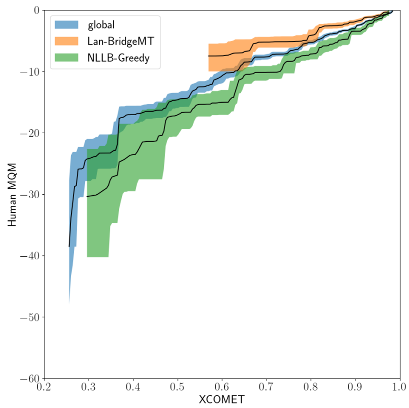

Figure 1: Average Human Ratings associated with *XCOMET* scores on Chinese to English (*zh-en*) WMT 23 data. We show scores for all system in aggregate (global) and two individual systems.

Consider Figure 1, which shows the expected human rating for each score of the *XCOMET* metric (the best metric in the WMT23 metrics task, with a very high segment-level correlation of 0.65 for the zh-en language pair) (Freitag et al., 2023). That is, for each possible value that *XCOMET* may assume, we show the expected human rating and the 95% confidence interval (computed using Isotonic Regression and bootstrap sampling; see Sections 2 and 3 for the details). The global curve (blue) shows the average human score for each metric score if computed over all systems under evalua-

[1]Other types of ratings exist, in particular preference ratings (Belz and Kow, 2010).

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 87–99

tion in the WMT23 dataset (in standard correlation to human judgment evaluation, we only measure whether this curve is monotonic). In contrast, *Lan-BridgeMT* (best system according to humans) and *NLB-Greedy* (lowest-rated system according to humans) show the average human score for each metric score when computed on one separate system only. For instance, an *XCOMET* score of 0.7 corresponds to an average Human-MQM score of -5.2 for *Lan-BridgeMT*, and Human-MQM score of -10.2 for *NLB-Greedy*.

This leads to the following consequence: For *Lan-BridgeMT*, higher human scores are associated with lower metric scores than the global curve, which leads to an underestimation of *Lan-BridgeMT*'s performance, according to *XCOMET*. The opposite effect is visible for *NLB-Greedy*, which is overestimated and, in fact, gains 3 ranks (from 15th to 12th place) when comparing the metric and human ranking (see also Table 1 in Section 3). Thus, a metric that exhibits a high global segment-level correlation to human judgments can lead to wrong system-level rankings. This observation leads us to the central claim of this paper: **The cause of the discrepancy between the correlation on the segment level and the final system ranking is due to the metric's dependency of the system under evaluation.**

The main position of this paper is that when evaluating a novel metric, one ought to measure the dependency on the system under evaluation as well, alongside the correlation to human judgment. In the following, we will formalize this dependency of the relation between human and metric ratings on the system under evaluation and derive a measure for quantifying this effect.

2 Averaging Metric Scores

Assume we are given a set of K machine translation systems π_k to evaluate. A translation system maps an input sentence $i \in \mathcal{I}$ in a fixed source language to an output sentence $o \in \mathcal{O}$ in a fixed target language: $\pi_k : \mathcal{I} \to \mathcal{O}$. The usual human evaluation scenario involves curating a test set of N inputs $\mathcal{T} = \left\{i^{(j)} | 1 \leq j \leq N\right\} \subset \mathcal{I}$ for which we collect the output of each system π_k for each input $i \in \mathcal{T}$, and then ask human annotators to produce ratings. This results in a set of ratings $\left\{(h_1^{(j)}, \ldots, h_k^{(j)}, \ldots, h_K^{(j)}) | 1 \leq j \leq N\right\}$, where $h_k^{(j)} \in \mathbb{R}$ is a scalar rating provided by human annotators measuring the quality of the translation provided by π_k for input $i^{(j)}$. We will assume that higher human ratings indicate higher translation quality. In this setting, it is natural to measure the overall quality of system π_k by the average human rating it achieves $\hat{\mu}_k^H = \frac{1}{N} \sum_{j=1}^N h_k^{(j)}$. This is an estimator of the expected human rating $\mu_k^H = \mathbb{E}[h_k]$ achieved by π_k for any input in $\mathcal{I}$, assuming that $\mathcal{T}$ is appropriately chosen.

In many cases, we want to replace human raters with an automated scalar metric $M : \mathcal{I} \times \mathcal{O} \to \mathbb{R}$, which maps an input and translation to a scalar value. For our test set $\mathcal{T}$, we can collect all metric ratings $\left\{(m_1^{(j)}, \ldots, m_k^{(j)}, \ldots, m_K^{(j)}) | 1 \leq j \leq N\right\}$, where $m_k^{(j)} = M(i^{(j)}, \pi_k(i^{(j)}))$, the metric rating for input $i^{(j)}$ and translation by π_k. In this case, it is common to use the sample average $\hat{\mu}_k^M = \frac{1}{N} \sum_{j=1}^N m_k^{(j)}$ to measure the quality of system π_k, which is an estimator of the expected metric rating $\mu_k^M = \mathbb{E}[m_k]$ achieved by π_k.

The goal of automated evaluation is to use $\hat{\mu}_k^M$ as a proxy measure for μ_k^H, in particular, to rank the systems $\pi_1, \ldots, \pi_K$ according to their performance. In the following, we will study the relationship between μ_k^H and μ_k^M, which is expressed by an unknown function f_G that maps from the metric scale to the human scale. There are two requirements to this function: first, that it is monotonic (i.e., that it respects the order of the metric scale), and second, that it does not depend on the system under evaluation π_k (i.e., that it is the same for all systems) The goal is to find the relation between μ_k^H and μ_k^M. The idea is to express $\mathbb{E}[h_k]$ in terms of an expectation over metric ratings as follows (for full derivation, see Appendix A):

$$\mathbb{E}[h_k] = \mathbb{E}_{p_k(m)}[\mathbb{E}_{p_k(h)}[h|m]] \quad (1)$$

The crucial element of Equation 1 is the conditional expectation $\mathbb{E}_{p_k(h)}[h|m]$. Here we consider the expectation according to $p_k(h)$, the distribution of human ratings for system π_k. Equation 1 describes the relationship between μ_k^H and μ_k^M by expressing the expected human rating in terms of an expectation over metric ratings. We interpret this element as a function f_k, which takes a metric rating as input and returns the expected human rating. Equation 1 yields a function f_k for each system separately, which is not necessarily the same across systems. At this point, we can restate the introductory discussion using our formalism. When averaging metrics $\hat{\mu}_k^M$ to rank systems, we implicitly assume that there is a global function f_G that is equal to all the system-specific functions f_k, i.e., $f_G = f_1 = \cdots = f_K$, and thus, only measure if f_G is monotonic (through correlation to human judgments). However, as shown in Figure 1, this assumption does not hold in practice (where blue is f_G, and we have an f_k for the two other systems respectively). To show that this is insufficient, we consider the effects of violating the assumption. Let us assume $f_1 \neq f_2$, but both are monotonic. Consider the extreme example that $\mu_1^M = \mu_2^M$, i.e., systems π_1 and π_2 are of the same quality under the metric. However, consider the case $f_1(m) = f_2(m) + C$, $C > 0$. Then $\frac{1}{N}\sum_j f_1(m_1^{(j)}) = C + \frac{1}{N}\sum_j f_2(m_1^{(j)}) > \frac{1}{N}\sum_j f_2(m_2^{(j)})$, thus, yielding that π_1 is better than π_2 in human space. This shows the necessity of measuring the monotonicity of a global function f_G and the dependence of the metric on the systems under evaluation.

We first introduce the Expected Deviation (ED), which measures the difference between f_G and f_k for all $k \in \{1 \ldots K\}$, which tells us how much a system is over-or-underestimated according to the metric. That is the difference

$$ED(k) = \frac{1}{N}\sum_{j=1}^{N} f_G(m_k^{(j)}) - \frac{1}{N}\sum_{j=1}^{N} f_k(m_k^{(j)}) \quad (2)$$

This is equivalent to $\mu_k^G - \mu_k^H$, where $\mu_k^G = \frac{1}{N}\sum_{j=1}^{N} f_G(m_k^{(j)})$, thus, we measure the difference between the average rating according to the global function and the average rating of the system-specific function, which corresponds to the human rating-average. Note that a mis-ranking occurs if one system is severely overrated while another is severely underrated. Thus we define the system dependence score *SysDep*$(\mathcal{M})$ as the worst case of this effect:

$$SysDep(\mathcal{M}) = max_{\pi_k} ED(k) - min_{\pi_k} ED(k) \quad (3)$$

3 Experiments

Estimating the Conditional Expectation. Even though the functions f_k and f_G are unknown in general, we can estimate them from data. We will use *Isotonic Regression (IR)* (Barlow and Brunk, 1972) for this purpose, which estimates a monotonic function $\hat{f}_k$ minimizing $\sum_j(\hat{f}_k(m_k^{(j)}) - h_k^{(j)})^2$. To estimate f_G, we utilize the same approach, pooling all paired data from all systems. To compute the *SysDep* of a metric, we compute the ED for each MT system under that metric. For this, we compute the average human rating $\hat{\mu}_k^H = \frac{1}{N_H}\sum_{j=1}^{N_H} h_k^{(j)}$, the average metric rating $\hat{\mu}_k^M = \frac{1}{N_M}\sum_{j=1}^{N_M} m_k^{(j)}$, as well as average remapped rating $\hat{\mu}_k^G = \frac{1}{N_M}\sum_{j=1}^{N_M} \hat{f}_G(m_k^{(j)})$ for each MT system. We provide our code in Appendix E.

Data. We rely on data from the WMT 23 Metrics shared task (Freitag et al., 2023). The data includes translations for 3 language pairs: English to German (*en-de*), Hebrew to English (*he-en*), and Chinese to English (*zh-en*). The translations were produces by 12-15 systems (depending on the language pair) which participated in the general MT task (Kocmi et al., 2023). Human ratings are available in the form of MQM annotations (Lommel et al., 2014), which are based on error-span annotations by experts that are subsequently transformed into a numeric value by assigning scores to errors based on their severity. Here, we will present results for the *XCOMET* (Guerreiro et al., 2023) metric (best metric according to correlation to human judgments) and the *zh-en* language pair, where we have access to $N_M = 1976$ segments per system rated by the metric and $N_H = 1177$ of these segments rated with human MQM ratings. Results for the other language pairs and an additional metric are shown in Appendix B. To estimate the conditional expectation functions f_k, we use the 1177 paired ratings for each system π_k. We employ $B = 200$ bootstrap samples of the paired data to fit B IR models. Our estimate, $\hat{f}_k$, represents the average of these B IR models. In Figure 1, we also present the range between the 2.5% and 97.5% percentiles.

Results. We show the results in Table 1. We can see that the ED ranges from -0.82 to 1.996, thus yielding a *SysDep* score of 2.816. We see that both *Lan-BridgeMT* and *GPT4-5shot* are underrated by the metric (negative *ED*), but *Lan-BridgeMT* more so, enough to invert their order. At the bottom of the ranking, we see a relatively large absolute *ED*. Ranking errors reflect an interplay between the systems' rating gap and the *ED*s. For example, *Online-A* loses 2 ranks according to the metric even though it has the lowest absolute *ED*. We also note that even though $\hat{f}_G$ is monotonic, the ranking between the metric and the remapped scores does not match completely. This can be attributed to

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
Lan-BridgeMT	-2.100	1	0.889	2	-2.920	2	*-0.820*
GPT4-5shot	-2.305	2	0.893	1	-2.800	1	-0.494
Yishu	-3.231	3	0.880	4	-3.179	4	0.052
ONLINE-B	-3.385	4	0.879	5	-3.188	5	0.197
HW-TSC	-3.398	5	0.883	3	-3.080	3	0.318
ONLINE-A	-3.785	6	0.856	8	-3.812	8	-0.027
ONLINE-Y	-3.792	7	0.868	6	-3.479	6	0.313
ONLINE-G	-3.857	8	0.864	7	-3.607	7	0.250
ONLINE-W	-4.062	9	0.848	9	-4.165	10	-0.103
ZengHuiMT	-4.232	10	0.846	10	-4.140	9	0.092
IOL-Research	-4.586	11	0.843	11	-4.251	11	0.335
ONLINE-M	-5.433	12	0.820	15	-4.907	15	0.526
ANVITA	-6.078	13	0.830	13	-4.602	13	1.475
NLLB-MBR-BLEU	-6.360	14	0.825	14	-4.726	14	1.634
NLLB-Greedy	-6.574	15	0.831	12	-4.578	12	**1.996**

Table 1: System rankings and average rating of WMT 23 *zh-en* systems according to *XCOMET*. The lowest score is in italics, and the highest is in bold.

the uncertainty introduced by bootstrapping and extrapolating to the unpaired metric ratings. It can be seen for *ONLINE-W* and *ZengHuiMT*, which have similar metric ratings.

Overall, our results show that although there is a highly monotonic function between the *XCOMET* scale and the human scale, *XCOMET* exhibits a high dependency on the system under evaluation, thus yielding an inconsistent ranking between humans and *XCOMET*.

4 Related Work

The derivation in Section 2 closely follows Wu and Resnick (2024), who provide the same argument in the context of binary prevalence estimation. In our case, the conditional expectation $\mathbb{E}[h|m]$ plays the same role as the calibration curve in their framework. Under that lens, the Expected Deviation is analogous to the Expected Calibration Error (Posocco and Bonnefoy, 2021). Following the same analogy, evaluating a new MT system is similar to applying a classifier to a new domain.

Previous studies by Deriu et al. (2023) and von Däniken et al. (2022) have highlighted that metric performance depends on the system under test. They employed a Bayesian framework to determine the proportions of binary or preference human ratings from metric scores; critically relying on confusion matrices estimated for each MT system. In this discrete rating context, these confusion matrices represent the same concept as $\mathbb{E}[h|m]$. In follow-up work, von Däniken et al. (2024) find that some metrics disproportionately favor certain MT systems over others compared to human preference ratings. Our finding provides a plausible explanation.

Chaganty et al. (2018) shows how to combine human ratings and metric ratings to derive an unbiased estimate of the true expected human rating μ^H while reducing the number of annotations needed. The proposed control variates estimator is based only on human and metric scores for a given MT system, even when estimating their correlation, thus avoiding the problem we describe.

Wei and Jia (2021) consider disagreements in the ordering of systems when using μ_k^M instead of μ_k^H. In particular they study the sign error, cases where $sign(\mu_1^M - \mu_2^M) \neq sign(\mu_1^H - \mu_2^H)$. They apply a bias variance decomposition to this error and find that while the human estimator is unbiased, it exhibits high variance while the opposite is the case for metrics. Our *SysDep* score presents a way to quantify this bias.

5 Conclusion

In this paper, we emphasize the importance of ensuring that automated metrics treat all MT systems consistently, a factor overlooked in current evaluations. By mapping metric scores to the human rating scale, we estimate how much a metric misjudges individual system performance. We compute the range of these deviations to assess how consistently a metric treats different systems. In Appendix C, we re-evaluate WMT23 metrics from this perspective. Additionally, in Appendix D, we confirm that these results stem from systematic differences in how metrics treat systems by measuring deviations within splits of a single system's ratings.

Limitations

This paper is intended to explore a supplementary aspect of the evaluation of automated metrics. The *SysDep* measure we developed will hopefully provide a starting point for the development of more refined evaluation of the way metrics treat different systems differently.

Our experiments are based solely on data from the WMT23 Metrics shared task. To further solidify our findings a larger scale study with more domains and larger sample sizes are needed.

While we provide a way to measure the system dependence of a metric, we do not provide any suggestions on how to develop metrics that minimize the *SysDep*.

Acknowledgments

This work was supported by the Swiss National Science Foundation (SNF) within the project "Unified Model for Evaluation of Text Generation Systems (UniVal)" [200020_219819].

References

R. E. Barlow and H. D. Brunk. 1972. The isotonic regression problem and its dual. *Journal of the American Statistical Association*, 67(337):140–147.

Anja Belz and Eric Kow. 2010. Comparing rating scales and preference judgements in language evaluation. In *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Arun Chaganty, Stephen Mussmann, and Percy Liang. 2018. The price of debiasing automatic metrics in natural language evalaution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Melbourne, Australia. Association for Computational Linguistics.

Jan Deriu, Pius von Däniken, Don Tuggener, and Mark Cieliebak. 2023. Correction of errors in preference ratings from automated metrics for text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6456–6474, Toronto, Canada. Association for Computational Linguistics.

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. 2021. 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.

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. *Preprint*, arXiv:2310.10482.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *Preprint*, arXiv:2302.09210.

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.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. 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.

Tom Kocmi, Christian Federmann, Roman Grundkiewicz, Marcin Junczys-Dowmunt, Hitokazu Matsushita, and Arul Menezes. 2021. To ship or not to ship: An extensive evaluation of automatic metrics for machine translation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 478–494, Online. Association for Computational Linguistics.

Arle. Language Technology Lab) Lommel, Hans. Language Technology Lab) Uszkoreit, and Aljoscha. Language Technology Lab) Burchardt. 2014. Multidimensional quality metrics (mqm) : a framework for declaring and describing translation quality metrics. *Tradumàtica*, (12):455–463.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nicolas Posocco and Antoine Bonnefoy. 2021. Estimating expected calibration errors. In *Artificial Neural Networks and Machine Learning – ICANN 2021*, pages 139–150, Cham. Springer International Publishing.

Brian Thompson and Matt Post. 2020a. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020b. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation*, pages 561–570, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Pius von Däniken, Jan Deriu, Don Tuggener, and Mark Cieliebak. 2022. On the effectiveness of automated metrics for text generation systems. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1503–1522, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Pius von Däniken, Jan Deriu, Don Tuggener, and Mark Cieliebak. 2024. Favi-score: A measure for favoritism in automated preference ratings for generative AI evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4437–4454, Bangkok, Thailand. Association for Computational Linguistics.

Johnny Wei and Robin Jia. 2021. The statistical advantage of automatic NLG metrics at the system level. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6840–6854, Online. Association for Computational Linguistics.

Siqi Wu and Paul Resnick. 2024. Calibrate-extrapolate: Rethinking prevalence estimation with black box classifiers. *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):1634–1647.

Yangjian Wu and Gang Hu. 2023. Exploring prompt engineering with GPT language models for document-level machine translation: Insights and findings. In *Proceedings of the Eighth Conference on Machine Translation*, pages 166–169, Singapore. Association for Computational Linguistics.

A Full Derivation

In Equation 4, we give the full derivation of Equation 1 in Section 2. In the following $p_k(h)$ is the density of human ratings for system π_k, $p_k(m)$ is its density of metric ratings, and $p_k(h, m)$ the joint density.

$$
\begin{aligned}
\mathbb{E}[h_k] &= \int_{-\infty}^{\infty} h p_k(h) \mathrm{d}h \\
&= \int_{-\infty}^{\infty} h \left[\int_{-\infty}^{\infty} p_k(h, m) \mathrm{d}m \right] \mathrm{d}h \\
&= \int_{-\infty}^{\infty} h \left[\int_{-\infty}^{\infty} p_k(h|m) p_k(m) \mathrm{d}m \right] \mathrm{d}h \\
&= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h p_k(h|m) p_k(m) \mathrm{d}m \mathrm{d}h \\
&= \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} h p_k(h|m) \mathrm{d}h \right] p_k(m) \mathrm{d}m \\
&= \int_{-\infty}^{\infty} \mathbb{E}[h_k|m] p_k(m) \mathrm{d}m \\
&= \mathbb{E}_{p_k(m)}[\mathbb{E}_{p_k(h)}[h|m]]
\end{aligned}
\tag{4}
$$

B Additional Results

Here we extend our experiment from Section 3 to additional language pairs and metrics of WMT 23. For the *en-de* language pair $N^H = 460$ and $N^M = 557$ and for *he-en* $N^H = 820$ and $H^M = 1910$. We show the results for *XCOMET* for each language pair in Tables 2, 3, and 4 (Note that Table 4 is the same as Table 1 in Section 3). We also include results for *GEMBA-MQM* (Kocmi and Federmann, 2023), which is a reference free metric based on prompting LLMs. The results can be seen in Tables 5, 6, and 7.

C Evaluating the System Dependence of WMT23 Metrics

In Section 2, we introduced the *SysDep* score. It measures the worst case in the difference of expected deviations (ED), which measures the difference between the average human rating we expect to see based on metric ratings and assuming a single global f_G and the true average human rating for a system π_k. To measure the system dependence of a metric across a set of systems $\pi_1, \ldots, \pi_K$, we compute the range of the individual ED: $SysDep = max_{\pi_k} ED(k) - min_{\pi_k} ED(k)$. We noted in Section 3 that $ED(k)$ alone is not enough to know whether system π_k will be ranked incorrectly, it depends on the true margin to the other systems, and their dependencies. By measuring the range, we consider the worst case when comparing two systems. We show the dependency ranges for all WMT23 metrics on all language pairs in Table 8. We notice that the values for *en-de* are large than the others. This is due to a larger range of human rating averages for this language pair (see also Tables 2–7 in Appendix B). We therefore also do not aggregate across language pairs.

Variants of *MetricX-23* (Juraska et al., 2023) perform best on *en-de* and *he-en*, while *GEMBA-MQM* has the lowest range for *zh-en*. The reference-free *prismSrc* (Thompson and Post, 2020a,b) metric performs worst on *en-de* and *zh-en*. The baseline *Random-sysname* (Freitag et al., 2023) performs worst for *he-en*. This baseline is an interesting case, as it is the prototypical example of a metric where every f_k is different. It assigns a fixed score to each system based on its name and adds Gaussian noise to this value to assign segment level scores. Therefore each f_k will be a different constant function.

D Intra-System Variability

In order to confirm that the observed *SysDep* scores are indeed due to a metric systematically treating systems differently and not due to variance in ratings, we will measure the maximum intra-system scores. For this, we use ratings from a single system and split them into 2 equal sized parts 10 times with different random seeds. This simulates a setting with 20 systems with half the sample size of the original setting. We then compute the *SysDep* score.

In Table 9, we show the maximum intra-system *SysDep* score computed this way over all systems for a given metric and language pair. We observe that for *he-en* and *zh-en* all scores are lower than the minimum between system *SysDep* reported in Appendix C. This confirms that in those cases metrics treat different systems differently. For the *en-de* language pair, we observe that while in many cases the intra-system score is lower than the *SysDep* between systems for the same metric and language pair, this is not always the case. This could be due to the metrics treating systems more equally for this language pair, or the relatively small sample sizes for *en-de* compared to the other language pairs.

E Estimating Conditional Expectations

In Section 3, we gave a brief overview of how to compute estimates for the functions f_k and f_G. In Listings 1 and 2, we show our *python* implementation. To estimate the system-level $\hat{f}_k$, we call the *.fit* method with human and metric ratings for system π_k. To evaluate the function $\hat{f}_k$, we use the *.conditional_expectation* method. To estimate the global function $\hat{f}_G$, we use the *.fit* method with paired human and metric ratings for all systems. We compute the remapped rating $\hat{\mu}_k^G$ by first fitting $\hat{f}_G$ and then using the *.remapped_expectation*

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
GPT4-5shot	-3.724	1	0.882	2	-4.768	1	-1.044
ONLINE-W	-3.950	2	0.883	1	-4.821	2	-0.871
ONLINE-B	-4.711	3	0.871	3	-5.272	3	-0.560
ONLINE-Y	-5.643	4	0.858	4	-5.909	4	-0.266
ONLINE-A	-5.668	5	0.853	5	-6.152	5	-0.483
ONLINE-G	-6.574	6	0.834	6	-7.079	6	-0.505
ONLINE-M	-6.936	7	0.830	7	-7.399	7	-0.462
Lan-BridgeMT	-8.670	8	0.801	9	-8.670	9	-0.000
ZengHuiMT	-9.255	9	0.790	11	-9.387	11	-0.132
NLLB-Greedy	-9.543	10	0.812	8	-8.405	8	1.138
NLLB-MBR-BLEU	-10.794	11	0.797	10	-9.005	10	1.789
AIRC	-14.228	12	0.724	12	-13.658	12	0.570

Table 2: System rankings and average rating of WMT 23 *en-de* systems according to *XCOMET*.

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
GPT4-5shot	-1.333	1	0.913	2	-1.690	2	-0.358
ONLINE-A	-1.381	2	0.908	3	-1.817	3	-0.436
ONLINE-B	-1.546	3	0.916	1	-1.635	1	-0.089
GTCOM-Peter	-1.886	4	0.904	4	-1.916	4	-0.030
UvA-LTL	-1.919	5	0.893	6	-2.193	6	-0.274
ONLINE-G	-2.055	6	0.895	5	-2.137	5	-0.082
ONLINE-Y	-2.349	7	0.881	8	-2.511	8	-0.162
ZengHuiMT	-2.382	8	0.889	7	-2.294	7	0.088
Samsung-Res.-Ph.	-3.234	9	0.874	9	-2.666	9	0.568
NLLB-MBR-BLEU	-3.678	10	0.869	11	-2.805	11	0.872
NLLB-Greedy	-3.790	11	0.872	10	-2.714	10	1.076
Lan-BridgeMT	-3.793	12	0.867	12	-2.823	12	0.971

Table 3: System rankings and average rating of WMT 23 *he-en* systems according to *XCOMET*.

method on the metric ratings for system π_k. We rely on the *Isotonic Regression* implementation from *scikit-learn* (Pedregosa et al., 2011) [2] and numerical utility functions from *numpy* (Harris et al., 2020).

[2] https://scikit-learn.org/stable/modules/generated/sklearn.isotonic.IsotonicRegression.html

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
Lan-BridgeMT	-2.100	1	0.889	2	-2.920	2	*-0.820*
GPT4-5shot	-2.305	2	0.893	1	-2.800	1	-0.494
Yishu	-3.231	3	0.880	4	-3.179	4	0.052
ONLINE-B	-3.385	4	0.879	5	-3.188	5	0.197
HW-TSC	-3.398	5	0.883	3	-3.080	3	0.318
ONLINE-A	-3.785	6	0.856	8	-3.812	8	-0.027
ONLINE-Y	-3.792	7	0.868	6	-3.479	6	0.313
ONLINE-G	-3.857	8	0.864	7	-3.607	7	0.250
ONLINE-W	-4.062	9	0.848	9	-4.165	10	-0.103
ZengHuiMT	-4.232	10	0.846	10	-4.140	9	0.092
IOL-Research	-4.586	11	0.843	11	-4.251	11	0.335
ONLINE-M	-5.433	12	0.820	15	-4.907	15	0.526
ANVITA	-6.078	13	0.830	13	-4.602	13	1.475
NLLB-MBR-BLEU	-6.360	14	0.825	14	-4.726	14	1.634
NLLB-Greedy	-6.574	15	0.831	12	-4.578	12	**1.996**

Table 4: System rankings and average rating of WMT 23 *zh-en* systems according to *XCOMET*.

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
GPT4-5shot	-3.724	1	-2.447	1	-4.123	1	-0.399
ONLINE-W	-3.950	2	-3.429	2	-4.822	2	-0.872
ONLINE-B	-4.711	3	-4.048	3	-5.383	3	-0.672
ONLINE-Y	-5.643	4	-4.424	4	-5.832	5	-0.189
ONLINE-A	-5.668	5	-4.567	5	-5.826	4	-0.158
ONLINE-G	-6.574	6	-6.018	6	-7.047	6	-0.473
ONLINE-M	-6.936	7	-6.217	7	-7.113	7	-0.177
Lan-BridgeMT	-8.670	8	-8.197	8	-8.891	9	-0.221
ZengHuiMT	-9.255	9	-8.357	9	-8.867	8	0.388
NLLB-Greedy	-9.543	10	-10.043	10	-9.683	10	-0.140
NLLB-MBR-BLEU	-10.794	11	-10.724	11	-10.352	11	0.442
AIRC	-14.228	12	-13.941	12	-12.526	12	1.702

Table 5: System rankings and average rating of WMT 23 *en-de* systems according to *GEMBA-MQM*.

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
GPT4-5shot	-1.333	1	-1.923	1	-1.377	1	-0.045
ONLINE-A	-1.381	2	-3.850	2	-1.882	2	-0.501
ONLINE-B	-1.546	3	-4.108	3	-1.969	3	-0.423
GTCOM-Peter	-1.886	4	-4.859	4	-2.144	4	-0.258
UvA-LTL	-1.919	5	-5.628	6	-2.312	6	-0.393
ONLINE-G	-2.055	6	-5.240	5	-2.281	5	-0.225
ONLINE-Y	-2.349	7	-6.885	8	-2.677	8	-0.328
ZengHuiMT	-2.382	8	-6.032	7	-2.484	7	-0.102
Samsung-Res.-Ph.	-3.234	9	-8.545	12	-2.954	12	0.280
NLLB-MBR-BLEU	-3.678	10	-8.075	9	-2.817	10	0.861
NLLB-Greedy	-3.790	11	-8.261	10	-2.813	9	0.977
Lan-BridgeMT	-3.793	12	-8.469	11	-2.840	11	0.953

Table 6: System rankings and average rating of WMT 23 *he-en* systems according to *GEMBA-MQM*.

	Human		Metric		Remapped		Exp. Deviation
	$\hat{\mu}_k^H$	R	$\hat{\mu}_k^M$	R	$\hat{\mu}_k^G$	R	ED
Lan-BridgeMT	-2.100	1	-1.949	2	-2.419	2	-0.319
GPT4-5shot	-2.305	2	-1.601	1	-2.199	1	0.106
Yishu	-3.231	3	-4.790	5	-3.492	5	-0.261
ONLINE-B	-3.385	4	-4.717	4	-3.489	4	-0.104
HW-TSC	-3.398	5	-4.367	3	-3.336	3	0.062
ONLINE-A	-3.785	6	-5.568	8	-3.838	9	-0.053
ONLINE-Y	-3.792	7	-5.453	7	-3.611	6	0.181
ONLINE-G	-3.857	8	-5.275	6	-3.724	7	0.134
ONLINE-W	-4.062	9	-5.760	9	-3.772	8	0.290
ZengHuiMT	-4.232	10	-6.337	10	-4.089	11	0.143
IOL-Research	-4.586	11	-6.511	11	-4.067	10	0.519
ONLINE-M	-5.433	12	-9.115	12	-4.899	13	0.534
ANVITA	-6.078	13	-9.440	13	-4.844	12	1.234
NLLB-MBR-BLEU	-6.360	14	-11.339	15	-5.379	15	0.981
NLLB-Greedy	-6.574	15	-11.282	14	-5.312	14	1.262

Table 7: System rankings and average rating of WMT 23 *zh-en* systems according to *GEMBA-MQM*.

```python

from typing import Self, Tuple

import numpy as np
from numpy.random import Generator, default_rng

from sklearn.isotonic import IsotonicRegression


class BootstrapIsotonic:

    def __init__(
        self,
        n_bootstrap: int = 200,
        rng: int | Generator = 0xdeadbeef,
    ):
        self.n_bootstrap = n_bootstrap
        self.rng = default_rng(rng)
        self.models = []

    def fit(
        self,
        human_ratings: np.ndarray[float],  # 1d array of human ratings
        metric_ratings: np.ndarray[float], # 1d array of matching metric ratings
    ) -> Self:
        # fit a model of f_k or f_G, i.e. the conditional expectation of human ratings given metric ratings
        # to get model a system-level function f_k, use only human_ratings for that given system k
        # to model the global function f_G, use data from all systems

        assert len(human_ratings) == len(metric_ratings)
        n_samples = len(human_ratings)

        for _ in range(self.n_bootstrap):
            bootstrap_indices = self.rng.choice(
                np.arange(n_samples),
                n_samples,
                replace=True
            )
            isotonic_model = IsotonicRegression(
                y_min=None,  # MQM scores range from large negative to 0
                y_max=0.,
                increasing=True,  # metric has positive correlation
                out_of_bounds='nan',  # don't extrapolate
            )
            isotonic_model.fit(
                X=metric_ratings[bootstrap_indices],
                y=human_ratings[bootstrap_indices],
            )
            self.models.append(isotonic_model)

        return self

    def _predict_bootstrap(
        self,
        m: np.ndarray[float]  # 1d array of metric ratings
    ) -> np.ndarray[float]:
        # helper function getting predictions from each model, returns 2d array of size [n_bootstrap, len(m)]
        result = np.zeros((self.n_bootstrap, len(m)), dtype=float)
        for bix, model in enumerate(self.models):
            result[bix, :] = model.predict(m)
        return result
```

Listing 1: Part 1 of the *python* code to estimate $\hat{f}_k$, $\hat{f}_G$, and $\hat{\mu}_k^G$.

```python

    def conditional_expectation(
        self,
        metric_ratings: np.ndarray[float],  # 1d array of metric ratings
    ) -> np.ndarray[float]:
        # this computes the function f_k or f_G (depending on what data we fitted on)
        bootstrap_predictions = self._predict_bootstrap(metric_ratings)
        return np.nanmean(bootstrap_predictions, axis=0)

    def confidence(
        self,
        metric_ratings: np.ndarray[float],  # 1d array of metric ratings
    ) -> Tuple[np.ndarray[float], np.ndarray[float]]:
        # this computes the confidence bounds around f_k or f_G in Figure 1
        bootstrap_predictions = self._predict_bootstrap(metric_ratings)
        lower = np.nanpercentile(bootstrap_predictions, 2.5, axis=0)
        upper = np.nanpercentile(bootstrap_predictions, 97.5, axis=0)
        return lower, upper

    def remapped_expectation(
        self,
        metric_ratings: np.ndarray[float], # 1d array of metric ratings
    ) -> float:
        # used to compute remapped system scores in Table 1.
        expected_human_ratings = self.conditional_expectation(metric_ratings)
        return np.nanmean(expected_human_ratings)
```

Listing 2: Part 2 of the *python* code to estimate $\hat{f}_k$, $\hat{f}_G$, and $\hat{\mu}_k^G$.

	en-de	he-en	zh-en
BERTscore	7.18	1.73	3.87
BLEU	9.02	2.06	4.23
BLEURT-20	3.68	1.66	3.35
Calibri-COMET22-QE	3.68	2.06	3.30
Calibri-COMET22	4.24	1.64	3.40
chrF	8.11	1.92	4.29
COMET	4.29	1.64	3.35
CometKiwi	3.84	1.95	3.02
CometKiwi-XL	3.77	1.98	3.01
CometKiwi-XXL	3.68	1.82	2.92
cometoid22-wmt21	5.44	2.11	3.30
cometoid22-wmt22	5.17	2.09	3.21
cometoid22-wmt23	4.66	1.81	3.20
docWMT22CometDA	3.87	1.65	3.37
docWMT22CometKiwiDA	4.53	1.83	2.76
eBLEU	9.49	2.08	4.29
embed-llama	7.07	2.13	4.21
f200spBLEU	8.42	2.01	4.23
GEMBA-MQM	2.57	1.48	**1.58**
instructscore	3.59	1.53	3.68
KG-BERTScore	4.24	1.88	3.04
MaTESe	5.98	1.49	3.16
mbr-metricx-qe	3.69	1.58	2.39
MEE4	8.48	1.88	4.21
MetricX-23-b	2.26	1.29	2.81
MetricX-23-c	3.56	1.69	2.36
MetricX-23-QE-b	**2.11**	1.55	2.62
MetricX-23-QE-c	2.82	**1.21**	1.65
MetricX-23-QE	2.93	1.77	3.12
MetricX-23	2.57	1.33	3.04
mre-score-labse-regular	9.70	1.65	3.94
MS-COMET-QE-22	5.87	2.22	3.37
prismRef	8.71	1.79	3.97
prismSrc	*11.24*	2.48	*4.61*
Random-sysname	9.97	*2.52*	4.53
sescoreX	3.59	1.52	3.47
tokengram-F	8.17	1.93	4.29
XCOMET-Ensemble	2.83	1.51	2.82
XCOMET-QE-Ensemble	2.95	1.78	2.95
XCOMET-XL	3.39	1.59	3.20
XCOMET-XXL	2.71	1.48	2.99
XLsim	7.83	2.01	4.20
YiSi-1	5.95	1.60	3.65

Table 8: *SysDep* for each metric and language pair. We show the **minimum** and *maximum* for each language pair.

	en-de	he-en	zh-en
BERTscore	2.08	0.88	1.00
BLEU	2.20	0.95	0.99
BLEURT-20	2.68	0.89	0.87
Calibri-COMET22-QE	2.97	0.82	0.85
Calibri-COMET22	2.50	0.79	0.89
chrF	2.07	0.96	0.93
COMET	2.25	0.78	0.87
CometKiwi	3.01	0.83	0.81
CometKiwi-XL	2.90	0.87	0.80
CometKiwi-XXL	2.65	0.83	0.85
cometoid22-wmt21	2.81	0.76	0.89
cometoid22-wmt22	2.74	0.73	0.81
cometoid22-wmt23	2.58	0.79	0.86
docWMT22CometDA	2.32	0.83	0.95
docWMT22CometKiwiDA	2.61	0.88	0.95
eBLEU	2.49	0.91	0.98
embed-llama	2.18	0.92	1.14
f200spBLEU	2.10	0.94	0.95
GEMBA-MQM	2.89	0.93	0.88
instructscore	2.20	0.83	0.82
KG-BERTScore	2.90	0.85	0.82
MaTESe	2.71	0.77	0.80
mbr-metricx-qe	2.52	0.84	0.79
MEE4	2.15	0.91	0.94
MetricX-23-b	2.67	0.87	0.68
MetricX-23-c	2.89	0.90	0.77
MetricX-23-QE-b	2.67	0.80	0.71
MetricX-23-QE-c	2.25	0.80	0.81
MetricX-23-QE	2.46	0.80	0.74
MetricX-23	2.34	0.90	0.63
mre-score-labse-regular	2.27	0.91	0.93
MS-COMET-QE-22	2.17	0.93	0.90
prismRef	2.24	0.83	0.99
prismSrc	2.09	0.88	0.91
Random-sysname	2.36	0.96	0.95
sescoreX	2.20	0.86	0.86
tokengram-F	2.10	0.96	0.94
XCOMET-Ensemble	2.44	0.71	0.68
XCOMET-QE-Ensemble	2.30	0.72	0.72
XCOMET-XL	2.44	0.74	0.67
XCOMET-XXL	2.35	0.74	0.69
XLsim	2.71	0.90	0.94
YiSi-1	2.44	0.83	0.90

Table 9: Maximum intra-system *SysDep* score for all metrics and language pairs.