On the Blind Spots of Model-Based Evaluation Metrics for Text Generation

Tianxing He* Univ. of Washington goosehe@cs.w*.edu

Jingyu Zhang* Johns Hopkins Univ. jzhan237@jhu.edu

Tianle Wang Shanghai Jiao Tong Univ. wtl666wtl@sjtu.edu.cn

Sachin Kumar Carnegie Mellon Univ. sachink@cs.cmu.edu

Kyunghyun Cho New York Univ. kyunghyun.cho@nyu.edu

James Glass Mass. Institute of Technology glass@mit.edu

Yulia Tsvetkov Univ. of Washington yuliats@cs.washington.edu

Abstract

In this work, we explore a useful but often neglected methodology for robustness analysis of text generation evaluation metrics: stress tests with synthetic data. Basically, we design and synthesize a wide range of potential errors and check whether they result in a commensurate drop in the metric scores. We examine a range of recently proposed evaluation metrics based on pretrained language models, for the tasks of open-ended generation, translation, and summarization. Our experiments reveal interesting insensitivities, biases, or even loopholes in existing metrics. For example, we find that BERTScore ignores truncation errors in summarization, and MAUVE (built on top of GPT-2) is insensitive to errors at the beginning of generations. Further, we investigate the reasons behind these blind spots and suggest practical workarounds for a more reliable evaluation of text generation.[1]

1 Introduction

Automatic evaluation of machine-generated text (Celikyilmaz et al., 2020) has been a core research challenge in the field of natural language generation (NLG), as difficult as language generation itself. Encouraged by the phenomenal success of large-scale pretraining (Devlin et al., 2019), a recent series of work has proposed to base evaluation metrics on pretrained language models (PLMs) (Zhang et al., 2020; Yuan et al., 2021; Pillutla et al., 2021). For example, unlike the traditional BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) metrics measuring token overlap between the reference and hypothesis, BERTScore (Zhang et al., 2020) computes a similarity score between the contextualized embeddings of the hypothesis and the reference texts. PLM-based metrics have been shown to have higher correlations with human annotations for various tasks (Yuan et al., 2021), and are becoming increasingly popular in practice.

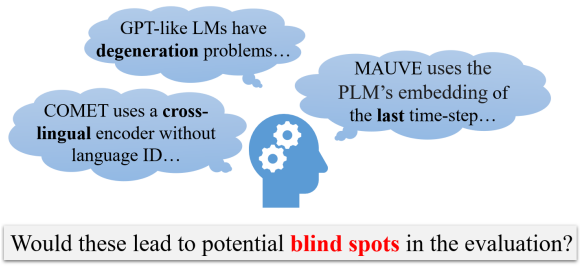


Figure 1: Motivation: The flaws of the underlying PLMs or certain design choices in the metrics could lead to potential blind spots in the evaluation. The goal of this work is to test for such blind spots.

However, PLMs have flaws. They could assign a high likelihood to degenerate, repetitive text (Holtzman et al., 2020) and could be insensitive to perturbations such as word order shuffling (Pham et al., 2021), negation (Ettinger, 2020), etc. These flaws, in combination with certain design choices, may lead to the downstream metrics based on such PLMs becoming brittle and open to manipulation (Figure 1).

To quantify this phenomenon, we develop a suite of *stress tests with synthetic data* for the robustness analysis of NLG metrics. In essence, we induce a variety of potential errors in clean text and measure the resulting drop in the metric scores. We cover a range of recently proposed and widely used PLM-based metrics for the tasks of open-ended generation, translation, and summarization. Our methodology facilitates full control over the synthesized error types, allowing us to test extreme or even adversarial scenarios that are not covered in the standard correlation-oriented evaluations (thus the name "stress test").

Our experiments reveal a number of glaring insensitivities, biases, and even loopholes in different metrics. Towards addressing these issues, we also

* Equal contribution. Both are corresponding authors. w* in the email refers to washington.

[1]Preprint. We will release our code and data at https://github.com/cloudygoose/blindspot_nlg by March 2023.

provide practical suggestions and workarounds for metric design choices as well as how the metrics should be reported.

Overall, we recommend that PLM-based metrics should not be used in isolation for evaluation. Rather, a combination of them should be reported so that they can cover each other's blind spots. Some prior works have alluded to some of these issues and even found workarounds to address them (Holtzman et al., 2020; Khandelwal et al., 2018). However, a large body of work on text generation uses these metrics off-the-shelf without considering their limitations. With this systematic analysis, we hope to raise awareness among researchers about the existence of these blind spots, and to exercise caution for more robust and reliable metric usage or development.

2 Methodology

For simplicity, in this section, let us assume a multi-reference translation dataset, where each sample has two reference translations produced by human translators, denoted *Ref-A* and *Ref-B*. We will generalize our methodology to other tasks in §3.

We begin by computing a "base" metric score by considering Ref-A as hypotheses and Ref-B as references. Since Ref-A is produced by human translators, we assume that it is less likely to contain translation errors than machine-generated text, and it should be assigned a high score by the metric. Due to these two assumptions, and to disambiguate from the reference set (Ref-B), we term it the *gold hypothesis* set.

For each test, we apply a synthesized error type (e.g., repetition or truncation) to the gold hypothesis set to construct a *noised hypothesis set*. We make sure that the amount or type of induced errors is sufficient to be distinctive from the original gold hypothesis (to be detailed in §5). The source texts and the references are left intact.

To determine whether a metric passes a test, a simple rank-based protocol is used: We claim that the metric *fails the test for this dataset* if the noised hypothesis set is not scored worse than the base score (from the gold set).[2] This rank-based protocol can be easily extended to the comparison of different gradations of the same noise type (controlled by hyper-parameters). For example, a 20%-truncation is expected to rank lower than a 10%-truncation, as more information is lost.

[2]As we will introduce in §4, all metrics except MAUVE are sample-level, and we compare the average score assigned to the gold/noised hypothesis set.

3 Tasks and Datasets

Our tests cover three ubiquitous text-generation tasks: open-ended generation, translation, and summarization. We now describe the dataset used for each task and the setting for gold hypotheses.

For open-ended generation, we use the WikiText-103 dataset (Merity et al., 2016). We randomly sample 2000 paragraphs longer than 256 tokens and conduct preprocessing (detailed in Appendix B). The samples typically contain seven or eight sentences. We divide them into two non-overlapping sets and set one as the references and the other as the gold hypotheses with 1000 samples each.

For summarization, we use the popular CNN-Dailymail (CNNDM) dataset (Hermann et al., 2015). Kryscinski et al. (2020) collected 10 additional human-annotated summaries (different from the CNNDM reference summary) for each of 100 samples in the CNNDM test set. We set the CNNDM reference summaries to be the gold hypotheses, and use these 10 annotations as references. Correspondingly, the multi-reference version of metrics are used. The gold hypotheses typically contain three sentences.

For translation, we use the evaluation dataset from the WMT21 metrics shared task (Akhbardeh et al., 2021). We only use the source text and reference translations. We report results on the German-English (De-En) language pair, which contains 1000 translation pairs. Most of the samples only contain one sentence. There are two human annotated references (human-A and human-B) for each sample. We use human-A as the gold hypothesis and human-B as the reference. We also repeat key experiments on the Chinese-English (Zh-En) data and obtain very similar observations. Therefore, we omit the Zh-En results for brevity.

4 Metrics

For open-ended text generation, we test GPT-PPL, MLM-PPL (Salazar et al., 2020), and MAUVE (Pillutla et al., 2021). We report the negated GPT/MLM-PPL so that all metric scores are the higher the better.

GPT-PPL denotes perplexity from the GPT2-large (Radford et al., 2019a) model. MLM-PPL is the masked language model perplexity from a RoBERTa-large model (Liu et al., 2019). We use

Blind Spot	Section	Affected Metrics
distant error	§5.1	MAUVE (-GPT2)
copy-source trick	§5.2	COMET-QE, BARTScore (-r/-f/-faithful), BERTScore (-r)
repetition	§5.3	GPT-PPL, MLM-PPL, BARTScore (all variants)
high-freq n-gram	§5.3	GPT-PPL, MLM-PPL
self-generation	§5.4	GPT-PPL, BARTScore (-faithful)
truncation	§5.5	BERTSc(-p/-f),BARTSc(-p/-f/-faithful),COMET-QE,ROUGE(-2/-L),MAUVE(-GPT2)
sentence switching	§5.5	MAUVE (-GPT2/-RoBERTa), BARTScore (-r)
BERT-diverge	App. G	COMET-QE
article removal	App. G	COMET-QE
noised punctuation	App. G	BARTScore (-r), ROUGE (-2/-L)
a few other fluency errors	App. G	BARTScore (-r)

Table 1: A category of the blind spots identified in this work for PLM-based metrics (detailed in §5).

a definition similar to the formulation in Salazar et al. (2020) and provide details in Appendix A. MAUVE is a reference-based metric that measures the similarity of the reference and candidate text distributions. It is computed using contextualized embeddings from PLMs. We consider MAUVE with GPT-2 large and RoBERTa-large features (denoted as MAUVE-GPT2/RoBERTa).

For translation and summarization, we test BERTScore (Zhang et al., 2020), MoverScore (Zhao et al., 2019), BARTScore (Yuan et al., 2021), COMET (Rei et al., 2020), PRISM (Thompson and Post, 2020), BLEURT (Sellam et al., 2020). Among these metrics, PRISM and BLEURT are only applied for translation. While COMET was originally proposed for translation, Kasai et al. (2022b) showed it has superior human correlation for CNNDM. Therefore, we also include it for summarization. We also include the traditional metrics BLEU (for translation), and ROUGE-2/L (for summarization).

Both BERTScore and BARTScore have variants for precision (-p), recall (-r), and f-measure (-f). In addition, BARTScore has a faithfulness (-faithful) variant. It has two model options, and we term them BARTScore-cnn or BARTScore-para.[3] By default, the metrics for translation and summarization are reference-based.[4] COMET and PRISM have a quality estimation (QE) variant (Specia et al., 2021), where users do not need to provide any reference. We term these variants COMET-QE and PRISM-QE.

In most cases, we directly use the released package or code for each metric and follow the recommended hyper-parameter or variant setting. We defer further implementation details and variant explanations to Appendix A.

[3]For BARTS-cnn, the Bart model is finetuned on the CNNDM dataset (Hermann et al., 2015). For BARTS-para, it is further finetuned on the ParaBank2 dataset (Hu et al., 2019).

[4]There is one exception: The faithfulness variant of BARTScore does not utilize reference.

5 Stress Tests and Results

We organize our findings into subsections where each of them contains a set of tests with the corresponding motivation, description, results, and implications (with practical workarounds). While we run each test for all metrics/tasks, we primarily discuss metrics found to be problematic for brevity.

We group and order our tests by their motivations: The *distant-error* (§5.1) and *copy-source* (§5.2) tests are motivated by certain metric design choices; The *repetition/freq-ngram* (§5.3) and *self-generation* (§5.4) tests are motivated by certain PLM properties; Finally, the general *fluency/consistency* (§5.5) tests mimic errors that human or machine writers could make. See Table 1 for a detailed categorization along with the metrics affected.

5.1 The Distant Error Test

MAUVE, as a metric based on distributional distance, is computed as a function of the last hidden representations of a PLM for the hypotheses and the references. Hence, it can vulnerable if the PLM is biased to encode only the local context, which has been observed in some former analysis (Khandelwal et al., 2018).

To test for this bias, we create synthetic errors by replacing a span of 10 consecutive tokens in different positions of the gold hypothesis with some erroneous token sequences. We experiment with (1) 10 random tokens from the vocabulary (2) randomly shuffled tokens of the original span. For each error sequence type, we create three different error position types by replacing the tokens at the very start, center middle, and very end of the gold hypotheses. A robust metric should give a significantly lower score to this clearly modified

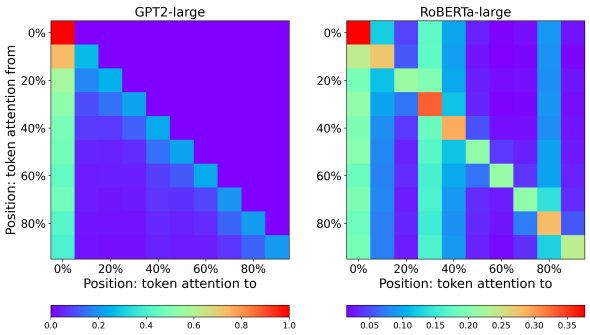


Figure 2: Attention distribution of GPT2-large and RoBERTa-large over the relative position in one data sample (averaged over transformer layers and attention heads). Each unit corresponds to 10% of tokens. More details and enlarged versions of this figure are shown in Figure 11 in Appendix C.

distribution of the hypotheses due to this obvious error.

As shown in Table 2, MAUVE-GPT2 shows only a marginal drop (less than 3%) or even an increase in scores for the shuffle errors in the start and middle positions. In comparison, MAUVE-RoBERTa penalizes errors in all positions severely and relatively equally, although we observe that the score drop is less significant for errors in the middle compared to errors in the start or end of the text.

Noise Type	MAUVE variant	
	GPT2	RoBERTa
Gold	0.957	0.979
Replace-Start	0.951 (-0.6%)	0.032 (-96.7%)
Replace-Middle	0.935 (-2.3%)	0.119 (-87.8%)
Replace-End	0.006 (-99.4%)	0.032 (-96.7%)
Shuffle-Start	0.999 (+4.4%)	0.407 (-58.5%)
Shuffle-Middle	0.999 (+4.4%)	0.787 (-19.7%)
Shuffle-End	0.020 (-97.9%)	0.315 (-67.8%)

Table 2: "Replace" indicates the token span is replaced with random tokens from the vocabulary. "Shuffle" means the tokens within the span are shuffled in-place. Results for the distant error test. MAUVE-GPT2 is insensitive to errors at the start and middle of hypotheses, while MAUVE-RoBERTa is more robust. The percentage shown is w.r.t the gold hypotheses.

We correlate this result with an *attention pattern analysis*. As shown in Figure 2, we observe that GPT2-large's attention is concentrated on the diagonal of the plot, which indicates GPT-2 mostly attends to the near history. In contrast, RoBERTa-large attends heavily to specific (probably important) token positions regardless of the current token position. In summary, the attention patterns provide evidence that GPT-2 features encode less long-range context compared to RoBERTa.[5] This pattern is typical across different data samples.

Implication Currently, the default feature used by MAUVE is from GPT-2, which as we show, ignores errors at the start or the middle of the generations. Our analysis indicates that MLMs such as RoBERTa could be a better choice (In all other tests we perform, these two feature options behave comparably). However, the distant error test is only one aspect, and a more comprehensive comparison (e.g., human correlation evaluation) beyond the scope of this work is needed.

5.2 The Copy-Source Trick

A number of metrics are based on the similarity between the hypothesis and the reference or source. Therefore, for tasks like summarization and translation, one could try to fool the metric by submitting a direct copy of the source text. We term it the copy-source trick.

Metric (task)	Gold	Copy-src
COMET-QE (wmt)	0.114	0.126
BARTSc-cnn-faithful (sum)	-1.376	-0.368
BARTSc-cnn-faithful-noavg (sum)	-82.95	-166.25
BERTScore-r (sum)	0.266	0.332
BERTScore-f (sum)	0.223	0.065

Table 3: Results of the copy-source trick. This simple trick could fool the metric and get scores higher than gold hypotheses. Some auxiliary results are in Table 11 (Appendix D).

As reported in Table 3, for both translation and summarization datasets, we find that COMET-QE, BERTScore-r, and BARTScore-cnn-faithful not just fail to account for this simple trick but in fact obtain higher scores than gold hypotheses (The case for other variants of BARTScore is similar, which is deferred to Table 11, Appendix D).

We attribute these behaviors to some of the metrics' design choices. (1) COMET-QE relies on a cross-lingual RoBERTa encoder, but it does not check the language ID of the hypothesis. (2) BARTScore, computed as a length-averaged log-likelihood, fails to account for the length of the hypothesis, which in this case is the entire source

[5] Besides this pattern, both GPT2-large and RoBERTa-large assign a large portion of attention to the very first token, which is also observed by Vig and Belinkov (2019). It is unclear to us why this happens.

article (and much longer than the desired summary). While removing the average operation is a natural remedy and indeed leads to a lower score for the noised hypothesis (shown by BARTS-cnn-noavg in the table), it is not ideal as it would also favor overly short summaries. (3) BERTScore-r's behavior on summarization, on the other hand, is not surprising since it is recall-oriented, and is alleviated by using the f-measure.

Implication The copy-source trick could be easily used to manipulate many metrics in a contest or a leaderboard. Fortunately straightforward solutions can counter this trick. For example, contest organizers can implement checks for similarity between submitted hypotheses and the source text and reject the matches. For translation, a language ID check would also be helpful. For log-probability-based metrics, it would be useful to check whether the length of the hypothesis is within the expected range.

5.3 The Repetition & Frequent N-Gram Test

It is well-known that GPT-like LMs suffer from a repetition problem—they tend to assign high likelihood to repetitive text (Holtzman et al., 2020). In addition, due to their statistical nature, they would favor frequent n-grams seen at training time. We design two tests to emulate these behaviors.

Test	Example
Rep-2	... allegiance to one's family, despite the turmoil and dissensions that occur. dissensions that occur. dissensions that occur.
Freq 4-gram	... in the middle of the site of the the course of the as part of the the top of the on the billboard hot in the summer of for the rest of

Table 4: Front-truncated examples of repetition (top) and the frequent n-gram (bottom) test on WikiText. Top-50 4-grams are used.

For the **repetition** test, we append to each gold hypothesis k copies of its last 4-gram to create a synthetic repetition problem (termed as Rep-k), with an example available in Table 4. For this test, a robust metric should give a lower score for Rep-k compared to gold, because synthetic repetition degrades quality.

The experimental results for the repetition test are shown in Table 5. The repetition problem plagues a wider range of models than expected. In addition to GPT-PPL, we find BARTScore-cnn, and MLM-PPL (based on RoBERTa) also prefer repetitive text. Other variants of BARTScore also face a similar problem and we defer the results to Table 12 (Appendix E).

Metric (task)	Gold	Repetition		
		Rep-10	Rep-20	Rep-30
BARTSc-cnn (sum)	-1.376	-1.486	-1.224	-1.091
BARTSc-cnn (wmt)	-2.168	-1.889	-1.721	-1.652
GPT-PPL (wiki)	-21.81	-15.48	-10.70	-8.080
MLM-PPL (wiki)	-2.635	-2.241	-2.019	-1.867
n-rep-4gram (wiki)	-0.007	-0.165	-0.287	-0.378

Table 5: Results for the repetition test. BARTScore-cnn, GPT-PPL, and MLM-PPL give higher scores for repetitive text. Negated rep-4gram (Welleck et al., 2020), which measures the diversity, is also reported.

For the **frequent n-gram** test, we first collect the top-k most frequent n-grams from the WikiText dataset. We then build synthetic hypotheses of length 256 by uniformly sampling n-grams from this collection and concatenating them. (see Table 4 for an example). To a human evaluator, these sequences are completely random and should get a lower score than the gold hypotheses.

Strikingly, as shown in Table 6 with 4-gram, we find that both GPT-PPL and MLM-PPL assign higher scores to the frequent n-gram sequences than gold. This gap further increases when we concentrate on more frequent n-grams. We present additional results with 3-gram in Appendix E.

Metric (task)	Gold	Freq 4-gram		
		Top-10	Top-50	Top-100
GPT-PPL (wiki)	-25.640	-4.456	-11.640	-18.160
MLM-PPL (wiki)	-2.994	-1.139	-2.469	-3.971
rep-4gram (wiki)	-0.019	-0.539	-0.199	-0.120

Table 6: Results for the frequent n-gram test. Both GPT-PPL and MLM-PPL deem the frequent 4-gram sequences as probable.

To illustrate this issue, we plot step-wise next-token probability given by the underlying GPT2-large model. As shown in Figure 3, the next-token probability exhibits a pattern that high-probability regions concentrate at the end of each 4-gram. For example, the "of" in the 4-gram "in the middle of" gets a higher probability than the first three tokens. We attribute this behavior to the LM's utilization of local context (Khandelwal et al., 2018). This pattern is similar to the 3-gram case, which is detailed in Appendix E.

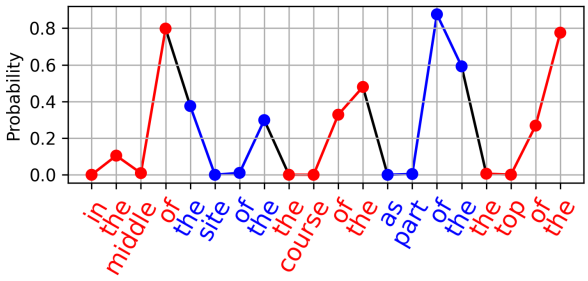


Figure 3: Step-wise next-token probability of a (partial) frequent 4-gram sequence given by GPT2-large, for GPT-PPL. The alternation between blue and red indicates the start of a new 4-gram.

Implication For metric users, it has been an established practice (especially for open-ended generation) to report diversity metrics like rep-4gram (Welleck et al., 2020) or n-gram entropy (Zhang et al., 2018), as shown in Table 5 and Table 6. But for metric developers, our results indicate that the degeneration issue can not be ignored even if the LM is not autoregressive.

5.4 The Self-Generation Bias

Log-probability-based metrics (e.g., GPT-PPL) are based on generative models such as GPT-2 (Radford et al., 2019b) or BART (Lewis et al., 2019). Meanwhile, these PLMs are also used as base models for developing new NLG systems (Yang and Klein, 2021; Beltagy et al., 2020). This leads to an interesting case in NLG: *The generative PLMs are used for both development and evaluation* (Yuan et al., 2021).

Naturally, we wonder whether this could cause some level of bias in the evaluation. In the following tests, we demonstrate this bias for the case of GPT-PPL and BARTScore.

For **GPT-PPL**, we construct a setting that mimics how it is used in practice: For the *generator*, we finetune GPT-2 models of different sizes (small, medium, and large), and use the finetuned model to generate continuations of prompts from the WikiText dataset. The details of finetuning are available in Appendix F. We use top-k sampling (Fan et al., 2018) with $k = 50$ to decode. For *evaluator*, we use GPT-2 models off-the-shelf.

For different combinations of generator and evaluator, the results are shown in Table 7. Conventional wisdom in the community is that the larger GPT model should generate higher-quality text, which correlates with the scores from the OPT-2.7b (Zhang et al., 2022) model. However, GPT2-small and medium perplexities violate these expectations, ranking generations from their own base models higher than those of larger models. We term this as *self-generation* bias.

Evaluator	Generator		
	GPT2-small wiki-ft	GPT2-med wiki-ft	GPT2-large wiki-ft
GPT2-small	-20.47	-23.98	-24.69
GPT2-med	-23.55	-16.97	-19.00
GPT2-large	-23.74	-18.61	-15.00
OPT-2.7b	-24.60	-18.88	-16.39

Table 7: Scores from GPT-PPL with different evaluator or generator. The evaluator model favours generation system based on itself.

BARTScore (Yuan et al., 2021) evaluates text generation quality as the log-probability of a seq2seq model. The default implementation relies on the finetuned BART-large model. Here, we test a hypothetical setting, where we base BARTScore on another popular PLM: T5 (Raffel et al., 2020). We use the BARTScore-cnn-faithful variant, and finetune all models on the CNNDM dataset (details given in Appendix F). Beam search with beam size 5 is used for decoding. The results are shown in Table 8. For this experiment, we do not assume the supremacy of one model over the other, as that requires more rigorous human evaluation.

Evaluator	Generator			
	BT-base	BT-large	T5-small	T5-base
BT-base	**-0.270**	**-0.361**	-0.367	-0.392
BT-large	**-0.357**	**-0.278**	-0.390	-0.389
T5-small	-0.359	-0.397	**-0.227**	-0.362
T5-base	-0.335	-0.344	**-0.331**	**-0.226**
nPPL	-4.323	-3.684	-4.903	-3.803
BS-para-p	-3.790	-3.762	-3.847	-3.786

Table 8: Scores from BARTScore-cnn-faithful using different PLMs as evaluator or generator on the summarization task. BT refers to BART and BS refers to BARTScore. Negated perplexity (nPPL) with the gold hypothesis are also reported for each model. In each row, scores marked by orange and bold are higher than scores marked by brown.

We observe an interesting but worrisome phenomenon: BART and T5 based evaluators strongly favor generators based on their own respective base models. This bias extends to different-sized variants of the base models as well. It is, however, less pronounced for the reference-based variant BARTScore-para.

Implication Overall, these results show that the log-probability-based metrics could be unfairly *biased* towards their underlying PLMs. Basing the metric on different PLM could give *inconsistent ranking* for the same set of systems.

Hence, practitioners should avoid situations where the generation system and the metric are based on the exact same PLM, or where systems based on different types of PLMs are compared with a metric based on one of them. In such cases, the scores should be complemented with additional evaluations from reference-based metrics.

While prior works follow this guideline by intuition (Liu et al., 2021), we show an explicit empirical analysis in support of this practice, which was previously lacking in the literature.

5.5 Fluency & Consistency Tests

The tests we discussed so far have been motivated by certain metric designs or properties of the underlying PLM that could be used to game the metrics. In this section, we move to more general tests, where we synthesize a range of perturbations that mimic human-errors and test the metrics' sensitivity to them.

5.5.1 Noise Types and Setup

Our tests cover two important aspects of natural language: fluency and consistency. Fluency tests focus on grammaticality, while consistency tests focus on temporal order, logic, or alignment with the source text.

Similar to previous sections, in each test we apply one type of noise the gold hypothesis. The noise can be regarded as an exaggeration of the errors human or machine writers could make. In total, we design 10 fluency tests and 8 consistency tests. For brevity, we only discuss a subset of them in this section (see Table 9).[6] The tests can generally be applied to all three tasks with a few exceptions (detailed in Appendix G). For example, we can not apply sentence switching to WMT as most samples only contain one sentence.

Most tests involve a hyper-parameter influencing the amount of noise added. This enables us to test how the metric behaves as we induce different levels of noise. To quantify the noise level, we define *noise-ratio*, which is based on the Levenshtein distance:

$$\frac{1}{|\mathcal{H}|}\sum_{h\in\mathcal{H}}\frac{\text{Levenshtein}(h', h)}{\text{len}(h)}, \quad (1)$$

where $\mathcal{H}$ is the set of gold hypotheses, and h' is the noised hypothesis. We employ the noise-ratio as a crude proxy to quantify the amount of noise across different noise types.[7]

For each noise type, a robust metric should give monotonically decreasing scores with an increasing noise-ratio. We claim a metric fails the test if it deviates from this expectation.

Finally, most noise types involve randomness. For each hyper-parameter, we report mean and standard-deviation over five runs with different random seeds. For each noise type and task, we set the hyper-parameters so that the gaps of noise-ratio between test points are close to or larger than 5%. The same set of random seeds and hyper-parameters are shared across all metrics.

5.5.2 Results

Results for a subset of metrics are shown in Figure 4. Unsurprisingly, most tests are passed by the metrics. However, the truncation and sentence switching tests give striking results. We will focus on these two tests in the following discussion, and defer more complete results and discussion to Appendix G.

A number of popular metrics fail the **truncation** test on the summarization task, including (some variants of) BARTScore, BERTScore, ROUGE, and COMET. It also affects open-ended generation (MAUVE-GPT2), which is discussed in Appendix G. This is surprising because truncation not only makes the hypothesis disfluent but also causes a serious loss of information.

The analysis in Figure 5(a) offers an insight into this undesirable behavior, where the values of three variants of BERTScore under the truncation test are plotted. We observe that precision increases with more truncation, canceling out the decrease in recall and leading to a non-decreasing f-measure. We conjecture that this happens due to the design of the task and datasets, where earlier parts of different summaries (of the same article) are more likely to overlap than the rest of the summaries. In Appendix G, we show a similar observation for

[6] A subset of the tests discussed in this section (e.g., truncation) overlaps with a very recent and independent work (Pimentel et al., 2022), but here we apply them to a wide range of metrics.

[7] One shortcoming of the Levenshtein distance is that it does not allow the switching operation. Therefore, for switching-based noise types, we divide the noise-ratio by 2.

Noise Type	Description
Truncation	A portion of tokens at the end of the hypothesis are removed. e.g., `She went to work.` → `She went`
Article Removal	A random portion of articles (the/a/an) in the hypothesis are removed.
Preposition Removal	A random portion of prepositions are removed. e.g., `She went to work.` → `She went work.`
Verb Lemmatization	A random portion of verbs in the hypothesis are lemmatized. e.g., `She went ...` → `She go ...`
Sentence Switching	Several random pairs of sentences in the hypothesis are switched, breaking temporal/logical order.
Sentence Replacement	Several sentences in the hypothesis are replaced by a random irrelevant sentence.
Negation	A random portion of sentences are negated. e.g., `She went ...` → `She did not go ...`

Table 9: Descriptions of a subset of the fluency tests (top) and consistency tests (bottom). Note that the truncation test not only breaks fluency but also causes loss of information (consistency). The complete set is described in Table 14 (Appendix G).

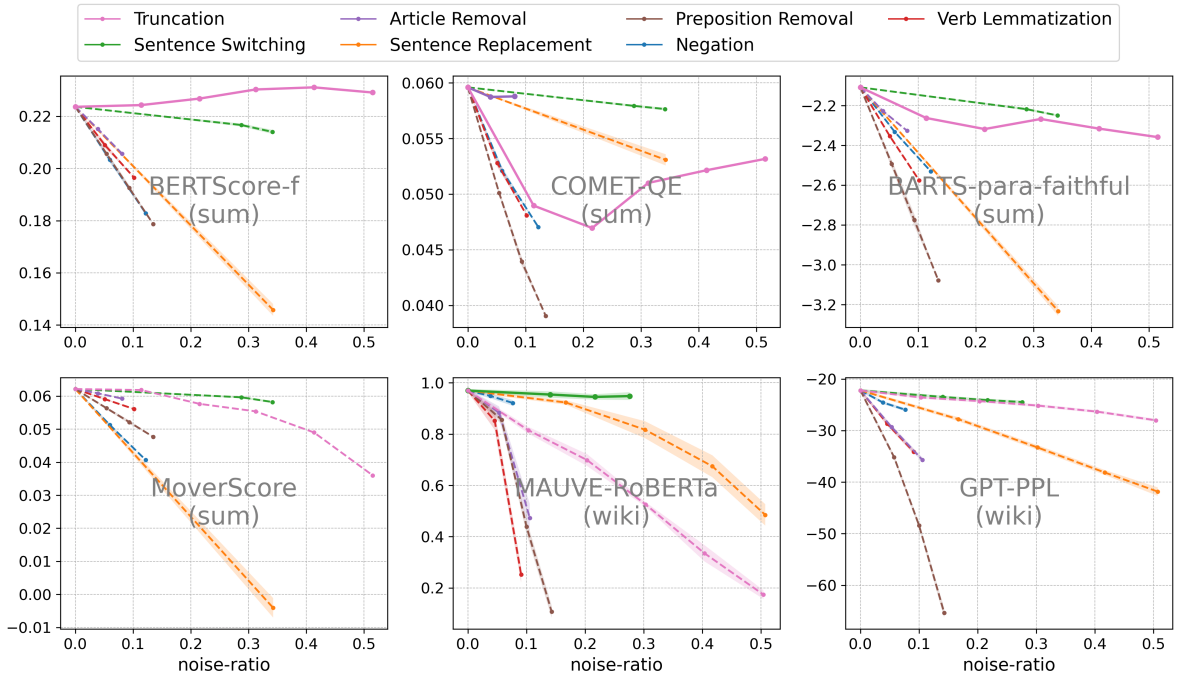
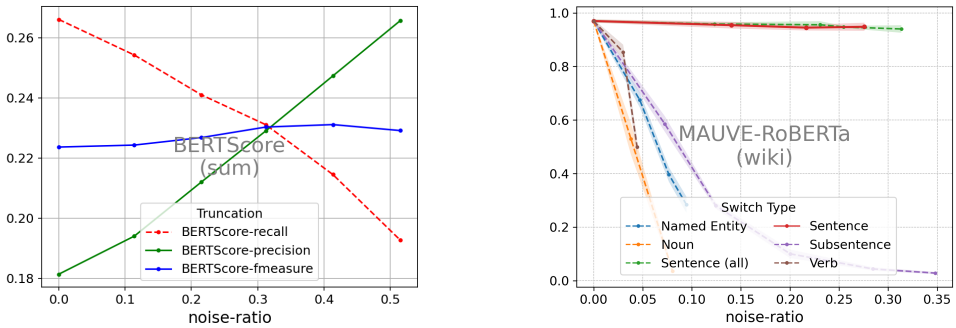


Figure 4: Selected results for fluency & consistency tests. For each plot, the x-axis is noise-ratio and the y-axis is the metric score. The point at noise-ratio zero is the score for the gold hypotheses. Non-monotonically-decreasing curves are highlighted in bold. Complete results are available in Appendix G.



(a) (b)

Figure 5: Analysis for truncation and sentence switching: (a) How the three variants of BERTScore react to the truncation test. (b) How MAUVE-RoBERTa reacts to different types of switch-based tests. "Sentence (all)" means that we do not fix the last sentence. The behavior of MAUVE-GPT2 is similar and we defer it to Figure 9 (Appendix G).

BARTScore-para. While reporting BERTScore-recall might remedy this issue, it is also not perfect, as we showed earlier that it fails the copy-source test.

In comparison, all metrics pass the truncation test in the translation task. We believe the reason is that in the WMT data, the gold hypothesis and the reference are highly similar (They mostly only differ by a few tokens). Therefore, it would be easier for the metrics to catch the loss of information.

Two metrics fail the **sentence switching** test: BARTScore-para-recall (sum), MAUVE-RoBERTa/GPT2 (wiki). This result is more striking for MAUVE, as the hypotheses in open-ended generation typically contain a number of sentences, and the temporal or logical order will be seriously disturbed by sentence switching. We give examples in Table 16 (Appendix G). Note that considering the distant error test of MAUVE, for the WikiText data, we intentionally do not switch the last sentence of the hypothesis paragraph.

To investigate more on this result, in Figure 5(b), we test switching different units of the hypothesis. Interestingly, MAUVE drops drastically for all other types of units.[8]

We do not test sentence switching for WMT as most samples only have one sentence. Future work may experiment with translation tasks with longer context (Tiedemann and Scherrer, 2017). Finally, for MoverScore, although the rank is right for both truncation and sentence switching, the slopes of the curves are relatively much flatter than other noise types, which might also be undesirable.

Implication Undesirable behaviors from the truncation test suggest that practitioners should either report all of the precision, recall, and f-measure for a complete picture or calibrate the f-measure to put more weight on recall than on precision.

The sentence switching test shows MAUVE's insensitivity to the temporal or logical order, while GPT-PPL has better behavior in that aspect. Therefore, it would be wise to use MAUVE in combination with GPT-PPL.

5.6 Can We Automate the Detection Process?

The tests we design rely on various intuitions including some level of understanding of the underlying PLM's behavior, or a detailed examination of the metric definitions. A natural next question is whether we can automate this process. Ideally, we would like an algorithm to search for a noising transformation function f of gold hypotheses that fools the targeted metric, while inducing perturbations visible to humans.

[8]We use {'**,**','**.**','**?**','**!**'} to deliminate sub-sentences.

Perturbation Examples
Around 21:30 **a** (→ **an**) 44 year old female car driver, ... *Relative BERTScore Change:* +0.37%
Before that **seven** (→ **eight**) coworkers had been ... *Relative BERTScore Change:* +0.28%
This (→ **These**) is waiting on a decision from the EuGH. *Relative BERTScore Change:* +0.52%
He (→ **They**) thinks that it makes sense ... *Relative BERTScore Change:* +0.17%

Table 10: Anomaly examples under automatic detection.

As a case study, we focus on BERTScore-f (wmt) and build a toy example using a discrete-space adversarial attack algorithm (Cheng et al., 2018; Li et al., 2020). Although it is only a preliminary attempt toward the ideal goal, the results show that it could be an interesting future direction.

On the high level, we design an enumeration-based algorithm that iteratively and greedily perturbs the hypothesis. Given a gold hypothesis h and source text s, the goal is to find a perturbed hypothesis h' that maximizes BERTScore(s, h', h),[9] subject to the noise-ratio being larger than a pre-specified value. i.e., the objective is to find a h' that BERTScore thinks is similar to h and aligns with the source s. The reference translations are not involved in this search.

In each perturbation step, we try two operations for each token in the current hypothesis: (1) Delete this token. (2) Replace this token with a token in a candidate set (detailed in Appendix H). Then, we select and apply the operation that maximizes BERTScore(s, h', h). This iteration is repeated until the desired noise-ratio is reached. One disadvantage of this approach is that we do not have a 100% guarantee that the perturbed hypothesis is indeed "bad". However, we do not observe empirical evidence of this weakness in the quantitative or qualitative examination.

Figure 10 (Appendix H) quantitatively demonstrates the effectiveness of the algorithm. Compared to BERTScore, the perturbations induce a large drop in a number of other metrics, implying that the perturbation is breaking the fluency/consistency of the gold hypotheses, and is

[9]The notation BERTScore(s, h', h) means that h' is inputted as the hypothesis, and h is inputted as the reference.

successfully fooling BERTScore.

We then inspect perturbed samples with high scores under BERTScore, with some examples in Table 10. The situation is especially common in articles (e.g., substitution of *a* and *an*), numbers (including the offset of date and time) and pronouns (e.g., substitution of *he*, *she*, *it* and *they*). While these substitutions are detrimental, they were not penalized by BERTScore. Incidentally, these patterns are not covered by our checks in Section 5.5, which demonstrates the value of this study.

Inspired by this, we attempt to design general noise transformation rules based on the observations (e.g., pronoun switching), and apply them to BERTScore. However, we find that these patterns do not generalize to the whole WMT dataset. One key reason is that the transformation is only effective in confusing BERTScore for a subset of the hypotheses, which might not be surprising due to the nature of the adversarial attack. We conclude that more research is needed to make this framework practical and we leave it to future work.

6 Discussion and Limitations

Since this work contains negative results, we devote this section to discussing limitations and preventing potential misunderstandings.

For Metric Users The results in this work should be regarded as **complementary** to the impressive human correlation results in the literature. For example, BLEU passes all our tests in WMT data, however, it is outperformed by PLM-based metrics in human correlation evaluations (Zhang et al., 2020). If a metric fails one of our tests, it only means the metric needs improvement on that particular aspect. Our main message is not to discourage the use of PLM-based metrics, nor to disvalue existing work by metric developers or users. **Instead, we suggest use the metrics with caution** (Mathur et al., 2020) **and with awareness of the blind spots.**

Our synthetic noise should be regarded as an exaggeration of potential textual errors found in practice, indicating that the related metric might also be insensitive in certain more general aspects. For example, the truncation test is related to the generation loss of information, and sentence switching is related to the broken temporal or logical disorder. Even if one can guarantee there is no truncation or switching errors, it does not mean the blind spots in the metrics can be ignored.

For Metric Developers While we have covered a large variety of stress tests in this work and we encourage future metric developers to use them for robustness analysis, the set is not exhaustive. That is, even if a metric passes all our tests, it does not guarantee that the metric is blind-spot-free. We also encourage developers to come up with novel tests targeting certain underlying property of their proposed metric (e.g., the distant error test we design for MAUVE). As a future direction, it is exciting to think about how to automate the stress-test process.

Another interesting future direction is to use the synthesized noised data to augment the finetuning of the metric or its underlying PLM, with the goal to make it robust against certain type of noise. For example, one could finetune RoBERTa with augmented data so that MAUVE passes the sentence switching test on WikiText. However, further analysis is required if the goal is to improve the metric in a more general sense (e.g., does it generalize to other datasets or other more subtle logical or temporal errors?).

Other Limitations We have primarily focused our analysis on similarity or log-probability-based metrics for NLG. There are a number of other important and interesting metrics which future work may analyze. For example, Deng et al. (2021) developed a family of interpretable metrics for various NLG tasks with the concept of information alignment. Recently, Zhong et al. (2022) proposed a multidimensional evaluator which re-frames NLG evaluation as a boolean question answering task. In addition, there are several task-specific metrics for paraphrase generation (Shen et al., 2022), image captioning (Hessel et al., 2021; Kasai et al., 2022a), dialogue (Mehri and Eskenazi, 2020), controlled text generation (Ke et al., 2022), etc. which might benefit from a similar analysis to ours.

In §5.5, we design a number of fluency or consistency noise types. It would be interesting to expand this set to be broader or more sophisticated (Ng et al., 2014). Also, there are other important aspects of text generation to consider, such as factuality (Wang et al., 2020; Pagnoni et al., 2021).

Last but not least, we evaluate our proposed stress tests only on English text. However, many language-specific properties can induce potential blind spots for metrics, especially for low-resource languages (Haddow et al., 2022) where PLMs may provide poor text representations. An important fu-

ture direction is expanding the tests to multilingual settings (Thompson and Post, 2020; Pires et al., 2019).

7 Related Work

Analysis of NLG Metrics In comparison to the vast literature on NLG metric development or benchmarking (Celikyilmaz et al., 2020; Gehrmann et al., 2021; Kasai et al., 2022b), the robustness analysis of PLM-based metrics is an underexplored area, where exisiting work focused on a small subset of metrics or a limited definition of robustness. For example, Vu et al. (2022) explore BERTScore's performance variation with changes in representation space and character perturbations. Kaster et al. (2021) propose a regression-based global explainability technique to disentangle metric scores along linguistic factors.

More related to our work, Hanna and Bojar (2021) conduct a fine-grained analysis of BERTScore on different error types with human-annotated datasets. Caglayan et al. (2020) demonstrate some curious phenomena for a range of metrics. Very recently, Sun et al. (2022) find that NLG metrics are not robust to dialects and have a preference for American English. In contrast, this work is more comprehensive in that we stress-test a wide range of popular metrics along many dimensions of noise.

Analysis of PLM This work takes inspiration from research analyzing the behavior of PLM's representations (Belinkov and Glass, 2019). Masked LMs such as BERT have been shown to be insensitive to word order (Pham et al., 2021), negation (Ettinger, 2020), and named entities (Balasubramanian et al., 2020). GPT-like models are shown to prefer repetitive text (Holtzman et al., 2020). Staliūnaitė and Iacobacci (2020) studies what types of linguistic knowledge BERT acquires with a focus on compositional and lexical semantics. There are also important lines of work on layer representation probing (Belinkov, 2022), or attention analysis (Dong et al., 2021).

Synthetic Data for NLP Model Analysis The use of synthetic data has been proven to be a powerful tool to analyze the capabilities of NLP models in tasks including natural language inference (McCoy et al., 2019; Naik et al., 2018), question answering (Ribeiro et al., 2019), reading comprehension (Sugawara et al., 2020) and text classification (Prabhakaran et al., 2019). Ribeiro et al. (2020) propose a task-agnostic methodology, which synthesizes a large number of examinations for NLP models. Ruder et al. (2021) subsequently extended this methodology to a multilingual setting. Goel et al. (2021) built a more complete model evaluation system by integrating subpopulations, transformations, evaluation sets, and adversarial attacks. This work follows the same high-level spirit, while our focus is on NLG metrics.

8 Conclusion

Using PLMs for NLG metrics is a double-edged sword. While the metrics benefit from the models' powerful representations, their black-box nature may cause unexpected behavior. This work shows that stress tests, complementary to the standard human correlation tests, are a powerful tool to cover corner cases, detect the metrics' blind spots, and point out aspects where the metric could improve.

As a major implication for metric users, we suggest using combinations of metrics so that they can cover each other's blind spots. While this has been an existing practice for a majority of work in the field, our results on the blind spots provide an explicit empirical argument for its importance. While we are still positive about the future of using PLM for NLG metrics, we call for more caution from both metric users and developers. More generally, a better understanding of the PLM itself is in need.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Sriram Balasubramanian, Naman Jain, Gaurav Jindal, Abhijeet Awasthi, and Sunita Sarawagi. 2020. What's in a name? are BERT named entity representations just as good for any other name? In *Proceed-*

ings of the 5th Workshop on Representation Learning for NLP, pages 205–214, Online. Association for Computational Linguistics.

Yonatan Belinkov. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219.

Yonatan Belinkov and James Glass. 2019. Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Ozan Caglayan, Pranava Madhyastha, and Lucia Specia. 2020. Curious case of language generation evaluation metrics: A cautionary tale. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2322–2328, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *ArXiv*, abs/2006.14799.

Minhao Cheng, Jinfeng Yi, Huan Zhang, Pin-Yu Chen, and Cho-Jui Hsieh. 2018. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. *CoRR*, abs/1803.01128.

Mingkai Deng, Bowen Tan, Zhengzhong Liu, Eric Xing, and Zhiting Hu. 2021. Compression, transduction, and creation: A unified framework for evaluating natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7580–7605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Dong, Chandra Bhagavatula, Ximing Lu, Jena D. Hwang, Antoine Bosselut, Jackie Chi Kit Cheung, and Yejin Choi. 2021. On-the-fly attention modulation for neural generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1261–1274, Online. Association for Computational Linguistics.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Anuoluwapo Aremu, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh Dhole, Wanyu Du, Esin Durmus, Ondřej Dušek, Chris Chinenye Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Yufang Hou, Yacine Jernite, Harsh Jhamtani, Yangfeng Ji, Shailza Jolly, Mihir Kale, Dhruv Kumar, Faisal Ladhak, Aman Madaan, Mounica Maddela, Khyati Mahajan, Saad Mahamood, Bodhisattwa Prasad Majumder, Pedro Henrique Martins, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, Andre Niyongabo Rubungo, Salomey Osei, Ankur Parikh, Laura Perez-Beltrachini, Niranjan Ramesh Rao, Vikas Raunak, Juan Diego Rodriguez, Sashank Santhanam, João Sedoc, Thibault Sellam, Samira Shaikh, Anastasia Shimorina, Marco Antonio Sobrevilla Cabezudo, Hendrik Strobelt, Nishant Subramani, Wei Xu, Diyi Yang, Akhila Yerukola, and Jiawei Zhou. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 96–120, Online. Association for Computational Linguistics.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the NLP evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 42–55, Online. Association for Computational Linguistics.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732.

Michael Hanna and Ondřej Bojar. 2021. A fine-grained analysis of BERTScore. In *Proceedings of the Sixth Conference on Machine Translation*, pages 507–517, Online. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2021. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages

7514–7528, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. 2019. Large-scale, diverse, paraphrastic bitexts via sampling and clustering. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 44–54, Hong Kong, China. Association for Computational Linguistics.

Jungo Kasai, Keisuke Sakaguchi, Lavinia Dunagan, Jacob Morrison, Ronan Le Bras, Yejin Choi, and Noah A. Smith. 2022a. Transparent human evaluation for image captioning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3464–3478, Seattle, United States. Association for Computational Linguistics.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Lavinia Dunagan, Jacob Morrison, Alexander Fabbri, Yejin Choi, and Noah A. Smith. 2022b. Bidimensional leaderboards: Generate and evaluate language hand in hand. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3540–3557, Seattle, United States. Association for Computational Linguistics.

Marvin Kaster, Wei Zhao, and Steffen Eger. 2021. Global explainability of BERT-based evaluation metrics by disentangling along linguistic factors. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8912–8925, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pei Ke, Hao Zhou, Yankai Lin, Peng Li, Jie Zhou, Xiaoyan Zhu, and Minlie Huang. 2022. CTRLEval: An unsupervised reference-free metric for evaluating controlled text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2306–2319, Dublin, Ireland. Association for Computational Linguistics.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294, Melbourne, Australia. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1910.13461*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. 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 6691–6706, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707, Online. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. 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.

Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. 2021. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? pages 1145–1160.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. In *Neural Information Processing Systems*.

Tiago Pimentel, Clara Meister, and Ryan Cotterell. 2022. On the usefulness of embeddings, clusters and strings for text generator evaluation.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5740–5745, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019a. Language models are unsupervised multitask learners.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019b. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Sebastian Ruder, Noah Constant, Jan Botha, Aditya Siddhant, Orhan Firat, Jinlan Fu, Pengfei Liu, Junjie Hu, Dan Garrette, Graham Neubig, and Melvin Johnson. 2021. XTREME-R: Towards more challenging and nuanced multilingual evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10215–10245, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Annual Meeting of the Association for Computational Linguistics*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. In *Proceedings of ACL*.

Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2022. On the evaluation metrics for paraphrase generation.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Chrysoula Zerva, Zhenhao Li, Vishrav Chaudhary, and André F. T. Martins. 2021. Findings of the WMT 2021 shared task on quality estimation. In *Proceedings of the Sixth Conference on Machine Translation*, pages 684–725, Online. Association for Computational Linguistics.

Ieva Staliūnaitė and Ignacio Iacobacci. 2020. Compositional and lexical semantics in RoBERTa, BERT and DistilBERT: A case study on CoQA. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7046–7056, Online. Association for Computational Linguistics.

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8918–8927.

Jiao Sun, Thibault Sellam, Elizabeth Clark, Tu Vu, Timothy Dozat, Dan Garrette, Aditya Siddhant, Jacob Eisenstein, and Sebastian Gehrmann. 2022. Dialect-robust evaluation of generated text.

Brian Thompson and Matt Post. 2020. 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.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Doan Nam Long Vu, Nafise Sadat Moosavi, and Steffen Eger. 2022. Layer or representation space: What makes BERT-based evaluation metrics robust? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3401–3411, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

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

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 1815–1825. Curran Associates, Inc.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

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. *CoRR*, abs/2210.07197.

Supplemental Materials

A Implementation Details of Metrics or Tests

MLM-PPL The high-level motivation for MLM-PPL (Salazar et al., 2020) is using a bidirectional masked language model to compute a quantity similar to next-token perplexity in autoregressive models by masking candidate tokens one by one and obtaining perplexity from masked token log probability instead of next-token log probability. We follow a similar formulation of the pseudo-perplexity in Salazar et al. (2020). Given a sequence $\boldsymbol{W} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{|\boldsymbol{W}|})$, we replace a token $\boldsymbol{w}_t$ with `[MASK]`, and predict it using all past and future tokens $\boldsymbol{W}_{\setminus t} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{t-1}, \boldsymbol{w}_{t+1}, \ldots, \boldsymbol{w}_{|\boldsymbol{W}|})$. Let $\log P_{\text{MLM}}(\boldsymbol{w}_t \mid \boldsymbol{W}_{\setminus t})$ denote the conditional log probability of predicting each token $\boldsymbol{w}_t$ given its context. MLM-PPL is defined as below:

$$\text{MLM-PPL}(\boldsymbol{W}) = \exp\left(-\frac{1}{|\boldsymbol{W}|}\sum_{t=1}^{|\boldsymbol{W}|}\log P_{\text{MLM}}(\boldsymbol{w}_t \mid \boldsymbol{W}_{\setminus t})\right).$$

MAUVE We use the default hyperparameter settings recommended in Pillutla et al. (2021). $c = 5$ is set for the scaling constant. For the quantization algorithm, we use k-means with 500 iterations and $n/10$ clusters, where n is the number of generations. We use two different embedding models, GPT2-large and RoBERTa-large, to create two variants of MAUVE.

BERTScore As suggested by Zhang et al. (2020), the f-measure variant of BERTScore is used for translation. However, the paper does not have recommendations for summarization. Therefore we test all three variants (precision, recall, f-measure).

BARTScore As introduced in Yuan et al. (2021), BARTScore has four variants to tackle different scenarios, and each variant defines a pair of input-output for BART: precision (reference to hypothesis), recall (hypothesis to reference), f-measure, and faithfulness (source to hypothesis).

As suggested by the paper, for translation we use the f-measure. However, for summarization, the recommendations are a bit vague. In the main sections, we mainly report the faithfulness variant as it is used by the paper for the SummEval dataset (which is based on CNNDM). We also test the other three variants and defer their results to the appendix.

In addition to BARTScore-cnn and BARTScore-para, BARTScore also has a *prompted* modeling option which we currently do not have the capacity to test. We leave it as future work.

ROUGE Following common practice, we use the f-measure of ROUGE-2 or ROUGE-L.

Test Implementation Our test code for translation or summarization is built upon the released code from BARTScore.[10] We also benefit from the Hugging Face library.[11] Some fluency and consistency tests are built using the spaCy library.[12] For the negation test, we utilize released code from the NLP CheckList (Ribeiro et al., 2020).[13]

B More Information on Datasets

For the gold hypotheses of the WikiText-103 dataset, we sample paragraphs with more than 256 tokens and conduct preprocessing to clean up dataset artifacts and special symbols. First, we trim extra space around {'**.**', '**,**', '**?**', '**!**', '**:**', '**;**', '(', ')', "**'s**", '**%**'}. Next, we remove the special token '@' in the dot '**@.@**' and hyphen '**@-@**' tokens. We also remove extra space around quotation marks. Finally, the text is truncated to the last full sentence under a total length of 256, which is to ensure the gold hypotheses are of similar length.

C Details on the Distant Error Test

In this section, we provide details about the *attention pattern analysis*. We input two random samples (non-cherry-picked) from the WikiText dataset to GPT2-large and RoBERTa-large and visualize the attention distribution over the relative position in the text. The sample is truncated to length 200 for the convenience of this analysis.

As shown in Figure 11, we average the attention distribution over all transformer layers and attention heads and then group 20 x 20 (attention-from and attention-to) tokens into one attention block for ease of presentation. We also include a high-granularity version where we group 2 x 2 tokens into one attention block.

[10] https://github.com/neulab/BARTScore.
[11] https://github.com/huggingface/transformers.
[12] https://github.com/explosion/spaCy.
[13] https://github.com/marcotcr/checklist.

D Auxiliary Results for the Copy-source Trick

In Table 11 we show auxiliary results of the copy-source trick. In addition to the discussion in Section 5.2, the recall and f-measure variants of BARTScore also fail this test.

For COMET, the scores from the copied source are very close to the gold hypothesis (marked in orange), which is also undesirable.

Metric (task)	GOLD	Copy-source
COMET(wmt)	0.531	-0.079
COMET-QE(wmt)	0.114	0.126
BertSc-r(sum)	0.266	0.332
BertSc-p(sum)	0.181	-0.177
BertSc-f(sum)	0.223	0.065
BartSc-cnn-p(sum)	-2.718	-3.022
BartSc-cnn-r(sum)	-3.249	-2.834
BartSc-cnn-f(sum)	-2.984	-2.928
BartSc-cnn-faithful(sum)	-1.376	-0.368
BartSc-cnn-failthful-noavg(sum)	-82.95	-166.25
BartSc-para-p(sum)	-4.023	-4.218
BartSc-para-r(sum)	-3.751	-2.948
BartSc-para-f(sum)	-3.887	-3.583
BartSc-para-faithful(sum)	-2.109	-0.874
COMET(sum)	-0.575	-0.584
COMET-QE(sum)	0.059	0.048

Table 11: Auxiliary results of the copy-source trick. In addition to the discussion in Section 5.2, for COMET the scores from the copied source are very close to the gold hypothesis (marked in orange), which is undesirable.

E Auxiliary Results for the Repetition & Frequent N-gram Tests

Table 12 contains auxiliary results for the repetition test. We observe that it affects all variants of BARTScore. Not surprisingly, the problem is less serious for the recall variant. As an illustrated example of the repetition test, Figure 6 shows the per-timestep next-token probability of a 4-gram repetitive text in the WikiText dataset, given by GPT-PPL. The first repetition of the 4-gram "hard to miss." has a slightly higher probability compared to the original ending. As this 4-gram is repeated more times, the probability given by GPT-PPL becomes increasingly higher.

In Table 13, results of frequent 4-gram and 3-gram tests are shown. We observe that it is easier for the frequent 4-grams to confuse the log-probability-based metrics. Per-timestep next-token probability plots for examples of a 4-gram and a 3-gram test are shown in Figure 3 and Figure 7,

Metric (task)	Gold	Repetition		
		Rep-10	Rep-20	Rep-30
B-cnn-f (wmt)	-2.168	-1.889	-1.721	-1.652
B-para-f (wmt)	-1.868	-1.956	-1.864	-1.839
BLEURT (wmt)	0.716	0.666	0.683	0.689
B-cnn-p (sum)	-2.718	-2.122	-1.675	-1.451
B-cnn-r (sum)	-3.249	-3.246	-3.251	-3.252
B-cnn-f (sum)	-2.984	-2.684	-2.463	-2.351
B-cnn-faithful (sum)	-1.376	-1.486	-1.224	-1.091
B-para-p (sum)	-4.023	-3.156	-2.630	-2.362
B-para-r (sum)	-3.751	-3.710	-3.693	-3.685
B-para-f (sum)	-3.887	-3.433	-3.162	-3.023
B-para-faithful (sum)	-2.109	-2.039	-1.759	-1.626
GPT-PPL (wiki)	-21.81	-15.48	-10.70	-8.080
MLM-PPL (wiki)	-2.635	-2.241	-2.019	-1.867

Table 12: Auxiliary results for the repetition test. "B-" refers to "BARTScore-".

respectively. In both cases, there are high probability regions concentrated at the end of each n-gram. For example, "the" in the 3-gram "side of the" gets a higher probability than the first two tokens, and "of" in the 4-gram "in the middle of" gets a higher probability than the first three tokens.

Metric (task)	Gold	Freq 4-gram		
		Top-10	Top-50	Top-100
GPT-PPL (wiki)	-25.640	-4.456	-11.640	-18.160
MLM-PPL (wiki)	-2.994	-1.139	-2.469	-3.971
rep-4gram (wiki)	0.019	0.539	0.199	0.120
Metric (task)	**Gold**	**Freq 3-gram**		
		Top-10	Top-50	Top-100
GPT-PPL (wiki)	-25.640	-5.650	-19.910	-27.410
MLM-PPL (wiki)	-2.994	-1.368	-4.224	-7.266
rep-4gram (wiki)	0.019	0.452	0.084	0.041

Table 13: Results of Frequent 4-gram and 3-gram tests.

F Details on the Finetuning (Self-Generation)

For GPT-PPL, we finetune the GPT-2 generators on the WikiText-103 training set for 2 epochs, with a learning rate of 1e-05 and a batch size of 16.

For BARTScore, we finetune the BART or T5 models on the CNNDM training set for 2 epochs, with a learning rate of 1e-05 and a batch size of 8.

G Auxiliary Description and Results of the Fluency and Consistency Tests

The full set of tests is described by Table 14. For the detailed hyper-parameter setting, please refer to our to-be-released code.

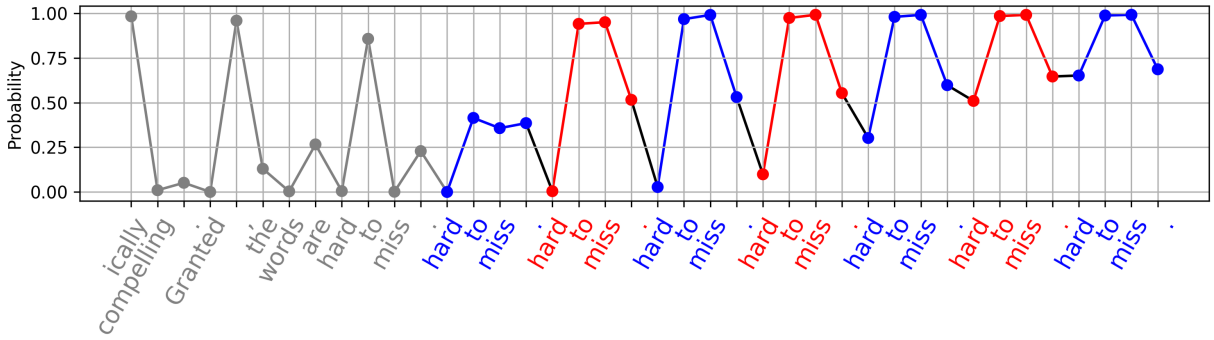


Figure 6: Per-timestep next-token probability of a 4-gram repetitive text sequence given by GPT-PPL.

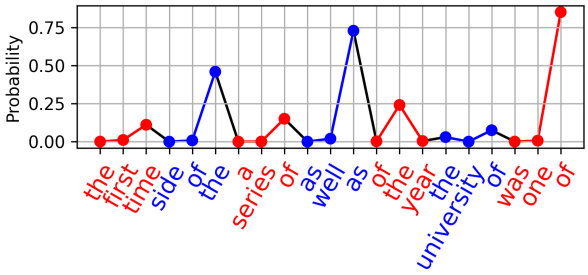


Figure 7: Per-timestep next-token probability of a frequent 3-grams sequence given by GPT-PPL.

In general, the tests can be applied to all three tasks. But there are exceptions due to the properties of the dataset: (1) We do not apply BERT-diverge to the WikiText data, as the task's nature is open-ended. (2) We can not apply sentence switching to WMT as most samples only contain one sentence. (3) Due to similar reasons, we do not apply verb or named entity switching and sentence replacement to WMT.

Compared to other tests, BERT-diverge is special in that its noise is generated automatically by an MLM, which is an interesting future direction for metric stress tests. One disadvantage of this approach is that we do not have a 100% guarantee that the perturbed hypothesis is indeed "diverged". However, we do not observe empirical evidence of this weakness in the quantitative (Most metrics drop drastically with this noise) or qualitative examination.

The **complete results** for the fluency and consistency tests are shown in Figure 13 for open-ended generation, Figure 12 for summarization, and Figure 14 for translation. For visibility, we plot fluency test and consistency tets separately for each metric. Failed tests are highlighted as bold lines.

Auxiliary Discussion of the Results We now discuss some interesting results which are not included in the main section.

For open-ended generation, both variants of MAUVE (-GPT2/-RoBERTa) fail the sentence switching test. Although MLM-PPL does not fail the test in terms of rank, the slope of the sentence switching curve is relatively much flatter than the other noise types, indicating an insensitivity.

Interestingly, while MAUVE-RoBERTa is robust to truncation, MAUVE-GPT2 only penalizes truncation in a binary manner. The score is much lower than gold for the first level of noise, but remains basically the same for other levels compared to the first level. This implies the GPT2 feature is not sensitive to the amount of information loss, which is problematic. From insights of the attention analysis (§5.1), we also attribute this to the locality of GPT2 embedding.

GPT-PPL and MLM-PPL are robust to truncation, but only penalize this error minimally as shown by the relatively flat slope of their truncation curves, which is not ideal.

For summarization, notably, BARTScore-cnn/para-r fails a number of fluency tests involving stop-words, prepositions, etc. This suggests extra caution is needed when developing recall-orientated log-probability-based metrics.

ROUGE-2 and ROUGE-L fail the truncation and noised punctuation tests. ROUGE2 also has a very marginal decrease in sentence switching, which is also undesirable.

Interestingly, BERT-diverge with COMET-QE is the only failure case for translation (The same set of BERT-diverge noise is shared across metrics). A few examples are given in Table 15. We observe that the semantics of the hypotheses are clearly diverged, however, the scores from COMET-QE do not drop.

In addition, COMET-QE also fails article removal on summarization.

Noise Type	Description
Truncation	A portion of tokens at the end of the hypothesis are removed. e.g., `She went to.`
Article Removal	A random portion of articles (the/a/an) in the hypothesis are removed. e.g., `She went to office.`
Preposition Removal	A random portion of prepositions are removed. e.g., `She went the office.`
Stop-word Removal	A random portion of stop-words are removed. e.g., `She went office.`
Verb Lemmatization	A random portion of verbs in the hypothesis are lemmatized. e.g., `She go to the office.`
Token Drop	A random portion of tokens are removed. e.g., `She to the offce.`
Repeated Token	A random portion of tokens are repeated once. e.g., `She went to to the office.`
Local Swap	A random portion of tokens are swapped with the token to the right of it. e.g., `She to went the office.`
Middle Swap	The left and right part of the sentence is swapped (The cut-off point is right in the middle of the length). This is to synthesize a wrong subject-verb-object (SVO) order. e.g., `To the office she went.`
Noised Punctuation	The punctuations {'**,**','**.**','**?**','**!**','**:**'} are noised. For example, commas are replaced by periods and vice versa. e.g., `She went to the office,`
Sentence Switching	Several random pairs of sentences in the hypothesis are switched, breaking temporal/logical order. e.g., `And she talked to her staff about Paris. She went to the office in Boston.`
Sentence Replacement	Several sentences in the hypothesis are replaced by a random irrelevant sentence. `This is an amazing game. And she talked to her staff about business.`
Negation	A random portion of sentences are negated. e.g., `She did not go to the office in Boston. And she talked to her staff about Paris.`
Generic Named Entity	A random portion of the named entities in the hypothesis are replaced by a generic phrase, destroying the information. e.g., `She went to the office in a place. And she talked to her staff about a place.`
Named Entity Switching	Several random pairs of named entities in the hypothesis are switched, breaking factuality. e.g., `She went to the office in Paris. And she talked to her staff about Boston.`
Verb Switching	Several random pairs of verbs in the hypothesis are switched. e.g., `She talked to the office in Boston. And she went to her staff about business.`
Noun Switching	Several random pairs of nouns in the hypothesis are switched. e.g., `She went to the staff in Boston. And she talked to her office about business.`
BERT-diverge	Several random tokens in the hypothesis are replaced one by one by sampling from the top-10 prediction of a masked language model (RoBERTa). At each step, one token at a random position is replaced by `[MASK]`, and inputed to RoBERTa for prediction. Since this process do not have access to the source text, the semantics of the hypothesis would gradually diverge. e.g., `She ran to the office in Boston. And she talked to her staff about business.`

Table 14: Descriptions of the fluency tests (top) and consistency tests (bottom). Note that the truncation test not only breaks fluency, but also causes loss of information (consistency). For fluency tests, the example gold hypothesis is "`She went to the office.`" For consistency tests, the example gold hypothesis is "`She went to the office in Boston. And she talked to her staff about Paris.`" The gold hypothesis here is only for ease of explanation and it does not exist in the datasets.

Analysis of Truncation In Figure 8, we show how different variants of BARTScore-para behave under the truncation test. We also observe that the recall variant behaves well,[14] while the precision and faithful variants are confused.

Analysis of Switching Figure 9 shows how MAUVE-GPT2 reacts to different types of switch-based tests. The behavior is similar to MAUVE-RoBERTa.

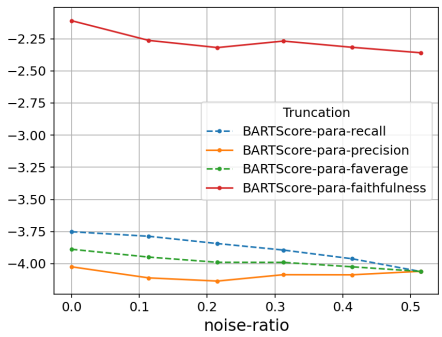


Figure 8: How the variants of BARTScore-para react to the truncation test for the summarization task.

H Auxiliary Description and Results of the Automatic Detection

H.1 Attack Algorithm Details

We fix the targeted LM as RoBERTa since BERTScore is based on it.

[14] But, BARTScore-para-recall fails the sentence switching test. Therefore, we recommend reporting the recall variant in combination with other variants.

BERT-Diverge Perturbation Examples
Gold: The biker still attempted to evade the car, however, brushed against the car at the rear end. **BERT-diverge:** The biker narrowly managed to evade the car, however nearly brushed against the car in the immediate area. *Relative COMET-QE Score Change:* +5.60%
Gold: A security service monitors the curfew. **BERT-diverge:** The security force enforced the laws. *Relative COMET-QE Score Change:* +2.95%
Gold: Greens and SPD blamed the State government for shared responsibility. **BERT-diverge:** Greens and others blamed the federal government for its failure. *Relative COMET-QE Score Change:* +18.61%

Table 15: Examples of noise from BERT-diverge on WMT data. The semantics have clearly diverged, however, the scores from COMET-QE do not drop.

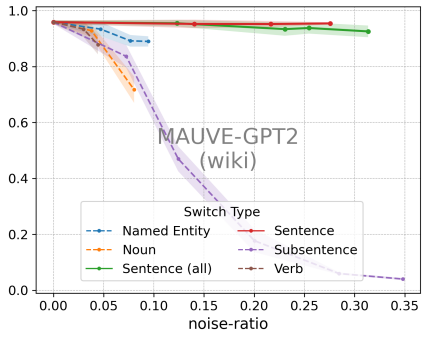


Figure 9: How MAUVE-GPT2 reacts to different types of switch-based tests. "Sentence (all)" means that we do not fix the last sentence.

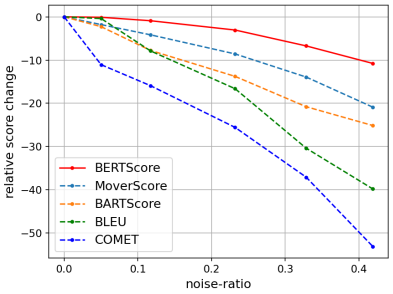


Figure 10: Relative score changes of some metrics under adversarial attacks against BertScore.

In our iterative perturbation algorithm, for a hypothesis $h = [w_1, \ldots, w_{\text{len}(h)}]$, we enumerate each token w_i in it, and design the following perturbations: (1) Delete the token. The perturbed hypothesis becomes $h' = [w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{\text{len}(h)}]$, (2) Substitute the token. We build the candidate token set C in two ways: (a) Use `[MASK]` to replace w_i, and employ the masked RoBERTa model to generate $k_1 = 8$ possible tokens $w' \in C_1$ with the highest scores (similar to BERT-diverge). (b) Utilize the word embedding in RoBERTa to find the $k_2 = 8$ possible tokens $w' \in C_2$ closest to w_i. And $C = C_1 \cup C_2$.[15] In this way, we can get $k_1 + k_2$ perturbed hypothesis $h' = [w_1, \ldots, w_{i-1}, w', w_{i+1}, \ldots, w_{len(h)}], w' \in C$. In our experiments, we set both k_1 and k_2 to eight.

H.2 Results

Results of the Attack Algorithm We test five different metrics on our adversarial perturbations. As shown in Fig. 10, compared to BERTScore, other metrics drop significantly, especially when the noise-ratio is small. When noise-ratio equals 11.72%, the relative score of BERTScore drops only 0.90%. Meanwhile, MoverScore, which shares the closest LM and calculation to BERTScore, drops 4.16%. This implies that the perturbation is breaking the fluency/consistency of the gold h.

[15] Some relatively meaningless substitutions, such as punctuation and uppercase/lowercase replacement will be filtered.

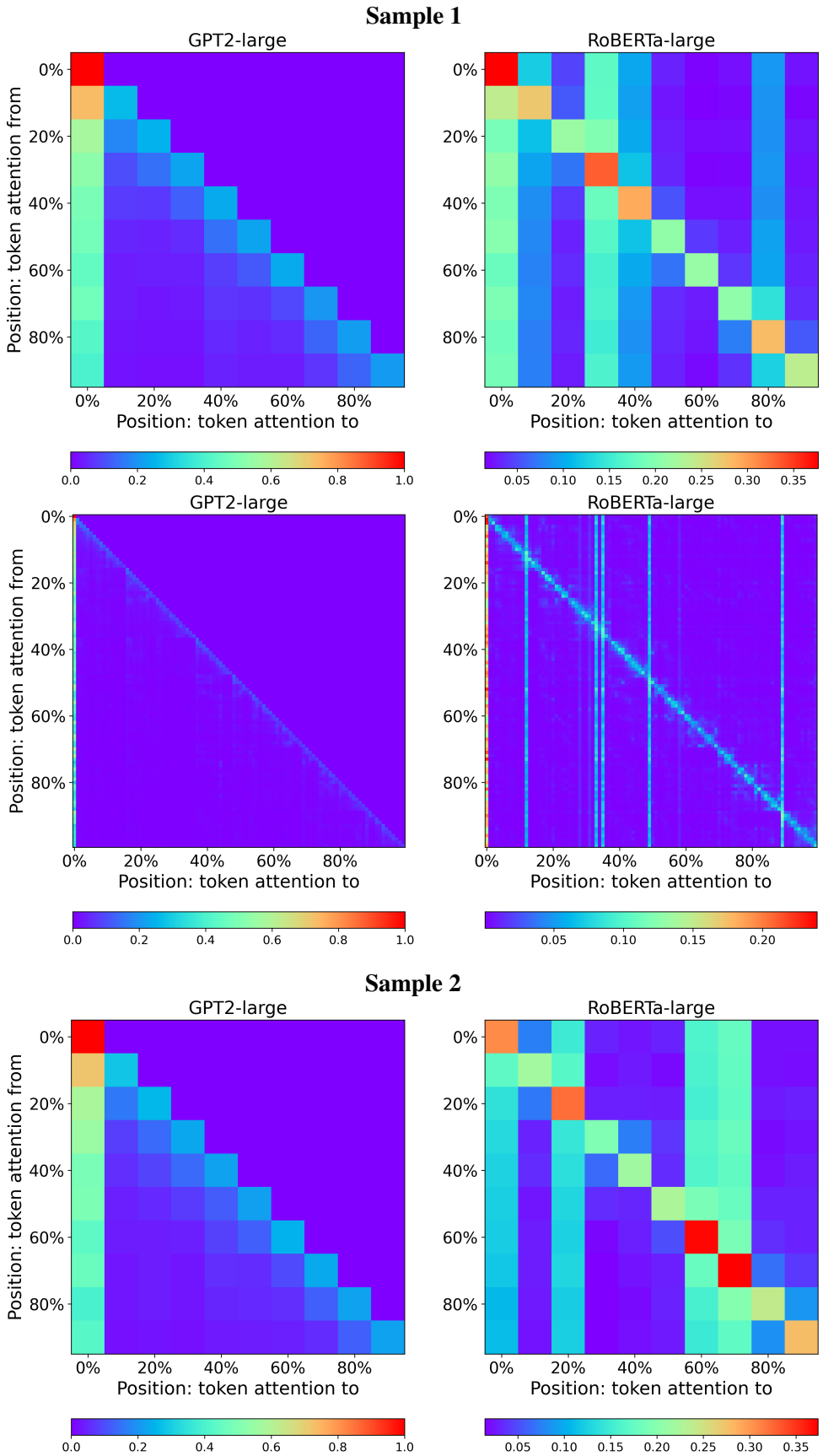


Figure 11: Attention distribution of GPT2-large and RoBERTa-large over the relative position in two random samples from the WikiText dataset. Top: Sample 1, low-granularity average. Middle: Sample 1, high-granularity average. Bottom: Sample 2, low-granularity average. Attention values are averaged over transformer layers and attention heads). This difference is typical and not a result of cherry-picking.

Noise Type	Example
Gold	The German invasion of Norway in 1940 led to Andersen's life once more taking a turn into illegal activities. His furniture workshop was used as a weapons depot by the Norwegian resistance movement, and he took part in looting German military stores. He was first arrested by the Germans after he had responded to rumours that he was a Nazi by writing the Norwegian national socialist party Nasjonal Samling's official publication Fritt Folk and stating that "although I have done many wrong things in my life, a Nazi I am not. Yours sincerely Johs. S. Andersen". The letter was published unedited by the newspaper, although Andersen was later arrested by the occupying authorities and sentenced to one year in prison, after spending half a year in detention. Using techniques he had learned during his earlier criminal career, Andersen managed to be transferred to prison hospital during his time in detention. While there he acquired false x-ray images and tuberculosis germs to fake illnesses in other captured resistance men who were on their way to interrogation. He also infected a German interrogator with malaria by contaminating his insulin.
Switched (6)	His furniture workshop was used as a weapons depot by the Norwegian resistance movement, and he took part in looting German military stores. Using techniques he had learned during his earlier criminal career, Andersen managed to be transferred to prison hospital during his time in detention. While there he acquired false x-ray images and tuberculosis germs to fake illnesses in other captured resistance men who were on their way to interrogation. The letter was published unedited by the newspaper, although Andersen was later arrested by the occupying authorities and sentenced to one year in prison, after spending half a year in detention. S. Andersen". He was first arrested by the Germans after he had responded to rumours that he was a Nazi by writing the Norwegian national socialist party Nasjonal Samling's official publication Fritt Folk and stating that "although I have done many wrong things in my life, a Nazi I am not. Yours sincerely Johs. The German invasion of Norway in 1940 led to Andersen's life once more taking a turn into illegal activities. He also infected a German interrogator with malaria by contaminating his insulin.

Table 16: Examples of sentence switching on the WikiText dataset. Six sentence pairs are switched. The switched hypothesis is incoherent on the high level. For example, the gold hypothesis discusses Andersen's life prior to the German invasion, his letter and arrest by the Germans, and finally his resistance against Nazis in his detention. However, in the switched hypothesis, sentences about different sub-topics are mixed together and it is difficult for a reader to grasp the meaning of this paragraph.

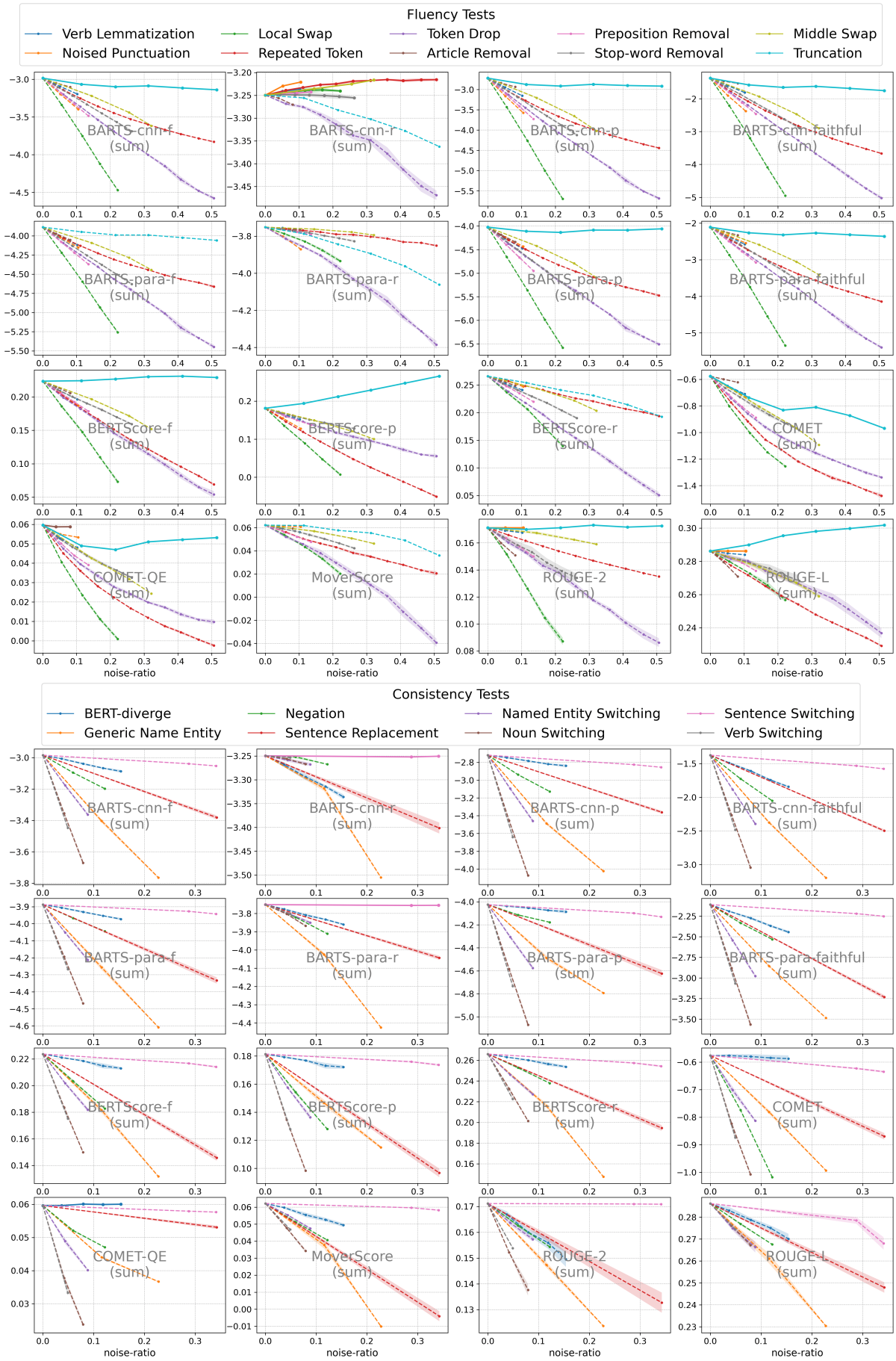


Figure 12: All results for fluency and consistency tests on the CNNDM dataset.

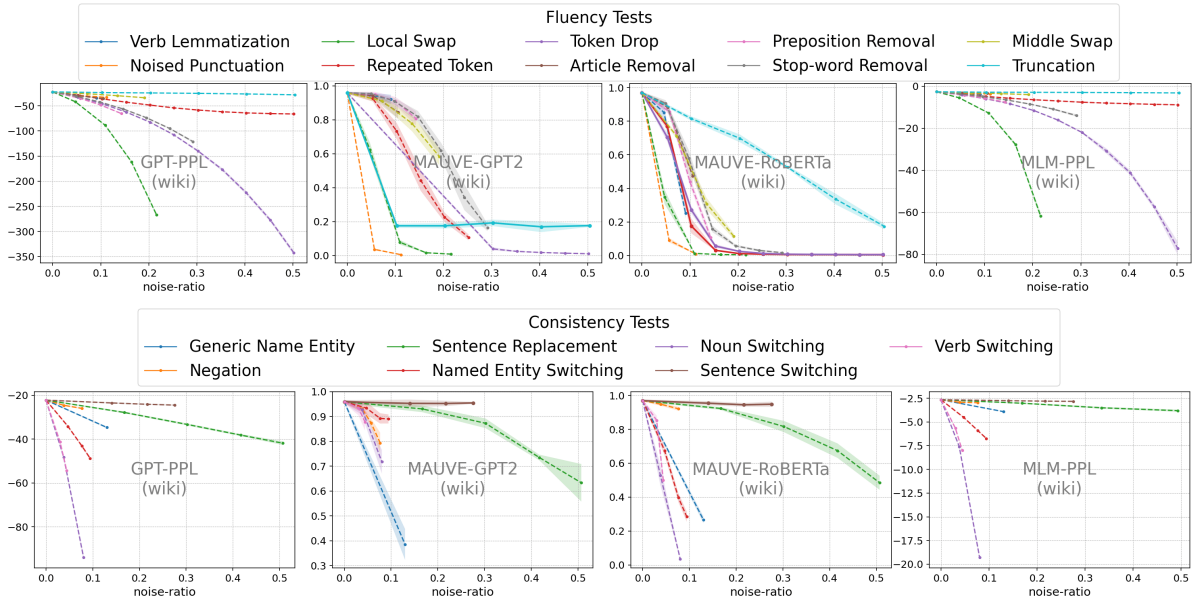


Figure 13: All results for fluency and consistency tests on the WikiText dataset.

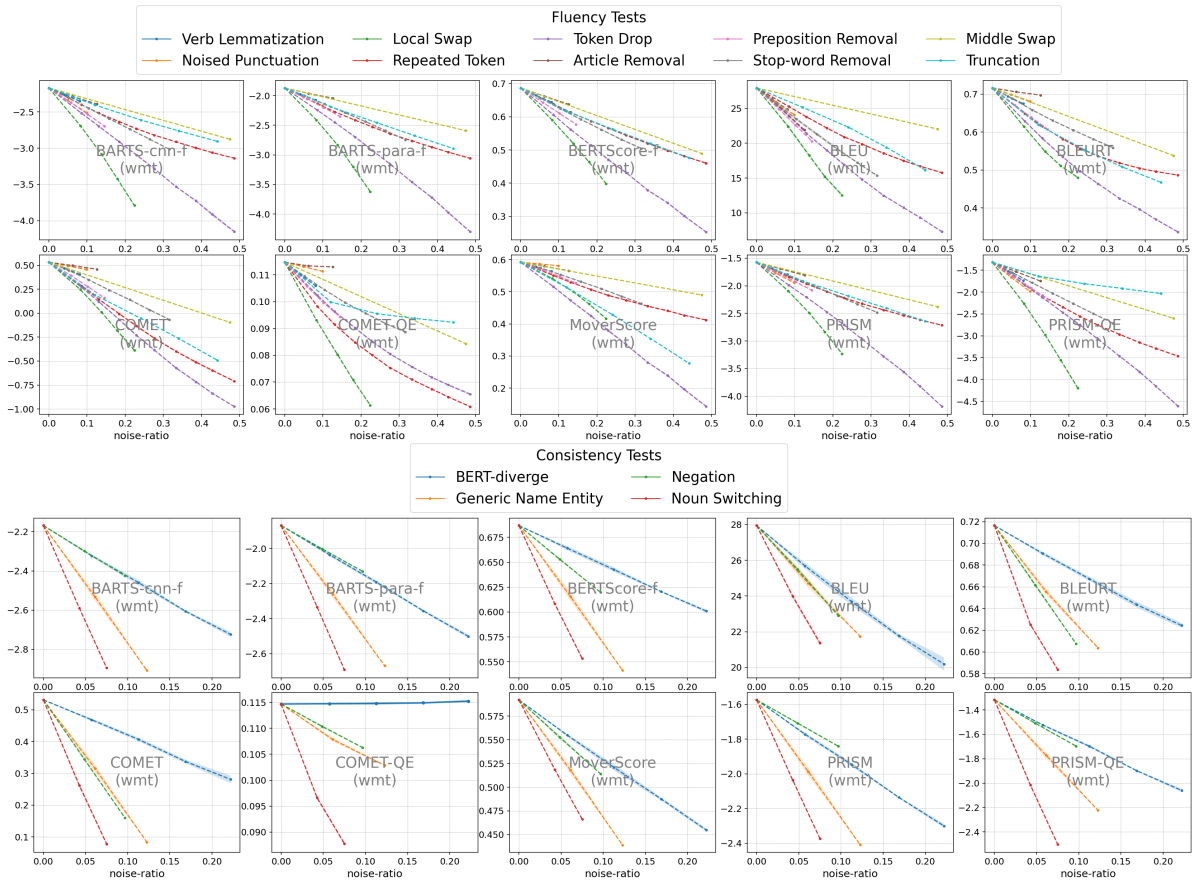


Figure 14: All results for fluency and consistency tests on the WMT dataset.