Characterizing the Confidence of Large Language Model-Based Automatic Evaluation Metrics

Rickard Stureborg[1,2] **Dimitris Alikaniotis**[1] **Yoshi Suhara**[3,*]
[1]Grammarly [2]Duke University [3]NVIDIA
rickard.stureborg@duke.edu
dimitrios.alikaniotis@grammarly.com
ysuhara@nvidia.com

Abstract

There has recently been a growing interest in using Large Language Models (LLMs) to evaluate NLP tasks automatically. Considerable research effort has been put into improving such systems towards achieving high correlations with human judgement. However, it is still unclear what level of correlation is good enough for practical applications of LLM-based automatic evaluation systems. This paper characterizes these LLM evaluators' confidence in ranking candidate NLP models and develops a configurable Monte Carlo simulation method. We show that even automatic metrics with low correlation with human judgement can reach high-confidence rankings of candidate models with reasonable evaluation set sizes (100s of examples). Further, we describe tradeoff curves between the LLM evaluator performance (i.e., correlation with humans) and evaluation set size; loss in correlation can be compensated with modest increases in the evaluation set size. We validate our results on RoSE, a text summarization dataset, and find our estimates of confidence align with empirical observations.[1]

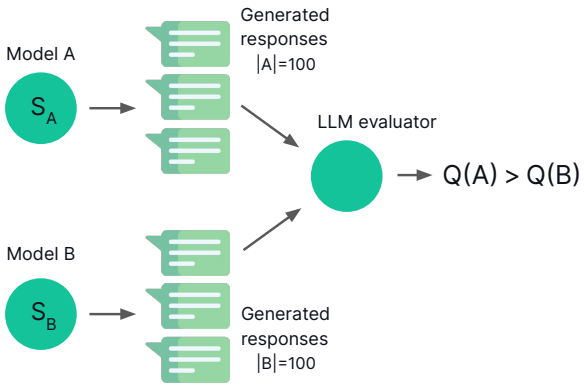


Figure 1: **Diagram of candidate model ranking procedure using an LLM evaluator**. Here, candidate models produce responses for the same evaluation set T of input prompts, and the mean scores by an LLM-based automatic evaluation metric are compared. Suppose that S_A is MV-BART and S_B is PEGASUS[2], and the evaluation set is each of their responses across 100 prompts. Our results indicate LLM evaluators would require approximately $r = .42$ correlation with human judgement to reach a 95% confidence in selecting the correct ranking.

1 Introduction

Automatic evaluation is a staple of Natural Language Processing (NLP) tasks, from the popular ROUGE score in text summarization to BLEU score in machine translation. These metrics often rely on human-written references, increasing the cost and effort of evaluation. Recently, Large Language Models (LLMs) have become commonly used evaluators because of their zero-shot capability in understanding the quality of texts (Wang et al., 2023; Huang et al., 2023). These methods, which we refer to as LLM-based automatic evaluation metrics (or LLM evaluators, in short), do not require human-written reference outputs and can be reconfigured to new tasks quickly. The goal of these automatic metrics is to replace human annotators in evaluating systems since human annotation is often expensive, slow, and difficult to manage (Stureborg et al., 2023). As such, well-performing auto-evaluators are the ones that correlate highly with human judgements.

However, relying on LLM evaluators as a replacement for human judgement comes with familiar challenges (Krishna et al., 2021; Schluter, 2017). Since the metrics are not perfect, we may want to further assess not just the correlation they have with human judgements (and our confidence in that value), but also the confidence one can have in downstream decisions we make using the metrics. A common use of LLM evaluators are to compare new, specialized Natural Language Gener-

*Work done while at Grammarly.
[1]Code available at github.com/rickardstureborg/llm-eval-confidence
[2]On SAMSum, MV-BART scores approximately 0.1 ACU better than PEGASUS.

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 2: Short Papers, pages 76–89

ation (NLG) models for a task. How confident can we be that the ranking given by an LLM evaluator is the same as the ranking human evaluators would give? Estimating this quantity is difficult, since it depends on many factors (evaluation set size, the evaluator's correlation with human labels, the magnitude of candidate models' performance gap, etc.), and available datasets often only compare a handful of candidate models at a time.

The research question we tackle in this paper is to characterize how likely (how confident) LLM evaluators are to predict correct pairwise rankings among candidate models. To estimate confidence, our work suggest a configurable monte carlo simulation developed based on empirical observations about LLM evaluators. We explore possible tradeoffs between factors affecting this confidence which could save on inference costs.

2 Methodology

The task we are interested in solving is to quantify the confidence in an LLM evaluator's decision when ranking two candidate models against each other. Figure 1 shows a useful diagram of how we compare two models or systems against one another. Appendix A includes a formal description of this task.

2.1 LLM-based Automatic Evaluation Metric

In order to evaluate our framework, we extend G-EVAL (Liu et al., 2023a), a state-of-the-art (SOTA) LLM evaluation method as our automatic metric. G-EVAL is an LLM-based automatic evaluation metric, specifically built on ChatGPT models. To determine the best models between candidates, we evaluate model responses over a validation dataset (described in §2.3) and use the mean score given from G-EVAL over this validation set to rank models against one another. The metric that our version of G-EVAL predicts is ACU, introduced by (Liu et al., 2023b). ACU is a recall-like metric which measures how many of the key facts (Atomic Content Units) are captured by the summary. The data annotation process for ACU leads to higher quality annotations (Liu et al., 2023b), and the underlying datasets labeled with this score have more diversity for a broader comparison of out-of-domain performance than traditional summary datasets such as SummEval. Further information is available in Appendix D. G-EVAL does not natively predict this metric, so we extend the system to do this by altering the prompt based on the language describing ACU from the original RoSE paper. Exact implementation is discussed in detail in Appendix H including our full prompt. We use `gpt-3.5-turbo-0301` and `gpt-4-0314` checkpoints in all of our experiments using OpenAI models.

2.2 Configurable Monte Carlo Simulation

We develop a methodology for finding the confidence in ranking two candidate models through a configurable Monte Carlo Simulation.[3] We produce synthetic "responses" from hypothetical candidate models S_A and S_B. These synthetic responses are simply denoted by their index in all responses generated (e.g. a_i or b_j) and simulate what the "true" score of a simulated response would be *if* it were given to a human for evaluation. We then simulate the automatic metric as trying to estimate this true score according to its known performance. Appendix B provides a rigorous description of the algorithm we use for configuring and running the simulation.

2.2.1 Assumptions

Access to human-labeled data is only required once. To simulate an automatic metric's behavior, we require knowing its performance as measured by the correlation with human judgements. This correlation can be known on a training set only and does not need to be known over the eventual dataset on which the automatic evaluator will be used to rank models. Specifically, we use Pearson's correlation r between the evaluations (scores) of the automatic metric and the evaluations (scores) given by humans.[4] A higher correlation with human judgement indicates better performance. This step is assumed to have been previously completed with training dataset H when building the automatic evaluator, as is standard when proposing a new automatic evaluation metric. Crucially, we require no actual human-labeled data for the candidate models S_A and S_B.

This human-labeled data can be used to determine the distribution of summary quality. This human-labeled data, originally required for evaluating the strength of an automatic metric, can be reused to learn the distribution of expected summary qualities. We achieve this through kernel den-

[3] Run simulation: `github.com/rickardstureborg/llm-eval-confidence`

[4] We define any deviation from human judgement as error.

sity estimation (KDE), which is a non-parametric method for estimating the probability density function. This resulting probability density estimate is used to sample simulated scores as the ground-truth for summary qualities.

There are no adversarial candidate models. We note that it is, of course, possible to construct an adversarial candidate model S_{adv} such that the correlation with human judgement of the automatic evaluator is different than the correlation assessed when building the automatic metric, which LLMs have been shown to be vulnerable to (Seth et al., 2023). It is an assumption of our methodology that the candidate models are approximately well-behaved in this respect. This is a limitation that we believe future work could improve upon by quantifying how adversarial attacks would affect our results[5], or by building more robust automatic evaluation metrics.

Bias of scores does not matter. Since we are using Pearson's correlation, the direction and magnitude of bias by an automatic metric has no effect on the correlation with human judgment, and is therefore left out of our configurations. This is further substantiated in Appendix G.

LLM-based automatic evaluators can be modeled as a noisy estimation of the human-preferred score. In this context, a noisy estimator takes the true human-labeled score and adds some noise to it to produce an imitated LLM-based automatic score. We find that gaussian noise is a reasonable approximation of LLM-based automatic evaluators based on empirical observations. To validate if these predictions can be approximated using a gaussian noisy estimation paradigm, we compare the absolute errors produced by G-EVAL-3.5 with the absolute errors produced by the noisy estimator in Figure 2.

2.3 Ranking Summarization Models

To empirically validate our simulated results, we focus on the RoSE benchmark introduced by Liu et al. (2023b). RoSE makes use of CNNDM (Nallapati et al., 2016), XSum (Narayan et al., 2018), and SAMSum (Gliwa et al., 2019)—covering a total of 23 summarization systems[6]. We use the CNNDM validation partition (8000 summaries) to inform all our choices in tuning our simulation and evaluate

[5]This could potentially be done through a "generalizability assumption" parameter in the simulation, which determines the bounds of how much r might deviate on the test set.

[6]Appendix I describes all summarization systems used.

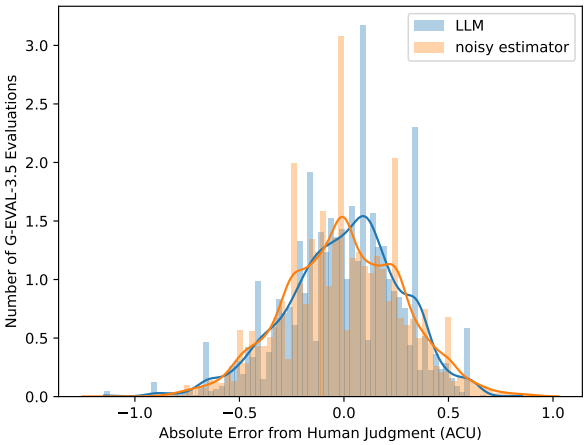


Figure 2: **Absolute errors of LLM evaluator and a noisy estimate on the ACU metric.** We approximate the absolute errors of LLM-based evaluators as a noisy estimator with gaussian noise in our simulation. The distributions of these errors are mostly aligned with what we see in the RoSE data. The blue and orange line are probability density estimates to better see how well-aligned the distributions are. Note that the bias of both metrics has been removed in this plot since it has no impact on the correlation.

the correlation with human judgement of our automatic metric system. This step is similar to the activities done by anyone building an automatic evaluation metric, and will always require human annotated data. We then use CNNDM, XSum, and SAMSum as our test set, and assume the average ACU metric for each system described in Liu et al. (2023b) as the true human-preferred ranking of the systems.[7] This allows us to investigate how our automatic metric compares against true human-sourced rankings. In practice, this step is usually performed on the data where there is no human-sourced rankings. Our work is attempting to investigate how confident one can be in decisions made from this step.

For testing purposes, we use CNNDM (12 systems, 6000 summaries), XSum (8 systems, 4000 summaries), and SAMSum (8 systems, 4000 summaries) as our test set, and assume the average ACU metric for each system described in Liu et al. (2023b) as the true human-preferred ranking of the systems.

3 Results and Discussion

The adapted G-EVAL-3.5 system performs much worse on ACU score than on the SummEval labels it was evaluated on, indicating the potential dif-

[7]Further discussion of this choice available in Appendix D

ficulty of adapting LLM evaluators to new tasks. The correlation with human judgements is given in Table 1.

Dataset	r	ρ	τ
CNNDM	0.22	0.22	0.17
XSum	0.14	0.13	0.10
SAMSum	**0.34**	**0.33**	**0.27**
Mean	0.24	0.23	0.18

Table 1: **Correlation between G-EVAL-3.5 and human judgement on ACU metric**. G-EVAL-3.5 performs much worse on scoring ACU than average performance on SummEval labels (coherence, consistency, fluency, relevance) of $\rho = 0.40$. Temperature was set to 0 for all experiments, and no tuning or prompt-engineering was done.

In Figure 4 and Appendix C.2 we show the relationship between N and r with confidence. As expected, larger evaluation set sizes and higher correlation with human judgement both lead to greater confidence. Our simulation results indicate that LLM evaluators are able to reach fairly high agreement despite low correlation. With an evaluation set of only 100 examples, models with just 0.2 correlation are able to correctly rank a 0.10 ACU difference with $\approx 80\%$ confidence. However, it should be noted that a difference of 0.10 ACU is substantial (10% of the entire range for the metric). Therefore there remains much room for improvement by LLM-based automatic evaluation metrics to discern nuanced performance differences with efficient evaluation set sizes.

Next we characterize the tradeoff between evaluation set size and correlation in Figure 3. We note that the tradeoff is steep with respect to correlation around the current performance of SOTA LLM evaluators (0.40-0.50). For relatively small increases in N one can trade away substantial correlation performance (which is much harder to come by than extra evaluation examples).

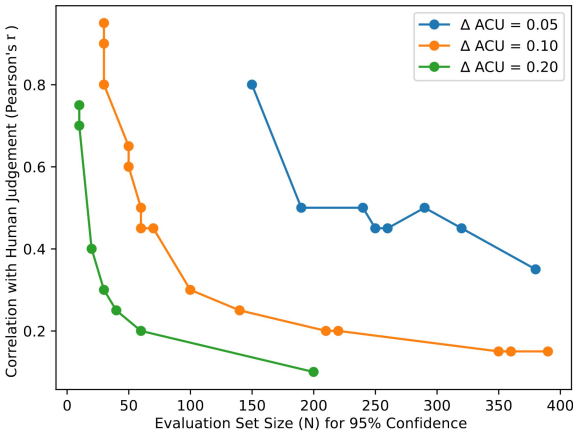


Figure 3: **Required evaluation set size N when correlation is r to reach 95% confidence for ranking a model pair with ΔACU expected quality difference.** This tradeoff between r and N can be exploited to lower overall evaluation costs. For example, OpenAI's GPT-4 is 20-30 times more expensive than GPT-3.5 Turbo, but correlates better on many tasks. Correlation performance that can be sacrificed by using a larger evaluation set as quantified by these curves. To reach 95% confidence for candidate models with ΔACU$= 0.10$, one can trade close to -0.20 (from ≈ 0.45 to ≈ 0.25) by gathering another $+40$ evaluation samples (≈ 60 to 100). Any point above or to the right of each line indicates more than 95% confidence and below the lines indicate less.

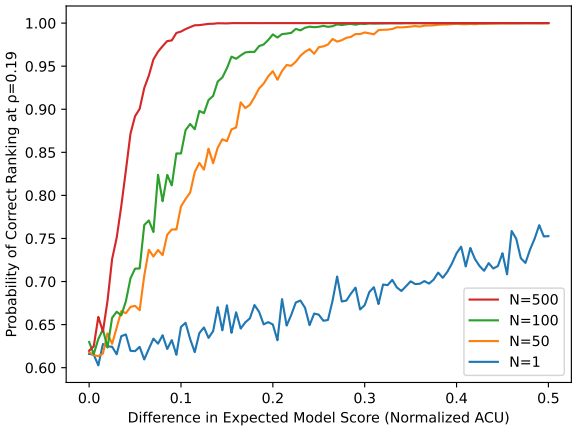


Figure 4: **Effect on confidence of increasing the evaluation set size N.** The plot shows confidence of an LLM-based automatic metric in ranking two summarization candidate models given their true expected quality differences.

As an evaluation of the simulations estimates of confidence, we compare our results to empirical observations from bootstrap sampling G-EVAL-3.5 predicted scores in Figure 5. SAMSum and XSum serve as out-of-domain test sets given that we trained our simulation on the validation set of CNNDM.

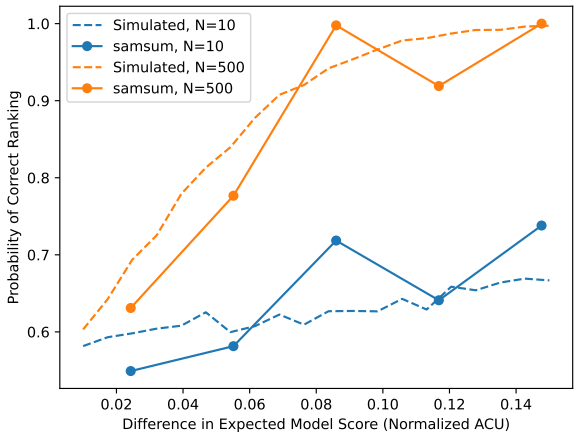


Figure 5: **Simulated vs Empirical results on SAMSum dataset.** Each point shows the probability of correctly ranking two candidate models using an LLM-based automatic evaluation metric. Generally, the empirical observations seem to match the simulated results well, although there is larger variance in the empirical results than the simulated, again highlighting the difficulty of the task. SAMSum serves as an out-of-domain test set since the monte carlo simulation was configured with data from the CNNDM validation set only, including the correlation value. Predictions by the G-EVAL were sampled with replacement and final datapoints were averaged into buckets of true ACU differences between the models being ranked.

4 Related Work

Automatic evaluation metrics are well established entities of NLP have been the subject of substantial research efforts. While many historical automatic metrics have been based on reference texts, some work (Zouhar et al., 2023) investigates metrics that score generations without any human-written references. Rei et al. (2020) investigates the use of neural frameworks for automatic evaluation.

Owczarzak et al. (2012) investigates the accuracy of ROUGE 1 and 2 scores in comparing summarization systems. Their work tries to identify the best metrics by ignoring system comparisons which have insignificant differences in performance. Their work does not consider more modern metrics such as LLM-based evaluators, nor quantifying the confidence in a given model ranking.

Liu et al. (2023b) investigate the statistical power of their dataset as a function of sample size, and analyze how metrics like ROUGE score's power compares to their proposed metric. This is a helpful analysis of their dataset's utility and the value of ACU as a metric, which we extend by explicitly investigating how likely an LLM-based evaluation system is to correctly rank models according to this metric.

Kocmi et al. (2021) look at automatic metrics for machine translation, examining how reliable such metrics are as compared with human judgements when ranking machine translation systems in pairs.

Deutsch et al. (2021) explore how precise estimates of correlation with human judgement are and find that confidence intervals of these reported correlations are wide. Their work focuses on many classic automatic evaluation metrics such as ROUGE and QAEval, but does not include newer LLM-based automatic metrics nor investigations of using the metrics to rank candidate models.

Similarly, Zhang and Vogel (2004) build a bootstrapping method for estimating confidence intervals of BLEU/NIST scores, and describe the effect of evaluation set size and number of reference translations on the confidence intervals of system-wide BLEU scores. These works do not investigate the tradeoffs between factors influencing confidence and cost.

5 Conclusion

Our work investigates the confidence of LLM evaluators in making downstream decisions by proposing a configurable monte carlo simulation. We show that even automatic metrics with low correlation to human judgement can reach high-confidence rankings of candidate models with modest evaluation set sizes (100s of examples). We also describe the exact tradeoff curves between this correlation and evaluation set size, so that cost of running inference can be minimized without sacrificing confidence. Our methods are validated by empirical observations on RoSE.

6 Limitations

Our work assumes that the human-labeled data is perfect. This is of course, false, since any annotation procedure is bound to find error and noise. We leave it to future work to combine the investigations into annotation error and introduce this source of error into our simulations. Likewise, our work does not investigate the imperfect measurement of the correlation value. Instead, our simulation assumes that this measured value is correct and can be trusted. Combining our work with that of others[8] may therefore be particularly suitable as a first step towards trusting the final confidence values

[8]Related work is discussed in Appendix 4.

given by our simulation. Empirically, this seems to not be very important as a source of error, since our simulation still describes the proportion of correct rankings we see in the RoSE data.

In researching closed-source LLMs such as those offered by OpenAI, there is little transparency regarding training data. It is therefore difficult to assess data contamination between training and testing sets. Given the publicly claimed knowledge cutoff date of GPT-* models (OpenAI, 2023), we believe the dataset proposed by Liu et al. (2023b) is unlikely to be part of the training data, thus making ACU a strong candidate for this analysis. However, the underlying text datasets, such as CNNDM may very well be part of their training data.

Some work points out that Large Language Model-based automatic evaluation metrics may exhibit other problematic behaviors (Li et al., 2024; Stureborg et al., 2024). Further work needs to be done to investigate the implications of such issues on characterizations of their confidence, especially in adjusting for their biases.

Acknowledgements

A special thanks to Bashar Alhafni for many useful discussions on text summarization, and Shao-Heng Ko for brainstorming mathematical approaches of determining the relationship between noise and correlation. We also thank the Grammarly research team for their generous support and feedback during this project. Thank you to the anonymous reviewers for their comments.

References

Shuyang Cao and Lu Wang. 2021. Cliff: Contrastive learning for improving faithfulness and factuality in abstractive summarization.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization.

Daniel Deutsch, Rotem Dror, and Dan Roth. 2021. A statistical analysis of summarization evaluation metrics using resampling methods.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization.

Alexander R. Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. SummEval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021. Language model as an annotator: Exploring dialogpt for dialogue summarization.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*. Association for Computational Linguistics.

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. Ctrlsum: Towards generic controllable text summarization.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is ChatGPT better than human annotators? potential and limitations of ChatGPT in explaining implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*. ACM.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Dragomir Radev, Yejin Choi, and Noah A. Smith. 2022. Beam decoding with controlled patience.

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.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Zhen Li, Xiaohan Xu, Tao Shen, Can Xu, Jia-Chen Gu, and Chongyang Tao. 2024. Leveraging large language models for nlg evaluation: A survey.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. G-eval: Nlg evaluation using gpt-4 with better human alignment.

Yixin Liu, Alexander R. Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023b. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation.

Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization.

Yixin Liu, Pengfei Liu, Dragomir Radev, and Graham Neubig. 2022. Brio: Bringing order to abstractive summarization.

Zhengyuan Liu and Nancy F. Chen. 2021. Controllable neural dialogue summarization with personal named entity planning.

Ye Ma, Zixun Lan, Lu Zong, and Kaizhu Huang. 2021. Global-aware beam search for neural abstractive summarization.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization.

Shashi Narayan, Yao Zhao, Joshua Maynez, Gonçalo Simoes, Vitaly Nikolaev, and Ryan McDonald. 2021. Planning with learned entity prompts for abstractive summarization.

OpenAI. 2023. Gpt-4 technical report.

Karolina Owczarzak, John M. Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9, Montréal, Canada. Association for Computational Linguistics.

Richard Yuanzhe Pang and He He. 2021. Text generation by learning from demonstrations.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Natalie Schluter. 2017. The limits of automatic summarisation according to ROUGE. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 41–45, Valencia, Spain. Association for Computational Linguistics.

Dev Seth, Rickard Stureborg, Danish Pruthi, and Bhuwan Dhingra. 2023. Learning the legibility of visual text perturbations. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3260–3273, Dubrovnik, Croatia. Association for Computational Linguistics.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators.

Rickard Stureborg, Bhuwan Dhingra, and Jun Yang. 2023. Interface design for crowdsourcing hierarchical multi-label text annotations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study.

Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong. 2021. Controllable abstractive dialogue summarization with sketch supervision.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Ying Zhang and Stephan Vogel. 2004. Measuring confidence intervals for the machine translation evaluation metrics. In *Proceedings of the 10th Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Baltimore, Maryland.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching.

Vilém Zouhar, Shehzaad Dhuliawala, Wangchunshu Zhou, Nico Daheim, Tom Kocmi, Yuchen Eleanor Jiang, and Mrinmaya Sachan. 2023. Poor man's quality estimation: Predicting reference-based MT metrics without the reference. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1311–1325, Dubrovnik, Croatia. Association for Computational Linguistics.

A Formal Problem Statement: Determining Confidence in Candidate Model Ranking

S_A and S_B are models that take an input prompt t and produces a response text. A and B are sets of generated responses a_i and b_i drawn from models S_A and S_B, respectively. These responses are generated using the same set T of prompts for each of the models. Thus, $|T| = |A| = |B|$. $Q(X)$—referred to as an LLM evaluator—is a scoring function that aggregates the scores across the responses $x_i \in X$. Q takes in a set X and returns a single score. Note, this formulation is aligned with already existing automatic evaluation metrics, as opposed to directly comparing the preference between two responses y_i and x_i from models S_X and S_Y. The benefit of this is that the evaluation can be carried out in parallel for several models, and comparisons can be made between models at any time later on.

For our experiments, Q is the mean individual score assigned to each response in the set X as determined by the automatic evaluation metric (described in §2.1). Our decision of which model is stronger is determined by comparing $Q(A)$ and $Q(B)$. If $Q(A) > Q(B)$, we say that S_A is a higher-quality summarizer than S_B as determined by our auto-evaluator. We will often refer to the size of the two sets A and B as N. In such cases we define that $N = |A| = |B|$.

We are then interested in estimating the probability that Q will correctly choose the better model between S_A and S_B. This will depend on factors such as Q's performance (correlation with human judgement) and the size N of the evaluation set, further described in §2.2.

B Simulation Algorithm

From the assumptions above, we describe Algorithm 1 to configure and run the monte carlo simulation using the initial training dataset of human-assigned scores H to a set of summaries. We set up our algorithm by defining the size N of the validation set that the simulated LLM evaluator will use, the ρ^* that we are interested in (potentially the measured correlation of an automatic metric we are investigating), and the range of differences in Summary model qualities Δ_{S_a,S_b} we want to investigate (as defined by expected ACU score).

Algorithm 1 Configure and Run the Simulation

$N \leftarrow 100$ ▷ Choose an evaluation set size
$\rho^* \leftarrow 0.19$ ▷ Choose correlation of interest
$\Delta_{S_a,S_b} \leftarrow \{0.01, 0.02, ..., 1.00\}$
$Q_\sigma(x) = x + \mathcal{N}(0, \sigma^2)$ ▷ Noisy estimator[4]

$f(x) \leftarrow \frac{1}{N}\sum_{i=1}^{N} K(x - h_i)$ ▷ KDE[9]
$\hat{\sigma} \leftarrow$ *s.t.* $\rho(H, Q_{\hat{\sigma}}(H)) = \rho^*$ ▷ Note[10]

for $\delta \in \Delta_{S_a,S_b}$ **do** ▷ avg quality difference δ
  for *number of model pair trials* **do**
    $A = \{x \,|\, x \sim f(x), \quad |A| = N\}$
    $B = \{x \,|\, x \sim f(x - \delta), |B| = N\}$
    for *number of evaluation trials* **do**
      Compute $Q_{\hat{\sigma}}(A), Q_{\hat{\sigma}}(B)$
      Record Δ_{S_a,S_b}
      Record mean scores by $Q_{\hat{\sigma}}$
      Determine model ranking
      Record if correct or not
    end for
  end for
end for

In total, $20,000$ samples are simulated ($M = 100$ generated model pairs $\cdot$ $V = 200$ generated evaluations) for each combination of N, ρ, and Δ_{S_a,S_b}. From these, the probability of correct decision (confidence) is calculated as the total number of correct decisions made divided by all samples generated:

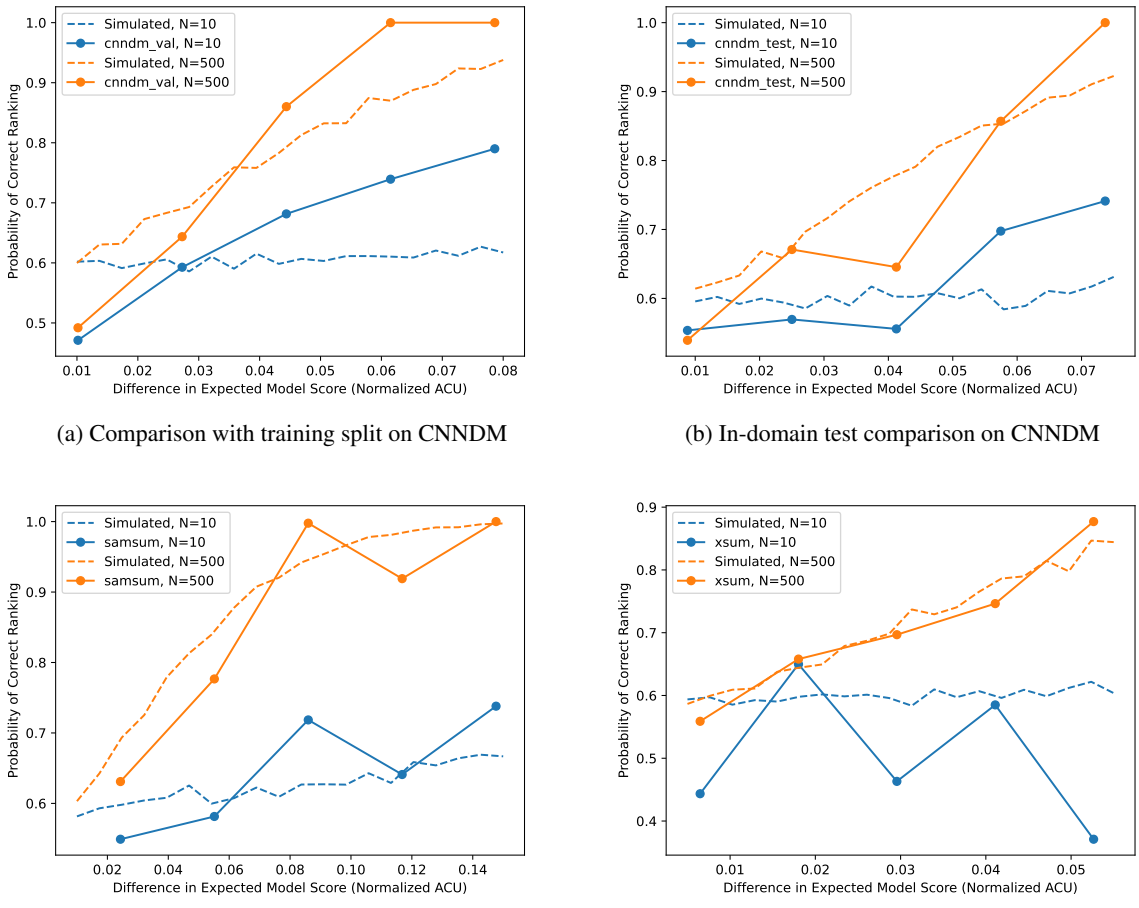
$$\frac{1}{MV}\sum_{i=1}^{M}\sum_{\cdot=1}^{V}\mathbb{1}[(Q_{\hat{\sigma}}(A_i) < Q_{\hat{\sigma}}(B_i)) = (0 < \delta)]$$

Note that this simulation models both aleatoric and epistemic uncertainties. Aleatoric (statistical) uncertainty is modeled by the selection of N true examples, while epistemic (systematic) uncertainty is modeled by the error introduced by an imperfect automatic metric.

C Additional Results

C.1 Comparing Empirical results with Simulated results based on Correlation-to-Noise Mapping Using Method 2

Results for Method 2 (from E of mapping from a correlation level to noise is given below. In this method, the mapping is explicitly calculated and no additional information from the training dataset is used, which yields worse results.



(a) Comparison with training split on CNNDM

(b) In-domain test comparison on CNNDM

(c) Out-of-domain test comparison on SAMSum

(d) Out-of-domain test comparison on XSum

Figure 6: **Comparing In- and Out-of-domain empirical results against the simulated results.** All simulations are based on $\rho = 0.19$, as this was the level of correlation G-EVAL-3.5 had on our training split (`cnndm_val`, top left).

[10]Find details on Kernel Density Estimate in Appendix F

[10]The determination of the appropriate value $\hat{\sigma}$ of the automatic evaluator to achieve a correlation of ρ^* is discussed in Appendix E

C.2 Correlation with Human Judgment ρ versus Confidence in Model Rankings

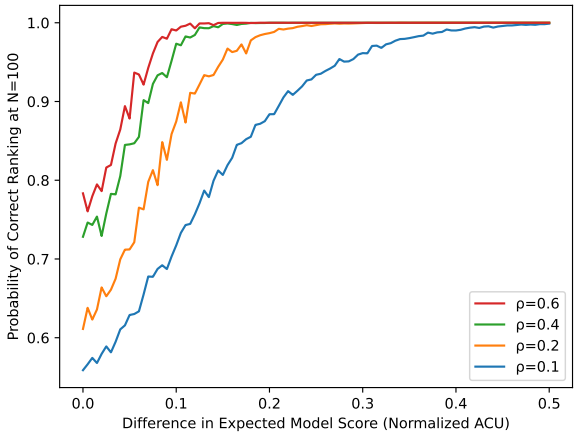


Figure 7: **Effect on confidence of increasing the correlation with human judgement ρ.** The plot shows confidence of an LLM-based automatic metric in ranking two summarization candidate models given their true expected quality differences.

C.3 Rankings Assigned by G-EVAL-3.5

Model	Human		G-EVAL-3.5	
	Rank	**ACU**	**Rank**	**ACU**
CTRLSum	1	44.6	3	3.15
GSum	2	44.5	2	3.19
BRIO	3	44.0	1	3.20
MatchSum	4	42.5	7	2.97
BRIO-Ext	5	41.7	5	3.01
SimCLS	6	40.5	4	3.06
BART	7	38.8	6	2.97
CLIFF	8	38.5	8	2.95
FROST	9	38.4	12	2.75
GOLD	10	38.1	10	2.88
PEGASUS	11	37.6	11	2.83
GLOBAL	12	36.4	9	2.88

Table 2: Mean ACU and Ranking assigned to CNNDM. G-EVAL-3.5 gives a score in the range 1-5, which is a different scale from the formulation in RoSE. We are only interested in relative comparisons. There were no ties, values are rounded.

Model	Human		G-EVAL-3.5	
	Rank	**ACU**	**Rank**	**ACU**
Ctrl-DiaSumm	1	49.0	3	3.216
MV-BART	2	47.7	2	3.226
PLM-BART	3	43.7	4	3.194
BART	4	42.9	1	3.230
CODS	5	38.4	6	2.946
PEGASUS	6	37.0	5	3.120
S-BART	7	34.6	8	2.820
UniLM	8	32.7	7	2.834

(a) Mean ACU and Ranking assigned to SAMSum

Model	Human		G-EVAL-3.5	
	Rank	**ACU**	**Rank**	**ACU**
FROST	1	27.9	1	2.799
PATIENCE	2	27.1	2	2.798
BRIO-Ctr	3	26.4	3	2.781
BRIO-Mul	4	26.3	8	2.719
CLIFF$_P$	5	25.1	5	2.760
PEGASUS	6	24.8	4	2.772
BART	7	24.0	7	2.721
CLIFF$_B$	8	22.1	6	2.739

(b) Mean ACU and Ranking assigned to XSum[11]

Table 3: Rankings on SAMSum and XSum as assigned by Human annotators in the RoSE dataset and the ACU-extended G-EVAL prompt ran through GPT-3.5 Turbo.

D Using ACU to Determine the True Human-Preferred Ranking of Candidate Models

We use the mean Atomic Content Units (ACU), as introduced by Liu et al. (2023b), in order to determine the "true" human-preferred ranking of candidate models. We prefer ACU over other metrics since it is explicitly human-labeled (as opposed to other automatic evaluation metrics like *ROUGE*) and has shown to have higher inter-annotator agreement (Liu et al., 2023b) than directly annotating for qualities such as Coherence or Relevancy. This metric has been shown to serve more reliably as the source of human annotations (Liu et al., 2023b) whereas metrics such as those introduced in Summeval (Fabbri et al., 2021) have been criticized for inconsistent annotations even among expert annotators.

D.1 ACU as a Reference-free Metric

In our experiments, we prompt G-EVAL to provide predictions on ACU without any reference summaries, which strictly differs from the original formulation of ACU. This is intentional, since the point of building an automatic evaluator is to avoid relying on human annotations.

E Determining the Noise-Level of the Noisy Estimator

The goal of the noisy estimator is to simulate the behavior of an LLM-based automatic metric. If we know the correlation the metric has with human judgements, we can work backwards to determine an appropriate noise level for the noisy estimator such that it also approximately reaches this correlation. The noisy estimator takes in the true scores, adds some gaussian noise, and returns the sum. This is repeated for every datapoint in the training set. Our noisy estimator is formally defined as

$$Q_\sigma(x) = x + \mathcal{N}(0, \sigma^2)$$

By sampling multiple values of σ and computing the resulting correlation between human-labeled scores and the predicted scores from the noisy estimator, we can describe their relationship in Figure 8.

[11] In RoSE data, BRIO-Mul is labeled 'brio'and BRIO-Ctr is 'brio-ranking'

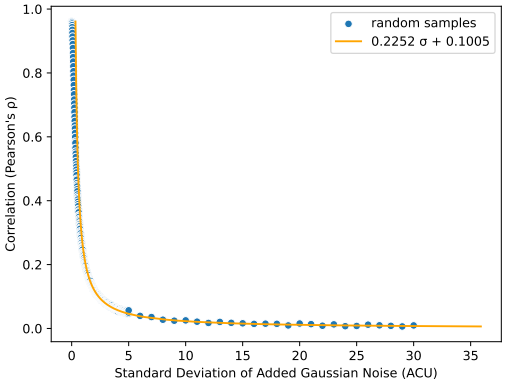


Figure 8: **Mapping between noise σ to correlation ρ as learned from the CNNDM validation set.** The relationship between Pearson correlation and gaussian noise is inverse.[**(TODO) fix the legend here, it shows a linear relationship**]

F Estimating PDFs using KDE on Training Data

We estimate the probability density functions (PDF) of true human-assigned scores in our monte carlo simulation using Kernel Density Estimation (KDE). We use a gaussian kernel. This helps us match the distribution of human-assigned scores in our simulation and ultimately influences the mapping between correlation and the level of noise that should be added (Appendix E).

G Metric Bias versus Spearman Correlation

In our experiments, we ignore the bias of our automatic evaluation metric since it does not have an impact on the overall correlation with human judgements. Below we carry out a simple analysis showing that this is the case. Here, 10,000 random values are generated as ground truth (X), and noisy estimations (Y) are produced by adding normal noise to X at a level of $\sigma = 1.0$. We then add different levels of bias to Y and calculate the resulting Spearman correlation between X and Y. The results are shown in the below table:

Bias	-100	-5	-1	-0.1	**0**	0.1	1	5	100
r	0.2855	0.2855	0.2855	0.2855	**0.2855**	0.2855	0.2855	0.2855	0.2855
ρ	0.2789	0.2789	0.2789	0.2789	**0.2789**	0.2789	0.2789	0.2789	0.2789

Table 4: Pearson's r and Spearman's ρ correlation between random values X and noisy estimations Y for different values of bias on Y.

H G-EVAL extension for Predicting ACU

Since G-EVAL was built specifically for the SummEval (Fabbri et al., 2021) attributes (Coherence, Consistency, Fluency, and Relevance), we extend these prompts to predict ACU as well. To do so, we simply copy-paste the description of what motivated the ACU from Liu et al. (2023b):

> *Salience is a desired summary quality that requires the summary to include all and only important information of the input article, [determined] by dissecting the summaries into fine-grained content units and defining the annotation task based on those units. Specifically, we introduce the Atomic Content Unit (ACU)[...], elementary information units [...] which no longer need to be further split for the purpose of reducing ambiguity in human evaluation.*

Additionally, the G-EVAL prompts explain the steps that the model should undertake to perform the evaluation. G-EVAL does not make available the prompts for generating these auto-CoT evaluation steps. Instead, we mimic this part of the prompt by paraphrasing the writing in Liu et al. (2023b) as well:

[T]he evaluation process is decomposed into two steps: (1) ACU Writing – extracting facts from one text sequence, and (2) ACU Matching – checking for the presence of the extracted facts in another sequence.

H.1 ACU Prompt

The final, zero-shot prompt used to predict ACU of a Summary given a Document is therefore:

```
You will be given one summary written for a given document.

Your task is to rate the summary on one metric.

Please make sure you read and understand these instructions carefully. Please
    keep this document open while reviewing, and refer to it as needed.

Evaluation Criteria:

ACU Salience (1-5) - a desired summary quality that requires the summary to
    include all and only important information of the input article. Salience
    can be determined by dissecting the summaries into fine-grained content
    units and defining the annotation task based on those units. Specifically,
    we introduce the Atomic Content Unit (ACU), elementary information units
    which no longer need to be further split for the purpose of reducing
    ambiguity in human evaluation. The evaluation process is decomposed into
    extracting facts from one text sequence, and checking for the presence of
    the extracted facts in another sequence.

Evaluation Steps:

1. ACU Writing - Read the document carefully and identify all Atomic Content
    Units (ACUs) and facts.
2. ACU Matching - Read the summary and compare it to the list of ACUs. Check
    what proportion of the extracted ACUs that the summary correctly covers.
3. Assign a score for ACU Salience on a scale of 1 to 5, where 1 is the lowest
    (covers very few of ACUs) and 5 is the highest (covers all important ACUs)
    based on the Evaluation Criteria.


Example:


Source Text:

{{Document}}

Summary:

{{Summary}}


Evaluation Form (scores ONLY):

- ACU Salience:
```

I Summarization Systems used as Candidate Models

The below table describes all systems used in this work and cites the original papers which introduced the methods.

System	Source
BART	Lewis et al. (2019)
BRIO	Liu et al. (2022)
BRIO-Ctr	Liu et al. (2022)
BRIO-Ext	Liu et al. (2022)
BRIO-Mul	Liu et al. (2022)
CLIFF	Cao and Wang (2021)
CLIFF$_B$	Cao and Wang (2021)
CLIFF$_P$	Cao and Wang (2021)
CODS	Wu et al. (2021)
Ctrl-DiaSumm	Liu and Chen (2021)
CTRLSum	He et al. (2020)
FROST	Narayan et al. (2021)
GLOBAL	Ma et al. (2021)
GOLD	Pang and He (2021)
GSum	Dou et al. (2021)
MatchSum	Zhong et al. (2020)
MV-BART	Chen and Yang (2020)
PATIENCE	Kasai et al. (2022)
PEGASUS	Zhang et al. (2020)
PLM-BART	Feng et al. (2021)
S-BART	Chen and Yang (2020)
SimCLS	Liu and Liu (2021)
UniLM	Dong et al. (2019)

Table 5: Summarization systems used as candidate models in our empirical experiments.