Demonstrations Aren't All You Need For Long-form Generation! Learning Task-Inherent Attribute Guidelines For Large Language Models

Anonymous ACL submission

Abstract

We study the sufficiency of demonstrations in enabling pre-trained large language models (LLMs) to implicitly learn the underlying task distribution for long-form generation. We prove the answer is no. For any long-form generation task, we show that if an LLM fails to initially grasp the task's language distribution, demonstrations alone are insufficient. This gap is caused by a lack of explicit task-language distribution characterization exposed to the model. Addressing this by capturing these distributions explicitly through task guidelines enhances model performance. We then present LongGuide, the first efficient algorithm that generates two types of guidelines as additional instructions for LLMs: *(i) Metric Guideline* (MG) that instructs models to optimize for selected metrics; and *(ii) Output Constraint Guideline* (OCG) that constrains generation at both the token and sentence levels. LongGuide automatically selects the most useful combination of guidelines, improving strong open- and closed-source LLMs by 5.39% and 6.58% under zero- and few-shot settings across seven tasks. Furthermore, LongGuide enhances LLMs beyond demonstrations, is learnable by weaker models to enhance stronger ones, and synergistically combines with prompt optimizers.

1 Introduction

In recent years, pre-trained large language models (LLMs) have demonstrated impressive instruction-based performance through zero- and few-shot learning capabilities (Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2022; Touvron et al., 2023; Jiang et al., 2023; Team et al., 2023). Notably, few-shot learning, termed as in-context learning (ICL), has proven highly effective and widely used (Dong et al., 2022). Let us first state ICL formally: the goal is to predict a token sequence Y, given another token sequence X, for a task T with a probability distribution P_T. In this context, ICL

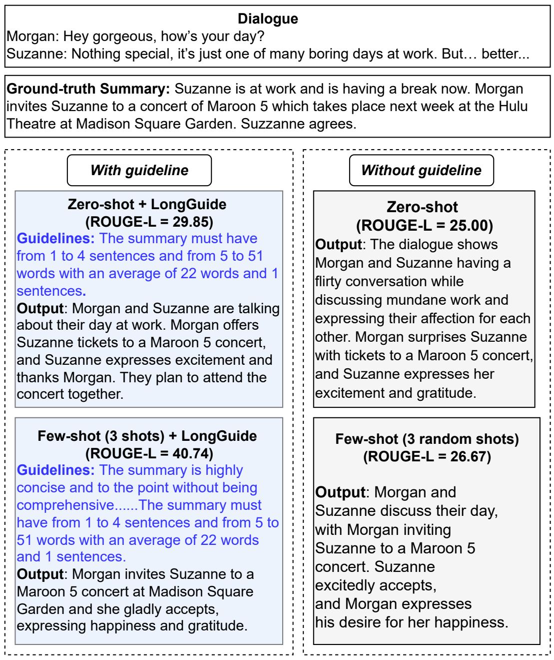


Figure 1: ChatGPT results on a SAMSum example (Gliwa et al., 2019) w/ & w/o LongGuide guidelines.

generation using an LLM $\mathcal{M}$ involves concatenating k demonstrations $\{(x_1, y_1), ..., (x_k, y_k)\}$. Let D_f denote the random variable for the demonstration token sequence. Then, we define $P_{\mathcal{M}}$ as $\mathcal{M}$'s output probability function as:

$$P_{\mathcal{M}}(Y|D_f, X) := \mathcal{M}(Y|Concat(x_1, y_1, ..., x_k, y_k), X) \quad (1)$$

where $D_f = Concat(x_1, y_1, ..., x_k, y_k)$. For simplicity, we omit formatting tokens of demonstrations and separator tokens between examples.

Several prior studies try to explain the ICL capabilities of LLMs, advocating for the sufficiency of well-chosen D_f as implicitly teaching the LLM to perform the tasks, especially those involving classification (Saunshi et al., 2020; Xie et al., 2021; Wang et al., 2024). Central to their theoretical analyses is a strong assumption that the model $\mathcal{M}$ accurately captures the underlying distribution of the task's language; i.e., $P_{\mathcal{M}}(X) = P_T(X)$.

However, this assumption is often not met, particularly with domain-specific terminologies (Yang et al., 2023a; Cheng et al., 2024), questioning the sufficiency of demonstrations. Furthermore, recent empirical studies highlight the deficiency of ICL in long-form generation tasks where answers are sentences or paragraphs such as summarization (Sun et al., 2023a). This poses significant gaps in our understanding of ICL's limitations and instructing LLMs to solve such tasks effectively.

We question the proficiency of demonstrations for long-form generation tasks. We prove that *for any long-form generation task, if a language model fails to grasp the task's language distribution initially, demonstrations cannot correct this deficiency*. We then hypothesize and empirically verify that LLMs do not fully transfer the text properties (language and format properties) of demonstrations to generated (long-form) answers. Based on this, we posit that instructing LLMs with explicit task guidelines that capture the text properties of the task comprehensively is essential for LLMs to enhance their performance. Fig. 1 illustrates such an example where instructing LLMs explicitly by guidelines carrying certain properties of the task output distribution leads to superior outcomes.

Motivated by this, we introduce LongGuide, a five-step, efficient guideline-learning algorithm that generates two streams of guidelines as supplementing instructions for LLMs from limited training data: *(i) Metric Guideline (MG)* directing models toward optimizing selected metrics on the task, motivated by prior studies in machine translation (Ranzato et al., 2015); *(ii) Output Constraint Guideline (OCG)* constraining generated outputs at both sentence and token levels, inspired by controllable generation studies (Fan et al., 2018a). Our method is related to prior studies in task instruction construction (Wang et al., 2022b) and enhancing LLM task understanding through task definitions (Yin et al., 2023). However, it differs by offering "post-hoc" instructions that guide LLMs to enhance responses based on learned quality and quantitative criteria.

LongGuide automatically identifies the optimal set of guidelines, resulting in significant overall performance enhancements for both open- and closed-source LLMs by 5.30% and 6.20%, respectively, across seven tasks including summarization, text simplification, translation, dialogue generation, table2text generation. Moreover, it learns guidelines from demonstrations boosting ICL performance, can be learned by weaker models to boost stronger models, and be developed concurrently and integrated with prompt optimization algorithms.

2 Demonstrations Alone Are Insufficient for Long-form Generation

Problem formulation. We define a long-form generation dataset with n data points as $D = \{\langle x, y\rangle_i\}_{i=1}^n$, where x and y respectively indicate the input context and ground truth *sentence- or paragraph-long* answer. Without loss of generality, X denotes the random variable for the input token sequence of x, and Y denotes the answer token sequence of y. An LLM $\mathcal{M}$ solving the task in the **instruction-based** setting is expected to generate Y given X and an input Instruction I.

2.1 Theoretical Analysis

Assumption 2.1. *For the test long-form generation task T that we consider, there exists $x \in \mathcal{X}$ for which $P_{\mathcal{M}}(X = x) \neq P_T(X = x)$, where $\mathcal{X}$ is the input token sequence space.*

Asm.-2.1 is equivalent to $\mathcal{M}$ does not fully capture T's true language distribution. We assume:

Assumption 2.2. *We define two probability functions as **functionally zero equivalent** if they act on the same input space and any arbitrary event causes both functions to be simultaneously zero or non-zero. We assume that P_T and $P_{\mathcal{M}}$ are **functionally zero equivalent**, i.e., $\forall x \in \mathcal{X}$, $P_{\mathcal{M}}(X = x) = 0 \Leftrightarrow P_T(X = x) = 0$.*

Note that Asm.-2.1 contradicts the common assumption $P_{\mathcal{M}}(X) = P_T(X)$ made by multiple prior studies (Xie et al., 2021; Min et al., 2022; Wang et al., 2024), while Asm.-2.2 is a relaxed version of that common assumption. With the above assumptions, we prove the following result:

Theorem 2.1. *For any demonstration token sequence $D_f \in \mathcal{D}$, the distribution $P_{\mathcal{M}}(X|D_f)$ does not fully approximate $P_T(X)$ i.e, there exists $x \in \mathcal{X}$ such that $P_{\mathcal{M}}(X = x|D_f) \neq P_T(X = x)$.*

where $\mathcal{D}$ is the demonstration token sequence space. The proof of Thm.-2.1 is presented in Appx.-A. In short, this proof shows that if a language model fails to grasp the generation task's language distribution (Asm.-2.1), demonstrations cannot correct this deficiency. This finding reveals flaws in our beliefs about demonstrations in ICL, suggesting we rethink methods to assist LLMs in characterizing their tasks in terms of language distribution, which is essential for long-form generation to ensure that outputs accurately reflect the task's requirements. This is the key distinction between long-form generation and classification, since in classification tasks, the output may not necessarily reflect the language properties of the input provided.

In practice, evaluating how accurately $\mathcal{M}$ captures the language distribution of a task T is highly challenging because the true distribution P_T is unknown. The widely adopted approach is by analyzing $\mathcal{M}$'s responses on testing samples of T using reference-based evaluation metrics such as ROUGE (Lin, 2004) and/or reference-free ones like Fluency (Fu et al., 2023; Zeng et al., 2020). While reference-based metrics are commonly used to assess $\mathcal{M}$'s performance on task T, reference-free metrics are typically employed to evaluate the linguistic properties of the answers.

Since $\mathcal{M}$ does not fully capture the task's language distribution even with D_f as input, hypothetically, it does not entirely transfer the linguistic properties of demonstrative outputs into the newly generated ones. This implies the existence of at least one reference-free language evaluation metric whose scores on $\mathcal{M}$'s generated answers do not wholly result from its score distribution of demonstrative answers. From our empirical explorations verifying this hypothesis in Appx.-C.1, we further discover that not only language properties but the text formatting properties (e.g., # of sentences) are not fully transferred from demonstrations. Therefore, we generalize this hypothesis. Denoting $\mathcal{G}_\mathcal{M} : \mathcal{X} \mapsto \mathcal{Y}$ as the generation function of $\mathcal{M}$, we propose the following theorem:

Theorem 2.2. *Suppose that there exists $x \in \mathcal{X}$ such that $P_\mathcal{M}(X = x) \neq P_T(X = x)$, for any finite set of demonstrations $\{(x_i, y_i)\}_{i=1}^k$, there exists at least one text property (language or format property) metric $E : \mathcal{X} \mapsto \mathbb{R}$ such that $\exists x \in \mathcal{X}$ so that $E(\mathcal{G}_\mathcal{M}(x)) \notin \{E(y_i)\}_{i=1}^k$.*

Thm.-2.2 is equivalent to there exists one language/format metric such that $\mathcal{M}$ cannot fully transfer its level from demonstrations to responses, regardless of how many finite demonstrations are used. Our proof is in Appx.-A with empirical supporting evidence in Appx.-C.1. This highlights a significant limitation of demonstrations: if they do not cover all possible outcomes of $\mathcal{G}_\mathcal{M}$ in $\mathcal{Y}$, which is often the case, they alone cannot enable $\mathcal{M}$ to fully integrate the text properties into responses.

Generalizing from the demonstrations to limited labeled data, we term this as the **text property transfer (PT) problem**: the challenge of ensuring that a model $\mathcal{M}$ can transfer specific desired text properties observed in a limited set of labeled data, such as demonstrations, to its responses. These findings partly explain why ICL is not an effective strategy for long-form tasks, as empirically found by (Sun et al., 2023a; Pu et al., 2023). We hypothesize that addressing PT problem enhances the instruction-based performance of $\mathcal{M}$. To formally study this hypothesis, we define text property task:

Definition 2.1. (Text property task) For a task $T \triangleq \{D, \mathcal{L}\}$ with the train dataset $D = \{(x_i^t, y_i^t)\}_{i=1}^n$, a text property task T' of T with a property measurement $f' : \mathcal{Y} \mapsto \mathbb{R}$ is defined as $T' \triangleq \{D', \mathcal{L}'\}$ such that $D' = \{(x_i^t, f'(y_i^t))\}_{i=1}^n$.

where $\mathcal{L}$ and $\mathcal{L}'$ are the learning objectives of T and T' respectively, and f' can be any *reference-free* language property or format property measurement. Let us denote the long-form text generation objective of a language model $\mathcal{M}$ for a task T as $\min_{\theta\in\Theta} \mathcal{L}(\theta, T)$, with θ is a tunable factor of $\mathcal{M}$ (such as its parameters or input instruction), and Θ is its space. With Definition 2.1, we propose:

Hypothesis 2.1. *(**LongGuide Hypothesis**) We hypothesize that T can be decomposed into r well-chosen **text property tasks** $T_1, ..., T_r$ with corresponding objectives $\mathcal{L}_1, .., \mathcal{L}_r$ such that when r is large enough, $T \approx T_1 \oplus ... \oplus T_r$. By jointly optimizing r text property task objectives $\mathcal{L}_1, .., \mathcal{L}_r$, we can approximately optimize the original task loss $\mathcal{L}$:* $\arg\min_{\theta\in\Theta} \sum_{i=1}^r \mathcal{L}_i \approx \arg\min_{\theta\in\Theta} \mathcal{L}$.

When it comes to our instruction-based objective, θ becomes I. Essentially, Hyp.-2.1 proposes a solution to enhance the performance of LLMs by addressing the PT problem, which involves optimizing responses based on the text property distributions present in the training data. We provide an empirical evidence supporting it in §4.1. Note that our proposed hypothesis differs from previous performance optimization approaches (Ranzato et al., 2015; Wieting et al., 2019), which primarily focus on single reference-based metrics like BLEU (Papineni et al., 2002), as well as generalizes prior efforts to optimize certain reference-free metrics to enhance model performance, such as relevancy (Gao et al., 2019). Additionally, while Hyp.-2.1 offers an alternative to optimizing reference-based metrics with large datasets, it demonstrates superior effectiveness under limited data constraints when

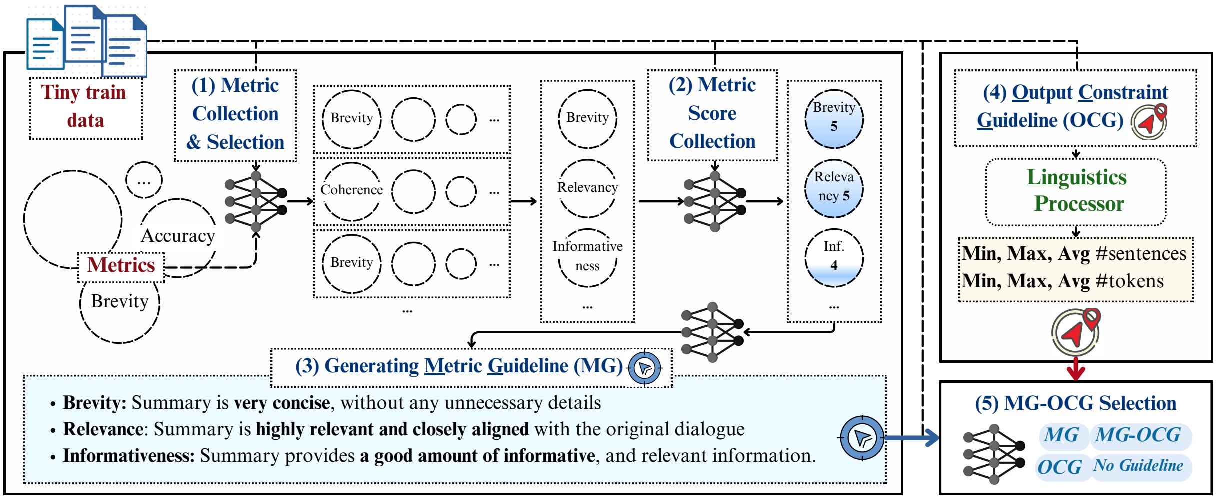


Figure 2: Overview of our LongGuide framework. Light blue and yellow boxes show the learned (in parallel) metric guideline (MG) and output constraint guideline (OCG).

we choose appropriate text property tasks. This is evident when we compare our method to one of the strongest prompt optimization algorithms in §4.1 which optimizes ROUGE directly.

Our experiments verifying Thm.-2.2 reveal that providing simple **guidelines** instructing LLMs to optimize certain text property metrics can enhance those properties in the responses, possibly because LLMs are optimizers (Yang et al., 2024). Based on Hyp.-2.1, our solution to enhance LLM performance by mitigating the PT problem is to develop a framework that automatically learns crucial guidelines as additional instructions for LLMs to optimize under the limited data constraints.

3 LongGuide

Motivation. Based on findings in §2, we propose LongGuide for improving LLM performance by addressing the PT problem. LongGuide self-employs crucial text property tasks to capture task distribution efficiently as shown in Fig. 2. To ensure our method is efficiently generalizable to new tasks, we only assume access to at most 50 task training samples: $D^{train} = \{(x_i^t, y_i^t)\}_{i=1}^n$, $n \leq 50$. We focus on two guidelines, concerning two streams of text properties: (1) reference-free evaluation metrics capturing the intrinsic properties of a text (*Metric Guideline (MG)*, Steps 1-3), and (2) output constraint metrics capturing the format that the generated text must adhere to (*Output Constraint Guideline (OCG)*, Step 4). Finally, LongGuide automatically evaluates different combinations of MG and OCG on D^{train} to determine the best guideline(s) as additional instructions for testing (Step 5). Let $\mathcal{M}$ be the LLM and $\mathcal{G}_{\mathcal{M}}$ be its generation function. Below, we outline 5 steps of LongGuide in detail and prompts are in Appx.-E.

Step 1: Metric Collection & Selection. We aim to select suitable metrics using the LLM $\mathcal{M}$ to reason their importance for training data batches. We first construct our pool of evaluation metrics S widely used for text generation tasks. S consists of 27 distinct metrics from 4 main sources (Appx.-Tab. 9 for details). Specifically, we include 3 metrics from ABC's of Communication (Wagner, 1963) evaluating communication skills. We follow previous work (Yuan et al., 2021; Fu et al., 2023) to include 12 more metrics for dialogue generation, summarization, data2text generation, and machine translation. We further propose 12 metrics for a broader spectrum. We do not collect the metrics' definitions as they may differ across tasks.

Given D^{train} and S, we perform K iterations to select the important metrics. At each iteration, we randomly sample a batch $B_i \subseteq D^{train}$ and instruct $\mathcal{M}$ to generate the top-5 most important metrics in S for evaluating the quality of the outputs in B_i: $T_i = \mathcal{G}_{\mathcal{M}}(I_M, B_i, S)$ with I_M being the instruction to $\mathcal{M}$ to generate top-5 metrics. We apply the top-5 constraint to prevent selecting too many metrics. The final set of metrics selected, denoted by M, consists of the metrics selected across all iterations *sorted in alphabetic order* to ensure consistent results across multiple runs: $M = sorted(T_1 \cup \dots \cup T_K) = \{M_1, ..., M_m\}$.

Step 2: Metric Score Collection. This step focuses on evaluating the selected metrics M on task data for comprehensively capturing the task properties. Motivated by prior studies (Wang et al., 2023a), we utilize $\mathcal{M}$ to score the metrics on a scale of 1–5. On i^{th} training sample $(x_i^t, y_i^t) \in D^{train}$, we infer $\mathcal{M}$ to evaluate y_i^t on the metrics: $scores_i = \mathcal{G}_{\mathcal{M}}(I_{score}, x_i^t, y_i^t, M) = \{s_{i_{M_1}}, ..., s_{i_{M_m}}\}$, where I_{score} is the instruction to score the metrics. We employ self-consistency (Wang et al., 2022a) to obtain the evaluation scores minimizing variance. The final scores, $scores_M = \{s_{M_1}, ..., s_{M_m}\}$, are the average of scores over all data outputs with $s_{M_j} = \frac{\sum_{i=1}^{n}(s_{i_{M_j}})}{n}$. We separate this step from metric selection in Step 1 because, once a metric is chosen, we aim to evaluate it on D^{train}, not just the samples that led $\mathcal{M}$ to select it.

Step 3: Generating Metric Guidelines. After obtaining $scores_M$, the goal of this step is to generate metrics' definitions moderated by $scores_M$, which serves as the Metric Guideline (MG): $G^M = \{d_{M_1}^{MG}, ..., d_{M_m}^{MG}\} = \mathcal{G}_{\mathcal{M}}(I_{MG}, scores_M, M)$, where I_{MG} is the instruction for $\mathcal{M}$ to generate the moderated definitions. We use the moderated metrics' definitions instead of $scores_M$ because they are more expressive. Fig. 2 illustrates an instance where "Inf." in the task "dialogue sum." achieving a score of $4/5$ is defined as "The summary provides a good amount of inf...". Essentially, G^M delineates the expected properties of the answers that $\mathcal{M}$ must uphold during generation.

Step 4: Output Constraint Guideline (OCG). Research on controlling long-form generation output has extensively proposed various constraints. These include constraints on the length, which are broadly applicable, as well as linguistic or keyword-based controls on the output, which are more specific to certain tasks (Fan et al., 2018a; Martin et al., 2020; He et al., 2022). Our target in this step is to propose a robust set of output constraints which are the universal applicability of LongGuide to any long-form generation tasks. We develop LongGuide specifically to learn six key output constraints, focusing on two distributions: #sentences and #tokens in ground-truth answers. These include minimum (MIN), maximum (MAX), and average (AVG) counts of sentences and tokens, which serve as basic exploratory statistics about length bins and specific expected values of these distributions. The Output Constraint Guideline (OCG) is formulated as G^{OC} = "The response must have from $\{MIN_s\}$ to $\{MAX_s\}$ sentences and from $\{MIN_t\}$ to $\{MAX_t\}$ tokens with an average of $\{AVG_s\}$ sentences and $\{AVG_t\}$ tokens.".

Step 5: MG–OCG selection. The inherent knowledge of various models for different tasks varies, leading to G^M and G^{OC} demonstrating varying degrees of enhancement. This step targets mitigating this by automatically selecting the best combination of guidelines for a given model. Specifically, we assess the model's performance on the limited training data D^{train} under 4 guideline settings G = {w/o guideline, G^M, G^{OC}, G^M & G^{OC}}. The best-performing combination on D^{train} is then the final LongGuide: $G^{best} = \arg\max_{g \in G}(performance(\mathcal{M}|I, g, D^{train}))$ with I being the task input instruction (§2).

4 Experiments

Task selection. We select 7 widely evaluated long-form generation tasks from 4 main categories: *summarization, text simplification, machine translation and generation*. The tasks are SAMSum (Gliwa et al., 2019), CNN/Daily Mail (3.0.0) (See et al., 2017) and XL-SUM (Hasan et al., 2021) for summarization, SWiPE (Laban et al., 2023) for text simplification, IWSLT-2017 en-ja (Cettolo et al., 2017) for machine translation, Synthetic-Persona-Chat (Jandaghi et al., 2023) for dialogue generation, and CommonGen-Challenge (Lin et al., 2020) for data-to-text generation. Our data preprocessing details are provided in Appx.-D.

Baselines and evaluation. Since LongGuide is the first method of self-learning guidelines as additional instructions for long-form generation, we compare it with the zero-/few-shot prompting baselines. We also compare it with one of the strongest prompt optimization algorithms, APO (Pryzant et al., 2023) which optimizes the input prompt on the D^{train}. More baselines are in §5.1 and Appx.-B.3. We empirically examine both open- and closed-source LLMs: *Mistral-7B-it v0.2* (Jiang et al., 2023) as an open-source model and *ChatGPT (gpt-3.5-turbo-1106)* (OpenAI, 2022) as a closed-source model. Both are among the strongest LLMs to date. Our main evaluation metric is ROUGE-L (Lin, 2004). The results we report are averaged over 3 runs, with 95% CI from t-tests.

4.1 Findings

LongGuide significantly mitigates the PT problem. We show that LongGuide effectively addresses the PT problem identified in §2. Our experimental results are presented in Tab. 1, conducted on

Models	Method	SAMSum ROUGE-L↑	SAMSum Avg.JS↓	CNN ROUGE-L↑	CNN Avg.JS↓	SWiPE ROUGE-L↑	SWiPE Avg.JS↓
Mistral-it (0.2)	Zero-shot (ZS)	$22.20_{\pm 0.43}$	0.10139	$19.23_{\pm 0.34}$	0.12623	$36.60_{\pm 0.59}$	0.05647
	ZS w/ OCG	$27.55_{\pm 0.98}$	0.04015	**22.46**$_{\pm 0.64}$	**0.07178**	$32.48_{\pm 1.91}$	0.06500
	ZS w/ MG	$27.81_{\pm 1.17}$	0.03880	$18.35_{\pm 0.60}$	0.14130	**38.21**$_{\pm 1.72}$	**0.05496**
	ZS w/ MG-OCG	**28.35**$_{\pm 1.66}$	**0.03746**	$22.05_{\pm 0.84}$	0.07885	$35.47_{\pm 2.89}$	0.05538
	ZS w/ LongGuide	**28.35**$_{\pm 1.66}$	**0.03746**	**22.46**$_{\pm 0.64}$	**0.07178**	**38.21**$_{\pm 1.72}$	**0.05496**
	Few-shot (FS)	$27.13_{\pm 0.26}$	0.05018	$17.56_{\pm 0.63}$	0.08436	$39.47_{\pm 0.45}$	0.04691
	FS w/ OCG	$27.84_{\pm 0.88}$	0.03362	$15.20_{\pm 5.28}$	0.09218	$29.54_{\pm 1.90}$	0.05961
	FS w/ MG	$27.50_{\pm 2.08}$	0.03518	$18.13_{\pm 5.28}$	0.08301	**41.36**$_{\pm 1.37}$	**0.04503**
	FS w/ MG-OCG	**30.65**$_{\pm 0.88}$	**0.03184**	**19.19**$_{\pm 0.49}$	**0.08139**	$38.56_{\pm 1.39}$	0.05289
	ZS w/ LongGuide	**30.65**$_{\pm 0.88}$	**0.03184**	**19.19**$_{\pm 0.49}$	**0.08139**	**41.36**$_{\pm 1.37}$	**0.04503**

Table 1: Avg. Jensen–Shannon divergence scores across distributions of text properties of generated answers vs ground truths (ChatGPT judge): (1) the trends of ROUGE-L and Avg. JS is nearly identical, supporting our proposed Hyp.-2.1; (2) LongGuide significantly mitigates the PT problem.

		Summarization			Simplification	Translation	Dialogue Generation	Table2Text
	Method	**SAMSum**	**CNN (3.0.0)**	**XL-Sum**	**SWiPE**	**IWSLT17 en-ja**	**Synthetic Persona**	**CommGen-Chall.**
	#shots (random)	3	3	5	3	3	5	5
Mistral-it (0.2)	Zero-shot (ZS)	$22.20_{\pm 0.43}$	$19.23_{\pm 0.34}$	$9.19_{\pm 0.03}$	$36.60_{\pm 0.59}$	$13.12_{\pm 1.39}$	$12.76_{\pm 1.54}$	$10.12_{\pm 0.02}$
	ZS w/ APO	$23.77_{\pm 1.88}$	$19.53_{\pm 2.08}$	$12.06_{\pm 1.55}$	$36.92_{\pm 1.81}$	$14.45_{\pm 1.84}$	$10.66_{\pm 1.08}$	$11.21_{\pm 2.02}$
	ZS w/ LongGuide	**28.35**$_{\pm 1.66}$	**22.46**$_{\pm 0.64}$	**14.38**$_{\pm 0.15}$	**38.21**$_{\pm 1.72}$	**16.53**$_{\pm 0.59}$	**14.69**$_{\pm 1.08}$	**25.20**$_{\pm 1.89}$
	% gain over ZS	+6.15	+3.23	+5.19	+1.61	+3.41	+1.93	+15.08
	Few-shot (FS)	$27.13_{\pm 0.26}$	$17.56_{\pm 0.63}$	$9.79_{\pm 0.18}$	$39.47_{\pm 0.45}$	$12.69_{\pm 1.82}$	$3.56_{\pm 0.36}$	$3.98_{\pm 0.17}$
	FS w/ APO	$26.23_{\pm 2.22}$	$18.18_{\pm 2.01}$	$11.99_{\pm 1.46}$	$39.55_{\pm 2.07}$	$14.08_{\pm 1.97}$	$4.26_{\pm 1.45}$	$5.45_{\pm 0.92}$
	FS w/ LongGuide	**30.65**$_{\pm 0.88}$	**19.19**$_{\pm 0.49}$	**15.23**$_{\pm 0.33}$	**41.36**$_{\pm 1.37}$	**16.62**$_{\pm 0.81}$	**5.25**$_{\pm 0.94}$	**25.05**$_{\pm 0.76}$
	% gain over FS	+3.52	+1.63	+5.44	+1.89	+3.66	+1.69	+21.07
ChatGPT	Zero-shot (ZS)	$23.83_{\pm 0.54}$	$20.12_{\pm 0.27}$	$10.80_{\pm 0.18}$	$45.09_{\pm 1.45}$	$36.13_{\pm 0.87}$	$19.46_{\pm 0.40}$	$24.21_{\pm 0.37}$
	ZS w/ APO	$25.05_{\pm 1.32}$	$20.34_{\pm 0.91}$	$12.19_{\pm 1.30}$	**46.32**$_{\pm 1.92}$	$37.74_{\pm 1.54}$	$19.91_{\pm 1.62}$	$23.63_{\pm 1.99}$
	ZS w/ LongGuide	**30.47**$_{\pm 1.57}$	**22.19**$_{\pm 0.65}$	**20.93**$_{\pm 0.52}$	$45.09_{\pm 1.45}$	**41.22**$_{\pm 0.46}$	**22.98**$_{\pm 2.65}$	**34.41**$_{\pm 1.01}$
	% gain over ZS	+6.64	+2.07	+10.13	+0.00	+5.09	+3.52	+10.20
	Few-shot (FS)	$22.21_{\pm 2.35}$	$14.51_{\pm 0.80}$	$11.42_{\pm 0.13}$	$33.72_{\pm 2.61}$	$31.93_{\pm 1.88}$	$16.10_{\pm 2.61}$	$22.08_{\pm 0.63}$
	FS w/ APO	$24.22_{\pm 2.33}$	$15.20_{\pm 2.19}$	$14.07_{\pm 3.05}$	$34.46_{\pm 2.01}$	$33.72_{\pm 3.20}$	$17.68_{\pm 1.80}$	$25.09_{\pm 3.15}$
	FS w/ LongGuide	**31.46**$_{\pm 1.34}$	**18.17**$_{\pm 1.32}$	**19.95**$_{\pm 1.38}$	**37.60**$_{\pm 2.85}$	**38.43**$_{\pm 2.37}$	**22.36**$_{\pm 0.89}$	**38.21**$_{\pm 3.70}$
	% gain over FS	+9.25	+3.66	+8.53	+3.88	+6.50	+6.53	+16.13

Table 2: Main experiments on summarization, text simplification, translation, and long-form question-answering tasks. LongGuide significantly outperforms APO on most of the tasks and enhances instruction-based performance of LLMs substantially.

3 datasets SAMSum, CNN, and SWiPE with Mistral. We use different combinations of LongGuide as additional instructions for the model under zero-shot and few-shot settings. For each task, we first have the set of selected text properties from LongGuide that the model needs to optimize, denoted as $\{M_1, ..., M_m, \#sentences, \#tokens\}$ (for the full lists, see Appx.-Tab. 12). We then measure the average of Jensen-Shannon divergence (Lin, 1991) between the property score distributions (judged by ChatGPT) between the generated answers and the ground truth answers, across all selected properties, denoted as $Avg.JS$: the lower the $Avg.JS$ value, the better the mitigation of the PT problem. From Tab. 1, we observe that LongGuide significantly reduces the $Avg.JS$ score compared to the baselines, showcasing the success of using guidelines as additional instructions to enhance property transfer. Moreover, across all benchmarks, the trend of ROUGE-L scores is nearly identical with $Avg.JS$, providing strong evidence verifying Hyp.-2.1. A case study is shown in Appx.-C.3.

LongGuide significantly boosts instruction-based performance of LLMs. Our main experiments with LongGuide on downstream tasks, as presented in Tab. 2, reveal four primary findings. Firstly, interestingly, for baselines, zero-shot performance is higher than few-shot performance for both models on average, and the gaps are especially large in Synthetic Persona and CommonGen-Challenge. We hypothesize that the models might have been partly exposed to the tasks' data during training, therefore, supplementing demonstrations into the prompts (few-shot) makes them out-of-distribution: when additional demonstrations are provided, the models often refuse to answer the queries. Meanwhile, LongGuide helps models overcome this issue for the few-shot setting. Secondly, LongGuide substantially improves zero- and few-shot baselines by 5.30% and 6.20% on average across models: improvement for few-shot prompting is surprisingly higher than in zero-shot, possibly because improving a stronger baseline is harder than a weaker one. Notably, LongGuide outperforms APO (Pryzant et al., 2023) in most benchmarks, especially under zero-shot, demonstrating that our strategy of optimizing reference-free property tasks (Hyp.-2.1) is significantly more effective than optimizing ROUGE-L on limited data. Thirdly, we observe that LongGuide achieves the highest improvements on CommonGen-Challenge with 15.62% and XL-SUM with 7.32%, and lowest improvement on SWiPE with 1.84% on average. These improvements are mainly because the answers generated by the baselines are often far

Models	Method	SAMSum	CNN (3.0.0)	XL-Sum	SWiPE	IWSLT17 en-ja	Synthetic Persona	CommGen-Chall.
Mistral-7B-it (0.2)	Zero-shot (ZS)	22.20$_{\pm 0.43}$	19.23$_{\pm 0.34}$	9.19$_{\pm 0.03}$	36.60$_{\pm 0.59}$	13.12$_{\pm 1.39}$	12.76$_{\pm 1.54}$	10.12$_{\pm 0.02}$
	ZS w/ OCG	27.55$_{\pm 0.98}\uparrow$	**22.46**$_{\pm 0.64}\uparrow$	**14.38**$_{\pm 0.15}\uparrow$	32.48$_{\pm 1.91}\downarrow$	**16.53**$_{\pm 0.59}\uparrow$	14.35$_{\pm 0.47}\uparrow$	24.16$_{\pm 0.11}\uparrow$
	ZS w/ MG	27.81$_{\pm 1.17}\uparrow$	18.35$_{\pm 0.60}\downarrow$	9.37$_{\pm 0.25}\uparrow$	**38.21**$_{\pm 1.72}\uparrow$	8.71$_{\pm 0.53}\downarrow$	12.53$_{\pm 0.58}\downarrow$	21.54$_{\pm 7.50}\uparrow$
	ZS w/ MG-OCG	**28.35**$_{\pm 1.66}\uparrow$	22.05$_{\pm 0.84}\uparrow$	13.64$_{\pm 0.38}\uparrow$	35.47$_{\pm 2.89}\downarrow$	15.76$_{\pm 1.85}\uparrow$	**14.69**$_{\pm 1.08}\uparrow$	**25.20**$_{\pm 1.89}\uparrow$
	MG-OCG selection	*MG-OCG*	*OCG*	*OCG*	*MG*	*OCG*	*MG-OCG*	*MG-OCG*
	Few-shot (FS)	27.13$_{\pm 0.26}$	17.56$_{\pm 0.63}$	9.79$_{\pm 0.18}$	39.47$_{\pm 0.45}$	12.69$_{\pm 1.82}$	3.56$_{\pm 0.36}$	3.98$_{\pm 0.17}$
	FS w/ OCG	27.84$_{\pm 0.88}\uparrow$	15.20$_{\pm 5.28}\downarrow$	12.22$_{\pm 1.19}\uparrow$	29.54$_{\pm 1.90}\downarrow$	**16.62**$_{\pm 0.81}\uparrow$	5.06$_{\pm 1.05}\uparrow$	**25.05**$_{\pm 0.76}\uparrow$
	FS w/ MG	27.50$_{\pm 2.08}\uparrow$	18.13$_{\pm 5.28}\uparrow$	11.80$_{\pm 2.06}\uparrow$	**41.36**$_{\pm 1.37}\uparrow$	8.67$_{\pm 0.62}\downarrow$	4.32$_{\pm 0.39}\uparrow$	14.58$_{\pm 2.24}\uparrow$
	FS w/ MG-OCG	**30.65**$_{\pm 0.88}\uparrow$	**19.19**$_{\pm 0.49}\uparrow$	**15.23**$_{\pm 0.33}\uparrow$	38.56$_{\pm 1.39}\downarrow$	15.83$_{\pm 0.95}\uparrow$	**5.25**$_{\pm 0.94}\uparrow$	5.94$_{\pm 1.00}\uparrow$
	MG-OCG selection	*MG-OCG*	*MG-OCG*	*MG-OCG*	*MG*	*OCG*	*MG-OCG*	*OCG*
ChatGPT (1106)	Zero-shot (ZS)	23.83$_{\pm 0.54}$	20.12$_{\pm 0.27}$	10.80$_{\pm 0.18}$	**45.09**$_{\pm 1.45}$	36.13$_{\pm 0.87}$	19.46$_{\pm 0.40}$	24.21$_{\pm 0.37}$
	ZS w/ OCG	29.19$_{\pm 0.77}\uparrow$	**22.39**$_{\pm 0.82}\uparrow$	**20.93**$_{\pm 0.52}\uparrow$	37.76$_{\pm 1.44}\downarrow$	38.86$_{\pm 1.11}\uparrow$	**22.98**$_{\pm 2.65}\uparrow$	**34.41**$_{\pm 1.01}\uparrow$
	ZS w/ MG	25.38$_{\pm 0.79}\uparrow$	20.37$_{\pm 0.41}\uparrow$	10.42$_{\pm 1.15}\downarrow$	45.06$_{\pm 2.96}\downarrow$	37.88$_{\pm 2.42}\uparrow$	19.91$_{\pm 0.59}\uparrow$	17.23$_{\pm 2.57}$
	ZS w/ MG-OCG	**30.47**$_{\pm 1.57}\uparrow$	**22.19**$_{\pm 0.65}\uparrow$	20.02$_{\pm 0.89}\uparrow$	41.38$_{\pm 4.91}\downarrow$	**41.22**$_{\pm 0.46}\uparrow$	20.95$_{\pm 1.91}\uparrow$	31.57$_{\pm 0.99}\uparrow$
	MG-OCG selection	*MG-OCG*	*MG-OCG*	*OCG*	*ZS*	*MG-OCG*	*MG-OCG*	*OCG*
	Few-shot (FS)	22.21$_{\pm 2.35}$	14.51$_{\pm 0.80}$	11.42$_{\pm 0.13}$	33.72$_{\pm 2.61}$	31.93$_{\pm 1.88}$	16.10$_{\pm 2.61}$	22.08$_{\pm 0.63}$
	FS w/ OCG	30.00$_{\pm 1.07}\uparrow$	**18.17**$_{\pm 1.32}\uparrow$	**19.95**$_{\pm 1.38}\uparrow$	16.68$_{\pm 1.29}\downarrow$	38.57$_{\pm 1.81}\uparrow$	**22.36**$_{\pm 0.89}\uparrow$	38.12$_{\pm 1.99}\uparrow$
	FS w/ MG	29.43$_{\pm 0.83}\uparrow$	15.45$_{\pm 2.16}\uparrow$	12.49$_{\pm 0.59}\uparrow$	19.36$_{\pm 1.40}\downarrow$	**39.45**$_{\pm 3.55}\uparrow$	18.64$_{\pm 0.49}\uparrow$	22.18$_{\pm 7.50}\uparrow$
	FS w/ MG-OCG	**31.46**$_{\pm 1.34}\uparrow$	14.84$_{\pm 2.58}\uparrow$	18.58$_{\pm 0.44}\uparrow$	**37.60**$_{\pm 2.85}\uparrow$	38.43$_{\pm 2.37}\uparrow$	19.47$_{\pm 1.20}\uparrow$	**38.21**$_{\pm 3.70}\uparrow$
	MG-OCG selection	*MG-OCG*	*OCG*	*OCG*	*MG-OCG*	*MG-OCG*	*OCG*	*MG-OCG*

Table 3: Ablation results on seven tasks. The gray rows represent baselines. The benefits of LongGuide's components vary across different models and tasks. The "MG-OCG selection" results are reported in Appx.-Tab. 10.

longer than the ground truths. LongGuide rectifies this issue by controlling the output length and quality, leading to significant performance gains. Finally, among the two models, interestingly, LongGuide improves Mistral by an average of 5.39%, while ChatGPT, commonly regarded as a stronger model, is improved by a larger margin, 6.58%. This suggests that LongGuide has the potential to benefit stronger models in the future.

Where do the improvements come from? To identify the primary source of improvements (whether from MG, OCG, or both), we present the results of LLMs with LongGuide's components in Tab. 3. Firstly, MG-OCG combination (*w/ MG-OCG*) is the most useful guideline for LLMs, observed to be the best 15 times, followed by OCG (*w/ OCG*) observed 10 times, and MG (*w/ MG*) twice. While these statistics underscore the effectiveness of combining MG-OCG, OCG particularly proves itself highly effective in tasks such as summarization, translation, and table-to-text generation. Secondly, MG and OCG individually improve most of the baselines, with OCG showing a slight overall advantage. This could be because while MG focuses on controlling the language properties of answers, it does not manage the output structure, sometimes resulting in longer/shorter answers than the ground truths. Exceptionally, on SWiPE, OCG affects all models, whereas MG shows particularly strong effectiveness with Mistral. Manual investigations reveal that ground-truth answers in SWiPE exhibit high variances in #sentences and #tokens which explains why OCG may not be effective for this benchmark. Thirdly, an interesting case is ChatGPT with few-shot prompting on SWiPE, where individual MG and OCG impair performance but

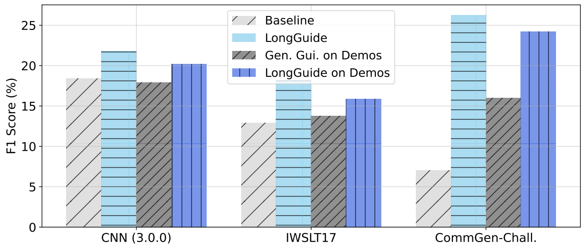


Table 4: LongGuide learned from demonstrations substantially enhances Mistral few-shot performance.

their combination enhances it. This shows evidence that MG and OCG complement each other. As discussed above, due to the uneven nature of answers in SWiPE, using MG or OCG alone may not work well for multiple samples, as MG and OCG only provide expected statistics. However, combining them could enhance performance by allowing them to complement each other. An illustrative SWiPE example of complement is in Appx.-Fig. 9.

5 Discussion

We discuss two key characteristics here, while Appendices B and C contain additional properties and analyses: (1) Understanding MG and OCG, the distributions of selected metrics and evaluated scores (Appx.-B.1); LongGuide is (2) transferable from weaker to stronger models (Appx.-B.2); (3) beneficial for non-instruct LLMs (Appx.-B.4); (4) synergistically combined with prompt optimizers (Appx.-B.3); (5) Extra ablation studies for #tokens and #sentences (Appx.-C.7); (6) Generalizability and customization of LongGuide (Appx.-C.8).

5.1 LongGuide Learns From Demonstrations To Boost ICL Performance

Here, we revisit the question posed in §2 and demonstrate that LongGuide learned from demon-

Methods	SAMSum	SWiPE	CommGen-Chall.
Zero-shot (ZS)	$22.20_{\pm 0.43}$	$36.60_{\pm 0.59}$	$3.98_{\pm 0.17}$
ZS w/ LongGuide	$\mathbf{28.35}_{\pm 1.66}$	$\mathbf{38.21}_{\pm 1.72}$	$\mathbf{25.20}_{\pm 1.89}$
ZS w/ LongGuide w/o step 2	$26.99_{\pm 1.61}$	$36.90_{\pm 1.91}$	$25.03_{\pm 2.01}$
Few-shot (FS)	$27.13_{\pm 0.26}$	$39.47_{\pm 0.45}$	$3.98_{\pm 0.17}$
FS w/ LongGuide	$\mathbf{30.65}_{\pm 0.88}$	$\mathbf{41.36}_{\pm 1.37}$	$\mathbf{27.23}_{\pm 0.58}$
FS w/ LongGuide w/o step 2	$30.37_{\pm 1.07}$	$35.54_{\pm 1.10}$	$27.15_{\pm 1.09}$

Table 5: Main ablation study with Mistral with LongGuide when Step 2 is skipped.

strations can significantly enhance ICL performance. Our experiments using Mistral cover CNN, IWSLT17 en-ja, and CommGen-Chall. datasets. Our results, presented in Tab. 4, involve averaging the performance under zero- and few-shot settings. For "Baseline", no guideline is utilized. For "LongGuide on Demos", we train LongGuide on demonstrations used in Tab. 2, in contrast to the D^{train} for the case of "LongGuide". We add one more baseline, "General Guidelines (Gen. Gui.) on Demos", where we ask the models to generate general task guidelines from demonstrations. The performance is summarized in Tab. 4, with detailed component results in Appx.-Tab. 11. Specifically, LongGuide trained on D^{train} outperforms it on demonstrations, suggesting its possible scalability with more training data. Moreover, while Gen. Gui. slightly worsens the Baseline on CNN, both LongGuide and LongGuide on Demos notably surpass the Baseline, and Gen. Gui., highlighting the effectiveness of LongGuide in capturing task-specific properties, thereby enhancing ICL performance.

5.2 Main Ablation Studies

From Tab. 3, we identify the unique contributions of each step within LongGuide. Notably, omitting Step 1 transforms LongGuide into OCG, whereas excluding Step 3 yields MG, and skipping Step 4 yields MG-OCG. We now investigate LongGuide under the condition of skipping Step 2, Metrics' scores collection. Essentially, for selected metrics from Step 1, we directly task the models to optimize them for the generated answers. As discussed in §3 (and Appx.-B.1), Step 2 is crucial for accurately capturing the task output properties for generation and avoiding conflicts among the metrics selected by MG. We experiment with Mistral on SAMSum, SWiPE, and CommGen-Chall. datasets because for these datasets, the best guideline combination involves MG. The results are presented in Tab. 5. As expected, without Step 2, the model performs worse, particularly for SAMSum and SWiPE where the highest drops are shown in the zero-shot setting. A case study is provided in Appx.-Fig. 11.

6 Related Work

Automatic prompt design for long-form generation. Long-form generation tasks are essential and have been studied extensively (Li et al., 2024). With LLM advancements, adapting these models for such tasks using prompt-based methods is critical. However, prior works (Bang et al., 2023; Yang et al., 2023b; Hadi et al., 2023; Zhou et al., 2023b; Pan et al., 2024) highlight the limited efficacy of LLMs in producing outputs that resemble ground truths, as evaluated by ROUGE-L (Lin, 2004). Our approach autonomously composes additional contexts, integrating evaluation targets and constraints. Additionally, enhancing instructions for LLMs (Wang et al., 2022b; Yin et al., 2023; Wang et al., 2023b), automatic prompt optimization (Zhou et al., 2023a; Pryzant et al., 2023), and demonstration selection (Yang et al., 2023c; Qin et al., 2023) are related areas that can be developed in parallel & combined with ours (Appx.-B.3).

Prompting for controllable generation. Controllable generation during fine-tuning has been extensively studied (Fan et al., 2018a; Lakew et al., 2019; Martin et al., 2020; He et al., 2022). More recently, researchers have explored prompting methods to control LLM generation. For instance, (Sun et al., 2023b) found that LLMs struggle to meet fine-grained hard constraints, while (Fonseca and Cohen, 2024) proposed controlling stylistic features like keywords and narrative during generation, leading to improved LLM summarization outcomes. Although (Fonseca and Cohen, 2024) is closely related to our output constraint guideline (OCG), our approach goes beyond summarization features, as discussed in §3. We focus on universally applicable features across multiple tasks.

7 Conclusion

We provide a theoretical understanding of the deficiencies of demonstrations alone in instructing large language models (LLMs) on the language & format (text) properties of long-form generation tasks, supported by illustrative evidences. To address this, we propose LongGuide, an efficient, guideline-learning algorithm that automatically identifies the crucial text properties and converts them into textual guidelines for LLMs. LongGuide enhances the performance of LLMs on these tasks significantly and shows promise for various downstream applications with minimal data required.

Limitations

Our study has several limitations. One limitation of our theoretical analysis is that it focuses solely on the task language distribution which is $P_{\mathcal{M}}(X)$ or $P_{\mathcal{M}}(X|D_f)$ instead of the actual output distribution, which is $\arg\max_{y \in \mathcal{Y}} P_{\mathcal{M}}(Y = y \mid X)$ or $\arg\max_{y \in \mathcal{Y}} P_{\mathcal{M}}(Y = y \mid D_f, X)$. In our study, while leveraging the task language distribution allows us to hypothesize and highlight the limitations of demonstrations, shifting focus to the actual output distribution could yield more insights.

An additional limitation of LongGuide is that its learned guidelines are based on task-level and average statistics rather than sample-based details. We designed our framework at the task level to address limited data constraints, as we found that sample-based learning under these conditions leads to high errors. While task-level guidelines already demonstrate significant improvements for LLMs, sample-based guidelines could offer more tailored guidance, potentially leading to optimal results. Moreover, this average guidance approach may be ineffective for tasks with high variance in the statistics that LongGuide learns. In such cases, the final step of LongGuide can prevent performance decline by likely choosing no guideline. For example, we found this applies to Code2Text (Richardson et al., 2017) & StoryGeneration (Fan et al., 2018b).

Furthermore, LongGuide relies on models having a certain level of task knowledge to perform self-evaluation effectively, and LongGuide necessitates LLMs with strong instruction-following capabilities. However, we anticipate that cutting-edge AI language models will overcome this limitation both now and in the near future.

Lastly, the guidelines learned by LongGuide may not be useful for the tasks the models are trained on. This is because these guidelines might introduce out-of-distribution context relative to the training data, thereby reducing the effectiveness of the testing inference. For instance, while we see notable enhancements on the CommonGen-Challenge dataset (Lin et al., 2020), it's intriguing that we don't observe any improvements on the WebNLG (Gardent et al., 2017) and E2E NLG (Puzikov and Gurevych, 2018) datasets, despite their expected similarity. Given the popularity of these datasets, we suspect the models we tested may have been previously trained on them.

Ethical Considerations

This method could be misused to optimize prompts for harmful purposes such as generating misinformation, hate speech, or privacy violations. While our method is not intended for such uses, it is impossible to completely prevent misuse. Although our method could enhance the efficiency and efficacy of bad actors, we do not anticipate that LongGuide is inherently more effective in these negative contexts than in positive applications.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the IWSLT 2017 evaluation campaign. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 2–14, Tokyo, Japan. International Workshop on Spoken Language Translation.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2024. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob

Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Xuan Long Do, Yiran Zhao, Hannah Brown, Yuxi Xie, James Xu Zhao, Nancy F Chen, Kenji Kawaguchi, Michael Qizhe Xie, and Junxian He. 2024. Prompt optimization via adversarial in-context learning. *arXiv preprint arXiv:2312.02614*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Angela Fan, David Grangier, and Michael Auli. 2018a. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018b. 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.

Marcio Fonseca and Shay B Cohen. 2024. Can large language model summarizers adapt to diverse scientific communication goals? *arXiv preprint arXiv:2401.10415*.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. Jointly optimizing diversity and relevance in neural response generation. 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 1229–1238, Minneapolis, Minnesota. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 124–133, Santiago de Compostela, Spain. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Junxian He, Wojciech Kryscinski, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2022. CTRLsum: Towards generic controllable text summarization. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5879–5915, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Pegah Jandaghi, XiangHai Sheng, Xinyi Bai, Jay Pujara, and Hakim Sidahmed. 2023. Faithful persona-based conversational dataset generation with large language models. *arXiv preprint arXiv:2312.10007*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Philippe Laban, Jesse Vig, Wojciech Kryscinski, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. SWiPE: A dataset for document-level simplification of Wikipedia pages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10674–10695, Toronto, Canada. Association for Computational Linguistics.

Surafel Melaku Lakew, Mattia Di Gangi, and Marcello Federico. 2019. Controlling the output length of neural machine translation. In *Proceedings of the 16th*

International Conference on Spoken Language Translation, Hong Kong. Association for Computational Linguistics.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. Pre-trained language models for text generation: A survey. *ACM Computing Surveys*, 56(9):1–39.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, 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, Barcelona, Spain. Association for Computational Linguistics.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

OpenAI. 2022. Introducing chatgpt.

Rangeet Pan, Ali Reza Ibrahimzada, Rahul Krishna, Divya Sankar, Lambert Pouguem Wassi, Michele Merler, Boris Sobolev, Raju Pavuluri, Saurabh Sinha, and Reyhaneh Jabbarvand. 2024. Lost in translation: A study of bugs introduced by large language models while translating code. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pages 1–13.

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

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *arXiv preprint arXiv:2309.09558*.

Yevgeniy Puzikov and Iryna Gurevych. 2018. E2E NLG challenge: Neural models vs. templates. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 463–471, Tilburg University, The Netherlands. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Anirudh Dagar, and Wenming Ye. 2023. In-context learning with iterative demonstration selection. *arXiv preprint arXiv:2310.09881*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Kyle Richardson, Sina Zarrieß, and Jonas Kuhn. 2017. The Code2Text challenge: Text generation in source libraries. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 115–119, Santiago de Compostela, Spain. Association for Computational Linguistics.

Nikunj Saunshi, Sadhika Malladi, and Sanjeev Arora. 2020. A mathematical exploration of why language models help solve downstream tasks. *arXiv preprint arXiv:2010.03648*.

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

Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Wieting, Nanyun Peng, and Xuezhe Ma. 2023a. Evaluating large language models on controlled generation tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3155–3168, Singapore. Association for Computational Linguistics.

Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Frederick Wieting, Nanyun Peng, and Xuezhe Ma. 2023b. Evaluating large language models on controlled generation tasks. *arXiv preprint arXiv:2310.14542*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal

Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Sara P Wagner. 1963. The abc's of communication. *American Association of Industrial Nurses Journal*, 11(8):8–11.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is ChatGPT a good NLG evaluator? a preliminary study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, pages 1–11, Singapore. Association for Computational Linguistics.

Rui Wang, Hongru Wang, Fei Mi, Yi Chen, Ruifeng Xu, and Kam-Fai Wong. 2023b. Self-critique prompting with large language models for inductive instructions. *arXiv preprint arXiv:2305.13733*.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Fangkai Yang, Pu Zhao, Zezhong Wang, Lu Wang, Bo Qiao, Jue Zhang, Mohit Garg, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2023a. Empower large language model to perform better on industrial domain-specific question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 294–312, Singapore. Association for Computational Linguistics.

Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. 2023b. Exploring the limits of chatgpt for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*.

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Cao Liu, Jun Zhao, and Kang Liu. 2023c. Representative demonstration selection for in-context learning with two-stage determinantal point process. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5443–5456, Singapore. Association for Computational Linguistics.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3063–3079, Toronto, Canada. 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*.

Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. 2020. A survey on machine reading comprehension—tasks, evaluation metrics and benchmark datasets. *Applied Sciences*, 10(21):7640.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023a. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*.

Yongxin Zhou, Fabien Ringeval, and François Portet. 2023b. Can gpt models follow human summarization guidelines? evaluating chatgpt and gpt-4 for dialogue summarization. *arXiv preprint arXiv:2310.16810*.

A Proofs

A.1 Proof of Thm.-2.1

Proof of Thm.-2.1. We prove this theorem by contradiction. Suppose the negation of Thm.-2.1 is true, i.e., there exists a $D_1 \in \mathcal{D}$ such that $\forall X \in \mathcal{X}$, $P_{\mathcal{M}}(X|D_1) = P_T(X)$ *(S1)*.

Now, let us consider the event $X \cap D_1^c$ where D_1^c is the conjugate of event D_1, or $D_1^c = \mathcal{D} \backslash D_1$. We have $P_{\mathcal{M}}(X \cap D_1^c | D_1) = 0$. From the assumption of the negation statement *(S1)*, we derive $P_T(X \cap D_1^c) = 0$. From the Asm.-2.2 of equivalent zero probability, we have $P_{\mathcal{M}}(X \cap D_1^c) = 0$. Similarly, we can consider the event $X^c \cap D^c$ where X^c is the conjugate of X, we arrive at $P_{\mathcal{M}}(X^c \cap D_1^c) = 0$. Since the two $X \cap D_1^c$ and $X^c \cap D_1^c$ form a disjoint union of D_1^c, we derive $P_{\mathcal{M}}(D_1^c) = P_{\mathcal{M}}(X \cap D_1^c) + P_{\mathcal{M}}(X^c \cap D_1^c) = 0 + 0 = 0$. Since D_1 and D_1^c form a disjoint union of $\mathcal{D}$, we have $P_{\mathcal{M}}(D_1) = 1$.

Now, we consider the event of $X \in \mathcal{X}$. From the negation statement *(S1)*, we have $P_{\mathcal{M}}(X|D_1) = P_T(X)\ \forall X \in \mathcal{X}$. Since $X \cap D_1$ and $X \cap D_1^c$ form a disjoint union of X, we have $P_{\mathcal{M}}(X) = P_{\mathcal{M}}(X \cap D_1) + P_{\mathcal{M}}(X \cap D_1^c) = P_{\mathcal{M}}(X \cap D_1) + 0 = P_{\mathcal{M}}(X \cap D_1)$. We also have $P_{\mathcal{M}}(X|D_1) = \frac{P_{\mathcal{M}}(X \cap D_1)}{P_{\mathcal{M}}(D_1)}$ from Bayes's theorem, meaning that $P_{\mathcal{M}}(X|D_1) = P_{\mathcal{M}}(X \cap D_1) = P_{\mathcal{M}}(X)$ (since $P_{\mathcal{M}}(D_1) = 1$). Meanwhile, from the negation statement *(S1)*, we have $P_{\mathcal{M}}(X|D_1) = P_T(X)$, thus $P_{\mathcal{M}}(X) = P_T(X)$ for all $X \in \mathcal{X}$, which contradicts to our Asm.-2.1. Therefore, our negation statement *(S1)* is false, leading to Thm.-2.1 is true. □

A.2 Proof of Thm.-2.2

Proof of Thm.-2.2. We prove Thm.-2.2 by identifying a trivial text property function. However, for suitable language or format text property functions, we hypothesize that the condition of $\mathcal{M}$ capturing the language distribution must be satisfied. We assume that the demonstration string D_f does not capture all possible outcomes of $\mathcal{G}_{\mathcal{M}}$ in $\mathcal{Y}$, which is often the case.

Recall that the demonstration string D_f consists of demonstrations $\{(x_i, y_i) | i \in (1, k)\}$ as defined in §1. We consider a trivial reference-free evaluation function $A : \mathcal{Y} \mapsto \{0, 1\}$ defined as:

$$A(y) = \begin{cases} 0, & \text{if } y \in \{y_1, y_2, ..., y_k\} \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

Since $\exists x_0 \in \mathcal{X}$ such that $\mathcal{G}_{\mathcal{M}}(x_0) = \hat{y}_0 \notin \{y_1, y_2, ..., y_k\}$, by the definition of A, we obtain $A(\hat{y}_0) = 1$. Meanwhile, for all $(x_i, y_i) \in D_f$, we have $A(y_i) = 0$. This shows that the trivial attribute obtained from the function A is not transferred from the demonstrations to the testing output, verifying that $\mathcal{M}$ can not fully learn the attribute A from D_f. □

Methods	CNN (3.0.0)	IWSLT17 en-ja	CommGen-Chall.
ChatGPT Zero-shot (ZS)	$20.12_{\pm 0.27}$	$36.13_{\pm 0.87}$	$24.21_{\pm 0.37}$
ChatGPT ZS w/ Mistral's MG	$21.41_{\pm 0.62}\uparrow$	$39.66_{\pm 2.47}\uparrow$	$29.95_{\pm 23.66}\uparrow$
ChatGPT Few-shot (FS)	$14.51_{\pm 0.80}$	$31.93_{\pm 1.88}$	$22.08_{\pm 0.63}$
ChatGPT FS w/ Mistral's MG	$13.96_{\pm 11.50}\downarrow$	$32.34_{\pm 13.79}\uparrow$	$33.34_{\pm 13.56}\uparrow$
Mistral Zero-shot (ZS)	$19.23_{\pm 0.34}$	$13.12_{\pm 1.39}$	$10.12_{\pm 0.02}$
Mistral w/ ChatGPT's MG	$19.67_{\pm 0.71}\uparrow$	$7.98_{\pm 1.49}\downarrow$	$6.29_{\pm 1.06}\downarrow$
Mistral Few-shot (FS)	$17.56_{\pm 0.63}$	$12.69_{\pm 1.82}$	$3.89_{\pm 0.17}$
Mistral FS w/ ChatGPT's MG	$19.00_{\pm 7.82}\uparrow$	$11.86_{\pm 2.79}\downarrow$	$3.61_{\pm 0.38}\downarrow$

Table 6: LongGuide can be transferable from weaker to stronger models.

B LongGuide's Extra Preliminary Properties

B.1 Understanding MG and OCG

Metric guideline (MG) (Step 1-3). To understand better how models select metrics to address the PT problem, we provide the specific metrics selected by tasks in Appx.-Tab. 12 and plot Appx.-Fig. 6 showing the frequency of metrics being selected. Among the 27 metrics, common linguistic metrics such as "Clarity" are selected frequently, highlighting their importance in capturing essential linguistic properties of answers for most of the tasks. In contrast, task-specific metrics like "Creativity" are less commonly selected, possibly because they have a lesser impact on multiple tasks. By examining the average score of selected metrics (Appx.-Fig. 7), we find that common linguistic metrics receive predominantly high scores, as expected. However, task-specific metrics like "Creativity" demonstrate diverse scores across tasks, indicating their varying importance and relevance.

Additionally, we find that metrics within MG can conflict with each other. This underscores the importance of LongGuide's Step 2 in weighting the metrics to avoid conflicts. For example, if MG consists of both "Conciseness" and "Informativeness", a very concise summary can not be highly informative, and vice versa (see Appx.-Fig. 10 for an example).

Output constraint guideline (OCG) (Step 4). For OCG, our ablation studies in Appx.-C.7 show that both the token and sentence constraints are useful for LLMs, with the sentence constraint being dominant. We hypothesize that LLMs can control #sentences better than #tokens generated. This can be partly observed in Appx.-Fig. 4 when we provide guidelines controlling #sentences and #tokens.

MG and OCG are complementary and non-interchangeable. In most tasks, the MG and OCG complement each other rather than conflict. This is because the language metrics used to construct the MG primarily evaluate the quality and characteristics of responses rather than their structure aspects such as sentence and token count, which is the main focus of the OCG. Moreover, the MG and OCG are not interchangeable. One might question whether utilizing conciseness and brevity metrics can sufficiently alter the OCG or if the OCG can effectively encompass the MG guideline. Our answer is no. While the MG can steer LLMs towards brevity in responses, it lacks precise criteria for conciseness. Modern LLMs, often trained to generate verbose responses, may struggle to meet human standards about conciseness without explicit quantitative. In contrast, the OCG supplies quantitative metrics like bins and means, yet these statistics alone do not directly address linguistic qualities. Therefore, the MG and OCG complement each other by emphasizing different facets. We provide examples to illustrate our explanations (see Appx.-Fig. 8, Fig. 9).

B.2 LongGuide Can Be Transferable From Weaker To Stronger Models

We find that the guidelines learned by LongGuide are transferable from weaker to stronger models. A weaker model can learn the guidelines at a low cost, which can then be used to enhance the performance of stronger models. This is particularly advantageous because powerful models are often closed-source and expensive to query, whereas open-source models are weaker but free to use.

Methods	CNN (3.0.0)	IWSLT17	CommGen-Chall.
Zero-shot (ZS)	$19.23_{\pm 0.34}$	$13.12_{\pm 1.39}$	$10.12_{\pm 0.02}$
ZS w/ APO	$19.53_{\pm 2.08}$	$14.45_{\pm 1.84}$	$11.21_{\pm 2.02}$
ZS w/ adv-ICL	$18.87_{\pm 2.69}$	$15.01_{\pm 1.72}$	$13.12_{\pm 2.21}$
ZS w/ LongGuide	$22.46_{\pm 0.64}$	$16.53_{\pm 0.59}$	$25.20_{\pm 1.89}$
ZS w/ LongGuide w/ APO	$\mathbf{22.76}_{\pm 1.04}\uparrow$	$\mathbf{17.13}_{\pm 1.05}\uparrow$	$\mathbf{27.01}_{\pm 1.01}\uparrow$
ZS w/ LongGuide w/ adv-ICL	$21.97_{\pm 3.21}\downarrow$	$16.90_{\pm 2.15}\uparrow$	$26.18_{\pm 3.47}\uparrow$

Table 7: Guidelines learned by LongGuide are further optimized by discrete prompt optimization frameworks bringing even better performance, with Mistral.

Since the output constraint guideline (OCG) learned for each dataset is independent of the models and consistent across models, it is transferable. Interestingly, we also find that the metric guideline (MG) is transferable from weaker to stronger models on most benchmarks, though the reverse is not generally true. We demonstrate this through experiments on CNN (3.0.0), IWSLT17 en-ja, and CommGen-Chall, representing all the tasks. We used the MG generated by Mistral for experiments on ChatGPT and vice versa under both zero-shot and few-shot settings. Tab. 6 shows the results. We observe that using ChatGPT with Mistral's MG generally improves performance, except when using few-shot on the CNN dataset. In this exception, the few-shot demonstrations often cause the model to refuse to summarize, a problem that the MG cannot entirely correct.

Hypothesizing for this transferability from weaker to stronger models, we argue that while guidelines learned by LongGuide help models better capture the task distributions, guidelines learned by a stronger model may not be beneficial for the weaker model, as the weaker model might not consistently interpret them accurately. Conversely, with its superior text comprehension capabilities, the stronger model can generalize tasks more effectively even when working with less expressive guidelines, as learned by the weaker model.

B.3 LongGuide Can Be Compared & Combined With Automatic Prompt Optimization Algorithms

The metric and output constraint guidelines (MG and OCG) learned by LongGuide may not be fully optimized for LLMs. Hence, it's intuitive to suggest that LLMs could achieve even greater performance by adopting optimal guidelines. In this section, we illustrate that the guidelines learned by LongGuide can be further refined through discrete prompt optimization algorithms. This capability is advantageous for LongGuide, enabling its concurrent development and integration with automatic prompt optimization algorithms.

Experimental setups. We employ two strong prompt optimizers, APO (Pryzant et al., 2023) and adv-ICL (Do et al., 2024), in our experiments. Here is our methodology: we integrated the guidelines generated by LongGuide into the prompt, including the input instruction and demonstrations. Subsequently, we applied the prompt optimizers to refine the input instruction, demonstrations, and guidelines. Our experiments were conducted using Mistral on datasets including CNN, IWSLT 2017 en-ja, and CommonGen-Challenge. Based on our findings detailed in Tab. 3. Following our findings in Tab. 3, the guideline used for CNN and IWSLT 2017 en-ja is OCG, while for CommonGen-Challenge it is MG-OCG.

Main results. Our results are detailed in Tab. 7. In summary, when further optimizing the OCG using APO and adv-ICL for CNN and IWSLT 2017, we observed a slight improvement. This could be attributed to the OCG already being concise and straightforward, making it easier for models to grasp. However, for the CommonGen-Challenge dataset, which utilizes the MG-OCG guideline with more detail, APO and adv-ICL have a greater amount of material to optimize within the prompts. This led to a substantial improvement in performance compared to the other datasets.

Methods	CNN (3.0.0)	IWSLT17	CommGen-Chall.
Zero-shot (ZS)	$7.60_{\pm 0.58}$	$2.99_{\pm 0.83}$	$\mathbf{10.96}_{\pm 0.36}$
ZS w/ OCG	$6.60_{\pm 0.74}\downarrow$	$3.70_{\pm 0.29}\uparrow$	$10.12_{\pm 0.56}\downarrow$
ZS w/ MG	$\mathbf{9.04}_{\pm 1.02}\uparrow$	$\mathbf{5.39}_{\pm 0.93}\uparrow$	$8.55_{\pm 0.74}\downarrow$
ZS w/ MG-OCG	$8.38_{\pm 0.91}\uparrow$	$4.59_{\pm 0.97}\uparrow$	$7.99_{\pm 0.70}\downarrow$
ZS w/ LongGuide	$\mathbf{9.04}_{\pm 1.02}\uparrow$	$\mathbf{5.39}_{\pm 0.93}\uparrow$	$\mathbf{10.96}_{\pm 0.36}$
Few-shot (FS)	$3.14_{\pm 0.32}$	$3.44_{\pm 0.83}$	$4.67_{\pm 0.33}$
FS w/ OCG	$2.24_{\pm 0.21}\downarrow$	$3.86_{\pm 0.61}\uparrow$	$8.11_{\pm 0.63}\uparrow$
FS w/ MG	$\mathbf{3.24}_{\pm 0.26}\uparrow$	$6.65_{\pm 0.97}\uparrow$	$\mathbf{10.71}_{\pm 0.80}\uparrow$
FS w/ MG-OCG	$2.99_{\pm 0.29}\downarrow$	$\mathbf{7.88}_{\pm 0.91}\uparrow$	$9.39_{\pm 0.89}\uparrow$
FS w/ LongGuide	$2.24_{\pm 0.21}\downarrow$	$\mathbf{7.88}_{\pm 0.91}\uparrow$	$\mathbf{10.71}_{\pm 0.80}\uparrow$

Table 8: Performance of **Mistral-7B-v0.1** using LongGuide learned by **Mistral-7B-Instruct-v0.2**. We observe that LongGuide improves more than half of the experiments, showing its potential effectiveness in enhancing even non-instruct models, especially for the translation task.

B.4 LongGuide Can Improve Non-instruct Models

Using guidelines learned by LongGuide, we add more instructions to models. Therefore, we aim to examine whether non-instruct models can benefit from these guidelines. Our final conclusion is yes, LongGuide has strong potential to enhance non-instruct models.

Specifically, since non-instruct models might struggle to follow our instructions to generate the guidelines §7, we utilize the guidelines learned by an instruct model instead. We run our experiments with **Mistral-7B-v0.1**[1](Jiang et al., 2023) using the guidelines learned by Mistral-7B-Instruct-v0.2. The results are provided in Tab. 8. We observe that LongGuide improves more than half of the experiments, showing its potential effectiveness in enhancing even non-instruct models, especially for the translation task.

[1] https://huggingface.co/mistralai/Mistral-7B-v0.1

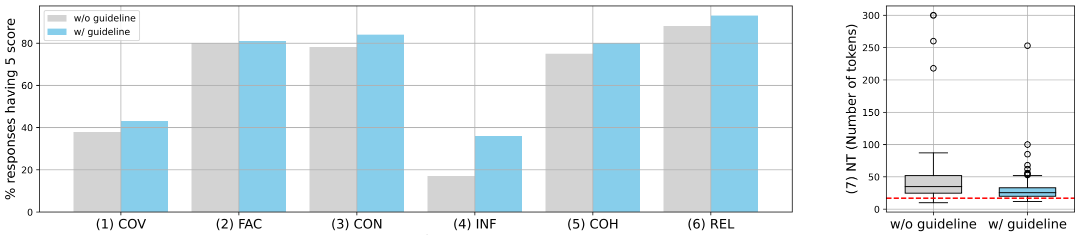


Figure 3: Evaluation results of Mistral on 100 SAMSum samples using ChatGPT-judge (with SC) across 7 metrics. We employed 5 demos having a score of 5 on metrics (1)-(6) and 17 tokens (mode of #tokens) on (7).

C Extra Results & Discussion

C.1 Empirical Illustrations of Thm.-2.2

Here, we demonstrate the empirical evidence of Thm.-2.2 on the dialogue summarization task.

Metrics. We follow Fu et al. (2023) to consider 6 metrics measuring the linguistic properties of the demonstrative answers and model responses for summarization: (1) Semantic Coverage (COV); (2) Factuality (FAC); (3) Consistency (CON); (4) Informativeness (INF); (5) Coherence (COH); (6) Relevance (REL). We use ChatGPT (OpenAI, 2022), an effective NLG evaluator (Wang et al., 2023a), to score these metrics on a scale of $1-5$. Since ChatGPT's evaluation can be unstable, we use Self-consistency (SC) (Wang et al., 2022a) with 3 sampling paths to obtain the score. We are also interested in measuring (7) Number of Tokens (NT) and (8) Number of Sentences (NS) in each response (we use NLTK lib.) since the lengthiness of the answers can significantly affect the models' performance (Fan et al., 2018a).

Methodology. Our main idea is that, *for a given metric, we select the demonstrations having the same score and evaluate whether the generated responses maintain that score*. We randomly select 100 samples from SAMSum (Gliwa et al., 2019) validation set for our evaluation. Due to the limited window size, we use $\{3, 5, 10\}$ samples from SAMSum training set as demonstrations. On each metric (1)-(6), all demonstrations selected have a perfect score of 5. For measurement (7) and (8), we select demonstrations having 17 output tokens for (7) NT and 2 sentences for (8) NS. We use Mistral-7B-it-v.02 (Jiang et al., 2023), one of the strongest open-source LLMs as the baseline.

We further add a simple guideline for each metric "The output must be highly {property}.", and we are curious whether a simple guideline, which strongly captures the distribution of the demonstrative property that we are interested in measuring, could help in maintaining that property better.

Main findings. From the results in Fig. 3, we observe several interesting findings. Firstly, on metrics (1)-(6), the model surprisingly maintains a perfect 5 score for every answer on *none of them*. Secondly, despite all demonstrations having 17 output tokens (the right-most chart), less than 5% the answers achieve this property. Fig. 4 also shows that, by adding a simple guideline, the percentages of answers maintaining the metrics are mostly improved and the variance of the number of output tokens is significantly reduced, verifying that adding guidelines is indeed helpful for models to maintain the properties better. Finally, more demonstrations do not significantly help, as different numbers of demonstrations yield similar trends across all metrics. As illustrated in Fig. 4, in the (1) COV case, a 5-demo setup improves performance, but increasing to a 10-demo setup drops performance to even below 3-demo case. In (2) FAC, (4) INF, and (6) REL, the 3-demo setup yields the best performance among the three cases. The 10-demo setup only shows a slight but insignificant improvement in (2) FAC and (5) COH. In summary, providing more demonstrations do not make significant differences. This indicates the necessity of enhancing the instructions rather than simply increasing the number of demonstrations, empirically proving Thm.-2.2.

C.2 LongGuide: Collected Metrics In Step 1

Tab. 9 presents our 27 metrics collected for LongGuide's Step 1. We first construct our pool of linguistic evaluation metrics S widely used for text generation tasks. S consists of 27 distinct metrics from 4 main sources (see Tab. 9 for details). Specifically, we collect 3 metrics from ABC's of Communication (Wagner,

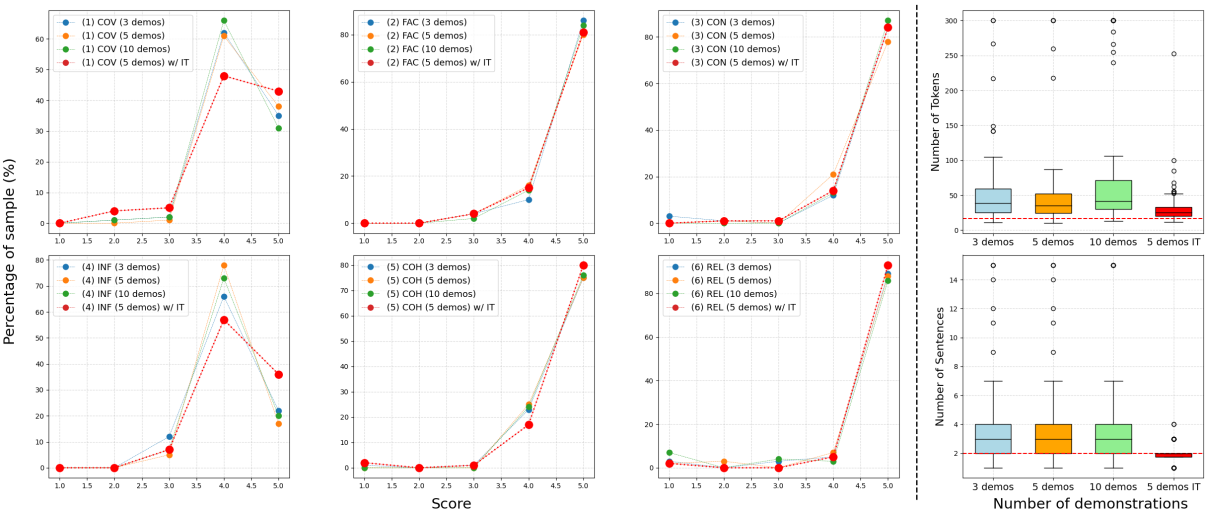


Figure 4: Evaluation results of Mistral on 100 SAMSum samples using ChatGPT-judge (with SC) across 8 metrics. We employed 3, 5, and 10 demos having a score of 5 on metrics (1)-(6) and 17 tokens (mode of #tokens) on (7) and 2 sentences (mode of #sentences) on (8). The result shows that different numbers of demonstrations would follow the same trends in results.

Source	Metrics	#
The ABC's of Communication (Wagner, 1963)	Accuracy, Brevity, Clarity	3
BARTScore (Yuan et al., 2021)	Relevance, Coherence	2
GPTScore (Fu et al., 2023)	Semantic Coverage, Factuality, Fluency, Informativeness, Consistency, Engagement, Specificity, Correctness, Understandability, Diversity	10
We propose	Completeness, Conciseness, Neutrality, Naturalness, Readability, Creativity, Rationalness, Truthfulness, Respect of Chronology, Non-repetitiveness, Indicativeness, Resolution	12
Total	27	27

Table 9: Evaluation metrics collected.

1963) evaluating clear communication. We then follow previous works (Yuan et al., 2021; Fu et al., 2023) to select 12 more metrics evaluating the dialogue response generation, text summarization, data-to-text generation, and machine translation. Finally, we propose 12 additional metrics found to be crucial for strong performance. We do not collect the metrics' definitions as they may differ across tasks.

C.3 Tab. 1: A Report Of JS Divergence Across All Metrics For SAMSum

Fig. 5 presents density plots of MG and OCG metrics selected by Mistral under the few-shot (FS) setting, measured on ground-truth, FS, and FS w/ LongGuide answers. For Jensen–Shannon divergence, the lower is better.

C.4 Tab. 2: CD-MG Selection Results of LongGuide

The numerical MG-OCG selection results on D^{train} are presented in Tab. 10, as also noted in Tab. 3. Overall, the performance of LongGuide on D^{train} closely mirrors its performance on the testing tasks in Tab. 3. The only discrepancy is for the IWSLT17 en-ja task with ChatGPT using few-shot prompting: the optimal guideline combination on D^{train} is MG-OCG (see Tab. 10), whereas the best on the testing set is MG (see Tab. 3).

C.5 LongGuide Can Generalize From Demonstrations: Numerical Results

Tab. 11 presents the numerical results of Tab. 4 in §5.1. Even with only 3-5 exemplars as demonstrations, LongGuide effectively derives MG and OCG guidelines, benefiting the model. In this case, D^{train} is the set of demonstrations, and the rest of LongGuide's steps remain unchanged.

			Summarization		Simplification	Translation	Dialogue Generation	Table2Text
Models	Method	SAMSum	CNN (3.0.0)	XL-Sum	SWiPE	IWSLT17 en-ja	Synthetic Persona	CommGen-Chall.
	#shots (random)	3	3	5	3	5	5	5
Mistral-7B-it	Zero-shot (ZS)	21.25	18.96	8.88	36.21	14.05	12.93	9.12
	ZS w/ OCG	27.43	**21.92**	**14.22**	31.19	**16.93**	12.99	20.67
	ZS w/ MG	27.68	18.02	10.26	**36.74**	11.06	13.74	19.98
	ZS w/ MG-OCG	**28.34**	21.63	13.90	35.12	15.49	**14.14**	**20.87**
	MG-OCG selection	*MG-OCG*	*OCG*	*OCG*	*MG*	*OCG*	*MG-OCG*	*MG-OCG*
	Few-shot (FS)	25.55	17.30	9.85	39.29	13.52	6.19	4.01
	FS w/ OCG	27.31	16.45	12.47	29.85	**17.58**	6.45	**20.50**
	FS w/ MG	27.88	18.47	12.01	**41.07**	14.09	6.47	11.16
	FS w/ MG-OCG	**30.01**	**19.87**	**14.89**	39.40	17.02	**8.06**	5.18
	MG-OCG selection	*MG-OCG*	*MG-OCG*	*MG-OCG*	*MG*	*OCG*	*MG-OCG*	*OCG*
ChatGPT	Zero-shot (ZS)	24.21	19.54	10.78	**45.11**	36.22	19.68	24.23
	ZS w/ OCG	28.81	21.88	**20.66**	37.58	38.45	**23.09**	**35.04**
	ZS w/ MG	25.12	20.02	10.42	45.09	37.72	19.81	18.50
	ZS w/ MG-OCG	**29.79**	**21.99**	19.91	42.72	**41.50**	20.82	30.09
	MG-OCG selection	*MG-OCG*	*MG-OCG*	*OCG*	*ZS*	*MG-OCG*	*MG-OCG*	*OCG*
	Few-shot (FS)	27.44	13.77	12.11	33.30	28.76	17.12	24.12
	FS w/ OCG	29.98	**17.55**	**19.26**	16.22	35.73	**21.50**	36.51
	FS w/ MG	28.89	14.03	12.75	19.14	36.09	19.12	21.99
	FS w/ MG-OCG	**30.65**	13.12	18.64	**37.24**	**36.22**	18.99	**38.33**
	MG-OCG selection	*MG-OCG*	*OCG*	*OCG*	*MG-OCG*	*MG-OCG*	*OCG*	*MG-OCG*

Table 10: MG-OCG selection results on D^{train} set for the main experiments in Tab. 2.

Methods	CNN (3.0.0)	IWSLT17 en-ja	CommGen-Chall.
Zero-shot (ZS)	$19.23_{\pm 0.34}$	$13.12_{\pm 1.39}$	$10.12_{\pm 0.02}$
ZS w/ CD trained on D^{train}	$\mathbf{22.46}_{\pm 0.64}$	$16.53_{\pm 0.59}$	$24.16_{\pm 0.11}$
ZS w/ MG trained on D^{train}	$18.35_{\pm 0.60}$	$8.71_{\pm 0.53}$	$21.54_{\pm 7.50}$
ZS w/ CD-MG trained on D^{train}	$22.05_{\pm 0.84}$	$15.76_{\pm 1.85}$	$25.20_{\pm 1.89}$
ZS w/ LongGuide trained on D^{train}	$22.46_{\pm 0.64}$	$16.53_{\pm 0.59}$	$25.20_{\pm 1.89}$
ZS w/ CD trained on Demos	$20.46_{\pm 0.10}$	$17.27_{\pm 1.83}$	$23.97_{\pm 0.47}$
ZS w/ MG trained on Demos	$18.33_{\pm 0.25}$	$8.63_{\pm 1.08}$	$18.98_{\pm 0.52}$
ZS w/ CD-MG trained on Demos	$19.16_{\pm 0.37}$	$14.00_{\pm 3.42}$	$24.46_{\pm 2.43}$
ZS w/ LongGuide trained on Demos	$20.46_{\pm 0.10}$	$14.00_{\pm 2.42}$	$24.46_{\pm 2.43}$
Few-shot (FS)	$17.56_{\pm 0.63}$	$12.69_{\pm 1.82}$	$3.98_{\pm 0.17}$
FS w/ CD trained on D^{train}	$19.17_{\pm 1.27}$	$\mathbf{19.86}_{\pm 2.93}$	$27.23_{\pm 0.58}$
FS w/ MG trained on D^{train}	$17.18_{\pm 2.01}$	$12.82_{\pm 0.15}$	$21.79_{\pm 5.20}$
FS w/ CD-MG trained on D^{train}	$21.18_{\pm 1.07}$	$18.70_{\pm 0.73}$	$25.43_{\pm 5.28}$
FS w/ LongGuide trained on D^{train}	$21.18_{\pm 1.07}$	$\mathbf{19.86}_{\pm 2.93}$	$27.23_{\pm 0.58}$
FS w/ CD trained on Demos	$16.88_{\pm 1.44}$	$19.40_{\pm 1.39}$	$\mathbf{28.28}_{\pm 0.69}$
FS w/ MG trained on Demos	$15.59_{\pm 0.59}$	$12.07_{\pm 2.68}$	$23.99_{\pm 4.66}$
FS w/ MG-CD trained on Demos	$19.89_{\pm 0.39}$	$17.78_{\pm 3.23}$	$27.41_{\pm 0.87}$
FS w/ LongGuide trained on Demos	$19.89_{\pm 0.39}$	$17.78_{\pm 18.43}$	$23.99_{\pm 4.66}$

Table 11: LongGuide learns the guidelines from only demonstrations with Mistral.

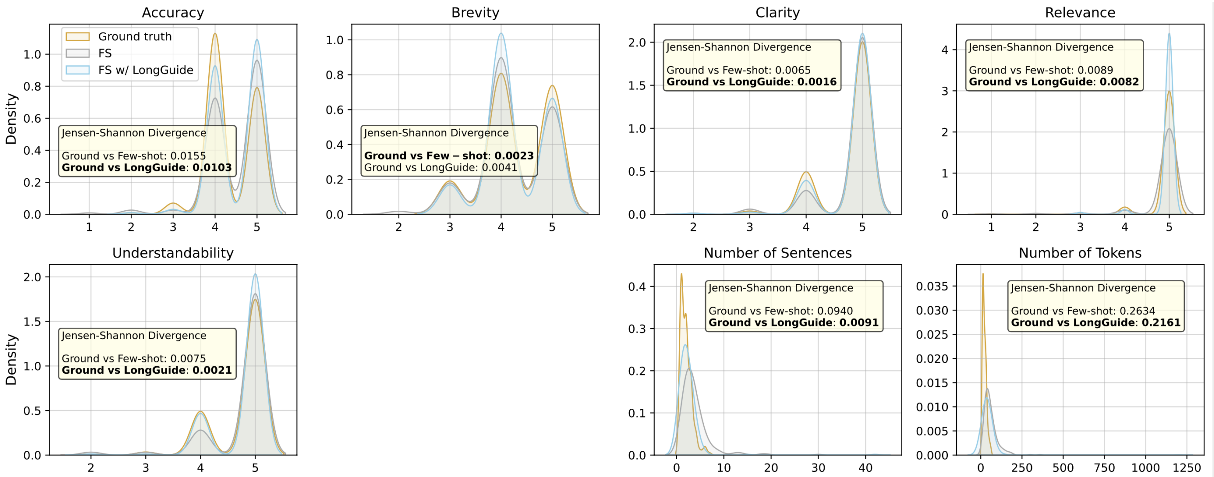


Figure 5: Density plots of MG and OCG metrics selected by Mistral under the few-shot (FS) setting, measured on ground-truth, FS, and FS w/ LongGuide answers. For Jensen–Shannon divergence, **lower is better**.

Task	Model	Selected Metrics
SAMSum	Mistral	['Accuracy', 'Brevity', 'Clarity', 'Relevance', 'Understandability']
	ChatGPT	['Accuracy', 'Brevity', 'Clarity', 'Relevance', 'Understandability']
CNN	Mistral	['Accuracy', 'Brevity', 'Clarity', 'Coherence', 'Completeness', 'Engagement', 'Readability', 'Relevance', 'Truthfulness', 'Understandability']
	ChatGPT	['Accuracy', 'Brevity', 'Clarity', 'Coherence', 'Completeness', 'Conciseness', 'Engagement', 'Neutrality', 'Readability', 'Relevance', 'Specificity']
XLSum	Mistral	['Accuracy', 'Brevity', 'Clarity', 'Coherence', 'Completeness', 'Consistency', 'Correctness', 'Diversity', 'Engagement', 'Factuality', 'Fluency', 'Indicative', 'Informativeness', 'Neutrality', 'Non-repetitiveness', 'Relevance', 'Resolution', 'Respect of Chronology', 'Semantic Coverage', 'Specificity', 'Understandability']
	ChatGPT	['Accuracy', 'Brevity', 'Clarity', 'Coherence', 'Completeness', 'Consistency', 'Correctness', 'Diversity', 'Engagement', 'Factuality', 'Fluency', 'Indicative', 'Informativeness', 'Neutrality', 'Non-repetitiveness', 'Rationalness', 'Relevance', 'Resolution', 'Respect of Chronology', 'Semantic Coverage', 'Specificity', 'Understandability']
SWiPE	Mistral	['Accuracy', 'Brevity', 'Clarity', 'Relevance', 'Understandability']
	ChatGPT	['Accuracy', 'Brevity', 'Clarity', 'Coherence', 'Conciseness', 'Consistency', 'Correctness', 'Readability', 'Understandability']
IWSLT17 en-ja	Mistral	['Accuracy', 'Clarity', 'Coherence', 'Consistency', 'Correctness', 'Factuality', 'Fluency', 'Relevance', 'Understandability']
	ChatGPT	['Accuracy', 'Clarity', 'Coherence', 'Consistency', 'Correctness', 'Factuality', 'Fluency', 'Relevance', 'Understandability']
Synthetic Persona	Mistral	['Accuracy', 'Brevity', 'Clarity', 'Coherence', 'Completeness', 'Consistency', 'Correctness', 'Diversity', 'Engagement', 'Factuality', 'Fluency', 'Indicative', 'Informativeness', 'Neutrality', 'Non-repetitiveness', 'Relevance', 'Resolution', 'Respect of Chronology', 'Semantic Coverage', 'Specificity', 'Understandability']
	ChatGPT	['Accuracy', 'Clarity', 'Coherence', 'Consistency', 'Correctness', 'Diversity', 'Engagement', 'Fluency', 'Indicative', 'Informativeness', 'Neutrality', 'Non-repetitiveness', 'Relevance', 'Resolution', 'Respect of Chronology', 'Specificity', 'Understandability']
CommGen-Chall.	Mistral	['Coherence', 'Conciseness', 'Fluency', 'Relevance', 'Understandability']
	ChatGPT	['Clarity', 'Coherence', 'Completeness', 'Conciseness', 'Consistency', 'Creativity', 'Engagement', 'Fluency', 'Naturalness', 'Relevance']

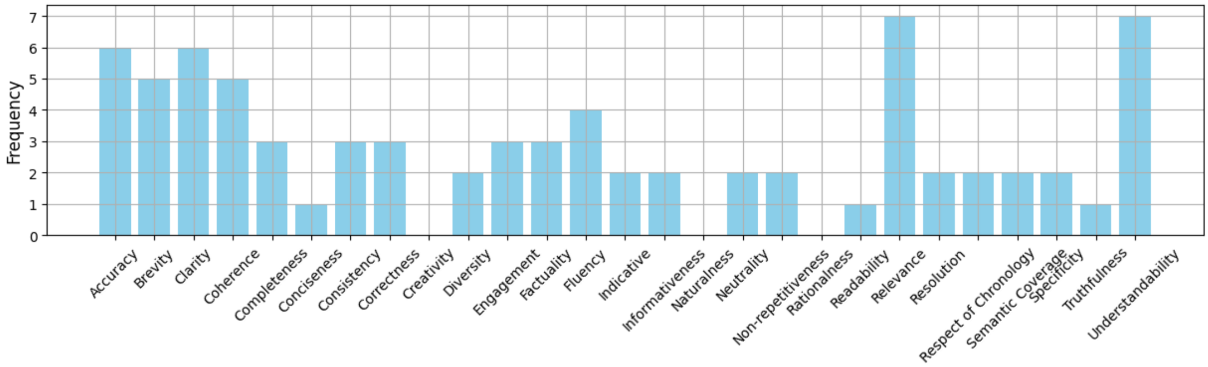
Table 12: Selected metrics by tasks by Mistral and ChatGPT.

C.6 Understanding MG and OCG: Which Metrics Were Selected The Most For MG?

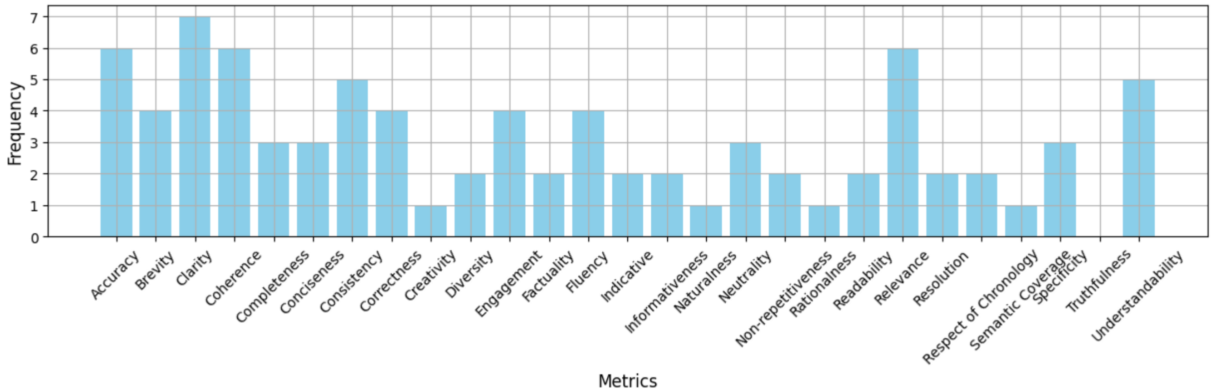
To understand better how models select metrics, we provide the specific metrics selected by tasks in Appx.-Tab. 12. Additionally, we plot Appx.-Fig. 6 showing the frequency distribution of metrics selected over 7 tasks. Among the 27 metrics collected in LongGuide's Step 1, it is evident that "Clarity" and "Relevance" are consistently prioritized highlighting their important roles in capturing linguistic properties of answers. Conversely, metrics like "Naturalness" and "Creativity" are less frequently selected, likely due to their lesser impact on task performance. Examining metric scores (Appx.-Fig. 7), we find that common linguistic metrics receive predominantly high scores, as expected. However, task-specific metrics such as "Informativeness" exhibit varied scores across tasks, reflecting their nuanced relevance.

C.7 Ablation Study: Without OCG's Token Or Sentence Information

Since OCG's token information and sentence information are the two types of information emphasized in OCG, we further investigate the importance of each type of information. The empirical experiments are conducted with Mistral (Jiang et al., 2023) on CNN, IWSLT-2017 en-ja, and CommonGen-Challenge. We present the results in Tab. 13. We observe that skipping OCG's token information or sentence information would hurt the performance. Specifically, the results drop more significantly when sentence information is omitted, and even fall below the Zero-shot score in CNN Few-shot with LongGuide and IWSLT17 en-ja Few-shot with LongGuide. The performance drops significantly in the CommonGen-Challenge Few-shot case, with a fall of 55.20%. Due to the volatility of the token count in a sentence, it is hard to estimate the other information with only one type of information given. Therefore, both types of information should be provided to better capture the text distribution.

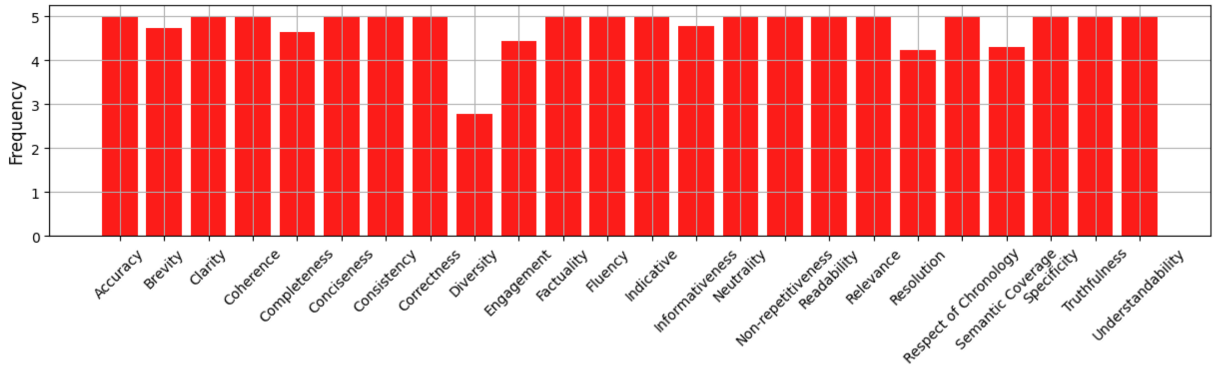


(a) Frequency of metrics selected by Mistral across datasets.

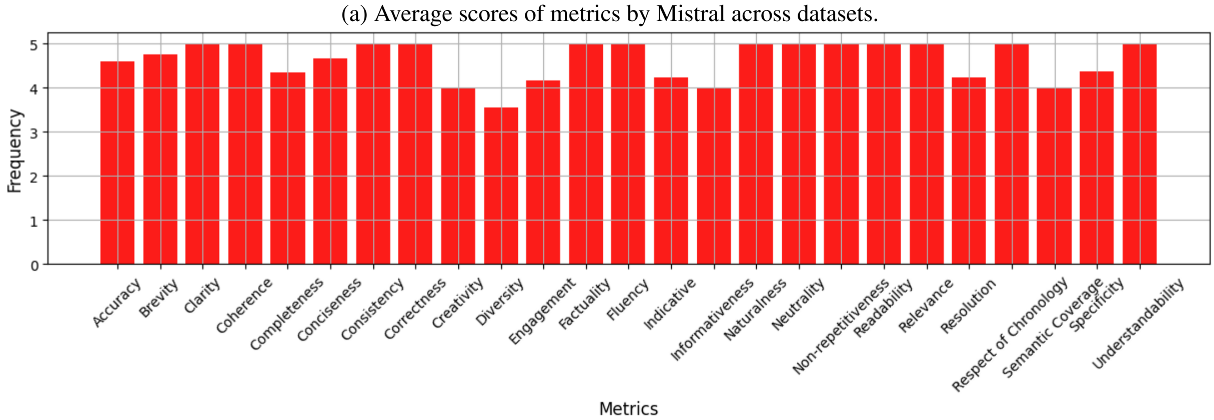


(b) Frequency of metrics selected by ChatGPT across datasets.

Figure 6: Frequency of metrics selected as the metric guideline.



(a) Average scores of metrics by Mistral across datasets.



(b) Average scores of metrics by ChatGPT across datasets.

Figure 7: Average scores of metrics as the metric guideline.

Methods	CNN (3.0.0)	IWSLT17 en-ja	CommGen-Chall.
Zero-shot (ZS)	19.23$_{\pm 0.34}$	13.12$_{\pm 1.39}$	10.12$_{\pm 0.02}$
ZS w/ LongGuide	**22.46**$_{\pm 0.64}$	**16.53**$_{\pm 0.59}$	**25.20**$_{\pm 1.89}$
ZS w/ LongGuide w/o Token Constraint	21.54$_{\pm 0.52}$↓	14.09$_{\pm 1.07}$↓	21.49$_{\pm 2.15}$↓
ZS w/ LongGuide w/o Sentence Constraint	20.92$_{\pm 0.23}$↓	10.02$_{\pm 4.17}$↓	13.32$_{\pm 0.73}$↓
Few-shot (FS)	17.56$_{\pm 0.63}$	12.69$_{\pm 1.82}$	3.98$_{\pm 0.17}$
FS w/ LongGuide	**21.18**$_{\pm 1.07}$	**19.86**$_{\pm 2.93}$	**27.23**$_{\pm 0.58}$
FS w/ LongGuide w/o Token Constraint	20.30$_{\pm 1.46}$↓	19.75$_{\pm 1.47}$↓	20.30$_{\pm 1.46}$↓
FS w/ LongGuide w/o Sentence Constraint	15.89$_{\pm 2.26}$↓	12.57$_{\pm 2.99}$↓	12.20$_{\pm 3.91}$↓

Table 13: Mistral results when omitting OCG's Token or Sentence Information, showing the importance of OCG's Token and Sentence information

C.8 Generalizability & Customization of LongGuide

LongGuide can be generalized in many ways. For example, one can always customize the metrics selected by MG and extend more constraints for OCG to suit downstream tasks. For instance, in summarization tasks, we can limit the pool of metrics selected by MG to those commonly used for evaluating summaries. Additionally, we can introduce more constraints for OCG, such as specifying keywords, the number of verbs, nouns, and so on (Fan et al., 2018a; Lakew et al., 2019; Martin et al., 2020). These customizations can make LongGuide more adept at handling downstream tasks. Additionally, the metric guideline (MG) and output constraint guideline (OCG) learned by LongGuide might not be optimal for LLMs, particularly the MG as discussed in Appx.-B.3. Further optimization methods for these guidelines can be implemented to better align them with the capabilities of specific LLMs, enhancing their performance.

D Implementation Details

Task benchmark preprocessing. We chose the newest versions of the above datasets. For each dataset except Synthetic-Persona-Chat, we sample 200 samples from the test set for our evaluation, following Bai et al. (2023), and 50 random samples from the train set for D^{train}. For Synthetic-Persona-Chat, we randomly sample 25 dialogues from its test set for our evaluation (678 utterances in total) and 3 dialogues from its train set where 50 random utterances are selected for D^{train}.

Prompting baselines' hyperparameters. We present the implementation and hyperparameters' details for our proposed LongGuide as well as prompting baselines below.

- **LongGuide.** We set the batch size is 5 and number of iterations is also 5 for LongGuide's step 1. For steps 2, 3, and 4, no hyperparameter involves. For the evaluations by Self-consistency (Wang et al., 2022a), we sample 3 results.
- **APO (Pryzant et al., 2023).** We set the number of optimization iterations is 5. We use 1 sample with the lowest ROUGE-L score as the error sample for generating gradients, following (Do et al., 2024). At each iteration, 5 textual gradients are generated, and 5 new prompts are sampled from textual gradients. Finally, 1 paraphrase of the input prompt is sampled at each optimization iteration.
- **adv-ICL (Do et al., 2024).** We use 3 iterations with a batch size of 5 as suggested by (Do et al., 2024). At each iteration, the number of new prompts sampled is 5.

Models' hyperparameters. The models' hyperparameters are presented below.

- **ChatGPT.** We use *gpt-3.5-turbo-1106* for our experiments. We use a window size of 1500 and Nucleus Sampling (Holtzman et al., 2019) as our decoding strategy with a p value of 1. We use the system role as "You are a helpful assistant!".
- **Mistral-7B-it-v0.2.** We use a window size of 1500, and Sampling decoding strategy (Holtzman et al., 2019) ($do\_sampling = True$). We load the model from Huggingface Transformers library (Wolf et al., 2020) with the model id is "mistralai/Mistral-7B-Instruct-v0.2". We do not set any explicit system role.

		Summarization			Simplification	Translation	Dialogue Generation	Table2Text
Models	**Method**	**SAMSum**	**CNN (3.0.0)**	**XL-Sum**	**SWiPE**	**IWSLT17 en-ja**	**Synthetic Persona**	**CommGen-Chall.**
	#shots (random)	3	3	5	3	5	5	5
Mistral	#tokens consumed	642	1110	811	1020	915	855	939
	US$ consumed	0	0	0	0	0	0	0
ChatGPT	#tokens consumed	1866	7683	4863	2380	1370	1344	1272
	US$ consumed	insignificant	insignificant	insignificant	insignificant	insignificant	insignificant	insignificant

Table 14: Total number of tokens consumed and US$ consumed for models to learn the metric guideline (MG) and output constraint guideline (OCG).

E Prompt Templates & Analysis

Prompting templates for LongGuide. Let Q, C, I, D_f be the input query, context, instruction, and demonstration token sequence respectively (§1, §2), and G^{best} is the learned guideline(s), the prompt for $\mathcal{M}$ is formatted: "$\{I\}\backslash\texttt{n}\{D_f\}\backslash\texttt{n}\{C\}\backslash\texttt{n}\{Q\}\backslash\texttt{n}\{G^{best}\}$".

Prompting costs. Tab. 14 presents the total number of tokens consumed for models to learn the metric guidelines and output constraint guideline (OCG) for both models with the hyperparameters of LongGuide specified in Appx.-D. We observe that the number of tokens needed to learn the guidelines is insignificant, demonstrating that LongGuide is a cost-effective solution and potentially beneficial for a wide range of applications.

Prompt for step 1, metric selection. Below is the prompt we use for step 1 selecting metrics for a given task.

```
Select top-5 metrics that are the most important from the list below to evaluate
a special way of {TASK_NAME}. {str(PRE_DEFINED_ASSESSMEN_METRICS)}.
Here are some demonstrations of the task {TASK_NAME}: {DEMONSTRATION_STRING}.
Output your list of metrics in Python list format without any explanation:
[...].
```

Prompt for step 2, metric score collection. Below is the prompt we use for step 2 for evaluating selected metrics on the task.

```
You are given an input and an output of a {TASK_NAME} task.
Input: {input}
Output: {output}
Your task is to evaluate the following criteria on a scale of 1-5, with 1 being
worst and 5 being best.
{EVALUATION_FORMAT}
The definitions of the criteria are: {METRICS_DEFINITIONS}
Your output must be in Python dictionary format without explanation.
```

Prompt for step 2, collecting METRICS_DEFINITIONS. Below is the prompt we use for step 2 collecting METRICS_DEFINITIONS for step 2.

```
Define the list of following metrics in details as the quality of the translation
expected for the {TASK_NAME} task.
{metrics}
Give me the list in bullet points.
```

Prompt for step 3, generating metric guideline (MG). Below is the prompt we use for step 3, generating the metric guideline (MG).

```
Now you are given the following metrics: {metrics_string} for the {TASK_NAME}
task.
Based on these scores on a scale of 5 for the quality of the summary:
{str(metrics_collected_scores)}, define the expected quality of the translation
for each metric in natural language. Give me the list in bullet points.
```

F Examples

Input (Dialogue)

Rob: Are we meeting up 2morrow?
Eve: How about Sunday
Eve: stores are open so we can go then
Rob: Ok I just don't know what time they close
Eve: we have ot go in the moring
Eve: I have some stuff to do around 3
Rob: ok

Ground-truth Answer (Summary)

Rob and Eve will meet on Sunday morning to go to the shops. Eve has something to do at about 3.

Zero-shot (ROUGE-L = 30.77)

Output: Rob and Eve discuss meeting up, with Eve suggesting they go on Sunday since stores are open. Rob agrees but is concerned about store closing times. Eve suggests going in the morning as she has plans in the afternoon. Rob agrees to the plan.

Zero-shot + OCG (ROUGE-L = 36.73)

Guidelines: The summary must have from 1 to 4 sentences and from 5 to 51 words with an average of 22 words and 1 sentences**.**

Output: Rob and Eve plan to meet on Sunday when stores are open, but Rob is unsure about the closing time, so they decide to go in the morning.

Zero-shot + MG (ROUGE-L = 36.07)

Guidelines: The summary effectively captures the main points while being highly concise**...**

Output: Rob and Eve discuss meeting up, considering Sunday due to stores being open. They plan to go in the morning as Eve has plans later. Rob expresses un-certainty about store closing times, to which Eve suggests going in the morning.

Zero-shot + MG-OCG (ROUGE-L = 44.00)

Guidelines: The summary must have from 1 to 4 sentences and from 5 to 51 words with an average of 22 words and 1 sentences**.** The summary effectively captures the main points while being highly concise**...**

Output: Rob and Eve plan to meet on Sunday to take advantage of store hours; they aim to go in the morning, allowing Eve time for personal activities around 3.

Figure 8: A sample from SAMSum dataset where MG and OCG supplement each other and are not interchangeable to increase the performance in final answer.

Input (r_content)

Yehuda Nir (March 31, 1930 – July 19, 2014) was a Polish-born American Holocaust survivor, psychiatrist and author of The Lost Childhood. Nir posed as a Roman Catholic and learned Latin to escape Nazi persecution in Poland during World War II. Nir's ordeal led him to a career as a psychiatrist, specializing in the treatment of post-traumatic stress disorder and severely ill children. He immigrated to the United in 1959 to complete medical residencies in New York City and Philadelphia. He served as the chief of child psychiatry of Memorial Sloan-Kettering Cancer Center from 1979 until 1986.
Nir was born Juliusz Gruenfeld in Lvov, Poland, (present-day Ukraine) on March 31, 1930. He later changed his name to "Nir" after World War II since "Gruenfeld" has German origins. Nir means plowed fields in Hebrew.
Nir released a memoir of his experience during the Holocaust, "The Lost Childhood" in 1989. A second edition was reprinted by Scholastic Press in 2002. The Lost Childhood is now used as part of the high school curriculum throughout the United States. He also published four self-help books focusing on relationships, including "Not Quite Paradise: Making Marriage Work" and "Loving Men for All the Right Reasons."
Yehuda Nir died at his home in Manhattan, New York City, on July 19, 2014, at the age of 84. His funeral was held at Riverside Memorial Chapel on the Upper West Side with burial on the Mount of Olives in Jerusalem.

Ground truth answer (s_content)

Dr. Yehuda Nir (31 March 1930 – 19 July 2014) was an Polish-born Israeli-American psychiatrist. He was known for surviving the holocaust. Nir served as a chief of child psychiatry at Memorial Sloan-Kettering Cancer Center from 1979 to 1986. He was born in Lvov, Poland. Nir died in Manhattan, New York, aged 84. **(53 tokens)**

Zero-shot answer (ROUGE-L = 21.43)

Yehuda Nir, born as Juliusz Gruenfeld in Lvov, Poland (now Ukraine) in 1930, survived the Holocaust by posing as a Roman Catholic and learning Latin. He later changed his name to "Nir" as a reference to plowed fields in Hebrew. Nir became a psychiatrist, focusing on treating post-traumatic stress disorder and severely ill children. He moved to the U.S. in 1959 to finish medical residencies in New York City and Philadelphia. Nir served as the head of child psychiatry at Memorial Sloan-Kettering Cancer Center from 1979 to 1986.
Nir published a memoir of his Holocaust experiences, titled "The Lost Childhood," in 1989. The book has been included in high school curriculums in the U.S. since then, and a second edition was published by Scholastic Press in 2002. In addition, he authored four self-help books about relationships, such as "Not Quite Paradise: Making Marriage Work" and "Loving Men for All the Right Reasons."
Nir passed away on July 19, 2014, at the age of 84, in his Manhattan home. His funeral took place at Riverside Memorial Chapel on the Upper West Side, while he was buried on the Mount of Olives in Jerusalem. **(192 tokens)**

Output constraint guideline (OCG)

*The summary must have from **1 to 17 sentences** and from **16 to 426 words** with an average of **80 words and 4 sentences.***

Only OCG (ROUGE-L = 15.79)

Yehuda Nir, born Juliusz Gruenfeld in Poland in 1930, survived the Holocaust by posing as a Roman Catholic and learned Latin. He later became a psychiatrist, focusing on PTSD and severely ill children in the US. Nir published "The Lost Childhood" about his experiences and self-help books on relationships. He died in 2014, with a funeral in New York and burial in Jerusalem.
Born in Lvov, Poland, in 1930 as Juliusz Gruenfeld, Yehuda Nir posed as a Catholic during the Holocaust and used his knowledge of Latin to hide. After immigration to the US in 1959, he specialized in child psychiatry and PTSD treatment.
Nir published his memoir, "The Lost Childhood," in 1989 and it is now used in US high schools. Additionally, he wrote four self-help books, including "Not Quite Paradise" and "Loving Men for All the Right Reasons."
Yehuda Nir, aged 84, passed away in Manhattan, New York, in 2014. His funeral was held at Riverside Memorial Chapel and he was buried on the Mount of Olives in Jerusalem. **(171 tokens)**

Metric guideline (MG)

- The simplified text retains the original meaning and information fully and without error.
- The simplified text is succinct and shorter than the original text, but not excessively so as to lose essential meaning or clarity.....

Only MG (ROUGE-L = 16.09)

Yehuda Nir, born Juliusz Gruenfeld in Poland in 1930, changed his name after World War II to the Hebrew term "Nir" meaning "plowed fields." Escaping Nazi persecution as a Roman Catholic with learned Latin, Nir became a psychiatrist specializing in PTSD and severely ill children. He immigrated to the US in 1959 for medical residencies in New York and Philadelphia.
Nir published a memoir of his Holocaust experiences, "The Lost Childhood," in 1989 and a second edition in 2002. Now used in US high schools, this memoir highlights his harrowing journey. Additionally, he wrote self-help books on relationships, including "Not Quite Paradise" and "Loving Men." Nir died at 84 in Manhattan in 2014 and was interred in Jerusalem. **(118 tokens)**

LongGuide (ROUGE-L = 27.82)

Dr. Yehuda Nir, born Juliusz Gruenfeld in 1930 in Lvov, Poland, survived the Holocaust by posing as a Catholic. He moved to the U.S. in 1959, becoming a psychiatrist specializing in PTSD and child psychiatry. His memoir, The Lost Childhood, published in 1989, is used in high schools. Nir died in Manhattan in 2014, with his burial in Jerusalem. **(59 tokens)**

Figure 9: An example of SWiPE (Laban et al., 2023) where the record contains fewer tokens than the expected average. This reduces the effectiveness of OCG and MG individually, but their combination could enhance performance.

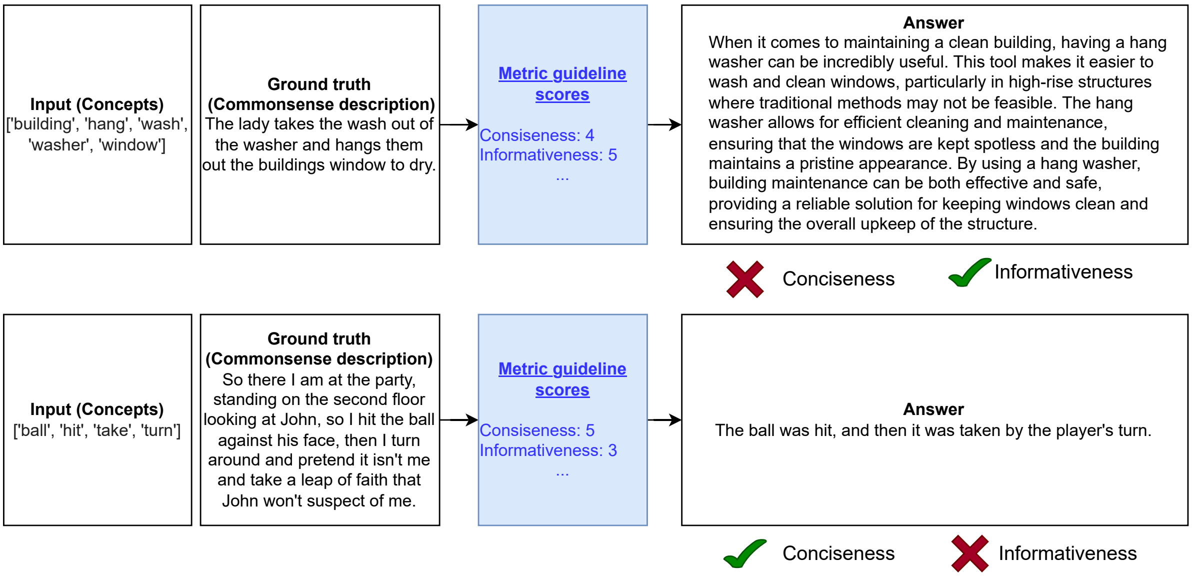


Figure 10: A CommonGen-Challenge example (Lin et al., 2020), where output with high Conciseness score could have low Informativeness score and vice versa

Input (Dialogue)

George: Hi Donna. I've been trying to catch you.
Donna: What about?
George: A rather delicate matter.
Donna: Did you catch AIDS?
George: Very funny!
Donna: It is, isn't it?
George: I don't think so.
Donna: Too bad. So what do you want?
George: Could we meet and discuss it somewhere?
Donna: Like where?
George: Like in a coffee shop or somewhere.
Donna: You sure a coffee shop is better to discuss delicate matters.
George: Come to think of it, you are right. It's not.
Donna: See? So what's up?
George: Couldn't you come to my place tonight?

Ground truth answer (Summary)

Donna will pay George a visit tonight to discuss a personal matter.

Zero-shot answer (ROUGE-L = 16.22)

George attempts to make arrangements to meet Donna to discuss a personal matter, but she is skeptical and makes a joke about the topic being AIDS. George finds her humor inappropriate and denies it. Donna presses for information on the discussion topic, and George suggests they meet at his place that night since a coffee shop may not be an appropriate setting.

Metric guideline scores w/ step 2

Accuracy: 5, Brevity: 5, ***Clarity: 4,*** *Correctness: 5, Relevance: 5, Understandability: 5,* ***Informativeness: 4***

LongGuide (ROUGE-L = 21.62)

George proposed discussing a private matter with Donna, suggesting a coffee shop but later acknowledging its unsuitability; he then invited her to his place instead.

Metric guideline scores w/o step 2

Accuracy: 5, Brevity: 5, ***Clarity: 5,*** *Correctness: 5, Relevance: 5, Understandability: 5,* ***Informativeness: 5***

LongGuide Without Step 2 (ROUGE-L = 17.24)

George asked Donna to discuss a delicate matter with him, initially suggesting a coffee shop. However, Donna pointed out that a coffee shop might not be the best place for such a conversation, leading George to reconsider and propose coming to his place that night instead.

Figure 11: A SAMSum example, where skipping step 2 worsens the performance due to lack of clarity in metrics