How Much Annotation is Needed to Compare Summarization Models?

Chantal Shaib[1*] **Joe Barrow**[3*] **Alexa F. Siu**[2]
Byron C. Wallace[1] **Ani Nenkova**[2]
[1]Northeastern University, [2]Adobe Research, [3]Pattern Data
{shaib.c, b.wallace}@northeastern.edu
{asiu, nenkova}@adobe.com
joe.barrow@patterndataworks.com

Abstract

Modern instruction-tuned models have become highly capable in text generation tasks such as summarization. Given the regularity with which new model variants are now released, an increasingly practical problem entails choosing the best (zero-shot) summarization model for a particular domain confidently, but with minimal effort. In this work we empirically investigate the test sample size necessary to select a preferred model in the context of news summarization. Our results reveal that comparative evaluation converges quickly for both automatic and human evaluation, with clear preferences for a system emerging from under 100 examples. Collected human preference data allows us to quantify how well automatic scores can reproduce preference rankings across a variety of downstream summarization tasks. We find that while automatic metrics are stable at smaller sample sizes, only some automatic metrics are able to moderately predict model win rates according to human preference.

1 Introduction

Instruction fine-tuned language models are highly capable summarizers, and new such models are now released often. Continuously comparing such models using large, reference-based benchmark assessments is a costly task, especially if one wants to use them in a new domain. Here we demonstrate on (English) new summarization data that—with respect to both human and automatic evaluations—preferences toward a summarization model emerge over test sets of about 50 samples. Collecting human judgements, GPT evaluations, or (if possible) manually composed references for this size dataset is reasonable. Further, we evaluate GPT evaluations and two popular reference-based evaluations, ROUGE-1 and BERTScore, in terms of their ability to predict human preferences on a set of 36 testing contexts. We collect human judgements in the context of three different summarization tasks and three sources of input. For these variations, we compute the accuracy of automated scores to reproduce human preferences between pairs of systems.

2 Background

Our goal is to establish the amount of test data needed to decide which of two summarization models produces better summaries for a given distribution over inputs (i.e., different sources of text to be summarized) and different task contexts for which the summary is to be used.

It is common to approach evaluation as a rate-then-compare task in which outputs from systems are rated for quality on a scale, and then average scores are used to compare systems. But it is well known that inputs may differ considerably in difficulty (Nenkova and Louis, 2008). Paired tests for statistical significance, that evaluate the differences of scores between two systems on the same input is the basis for comparison are therefore more appropriate (Rankel et al., 2011; Dror et al., 2018). Most contemporary work has embraced this approach, largely abandoning scoring of outputs and instead soliciting preferences among two or more choices (Novikova et al., 2018). Given developments in LLMs, pairwise win rates have become the *de facto* standard for reporting comparisons between instruction tuned models. In this work we similarly adopt win rate to compare systems, and we empirically identify the smallest test set size that reliably reveals preferences.

Most closely related to our work is the study on estimating power of tests for statistical significance, i.e., the minimum test size necessary to detect statistical differences of a given size (Card et al., 2020). Our work is aligned with the main question of this prior work, but we present empirical estimates of differences between systems without making any assumptions of tests to be used or size of effect we

[*]Work completed while at Adobe Research.

Proceedings of the Third Workshop on Bridging Human–Computer Interaction and Natural Language Processing (3rd HCI+NLP Workshop), pages 51–59
June 21, 2024

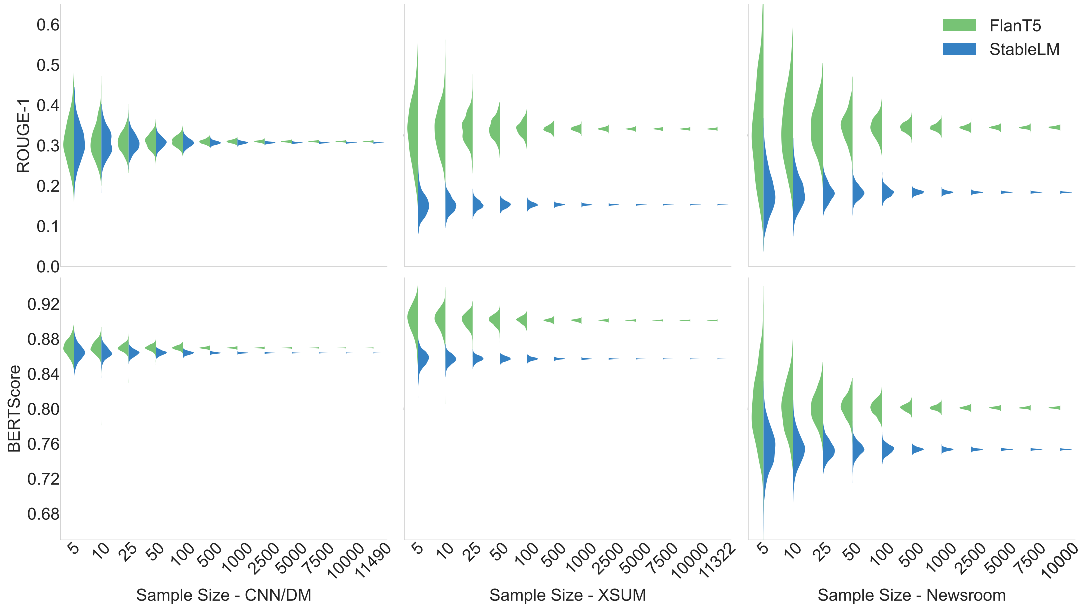


Figure 1: Distributions of average ROUGE-1 and BERTScores across 1000 re-samples. Differences between systems emerge clearly and quickly for XSUM and Newsroom.

want to detect. Our empirical findings may inform future work on power estimation.

Prior related work proposes ways of carrying out evaluations, either automatically or manually (Laban et al., 2022a; Zhang* et al., 2020; Fabbri et al., 2022; Zhong et al., 2022; Liu et al., 2022), and measuring the correlations between system rankings produced by human and automatic evaluations on a given benchmark (Gehrmann et al., 2023). We do not propose new evaluation methods, but rather introduce a method for validating automatic evaluations that does not rely on a benchmark, and instead measures the accuracy of automatic scores in reproducing human judgements across different input distributions and intended use-cases.

3 Unnecessarily Large Benchmarks

We first compare two models, FlanT5-XXL (Chung et al., 2022) and StableLM (Andonian et al., 2021) via automatic scores over three news summarization benchmarks: CNN/DM (See et al., 2017; Hermann et al., 2015), XSUM (Narayan et al., 2018), and Newsroom (Grusky et al., 2018). We use the test set splits of these datasets from Huggingface.[1]

CNN/Daily Mail and XSUM contain about 10K test inputs. The Newsroom test set split has over 100k samples. For efficiency, we randomly sample 10k examples from this set to scale it down to a size comparable to the other two datasets. We then generate summaries with FlanT5 and StableLM for all articles in the test sets, using the summarization prompts that these models have been trained on (see Appendix A). For each test split we sample 1000 times with replacement smaller test set sizes ranging from $[5, \texttt{len}(dataset)]$. We evaluate the two models with the commonly used ROUGE-1 (Lin, 2004) and BERTScore (Zhang* et al., 2020).[2] Both scores compare a summary with a human-written reference summary. ROUGE does so using tokens, while BERTScore relies on embeddings. We show score variations for FlanT5 and StableLM across the three datasets in Figure 1. For all three datasets, a preference for one of the models emerges early: The winning model as scored over 10k test points emerges after just 25-50 samples.

Given these findings, we collect human judgements on 100 samples from each of the data sources, varying the task context in which the judgement is made. We also add GPT-4 as another summarization model to be evaluated, and later report the accuracy of GPT-based evaluation against the aggregated human judgements.

4 Human Preferences

We hire three individuals on Upwork (Appendix F) for CNN/DM and Newsroom, and one for XSUM. We select 100 inputs for annotation from each dataset, which given the trends we observed in

[1] https://huggingface.co/docs/datasets/index

[2] We also report BLEU (Papineni et al., 2002) and SummaC-ZS (Laban et al., 2022b), in Appendix B.

the previous section, would be sufficient to reveal human preference.[3]

We also add summaries produced by GPT-4 for evaluation on the smaller dataset. FlanT5, StableLM, and GPT-4 represent encoder-decoder, decoder-only (open-source), and decoder-only (closed-source) models, respectively.

We instruct annotators to rank the summaries for each input in order of preference. This is a typical evaluation setting in which win rates—the percentage of input for which the model was preferred over the other—provide the clearest score for each model pair.

We provide three different scenarios to measure how preference may change based on context: *(i)* Rank the summaries in order of preference; *(ii)* Assuming you are monitoring the news for important world events, rank the summaries in order of preference; *(iii)* Which summary best captures the main details of the event being reported on? *(iv)* Which summary contains the fewest unnecessary details?

For GPT-4, we append the summaries with the instructions and provide these as prompts to the model.

4.1 Stability of Preference

First, we look to confirm whether smaller test samples are sufficient to make the same conclusion as with a larger sample. We apply the same procedure described in Section 3, where we resample 1000 test sets of size 25 and 50 from the 100 for which we have human judgements. Figure 2 shows the win rates for the CNN/Daily Mail test set for each of the three pairs of models, on the full test set of 100 samples, as well as the min, max and average win rate recorded across the 1000 smaller test sets.

While there is some variation in the strength of the preference for a model, the overall preference is preserved in the smaller samples. In only one case—the comparison between FlanT5 and StableLM—does the overall preference change for the minimum value of win rates from the one thousand samples of size 25. With 50 samples in the evaluation set, all three of the minimum, maximum and average win rates lead to the same conclusion about which system in the pair is better as that from the full 100 sample test set.

Similarly for the other two datasets, Newsroom and XSUM, none of the overall preferences flip for test sets of size 50 and only one minimum value for the 25 samples flips the preference. We provide the complete tables in Appendix C.

These results indicate that even under human evaluation, smaller test set samples (n=50) are adequate to conclude which is the preferred summarization model.

In many cases, the strength of the preference may be of interest. As shown in the variation between the minimum and maximum win rates, the strength as captured by win rates can vary considerably depending on the test set. We leave for future work analysis of the test size required to obtain reliable conclusions about the strength of the preference.

4.2 Human Preference Varies by Task and Input Source

We now turn to comparing model preferences relative to downstream task use.

Figure 3 shows the variation of aggregated preferences on the full 100 sample test set for CNN/Daily Mail. The context of the task can dramatically change the win rates for a given model. When contextualized in a specific use-case, human preferences flip from the overall rating for two out of the three model comparisons.

The overall win rate for StableLM over FlanT5 is 54%, indicating a weak preference for StableLM. In the world event use case however, the win-rate for FlanT5 increases to 53%, flipping to a preference for FlanT5. Similarly, the win rate of StableLM over GPT-4 in the overall condition is 21% but flips to 76% in the main details setting. The win rates of FlanT5 over GPT-4 remain stable across all tasks, always in favor of GPT-4.

Similarly, win rates according to the aggregate human preference for two systems vary depending with the source of data. In the next section we discuss how this observed variability changes the approach to validation of automatic evaluations.

5 Validating Automatic Evaluation

We presented qualitative evidence that the context in which preferences are made change the human preferences dramatically. We also provided clear examples of cases when human preference for the same two models can flip depending on the context. This judgement variability poses a novel requirement for validating automatic evaluation approaches. We cannot combine win rates across settings and compute correlations between human preferences and automatic scores because these

[3] See Appendix F for details about cost and hours for all annotations.

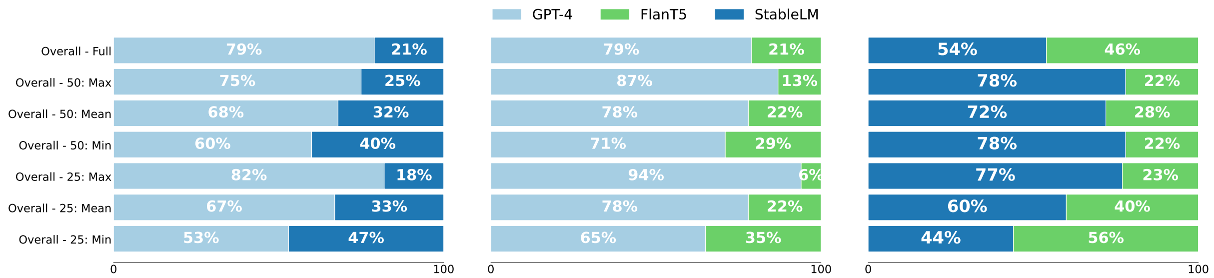


Figure 2: Aggregated annotator win rates for the CNN/DM dataset for the overall metric. Model preferences remain fairly stable across all sample sizes except in one case for sample size of 25.

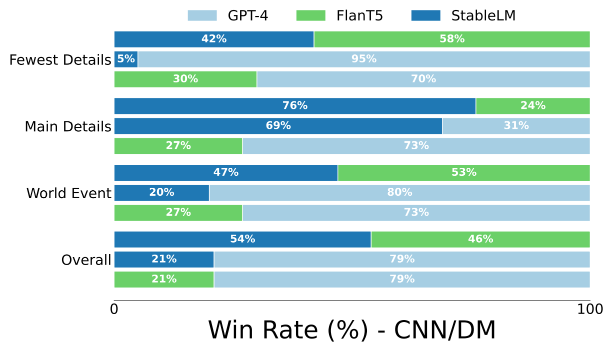


Figure 3: Aggregated annotator win rates across all metrics. Model preferences can change depending on the task setting.

come from different distributions. We do, however, have a sufficient number of pairs for comparison: 3 models evaluated on 3 sources of data, on 4 context of use. This yields 9 overall preferences and 27 contextually dependent preferences.

For four automatic methods for evaluation, we compute the accuracy of the automatic score in reproducing human preferences. Specifically, we compute the percentage of pairwise comparisons for which the automatic evaluation agrees with the human win rates on which system is the better one. This is a coarse requirement because it does not capture the size of the win rate. For example the win rate of one system over another in human preferences is 51% but an automatic score predicts that its win rate is 79%, the automatic score will be considered accurate.

Table 1 shows the accuracy for four automatic evaluations: ROUGE-1, BERTScore, G-Eval, and GPT-4 as an annotator. In the case of GPT-4 as an annotator, we provide GPT-4 with the exact same instructions as the human annotators. For the first three approaches, a win for a model is declared if the score assigned by the method for this input is higher than that for the other model. In cases

Metric	Accuracy (%)
ROUGE-1	78
BERTScore	56
G-Eval	44
GPT-4 (as annotator)	78

Table 1: Accuracy of automatic metrics compared to human evaluations. GPT-4 as-an-annotator and ROUGE-1 score have the highest accuracy in predicting which model is selected by human annotators in each task setting.

when the scores for an input are the same, there is a tie. In the fourth case, using GPT-4 as an annotator provides ratings, so the wins are decided by the ranking returned by GPT-4 (rather than a proxy score). In this case, there are no ties because the annotators were asked to do a forced choice comparison. We find that ROUGE-1 and GPT-4 as an annotator are able to moderately predict the aggregated human preferences across the different tasks, compared to BERTScore and G-Eval which are not able to do so as reliably.

6 Conclusions

We presented automatic and human evaluations designed to establish the minimum amount of data necessary to choose between contemporary summarization models. Comparative evaluations establish which model performs better with test sets of 50 inputs. For human evaluation, a test size of 50 is sufficient to confidently establish which of two models people prefer. Human preference varies, however, depending on the intended use of the summary and on the source of data for summarization. This variation calls for new methods for validating automatic scores. We find that all four automatic evaluations predict preferences better than chance but lead to erroneous conclusions for many pairwise comparisons.

Limitations

We only evaluate over benchmark news datasets, where it is possible that our observations may not be reflected in other, more niche domains. In part, this choice is due to lack of availability of quality summarization datasets with references (and further motivating the need for evaluation over small samples), however it is important for future work to consider more specialized cases. Another limitation is that we do not collect human annotations nor GPT-4 summaries over the entire test set splits. This poses a challenge as collecting these evaluations and summaries over such a big dataset is costly.

References

Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Shivanshu Purohit, Tri Songz, Wang Phil, and Samuel Weinbach. 2021. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch.

Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With Little Power Comes Great Responsibility. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9263–9274.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Sebastian Gehrmann, Elizabeth Clark, and Thibault Sellam. 2023. Repairing the Cracked Foundation: A Survey of Obstacles in Evaluation Practices for Generated Text. *Journal of Artificial Intelligence Research*, 77:103–166.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2022a. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022b. SummaC: Re-Visiting NLI-based Models for Inconsistency Detection in Summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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. 2022. Revisiting the Gold Standard: Grounding Summarization Evaluation with Robust Human Evaluation. *arXiv*.

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. *ArXiv*, abs/1808.08745.

Ani Nenkova and Annie Louis. 2008. Can you summarize this? identifying correlates of input difficulty for multi-document summarization. In *Proceedings of ACL-08: HLT*, pages 825–833, Columbus, Ohio. Association for Computational Linguistics.

Jekaterina Novikova, Ondrej Dusek, and Verena Rieser. 2018. Rankme: Reliable human ratings for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 72–78. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Peter Rankel, John Conroy, Eric Slud, and Dianne O'Leary. 2011. Ranking human and machine summarization systems. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 467–473, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Model	Prompt
FlanT5	`[TEXT]`\nWhat is a one-paragraph summary of the above article?
StableLM	<|SYSTEM|># StableLM Tuned (Alpha version) - StableLM is a helpful and harmless open-source AI language model developed by StabilityAI. - StableLM is able to facilitate human communication by providing a summary of a given text. - StableLM is able to provide summaries that are useful and relevant to the given text. <|USER|> `[TEXT]`. Summarize the given piece of text. <|ASSISTANT|>
GPT-4	"role": "user", "content": "`[TEXT]` \n\n Summarize the above text. \n\n"

Table 2: Input and prompt structure for each summarization model. `[TEXT]` is replaced with the article to be summarized.

Appendix

A Summarization Prompt Details

For the summarization prompts, we use prompts and input structures that the models have been trained on. Table 2 shows the input for each model, where `[TEXT]` is replaced with the article to be summarized.

B BLEU and SummaC-ZS

Figure 4 shows the distributions of averaged BLEU and SummaC-ZS scores over all three datasets. BLEU scores have trouble capturing meaningful scores across longer inputs as seen with StableLM. SummaC-ZS uses NLI-models to score sentence-level information – similar to ROUGE-1 and BERTScore, we can start differentiating models earlier than the full sample size.

C Human Evaluation Win Rates and Sample Sizes: XSUM and Newsroom

We provide the aggregated win rates across annotators for XSUM (Figure 5) and Newsroom (Figure 6). Both datasets show the same trend as in Figure 2, where the win rate pair ranking is preserved in the minimum, maximum, and average win rates across 1000 trials. This holds across sample sizes of 50, but not in *all* cases with sample size of 25.

D Human Evaluation Win Rates and Tasks: XSUM and Newsroom

Similar to Figure 3, we show the win rates across different tasks for XSUM and Newsroom in Figure 7. These results support the finding that preference changes between downstream scenarios.

CNN/DM		
Annotators	Factuality κ	Text Quality κ
1, 2	0.522	0.053
1, 3	0.249	0.539
2, 3	0.133	-0.081

Table 3: Agreement scores, Cohen's kappa.

E Annotator Agreement on Text Quality and Factuality

For CNN/DM we report the agreement scores over factuality and text quality questions that we collect in our surveys in Table 3. We expect the agreement scores for factuality to be much higher; it is possible that this is an indicator for different tolerance for minor errors (e.g., vague wording) or may be indicative of the cognitive load involved in judging factuality. Similarly for text quality, the threshold for artifacts or other issues may differ between annotators.

F Annotation Details

Costs We hired seven professional proofreaders from Upwork, who were each recruited to read 100 articles and rank 3 summaries per article. We paid each annotator a flat fee of $325 to evaluate the summaries When asked for a time estimate after they completed, responses ranged between 10 and 13 hours to complete the study, meaning annotators were compensated at roughly $25-$30 per hour. The annotators typically completed the work over one to three days.

Annotation Platform We hire annotators on Upwork[4]. We presented the annotators with a custom interface for ranking the summaries and answering questions, shown in Figure 8. Annotators were encouraged to take extended breaks during annotation to reduce task fatigue.

[4] `https://www.upwork.com/nx/enterprise-homepage/`

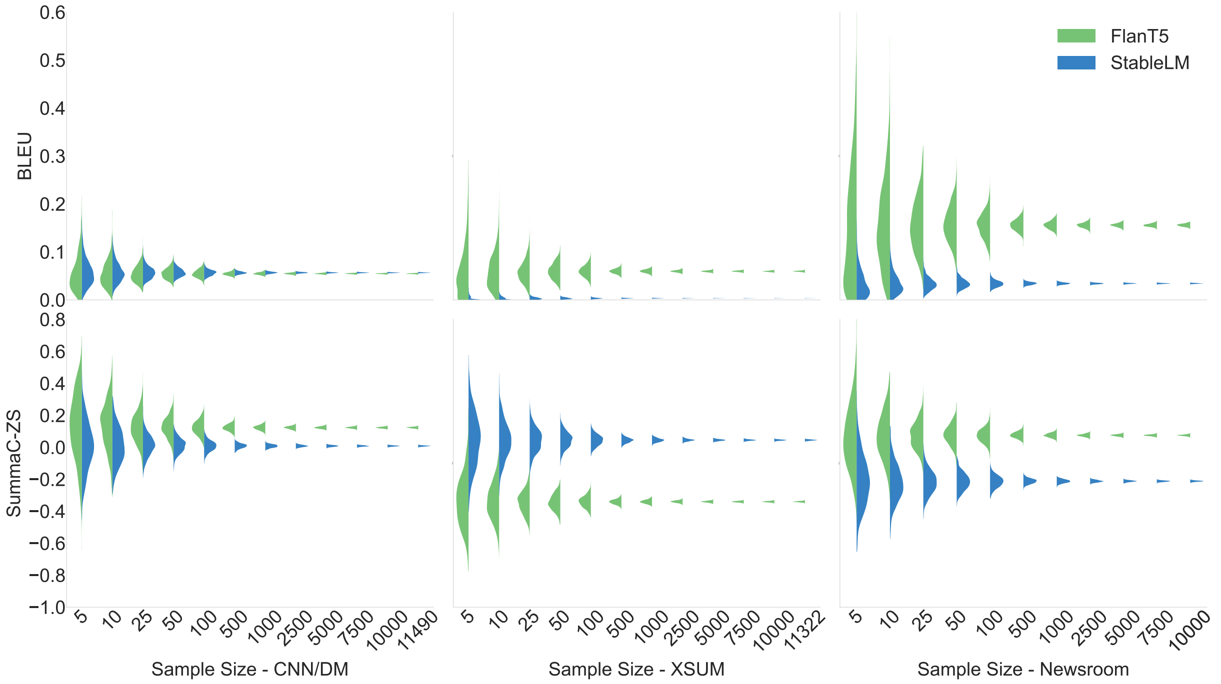


Figure 4: Distributions of averaged BLEU and SummaC-ZS scores across 1000 re-samples for CNN/DM, XSUM, and Newsroom.

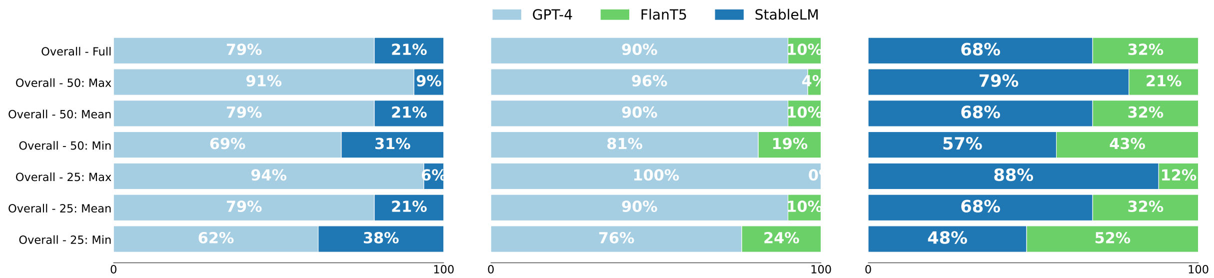


Figure 5: Win rates aggregated by annotators (XSUM).

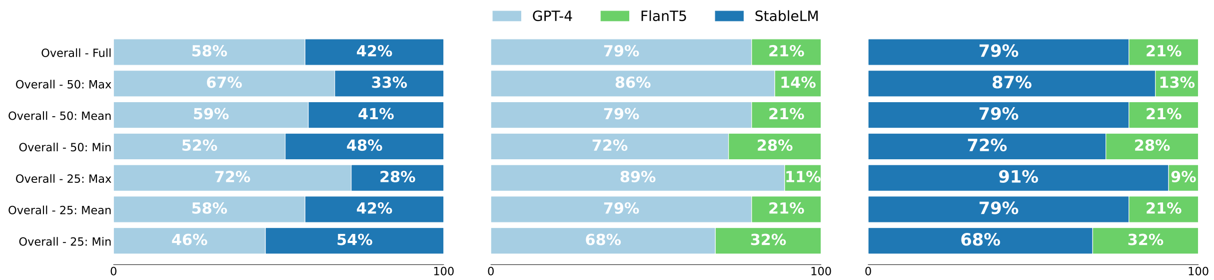


Figure 6: Win rates aggregated by annotators (Newsroom).

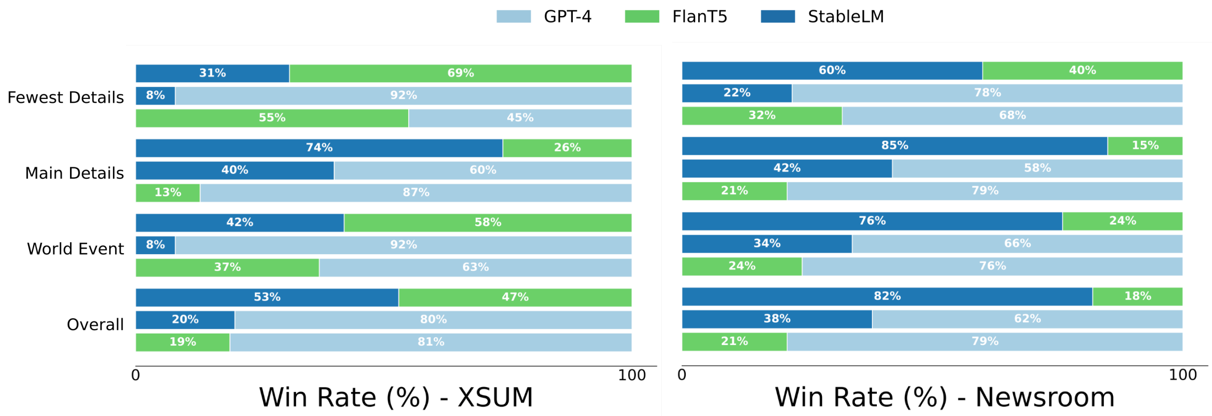
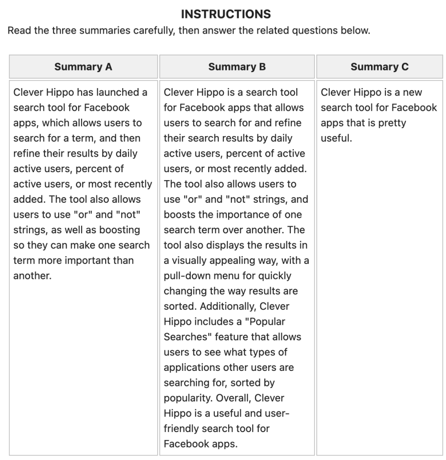
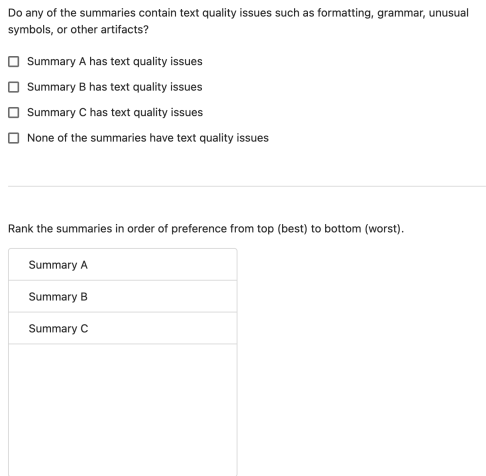


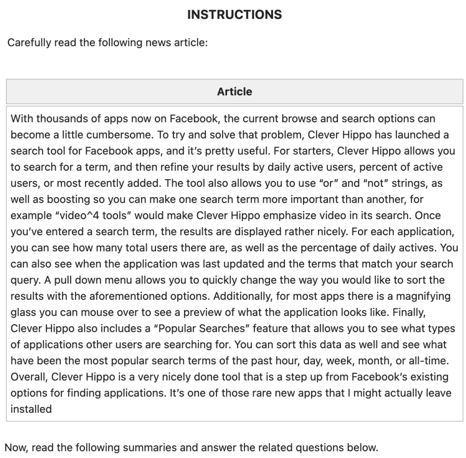
Figure 7: Aggregated annotator win rates across all metrics over the XSUM and Newsroom datasets.



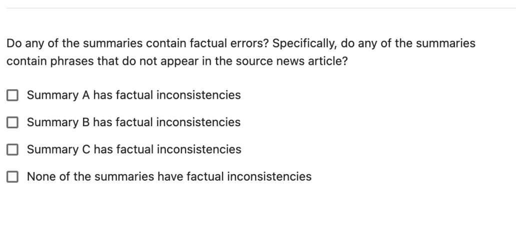
(a) Summaries, as presented to the annotators.



(b) Text quality issues and the ranking interface for the summaries. Each box with the summary label can be dragged-and-dropped into any order.



(c) Article, as presented to the annotator.



(d) Factuality questions asked about each summary.

Figure 8: The annotation interface. For each article, annotation happens across two pages. The first page contains the summaries (8a) and rankings (8b), and the second page contains the article (8c) and factuality questions (8d).