Improving Minimum Bayes Risk Decoding with Multi-Prompt

Anonymous ACL submission

Abstract

While instruction fine-tuned LLMs are effective text generators, sensitivity to prompt construction makes performance unstable and sub-optimal in practice. Relying on a single 'best' prompt cannot capture all differing approaches to a generation problem. Using this observation, we propose *multi-prompt decoding*, where many candidate generations are decoded from a prompt bank at inference-time. To ensemble candidates, we use Minimum Bayes Risk (MBR) decoding, which selects a final output using a trained value metric. We show multi-prompt improves MBR across a comprehensive set of conditional generation tasks (Figure 1), and show this is a result of estimating a more diverse and higher quality candidate space than that of a single prompt. Further experiments confirm multi-prompt improves generation across tasks, models and metrics.[1]

1 Introduction

Minimum Bayes Risk (MBR) decoding (Bickel and Doksum, 1977) improves the generation quality of large language models (LLMs) over standard, single-output decoding methods, such as beam search and sampling. MBR generates a set of candidates and selects the one with the highest expected utility, using all other hypotheses as references (see Fig. 2, left), following a simple intuition that a desirable output should be highly probable and consistent with others. MBR has been applied across a variety of NLP generation tasks (Amrhein and Sennrich, 2022; Shi et al., 2022; Suzgun et al., 2023; Jain et al., 2023). In particular, self-consistency (Wang et al., 2023), a special case of MBR, has become widely used to improve LLM reasoning capabilities by ensembling reasoning paths.

A central question to improve the generation quality of MBR decoding is how to balance between diversity and adequacy within the candidate set. Prior work has found success using sampling-based decoding to generate diverse hypotheses (Eikema and Aziz, 2020; Freitag et al., 2022a, 2023a). However, naively increasing the sampling temperature eventually degrades the quality of the candidates. Recently, instruction fine-tuned LLMs (Ouyang et al., 2022; Chung et al., 2022) have opened up the possibility of writing *prompts* in various formats to elicit higher diversity generations. As these models are observed to be sensitive to prompt design, a slight change in phrasing or the inclusion of more relevant example can significantly impact model quality and behavior (Srivastava et al., 2023; White et al., 2023).

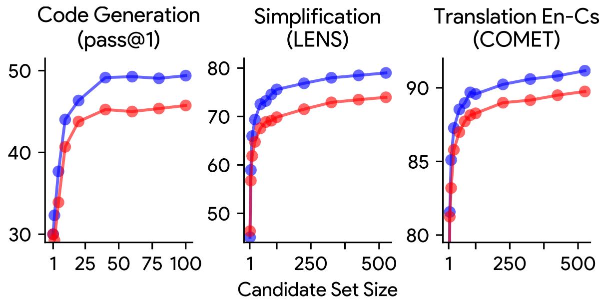

Figure 1: Multi-prompt and single prompt MBR results for code generation on HUMANEVAL, text simplification on SIMPEVAL, and translation on WMT '22 EN-CS generated with open-source 7B LLMs (details in §4).

Taking advantage of the prompt sensitivity of LLMs, we introduce multi-prompt MBR decoding, which samples candidates using a bank of human- or model-written prompts (see Figure 2, right). Intuitively, exploring a variety of prompts enables the generation of diverse, high quality hypotheses that provide a closer representation of the true output distribution. By guiding the model towards different regions of the output space, each prompt captures unique sequences that are coherent and relevant to the specific input example.

We experiment with three distinct generation tasks: text simplification (Maddela et al., 2023),

[1]Our experiment code, data and prompts are available at `https://anonymized_url`.

machine translation (Kocmi et al., 2022), and code generation (Chen et al., 2021). Each task assess the impact of different prompt components on multi-prompt MBR, such as instance-level prompts for code, task descriptions for simplification, and in-context examples for translation. To account for the relative quality between prompts, we develop different strategies for selecting prompts that outperform a baseline random choice: *sampling* prompts from a large prompt bank based on their usage on an unlabeled set of task data and *selecting* prompts using embedding-based heuristics without any examples.

We evaluate multi-prompt MBR on a broad range of LLMs including both open-source models like Llama 2 (Touvron et al., 2023) and state-of-the-art closed-source models such as GPT-4 (Achiam et al., 2023). Our results show multi-prompt MBR consistently improves single-prompt MBR across all three tasks and model scales, with gains of up to 7% on HumanEval (Chen et al., 2021) and 5 points of LENS on SIMPEVAL (Maddela et al., 2023). Figure 1 displays results for models at the 7B scale. Finally, we study the dynamics between different utility and evaluation metrics, revealing that multi-prompt MBR with one metric improves performance universally across metrics.

2 Preliminaries

Instruction fine-tuned LLMs are trained to follow arbitrary natural language task descriptions (Wei et al., 2022a). Given an input x and prompt ρ, an autoregressive language model π_θ parameterized by θ estimates an output sequence $y \sim \pi_\theta(x, \rho)$ using an decoding algorithm by sampling the next token conditioned on the input $\pi_\theta(y_i|y_{<i}, x, \rho)$. The decoding algorithm aims to generate y by maximizing the sequence likelihood over the language model distribution $\pi_\theta(y|x, \rho) = \Pi_{i=1}^{T} \pi_\theta(y_i|y_{<i}, x, \rho)$.

Minimum Bayes Risk Decoding. In practice, the highest likelihood sequence does not necessarily yield the highest quality generation (Jaeger and Levy, 2006). From this observation, MBR decoding (Bickel and Doksum, 1977; Eikema and Aziz, 2020) first samples a set of hypotheses $\mathcal{H}$ from the model π_θ, approximating the true distribution of output space $\mathcal{Y}$, then selects the output $\hat{y}_{MBR}$ that maximizes the expected utility (or minimizes the expected loss in traditional formulation) with respect to a set of references $\mathcal{R}$:

$$\hat{y}_{\text{MBR}} = \underset{y \in \mathcal{H}}{\arg\max} \left(\mathbb{E}_{\mathcal{H} \sim \pi_\theta} [U(y, \mathcal{R})] \right), \quad (1)$$

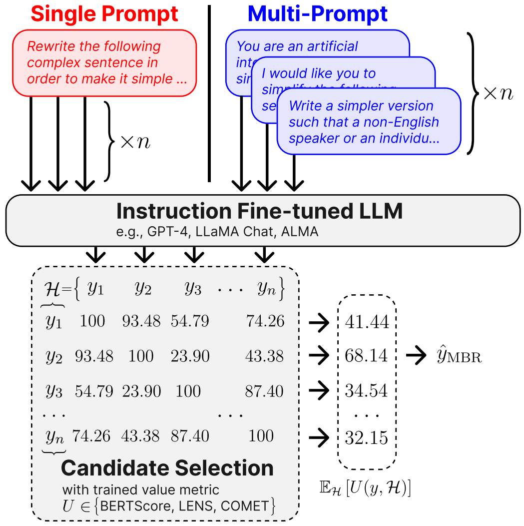

Figure 2: Multi-prompt MBR generates candidates using a human- or model-written prompt bank and selects the highest pairwise score with a trained value metric.

where $U(y, R) = \mathbb{E}_{y' \sim \mathcal{R}}[u(y, y')]$ and $u(y, y')$ is a utility function that evaluates hypothesis y against a reference y'. In practice, $\mathcal{R}$ is also sampled from the same model π_θ under the assumption that the model produces reliable outputs in expectation, and is usually set as identical to hypothesis set $\mathcal{H}$.

Many existing techniques to improve LLMs' performance such as self-consistency (Wang et al., 2023) and output ensemble (Kobayashi, 2018) are special cases of MBR. For instance, self-consistency can be viewed as MBR using the utility function $u(y, y') = \mathbb{1}[\text{ans}(y) = \text{ans}(y')]$, where $\text{ans}(y)$ is the answer extracted from the reasoning path y (Bertsch et al., 2023).

3 Multi-Prompt MBR Decoding

Prior work on MBR decoding primarily uses models trained or fine-tuned for a specific generation task (Freitag et al., 2022a; Fernandes et al., 2022). With instruction fine-tuned LLMs, the input x is contained within a structured prompt ρ, consisting of task instruction and/or in-context examples. Earlier studies have extensively documented that the design of the prompt has a dramatic impact on overall performance (Mishra et al., 2022; Khashabi et al., 2022; Lu et al., 2022; Sclar et al., 2023).

To investigate this phenomenon, we show in Figure 3a (bottom) the likelihoods and quality of samples from 10 prompts of varying performance for a text simplification task, measuring quality as the LENS metric score against a set of gold references. Greedy sampling ($\tau = 0$) estimates

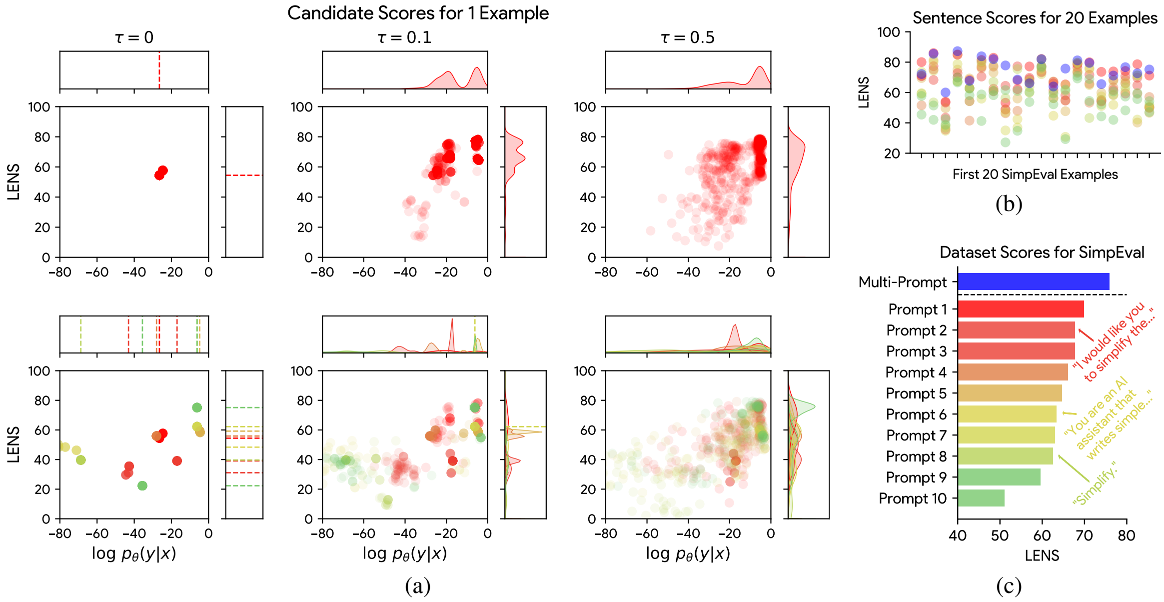

Figure 3: (a) LENS score and sequence probability for 1000 generations on a single text simplification example decoded from Llama 2 7B Chat with temperatures $\tau = [0, 0.1, 0.5]$ using a single prompt (top) and multiple prompts (bottom). As the temperature increases, we find each prompt estimates candidate sequences centered at different modes. (b) LENS scores of the best generation per-prompt for the first 20 sentences in SIMPEVAL, showing no single prompt produces the best overall output. (c) Dataset-level LENS performance of each prompt when performing single prompt MBR vs. multi-prompt MBR.

different sequences for each instruction, with single prompt (Figure 3a, top) generating a single sequence. As we increase temperature τ, generations from a single prompt simply exhibit noise centered around the mode of the highest likelihood sequence, while multi-prompt estimates a generations around modes uniquely defined by each prompt. For instance, one of the prompts (i.e., Prompt 9 highlighted in green) produces the highest quality generation for this one input sentence, despite having a low performance over the entire dataset. In fact, no prompt consistently produces the highest quality sequences, as illustrated in Figure 3b, rather prompts are most effective at different inputs.

Building upon these insights, we propose multi-prompt MBR decoding, depicted in Figure 2, where the MBR hypothesis set $\mathcal{H}$ consists of outputs sampled from n distinct prompts ρ:

$$\mathcal{H} = \bigcup_{i=1}^{n} \mathcal{H}_i, \text{ where } \mathcal{H}_i = \{y | y \sim \pi_\theta(x, \rho_i)\}. \quad (2)$$

Bertsch et al. (2023) show that MBR seeks the mode of some distribution q over a quality feature $\phi(y)$ applied to the output space rather than the mode of the model's distribution:

$$\hat{y}_{\text{MBR}} \approx \arg\max_{y \in \mathcal{H}} q(\phi(y)|x). \quad (3)$$

We hypothesize, in expectation, the mode of $\phi(y)$ across outputs from multiple prompts has higher downstream performance compared to that derived from a single prompt. This is empirically supported by our example, where Figure 3c shows that multi-prompt MBR outperforms individual single-prompt MBR across the full task dataset.

Although multi-prompt ensembles hypothesis spaces between prompts, some notion of objective quality still exists when constructing the prompt bank. As shown in Figure 3c, the majority of the 10 human-written prompts fall within a 10-point range of LENS scores when evaluated on the task dataset but a few prompts consistently produce low-quality generation. Therefore, to account for the hierarchy in prompt quality, we propose two methods for choosing the prompts used at generation time from a prompt bank $\mathcal{P}$: sampling from a learned distribution of prompts, based on a small unlabeled train set (§3.1); and selecting a subset of prompts based on heuristics in the absence of a train set (§3.2).

3.1 Prompt Sampling

In this approach, we first calculate the probability of each prompt $p(\rho)$ as the proportion of times that prompt generates the highest scoring output on a separate training set. At inference time, prompts are sampled with replacements from this learned probability distribution, and candidate outputs are then generated given these prompts.

Top-p Prompt Sampling. Inspired by the principle of nucleus sampling (Holtzman et al., 2020), our goal is to keep the prompts with high probability and truncate the least used prompts by setting their probabilities to zero. We define the top-p prompt set as the minimal set $\mathcal{P}_{\text{top-p}} \subseteq \mathcal{P}$ such that:

$$\sum_{i=0}^{|\mathcal{P}_{\text{top-p}}|} p(\rho_i) \geq p. \quad (4)$$

We then re-normalize the distribution of $\mathcal{P}_{\text{top-p}}$ and sample prompts from the new distribution:

$$p'(\rho) = \begin{cases} \frac{p(\rho)}{\sum_{\rho \in \mathcal{P}_{\text{top-p}}} p(\rho)} & \text{if } \rho \in \mathcal{P}_{\text{top-p}} \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

3.2 Prompt Selection

Prompt selection chooses a fixed subset $\mathcal{P}_{\text{best}} \subset \mathcal{P}$ of $|\mathcal{P}_{\text{best}}| = k$ prompts based on heuristics. Compared to sampling, this does not require an additional training set to evaluate prompt efficacy. We consider the following heuristics for selecting $\mathcal{P}_{\text{best}}$: prompts that have the closest similarity and greatest dissimilarity with others, and prompts that are randomly selected from each k-NN cluster, which is also useful when a training set is presented, allowing the selection of high-performing prompts within each cluster. We calculate the semantic (dis)similarity of prompts based on SentenceBERT (Reimers and Gurevych, 2019) embeddings.

4 Experiment Setup

In this section, we describe the experimental details for evaluating the efficacy of multi-prompt MBR decoding across tasks, prompt setups, models, and utility metrics, with results and analyses in §5.

4.1 Tasks & Datasets

Unlike previous work applying MBR to a single generation task (Shi et al., 2022; Eikema and Aziz, 2022), we deliberately select three unique tasks to demonstrate the universality of multi-prompt: text simplification with task-level instructions, code generation with example-level instructions, and machine translation with in-context examples.

Code Generation. We use HumanEval (Chen et al., 2021) benchmark, where models are tasked with generating a Python program given a description with unit tests. Since each example is a unique coding task, we generate a unique prompt bank for each input. Following Zhang et al. (2023), we reject empty, degenerate (e.g., `pass`, `return None`), or non-compiling programs before applying MBR.

Text Simplification. We use the SIMPEVAL$_{2022}$ test set (Maddela et al., 2023), containing complex sentences from Wikipedia, paired with human-written simplifications. The prompt bank is generated based on author-written examples (Table 4) and are used for the entire dataset.

Machine Translation. We purposely choose the EN $\rightarrow$ CS language pair from the WMT 22 (Kocmi et al., 2022) `newstest` corpus, ensuring its exclusion from the training data of recent translation LLMs or metrics (Xu et al., 2024). Results on additional language pairs are in Appendix C.2.

4.2 Constructing the Prompt Bank

Following existing work studying prompt sensitivity (Mizrahi et al., 2023; Gonen et al., 2023), our experiments rely on a small set of manually written seed prompts, and use an LLM to generate diverse paraphrases of prompts. Model-written prompts are generated using GPT-4 Turbo. For seed prompts, the authors manually write 10 for text simplification (Table 4) and use the original HUMANEVAL instruction from each example for code generation. Only the LLM-written prompts are used for multi-prompt. The translation prompts consist of randomly sampled in-context examples from previous WMT shared tasks. We use in-context examples for translation as the translation LLMs were not trained to follow task instructions.

For multi-prompt experiments, we select from the prompt bank with top-p prompt sampling (§5.2) using $p = 0.6$, where the prompt usage $p(\rho)$ is calculated using a held-out 20% split of each dataset. For our single prompt baselines, we aim to use strongest available prompt in the prompt bank, so we use the prompt with the highest usage $p(\rho)$. Human-written prompts and prompt generation instructions are included in Appendix A.

4.3 Models

Our main experiments are performed with Llama 2-7B Chat (Touvron et al., 2023) for simplification, ALMA-7B-R (Xu et al., 2024) for translation and CodeLLaMA-13B Instruct (Roziere et al., 2023) for code generation, all fine-tuned to follow instructions. In §5.3 we further explore a wide range of model architectures and sizes, including state-of-the-art and task-specific fine-tuned models. Unless otherwise specified, we generate the hypothesis set using nucleus sampling (Holtzman et al., 2020) with $\tau = 0.9, p = 0.95$. We include a detailed review of all models in this work in Appendix B.2.

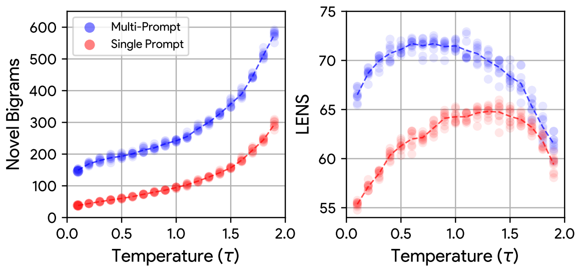

Figure 4: Candidate set diversity and LENS scores across temperatures for simplification task. At low temperatures, the increased candidate diversity from multi-prompt directly translates to improved performance.

4.4 Utility Metrics & Evaluation

Our core experiments use the trained LENS (Maddela et al., 2023) for simplification and COMET (Rei et al., 2020) for translation as the candidate selection metric. For code generation, we use MBR-EXEC (Shi et al., 2022), which executes each candidate program against a set of test cases, selecting the program with the highest agreement over all test cases' outputs. As in Zhang et al. (2023), we use the docstring examples as test cases for MBR-EXEC and evaluate with pass@1. Given the growing body of work on metric development, we verify our multi-prompt results across a broad range of utility and evaluation metrics in §5.4.

5 Experiment Results

We compare multi-prompt decoding to traditional MBR (§5.1), ablate the prompt sampling mechanism (§5.2), vary model architectures (§5.3), evaluate across utility metrics (§5.4) and finally evaluate multi-prompt on efficient MBR alternatives (§5.5).

5.1 How does multi-prompt MBR perform?

Multi-prompt Improves MBR. We report our main results in Figure 1, comparing single prompt and multi-prompt performance as the number of generated candidates increases, with detailed results in Figure 7 in App. C. Multi-prompt consistently outperforms standard MBR for all tasks.

Candidate Diversity $\not\Rightarrow$ Quality. To measure the impact of temperature on the candidate set quality, we report performance and diversity, as measured by novel bi-grams, across temperatures in Figure 4. For low temperatures, we find that multi-prompt generates a consistently more diverse candidate space, which directly translates to higher-quality generation. While single prompt MBR performance improves with temperature $\tau > 1$, despite generating an equal or greater diversity set than multi-prompt, multi-prompt MBR still produces higher quality candidates. As $\tau \to 2$, the quality of single and multi-prompt MBR begins to degrade as their candidate sets become too noisy to generate high-quality sequences. Framing the decoding process as each prompt estimating a unique distribution of candidate generations (§3), the ability of multi-prompt to achieve higher quality generation as a result of candidate set diversity is intuitively the byproduct of combining multiple candidate distributions defined by each instruction.

	pass@1	LENS	COMET
Single Prompt ($\|\mathcal{H}\|=100$)	48.78	74.67	88.93
Multi-Prompt + Prompt Sampling ($\|\mathcal{P}\|=100$)			
Random Selection	–	74.91*	89.98*
Prompt Sampling	–	78.29*	90.33*
Top-p Prompt Random	–	78.61*	90.11*
Top-p Prompt Sampling	–	**79.08***	**90.36***
Single Prompt ($\|\mathcal{H}\|=10$)	41.55	61.26	87.24
Multi-Prompt + Prompt Selection ($\mathcal{P}_{\text{best}} \subset \mathcal{P}$, $\|\mathcal{P}_{\text{best}}\|=10$)			
Random Selection	39.63	60.00	87.81*
k-NN Cluster Random	40.24	58.73	87.80*
Farthest Similarity	**44.51***	58.32	**88.14***
Closest Similarity	37.80	61.53*	87.73*
Highest Performance	–	62.43*	87.65
k-NN Cluster Performance	–	**66.12***	87.73*

Table 1: Results for prompt sampling using 100 prompts (top) and subset selection using 10 of 100 prompts (bottom). * = Statistically significant improvement with $p<0.05$. Sampling from a weighted, truncated distribution improves multi-prompt across candidate set sizes.

5.2 What is the impact of the prompt bank?

Sampling Prompts Improves Candidate Quality. Table 1 (top) reports results for multi-prompt across different prompt sampling methods for text simplification and translation. We perform a hypothesis test for the statistical significance of each variation of multi-prompt outperforming single prompt MBR using bootstrap sampling with 1000 iterations (Koehn, 2004). Note that, code generation results are omitted as a unique set of prompts is generated for each HumanEval example. We find sampling prompts by usage and truncating the top-p prompts improves multi-prompt over a random selection baseline, with top-p prompt sampling performing the best on both tasks.

A Higher Quality Prompt Bank Improves Multi-prompt. Table 1 (bottom) reports results for different prompt subset selection methods, which use heuristics to select a smaller set of prompts for multi-prompt to maximize performance. The best

	Single Prompt	Multi-prompt
Code Generation ($\|\mathcal{H}\|=20$) – HUMANEVAL (pass@1)		
StarCoder 2 15B	44.51	49.39 (+4.88)
CodeLlama 7B	37.80	40.85 (+3.05)
CodeLlama 13B	43.29	48.17 (+4.88)
CodeLlama 34B	45.73	52.44 (+6.71)
CodeLlama 70B	61.59	68.90 (+7.31)
GPT-3.5	68.29	73.78 (+5.49)
GPT-4	81.71	82.93 (+1.22)
Text Simplification ($\|\mathcal{H}\|=100$) – SIMPEVAL (LENS)		
Ctrl T5 3B	72.6	–
Ctrl T5 11B	74.4	–
Llama 2 7B Chat	75.71	80.38 (+4.67)
Llama 2 13B Chat	78.19	80.27 (+2.08)
Llama 2 70B Chat	82.21	83.28 (+1.07)
GPT-3.5	76.87	81.25 (+4.38)
GPT-4	76.47	81.56 (+5.09)
Translation ($\|\mathcal{H}\|=100$) – WMT '22 EN-CS (COMET)		
WMT '22 Winners	91.9	–
MS Translate API	90.6	–
ALMA 7B R	89.17	89.94 (+0.77)
ALMA 13B R	89.41	90.45 (+1.04)
GPT-3.5	91.27	91.35 (+0.08)
GPT-4	92.24	92.47 (+0.23)

Table 2: Metric scores for state-of-the-art systems compared to LLMs with multi-prompt using $|\mathcal{H}|$ candidates. Translation and simplification baselines are as reported in Hendy et al. (2023) and Maddela et al. (2023).

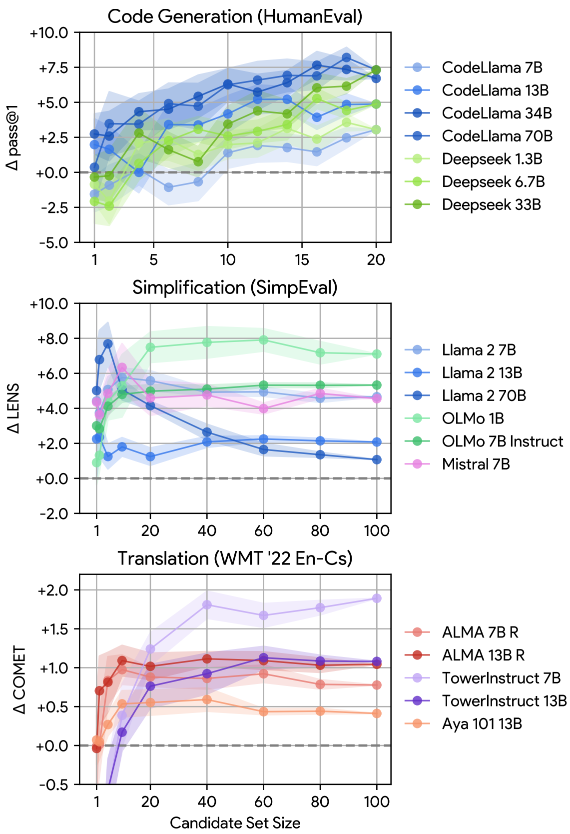

Figure 5: Δ metric improvement from single prompt to multi-prompt across model sizes and architectures, reported with a 95% CI bootstrapped over 20 iterations. For absolute performance, see Figure 10.

selection method for each task had a significant impact on performance when compared to a single prompt MBR (+2.9 pass@1, +4.9 LENS and +0.9 COMET). For text simplification, decoding with the 10 highest performing prompts is further improved by selecting prompts from a k-NN clustering of prompt embeddings, which enforces a dis-similarity between prompts. However, translation and code generation benefit from using the farthest similarity, or semantically distant prompts. These results highlight multi-prompt's sensitivity to the prompt construction, and shows that enforcing both diversity via multi-prompt and performance via prompt selection improves candidate generation. A direct comparison between prompt sampling and selection using the same candidate set size is included in Table 6 in Appendix C.3.

5.3 Does multi-prompt MBR improve quality across model architectures and sizes?

Multi-prompt Improves MBR Across Models. Figure 5 reports improvement of multi-prompt over single prompt across widely used LLMs as a Δ change in score, with per-model results in Appendix C.4. In all cases, multi-prompt outperforms single prompt using a sufficiently large candidate set size, showing an increasing or constant metric improvement. In fact, smaller models surpass their larger counterparts' single output decoding at large enough candidate set sizes (Fig. 10). For instance, CodeLlama 13B outperforms its 70B variant using multi-prompt with 18 candidates ($48.26 > 47.99$ pass@1) and TowerInstruct 7B outperforms 13B with 5 candidates ($81.73 > 80.14$ COMET).

LLMs with Multi-prompt Outperform Fine-tuned Models. Whether general-purpose, instruction fine-tuned LLMs outperform models trained on a specific generation task is still an active question (Qin et al., 2023), so we compare state-of-the-art results from each task dataset using single prompt MBR to instruction fine-tuned LLMs using multi-prompt MBR with top-p prompt sampling. In Table 2, we report previous SOTA results for each task: an 11B T5-based text simplification model with control tokens for simplification operations (Sheang and Saggion, 2021), the EN-CS results for the WMT '22 winning submission (Kocmi et al., 2022) and StarCoder 15B, a code infilling and generation LLM (Li et al., 2023), not explicitly trained to follow natural language instructions. LLMs surpass fine-tuned model performance when using

Evaluation Metric →

MBR Utility Metric ↓

Text Simplification (LLaMA 7B Chat)

	SARI	BERTSCORE	LENS	LENS-SALSA$_{RF}$	SLE$_{RF}$
SARI	+1.08*	+1.06*	+7.24*	+4.33*	+0.38*
BERTSCORE	+1.44*	+1.09*	+6.18*	+3.11*	+0.45*
LENS	-0.67	-0.05	+5.78*	+4.69*	+0.82*
LENS-SALSA^{RF}	-0.83	+0.35*	+8.10*	+4.65*	+0.97*
SLE^{RF}	-5.25	-4.71	+2.39*	-4.51	+1.05*

Translation (ALMA 7B)

	BERTSCORE	COMET-22	COMETKIWI$_{RF}$	XCOMET	METRICX	METRICX-QE$_{RF}$
BLEU	+0.34*	+0.47*	+0.67*	-0.14	+0.04	+0.11*
BERTSCORE	+0.51*	+1.59*	+1.68*	+2.48*	+0.22*	+0.29*
COMET-22	+0.71*	+0.89*	+1.72*	+3.29*	+0.13*	+0.18*
COMETKIWI^{RF}	+0.80*	+1.03*	+1.06*	+2.87*	+0.07*	+0.08*
XCOMET	+0.14	+0.85*	+0.84*	+3.34*	+0.09*	+0.04*
METRICX	+0.36*	+0.81*	+0.36	+3.93*	+0.07*	-0.04
METRICX-QE^{RF}	+0.60*	+1.68*	+2.11*	+5.31*	+0.08*	+0.03*

Table 3: Δ metric improvement from single prompt to multi-prompt across metrics. RF = Reference-free reranker. * = Statistically significant improvement with $p < 0.05$. For absolute performance, see Table 8.

multi-prompt, for instance Llama 2 13B shows +5.8 LENS over fine-tuned T5 11B.

5.4 Does multi-prompt MBR over-fit to the utility metric?

An inherent challenge of evaluating MBR is that the utility metric used to select candidates is typically also used for the final evaluation, in such cases it is difficult to attribute the metric improvement to higher quality generation (Bertsch et al., 2023). Given growing attention to metric development, we leverage various trained metrics to test whether multi-prompt using one utility metric improves performance cross all other utility metrics. We experiment with traditional overlap-based metrics, (BLEU, SARI), embedding similarity (BERTSCORE), small (~100M parameter) trained metrics with references (LENS, COMET-22) and without references (COMETKIWI, LENS-SALSA, SLE), and large (3B+ parameter) trained metrics (XCOMET, METRICX, METRICX-QE). These metrics represent diverse text evaluation approaches and encompass the full state of evaluation in both tasks. We include a full description of metric architectures in Appendix B.1.

Multi-prompt MBR Improves Across Metrics. Table 3 reports results for cross-metric evaluation, with the diagonal reflecting the traditional MBR evaluation setup (i.e., calculate MBR and evaluate using the same metric) and other cells indicate generalization from one metric to all others. Multi-prompt improves performance on most evaluation setups, with a few notable exceptions such as disagreement between trained and overlap-based metrics for simplification and COMET-based metrics for translation. For simplification, trained metrics' failure when evaluated by SARI and BERTSCORE may be a byproduct of the test set size, as these metrics typically require a substantial number of references for stable evaluation (Alva-Manchego et al., 2020), more than what are provided in SIMPEVAL. Interestingly, the magnitude of performance improvement is highly variable to the specific utility metric, with no clear relationship between the metric architecture and improvement of multi-prompt, but typically a lower baseline performance indicates multi-prompt performs better (Table 8 in Appendix for more details).

5.5 How does the metric type impact multi-prompt MBR?

As discussed by Fernandes et al. (2022), the MBR operation requires each candidate evaluate against every other candidate (i.e., $\mathcal{O}(n^2)$ comparisons), this becomes inefficient in practice for a large n, especially when using a trained utility metric. Therefore, we explore multi-prompt MBR alternatives using reference-free utility metrics:

- **Reranker ($\mathcal{O}(n)$).** Re-ranking directly estimates the quality of each candidate using a reference-free metric: $\hat{y}_{\text{MBR}} = \arg\max_{y \in \mathcal{H}} [\mathrm{U}(y)]$. We use the trained LENS-SALSA for simplification (Heineman et al., 2023) and COMET-MQM (Rei et al., 2021) for translation. For code generation, we use Code Reviewer (Shi et al., 2022), which calculates agreement between the per-token probability of the generation given the docstring and the original docstring given the generation. Reference-free re-ranking only requires n metric calculations to directly estimate quality.
- **Reranker + MBR ($\mathcal{O}(n + m^2)$).** We use a two-stage selection where we first rerank all n candidates and select the top m to use for MBR, where the cheap re-ranker can distill the candidate set and the expensive MBR metric performs the final selection, where $m \ll n$.
- **Multi-turn MBR ($\mathcal{O}(n^2 + m^2)$).** Similar to the previous approach, we perform MBR and then re-compute MBR using the top m candidates.

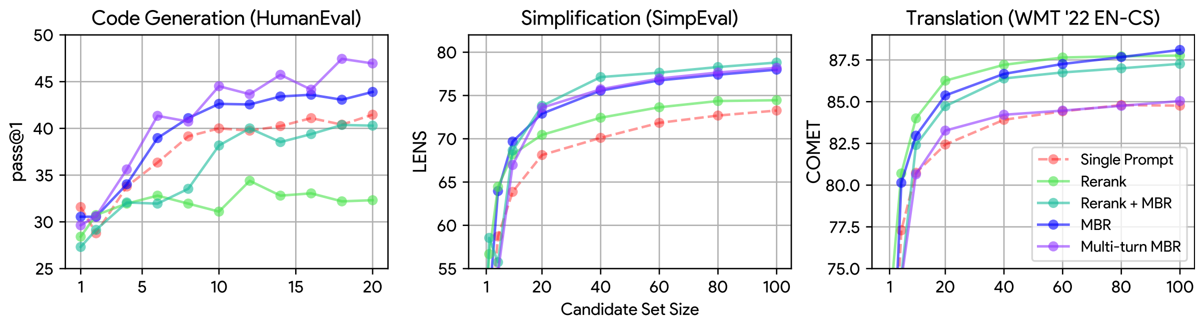

Figure 6: Alternative MBR formulations for multi-prompt across candidate set sizes for code generation, text simplification and translation. Efficient MBR methods show inconsistent results, dependent on task and metric.

Results. We report results across candidate selection methods in Figure 6, finding the multi-prompt achieves performance improvement across reference-based and reference-free metrics, yet the relative performance of methods varies between tasks. With text simplification, the methods first narrowing the candidate set ('Rerank + MBR') and iteratively performing MBR ('Multi-turn MBR') either match or out-perform vanilla MBR. We speculate the first pass may prune the lowest quality generations such that the second pass only considers a distilled candidate set, which better informs the MBR calculation. For translation, the more efficient re-ranker outperforms vanilla MBR, which follows recent work finding trained reference-based and reference-free MT metrics are approaching a similar quality (Freitag et al., 2023b). For code generation, the re-ranker under-performs MBR, which may be reflective of the performance of Code Reviewer compared to MBR-EXEC, as the latter has access to multiple test cases.

6 Related Work

Output Selection. Ensembling outputs across a generation set has become a widely used technique for improving LLM performance in classification tasks, such as using a majority vote over reasoning chains (Wang et al., 2023), or merging outputs from multiple models (Kobayashi, 2018; Martínez Lorenzo et al., 2023). This work applies the same underling concept to text generation by leveraging trained automatic evaluation metrics. To our knowledge, it is the first to propose a multi-prompt decoding scheme for text generation.

MBR Decoding. MBR decoding has been previously used to improve generation quality for machine translation (Kumar and Byrne, 2004; Eikema and Aziz, 2020; Müller and Sennrich, 2021) text simplification (Maddela et al., 2023), summarization and style transfer (Suzgun et al., 2023). Bertsch et al. (2023) highlight the growing popularity of MBR as a simple technique in machine translation and reporting shared tasks results. While our work is the first to propose generating the MBR hypothesis space using a prompt bank, Farinhas et al. (2023) perform preliminary experiments with paraphrases of a single sentence prompt, but found no difference in performance. Recent work argues sampling strategies like nucleus (Eikema and Aziz, 2022) or epsilon (Freitag et al., 2023a) offer slightly better performance over beam search for MBR, with this work extending their findings by attributing candidate set quality to sampling diversity.

Prompt Selection. Current work on prompting for text generation has instead focused on optimization, such as in-context example selection (Min et al., 2022), example ordering (Lu et al., 2022) and prompt selection (Gonen et al., 2023). Notably, Agrawal et al. (2023) show selecting in-context examples for MT by maximizing n-gram overlap between the source and examples improves few-shot performance. Zhou et al. (2023) experiment with LLMs as prompt generators, and Yang et al. (2023) show using LLMs to iteratively rewrite prompts on a development set can distill a single, high-performant prompt. Our work builds on LLM-written prompts and basic heuristics for distilling the prompt bank to further improve multi-prompt.

7 Conclusion

In this work, we propose multi-prompt, a generalized case of MBR for conditional text generation. Multi-prompt successfully ensembles outputs of instruction fine-tuned language models across prompt constructions and in-context examples. We highlight the importance of prompt selection and sampling when constructing the prompt bank with top-p prompt sampling and further verify our results across tasks, models and utility metrics.

Limitations

We limit our study of the prompt bank to a basic set of seed prompts and GPT-written paraphrases for each task. Notably, we do not study the impact of prompt formats (e.g., `passage:{}\n answer{}` vs. `Passage::{} Answer::{}`, Sclar et al., 2023), in-context example ordering (Lu et al., 2022) or example selection (Agrawal et al., 2023) on multi-prompt performance, although multi-prompt may extend to such methods. We leave the question of exhaustively constructing a prompt bank to future work, perhaps by extending work in prefix tuning (Li and Liang, 2021).

An inherent limitation of MBR is the increase in inference time, where we generate up to 100 samples in our experiments, and use a neural utility metric with either linear or quadratic comparisons between candidates. In practice, the generation time was significantly lowered by decoding in parallel and the use of efficient-memory attention techniques such as paged and flash attention used in the vLLM library (Kwon et al., 2023). The computational bottleneck for large candidate set sizes was instead evaluating the utility metrics across all pairs of generated candidates. To lower the number of metric comparisons, promising results have been demonstrated by pruning low-scoring candidates during the MBR process (Cheng and Vlachos, 2023), aggregating embedding representations of candidates (Vamvas and Sennrich, 2024) or selecting a subset of references for each candidate using heuristics on reference embeddings (Deguchi et al., 2024). Similarly, we show in §5.5 efficient alternatives to MBR such as using reference-free metrics largely preserve the benefits from multi-prompt.

Along with MBR, many widely used methods improving LLM abilities trade increased compute at inference time for higher performance, such as using chain-of-thought to decode a reasoning chain for a single answer or using self-consistency to selects an answer among multiple reasoning chains (Wei et al., 2022b; Wang et al., 2023).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (un) suitability of automatic evaluation metrics for text simplification. *Computational Linguistics*, 47(4):861–889.

Duarte M Alves, José Pombal, Nuno M Guerreiro, Pedro H Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, et al. 2024. Tower: An open multilingual large language model for translation-related tasks. *arXiv preprint arXiv:2402.17733*.

Chantal Amrhein and Rico Sennrich. 2022. Identifying weaknesses in machine translation metrics through minimum Bayes risk decoding: A case study for COMET. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1125–1141, Online only. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew Gormley. 2023. It's MBR all the way down: Modern generation techniques through the lens of minimum Bayes risk. In *Proceedings of the Big Picture Workshop*, pages 108–122, Singapore. Association for Computational Linguistics.

Peter J Bickel and Kjell A Doksum. 1977. *Mathematical statistics: Basic ideas and selected topics, volumes I-II package*. Chapman and Hall/CRC.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Julius Cheng and Andreas Vlachos. 2023. Faster minimum Bayes risk decoding with confidence-based pruning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12473–12480, Singapore. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Liam Cripwell, Joël Legrand, and Claire Gardent. 2023. Simplicity level estimate (SLE): A learned reference-less metric for sentence simplification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12053–12059, Singapore. Association for Computational Linguistics.

Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe, Hideki Tanaka, and Masao Utiyama. 2024. Centroid-based efficient minimum bayes risk decoding. *arXiv preprint arXiv:2402.11197*.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2022. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

António Farinhas, José G. C. de Souza, and André F. T. Martins. 2023. An empirical study of translation hypothesis ensembling with large language models. *Preprint*, arXiv:2310.11430.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. NTREX-128 – news test references for MT evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24, Online. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023a. Epsilon sampling rocks: Investigating sampling strategies for minimum Bayes risk decoding for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9198–9209, Singapore. Association for Computational Linguistics.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022a. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. 2023b. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, pages 578–628, Singapore. Association for Computational Linguistics.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022b. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Hila Gonen, Srini Iyer, Terra Blevins, Noah Smith, and Luke Zettlemoyer. 2023. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10136–10148, Singapore. Association for Computational Linguistics.

Naman Goyal, Jingfei Du, Myle Ott, Giri Anantharaman, and Alexis Conneau. 2021. Larger-scale transformers for multilingual masked language modeling. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 29–33, Online. Association for Computational Linguistics.

Nuno M Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André FT Martins. 2023. xCOMET: Transparent machine translation evaluation through fine-grained error detection. *arXiv preprint arXiv:2310.10482*.

David Heineman, Yao Dou, Mounica Maddela, and Wei Xu. 2023. Dancing between success and failure: Edit-level simplification evaluation using SALSA. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages

3466–3495, Singapore. Association for Computational Linguistics.

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

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

T Jaeger and Roger Levy. 2006. Speakers optimize information density through syntactic reduction. *Advances in neural information processing systems*, 19.

Siddhartha Jain, Xiaofei Ma, Anoop Deoras, and Bing Xiang. 2023. Self-consistency for open-ended generations. *arXiv preprint arXiv:2307.06857*.

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*.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960, Online. Association for Computational Linguistics.

Juraj Juraska, Mara Finkelstein, Daniel Deutsch, Aditya Siddhant, Mehdi Mirzazadeh, and Markus Freitag. 2023. MetricX-23: The Google submission to the WMT 2023 metrics shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 756–767, Singapore. Association for Computational Linguistics.

Marzena Karpinska, Nishant Raj, Katherine Thai, Yixiao Song, Ankita Gupta, and Mohit Iyyer. 2022. DEMETR: Diagnosing evaluation metrics for translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9540–9561, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. 2022. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3631–3643, Seattle, United States. Association for Computational Linguistics.

Hayato Kobayashi. 2018. Frustratingly easy model ensemble for abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4165–4176, Brussels, Belgium. Association for Computational Linguistics.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. 2023. Starcoder: May the source be with you! *arXiv preprint arXiv:2305.06161*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16383–16408, Toronto, Canada. Association for Computational Linguistics.

Abelardo Carlos Martínez Lorenzo, Pere Lluís Huguet Cabot, and Roberto Navigli. 2023. AMRs assemble! learning to ensemble with autoregressive models for AMR parsing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1595–1605, Toronto, Canada. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2023. State of what art? A call for multi-prompt LLM evaluation. *arXiv preprint arXiv:2401.00595*.

Mathias Müller and Rico Sennrich. 2021. Understanding the properties of minimum Bayes risk decoding in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 259–272, Online. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

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.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is ChatGPT a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Nuno M. Guerreiro, JosÃ© Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.

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

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code LLaMA: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

Kim Cheng Sheang and Horacio Saggion. 2021. Controllable sentence simplification with a unified text-to-text transfer transformer. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 341–352, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Freda Shi, Daniel Fried, Marjan Ghazvininejad, Luke Zettlemoyer, and Sida I. Wang. 2022. Natural language to code translation with execution. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3533–3546, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, and et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2023. Follow the wisdom of the crowd: Effective text generation via minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4265–4293, Toronto, Canada. Association for Computational Linguistics.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, et al. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.

Jannis Vamvas and Rico Sennrich. 2024. Linear-time minimum bayes risk decoding with reference aggregation. *arXiv preprint arXiv:2402.04251*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with ChatGPT. *arXiv preprint arXiv:2302.11382*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models. *arXiv preprint arXiv:2309.11674*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. *arXiv preprint arXiv:2401.08417*.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

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

Tianyi Zhang, Tao Yu, Tatsunori Hashimoto, Mike Lewis, Wen-tau Yih, Daniel Fried, and Sida Wang. 2023. Coder reviewer reranking for code generation. In *International Conference on Machine Learning*, pages 41832–41846. PMLR.

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

Human-Written Text Simplification Prompt
Rewrite the following complex sentence in order to make it easier to understand by non-native speakers of English. You can do so by replacing complex words with simpler synonyms (i.e. paraphrasing), deleting unimportant information (i.e. compression), and/or splitting a long complex sentence into several simpler ones. The final simplified sentence needs to be grammatical, fluent, and retain the main ideas of its original counterpart without altering its meaning.
Simplify the sentence please.
You are an artificial intelligence designed to simplify human written text. The text you are given will contain complex ideas, phrases or concepts and your job is to rewrite that text in a simple and easy to understand way. Your simplification should be completely fluent and retain the ideas of the simplification.
I would like you to simplify the following sentence such that the text is as concise and easy to read as possible.
Text simplification is an operation used in natural language processing to change, enhance, classify, or otherwise process an existing body of human-readable text so its grammar and structure is greatly simplified while the underlying meaning and information remain the same. Text simplification is an important area of research because of communication needs in an increasingly complex and interconnected world more dominated by science, technology, and new media. But natural human languages pose huge problems because they ordinarily contain large vocabularies and complex constructions that machines, no matter how fast and well-programmed, cannot easily process. However, researchers have discovered that, to reduce linguistic diversity, they can use methods of semantic compression to limit and simplify a set of words used in given texts. Please simplify the following sentence.
Please simplify the below sentence by using a combination of these three operations. Elaboration. An addition of meaningful, relevant and correct information, such as clarifying vague terminology, providing background information on an entity or subject, or explicating general world knowledge unknown to the audience. Generalization. A deletion of unnecessary, irrelevant or complicated concepts. Paraphrase. Swapping complex spans with equivalent, simpler alternatives. The final sentence should be grammatical, concise and easier to read compared to the original sentence.
You are an AI assistant that writes text simplification. Text simplification can be defined as any process that reduces the syntactic or lexical complexity of a text while attempting to preserve its meaning and information content. The aim of text simplification is to make text easier to comprehend for a human user, or process by a program. Please simplify the following sentence.
Simplify.
You are to act as a text simplification bot. As a text simplification bot, you will simplify the following sentence such that it is syntactically easier to read and semantically easier to understand. Please do not make the text more complex, longer or difficult for a reader.
I am writing a sentence, please take a look at this sentence and write a simpler version such that a non-English speaker or an individual with disabilities could better understand the sentence.

Table 4: Text simplification prompts used for the decoding experiment in Figure 3 and used as examples to write GPT-4 prompts for experiments in §5.

A Prompt Bank Construction

Table 4 contains the human-written prompts for text simplification. These human-written prompts are provided as examples to GPT-4 when automatically generating prompts for large-scale experiments in §5. For code generation, we extract the docstring in the original HUMANEVAL examples as the human-written prompt, and provide it as an example prompt to GPT-4. For machine translation, our few-shot examples were sampled randomly from the WMT `newstest19` test corpus (Barrault et al., 2019).

B Detailed System Descriptions

In this section, we include a full description of the generation models and utility metrics used in experiments throughout §5.3 and §5.4. All experiments were inference-based and were run on up to 4xN-

Prompt-Generation Instruction
Please write a variation of the following instruction for a coding task. You may be creative in proposing potential solutions, or explaining the nature of the task. Please do not write any examples. Example: `{example_prompt}` Prompt:
Create a prompt for a language model to simplify a sentence, this prompt will explain the text simplification task and instructions for how to perform the task. The prompt should be diverse, include a description of simplification and clearly state what is expected of the language model. Example: `{example_prompt_1}` Example: `{example_prompt_2}` Prompt:

Table 5: Instruction templates provided to GPT-4 when generating task instructions for code generation (top) and text simplification (bottom).

VIDIA A40 GPUs, depending on the requirements of the specific model or utility metric. The use of models, metrics and datasets in this project follows their respective licenses and intended use.

B.1 Utility Metrics

B.1.1 Code Generation

MBR-EXEC (Shi et al., 2022) executes candidate generations on a series of test cases, and selects the candidate with the highest agreement on its output with all other candidates. While the authors do not evaluate on HUMANEVAL, we replicate the setup in Zhang et al. (2023) by using the test cases in the docstring to calculate the agreement. We use a soft loss over all test cases, as many HUMANEVAL docstring examples are trivial or edge cases. If two candidates have the same MBR score, we break ties using the candidate with higher probability under the language model.

Code Reviewer (Zhang et al., 2023) attempts to find a consensus between the likelihood of the generated program $p(y|x)$ and the original docstring using a minified version of the generation $p(x|y)$. We use their implementation for rejecting degenerate samples, minifying code and calculating the reviewer score. We use the same models for generation and re-ranking.

B.1.2 Simplification

SARI (Xu et al., 2016) is an n-gram overlap based metric that compares edits on inputs, outputs and a bank of references.

BERTSCORE (Zhang et al., 2020) calculates a word-level cosine similarity of BERT embeddings. Alva-Manchego et al. (2021) find BERTSCORE is an adequate measure of quality generation, but that it does not correlate with simplicity.

LENS (Maddela et al., 2023) is a RoBERTa-based metric trained using human ratings of text simplification model outputs. The authors train on an adaptive loss to allow a high score for generations was close to *any* references, encouraging the metric to consider different simplification types.

Lens-Salsa (Heineman et al., 2023) extends the Lens architecture by fine-tuning on a dual sentence- and word-level quality objective. The authors show Lens-Salsa is more sensitive to specific edit operations, while not requiring any reference simplifications.

SLE (Cripwell et al., 2023) is a RoBERTa-based metric trained to estimate the simplicity of text, with the simplicity score defined as the difference in simplicity between the complex and simplified sentences. SLE was trained on 0-4 readability scores of news articles in the Newsela corpus (Xu et al., 2015), with an additional label softening for individual sentences in the corpus.

B.1.3 Translation

BLEU (Papineni et al., 2002) is an n-gram overlap based metric comparing a translation to a bank of references. BLEU remains a widely-used standard for automatic evaluation, despite lower correlation to human judgement compared to learned metrics (Freitag et al., 2022b). We use the ScareBLEU implementation (Post, 2018).

COMET (Rei et al., 2020) is a widely used RoBERTa-based metric, trained on direct assessments of simplification quality. For reference-free evaluation, we use the CometKiwi-XXL variant (Rei et al., 2022, 2023), trained to predict sentence- and word-level scores simultaneously.

xCOMET (Guerreiro et al., 2023) is a fine-tuned XLM-R model (Goyal et al., 2021) based on the CometKiwi architecture, but scaling the model size and training data, including with synthetic data created by randomly swapping n-grams or entire sentences with unrelated translations. We use the 11B xCOMET-XXL in our experiments.

MetricX (Juraska et al., 2023) is a recent fine-tuned 11B mT5-XXL (Xue et al., 2021) trained on DA data from 2015-20, MQM data from 2020-21 (Freitag et al., 2021) and synthetic data based on the MQM and DEMETR (Karpinska et al., 2022) taxonomies of translation errors. Notably, the MetricX architecture encodes both candidates and references together, while COMET encodes both separately and combines the outputs to calculate the final score. We also use the QE variant MetricX-QE trained without references. The WMT '22 test data used in this work is not included in the training data of any translation metrics we considered.

B.2 Model Architectures

B.2.1 Code Generation

StarCoder 2 (Li et al., 2023) is trained from-scratch on 4T tokens from 600+ programming languages. Although the model is not instruction fine-tuned, we see a slight performance improvement with multi-prompt, likely because comments and code descriptions are included in its pre-training.

CodeLLaMA (Roziere et al., 2023) is a fine-tuned Llama 2 model on 500B-1T tokens of code-related datasets, including Python, substantially outperforming the base Llama 2 model on HumanEval.

B.2.2 Simplification

Instruction Fine-tuned Models. We experiment with widely used instruction fine-tuned LLMs, aiming for a broad coverage of current models: Llama 2 Chat (Touvron et al., 2023), Gemma (Team et al., 2024) and Mistral (Jiang et al., 2023).

Fine-tuned Control T5 (Sheang and Saggion, 2021) is a T5-based text simplification model fine-tuned on the Wiki-Auto (Jiang et al., 2020) dataset of aligned English-Simple English Wikipedia articles. We use their same control token setup: <NC_0.95> <LS_0.75> <DR_0.75> <WR_0.75>.

B.2.3 Translation

ALMA-R (Xu et al., 2024) is a class of translation LLMs. The base ALMA (Xu et al., 2023) is a fine-tuned LLaMA model with text in each target language and then parallel translation data. ALMA-R is an extension trained on a contrastive preference loss to incorporate ratings of translation quality.

TowerInstruct (Alves et al., 2024) is a fine-tuned Llama 2 model on multi-lingual instructions, aiming to incorporate tasks beyond translation, such as paraphrasing, post editing and grammar error correction.

Aya 101 (Üstün et al., 2024) is an mT5-based model fine-tuned on multi-lingual data in 101 languages. While mT5 is instruction-following model, Aya is not fine-tuned on instruction data.

Additionally, we provide results from the WMT '22 winning submission, and the Microsoft Translate API, as reported in Hendy et al. (2023).

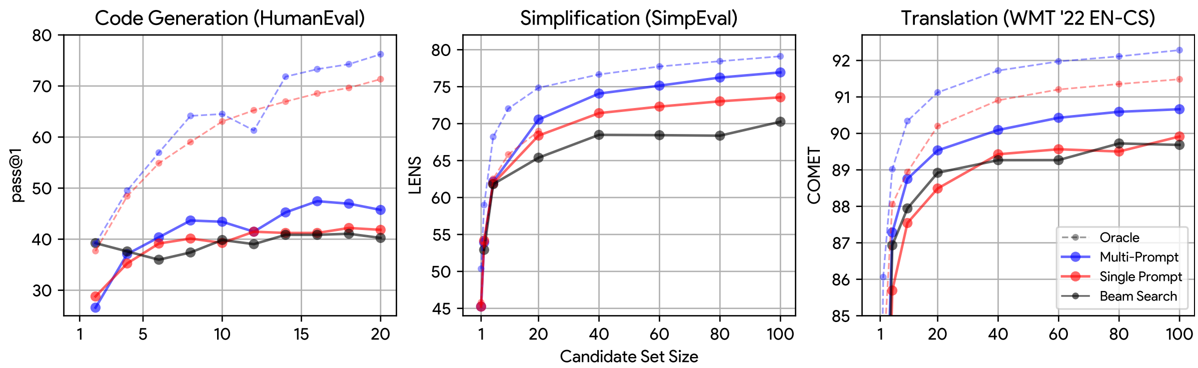

Figure 7: Multi-prompt, single prompt and beam search MBR decoding performance across candidate set sizes for code generation, text simplification and translation. Results averaged over 5 bootstrap iterations.

C Further Results

C.1 Beam Search & Oracle Performance

Following related work in MBR, we report upper-bound 'oracle' results (similar to Shi et al., 2022) and a lower-bound beam search baseline (similar to Freitag et al., 2023a) in comparison to our main results (Figure 1) in Figure 7.

Beam Search. The MBR candidate set historically has consisted of the top beam search candidates, but as language models have become better generators recent work has argued sampling leads to a better estimation of the hypothesis space (Freitag et al., 2023a). For this reason, we exclusively use nucleus sampling in §5, but we report beam search as a baseline in Figure 7, with a 'candidate set size' of n corresponding to the top n beam candidates, or n candidates with nucleus sampling for other results.

Oracle. As the final MBR performance can be impacted both by the quality of the candidate set and the choice of utility metric, we report an upper-bound performance by deliberately selecting the best candidate generations. Given a test set with gold-standard references $\mathcal{R}$, we define the oracle performance as the set of the highest scoring possible selection of candidates:

$$\text{Oracle}(\mathcal{R}^*) = \sum_{r \in \mathcal{R}^*} \max_{y \in \mathcal{H}} [U(y, r)] \qquad (6)$$

Since code generation is evaluated using pass@1, its oracle uses expected pass@k (Shi et al., 2022), which measures whether at least one candidate within the candidate set passes all unit tests $\mathcal{T}$:

$$\text{ExPass@}K = \mathop{\mathbb{E}}_{|\mathcal{H}|=K} \left[\max_{y \in \mathcal{H}} \min_{t \in \mathcal{T}} \mathbb{1}[t(y)] \right] \qquad (7)$$

Results. As oracle performance measures candidate set quality independent of the utility metric, we find an increase in oracle performance coincides with an improvement when using multi-prompt, indicating that a utility metric can naturally select candidates when the candidate set is higher quality. This suggests improving utility metrics may be a promising direction to bridge the gap between candidate quality and candidate selection. Beam search was a particularly strong baseline for small candidate set sizes, particularly for code generation, but beam search is not as sensitive to improvement as the candidate set size increases. Additionally, as code generation is evaluated using the binary pass@1 metric, rather than a scalar quality metric as used by translation and simplification, there is a large gap between MBR and oracle performance, also observed by Shi et al. (2022).

C.2 En-XX Translation Results

For brevity, we limit our multi-prompt experiments to only the English-Czech language pair, but report results across the full ALMA test set, including WMT '22 test data and a subset of NTREX (Federmann et al., 2022), in Figure 8, where we observe improvement with multi-prompt is dependent on the language pair. Generally, high resource languages (such as French, German, Russian) do not have a substantial difference, which may be a result of the low prompt sensitivity for such pairs.

C.3 Detailed Prompt Selection Results

To further compare prompt sampling and prompt selection with the same candidate set size, we replicate the same experiment as Table 1, but modify prompt selection (bottom) to use 10 candidates for *each* prompt, such that both sampling and selection use 100 candidates. We find similar results when comparing between prompt selection methods, where at least one selection method leads

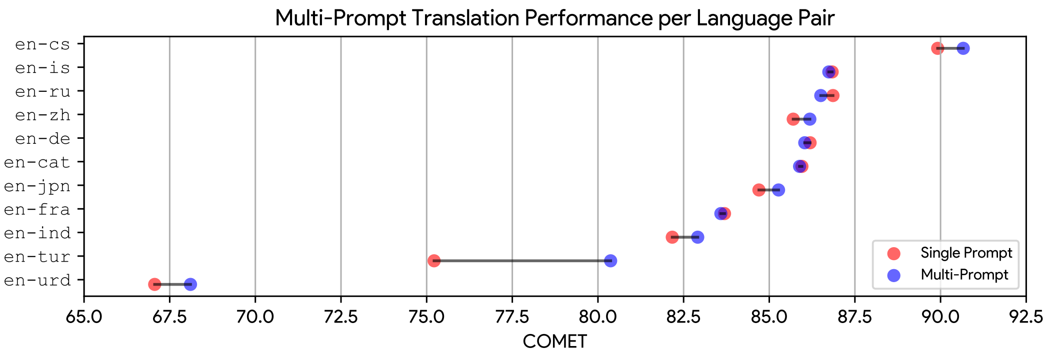

Figure 8: Multi-prompt and single prompt performance of ALMA 7B R across En-XX translation pairs. For low resource language pairs (e.g., Urdu, Turkish, Czech) we observe significant performance improvements, but not for most high resource pairs (e.g., French, German, Russian).

	pass@1	LENS	COMET
Single Prompt ($\|\mathcal{H}\|=100$)	48.78	74.67	88.93
Multi-Prompt + Prompt Sampling ($\|\mathcal{P}\|=100$, $\|\mathcal{H}\|=100$)			
Random Selection	–	74.91*	89.98*
Prompt Sampling	–	78.29*	90.33*
Top-p Prompt Random	–	78.61*	90.11*
Top-p Prompt Sampling	–	**79.08***	**90.36***
Single Prompt ($\|\mathcal{H}\|=100$)	48.78	74.67	88.93
Multi-Prompt + Prompt Selection ($\|\mathcal{P}_{\text{best}}\|=10$, $\|\mathcal{H}\|=100$)			
Random Selection	47.40	70.95	89.90*
k-NN Cluster Random	45.73	72.04	90.14*
Farthest Similarity	**49.17***	71.64	90.18*
Closest Similarity	45.73	72.17	**90.87***
Highest Performance	–	72.56	90.27*
k-NN Cluster Performance	–	**75.88***	90.43*

Table 6: Results for prompt sampling using 100 prompts (top) and subset selection with 100 candidates using 10 of 100 prompts (bottom). * = Statistically significant improvement with $p < 0.05$.

to a statistically significant improvement on each task. However, all prompt selection methods underperform prompt sampling. This underscores the benefit of the increased diversity from generating using a full prompt bank with multi-prompt.

C.3.1 Selected Prompts

To provide intuition into how our prompt sampling impacts the underlying prompt set, we run multi-prompt using 100 prompts, generating a single output from each prompt. Figure 9 reports the frequency that the generation from each prompt was selected as the MBR candidate among all prompts as a % over the full dataset. A flat distribution indicates that each prompt equally produces the final MBR generation, but we find a few prompts receive disproportionately more frequent usage, with some prompts never producing the MBR candidate. Interestingly, both tasks have a very different distribution of usage, perhaps as translation is using

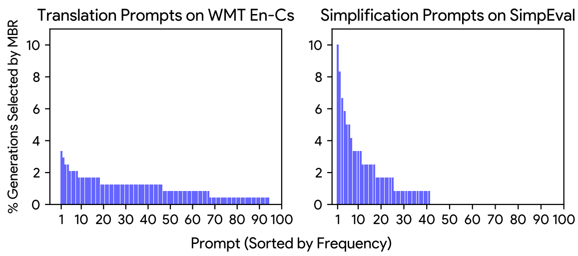

Figure 9: Distribution of prompt usage for 100 prompts. A few high quality prompts generate many final candidates, while many prompts are rarely (if ever) used.

few-shot examples, which may be less sensitive to performance as a natural language task instruction. We show examples of the top prompts for simplification and translation in Table 7.

C.4 Detailed Multi-Model Results

See Figure 10 contains separated results for multi-prompt and single prompt for each model, as reported in Figure 5 and discussed in §5.3.

C.5 Detailed Cross Metric Evaluation

Table 8 contains the full results for the MBR experiments across metrics as discussed in §5.4. While evaluating on the same metric used for MBR clearly improves performance the most (see entries on the diagonal), we find multi-prompt performed on any metric universally improves performance when evaluated on any other metric. Recent neural metrics, which achieve higher correlation with human judgements, also have a higher overall performance. Note, METRICX scores translations on a $[0, 25]$ scale corresponding to an MQM rating, where lower is better and SLE scores simplifications on a $[0, 4]$ corresponding to a Newsela simplification rating, where higher is better. For clarity, we negate the METRICX results in Table 3 such that all the green cells indicate a metric improvement.

Top 10 GPT-4 Generated Text Simplification Prompts (Sorted by No. Generations Selected)
Rewrite the following sentence in a simplified manner, making sure the same meaning and message are still conveyed clearly. The simplification should be done such that it can be read and understood easily by an individual who may not have knowledge of the English language or any disabilities that limit their understanding.
Please simplify the following sentence so that it is easy to understand by people with disabilities or those who are unfamiliar with English. Try to use shorter words, fewer clauses, and a simpler structure.
Simplify this sentence such that a non-English speaker or a person with disabilities is able to understand the sentence. Focus on replacing complex words and structures with simpler ones, while keeping the meaning intact. You can remove unnecessary words, break up longer phrases, and generally make the text more readable.
Text simplification is an important task in natural language processing for creating a simplified version of a sentence that conveys the same meaning as the original sentence but with less complex language. For this task, you will be given a sentence and asked to rewrite it using simpler words and structures so that a non-English speaker or an individual with disabilities can better understand it. Please use semantic compression to create a simplified version of the following sentence.
You are an artificial intelligence designed to simplify written text. The text you are given may be complex, and your job is to rewrite it in a way that a non-english speaker or an individual with disabilities could easily understand. While you simplify the text, you should make sure it is grammatically correct and retains the original meaning of the text.
You are an AI assistant tasked with creating a simpler version of a text. Text simplification can be defined as the reduction of the syntactic or lexical complexity of a text without changing its meaning. The aim of text simplification is to make the text easier to understand for a human or process by a program. Please simplify the following sentence.
Rewrite this sentence in a simple and easy to understand way. Make sure to retain the meaning and ideas of the original sentence while using shorter words and sentences.
Create a simpler version of the sentence below so that it can be better understood by non-English speakers or individuals with disabilities. Text simplification techniques should be used to reduce the complexity of the language while preserving the original meaning and information.
You are an AI assistant that writes text simplification. Text simplification can be defined as any process that reduces the syntactic or lexical complexity of a text while attempting to preserve its meaning and information content. The aim of text simplification is to make text easier to comprehend for a human user, or process by a program. Your task is to take the following sentence and produce a simplified version that would be easier for a non-English speaker or someone with disabilities to understand. Please simplify the sentence.
This prompt asks you to simplify the given sentence. In order to do so, reduce the sentence to its most basic and clear components. Remove unnecessary words, clauses, and phrases that can be inferred from the context. Use shorter, more concise words where possible. After simplifying, the resulting sentence should still convey the same essential message.
Top 5 Randomly Sampled Few-shot Translation Instructions (Sorted by No. Generations Selected)
Anglická věta: To do this, simply access your order page, tap 'Help and support' and choose the option 'Call rider'. Česká věta: Chcete-li to provést, jednoduše přejděte na stránku objednávky, klikněte na „Nápověda a podpora“ a vyberte možnost „Zavolat jezdci“. Anglická věta: A private mass and the national anthem preceded the ceremony, which featured a portrait of De Klerk between two candles and a choir decorated with white flowers. Česká věta: Soukromá mše a státní hymna předcházely tomuto ceremoniálu, který představil portrét De Klerka mez dvěma svíčkami a sbor ozdobený bílými květy. Anglická věta: After that, we cannot offer an estimate on delivery times as it comes down to individual country's postal service and customs if outside of the EU. Česká věta: Poté nemůžeme odhadnout dobu dodání, protože záleží na poštovních a celních službách v jednotlivých zemích, pokud se nacházejí mimo EU. Anglická věta: This item is an original American comic and is in English! Česká věta: Tato položka je originální americký komiks a je v angličtině! Anglická věta: If they cannot find you they will surely call. Česká věta: Pokud vás nenajdou, určitě zavolají.
Anglická věta: New Zealand's computer emergency response team was among the first to report that the flaw was being "actively exploited in the wild" just hours after it was publicly reported Thursday and a patch released. Česká věta: Tým Nového Zélandu pro reakci na počítačové ohrožení byl mezi prvními, kdo nahlásil, že tato závada se „aktivně divoce zneužívá“ jen pár hodin po tom, co byla veřejně nahlášena ve čtvrtek a byla vydána záplata. Anglická věta: Not sure, but I don't think we had any way of having them pay. Česká věta: Nejsem si jistý, ale nemyslím si, že bychom měli nějaký způsob,a by museli zaplatit. Anglická věta: Luckily, the guy was honest and rather than trying to charge the higher price, he sold me the tires for the price I had on my printout. Česká věta: Naštěstí byl ten chlapík čestný a než aby se pokoušel účtovat vyšší cenu, prodal mi pneumatiky za cenu, kterou jsem měl na mém výtisku. Anglická věta: The Cowboys just made sure Zeke and his teammates got that opportunity. Česká věta: Cowboys se právě postarali o to, aby Zeke a jeho spoluhráči tuto příležitost dostali. Anglická věta: Description Please scroll to the bottom of the listing for more pictures. Česká věta: Popis Pro více obrázků sjeďte na konec nabídky.
Anglická věta: This is on a quote only basis and you need to supply us with your address for a quotation. Česká věta: Tato služba je poskytována pouze na základě cenové nabídky dle vámi poskytnuté adresy. Anglická věta: Fed up completely, she asks "Are you even going to work today?" Česká věta: Totálně znechucená se ptá: „Budeš dnes vůbec pracovat?“ Anglická věta: So there was the usual gentle chaos that attends any gathering of toddlers. Česká věta: Takže nastal obvyklý mírný chaos, který provází každé setkání batolat. Anglická věta: We currently do not have the exact information on what happened to the rider as well as to your order. Česká věta: V současné době nemáme přesné informace o tom, co se stalo s jezdcem, stejně jako s vaší objednávkou. Anglická věta: UK media reported that "thousands" were eager to raise cash for the protesters by purchasing the gray T-shirt, which depicts an empty plinth with "Bristol" written above it. Česká věta: Média ve Velké Británii hlásila, že „tisíce lidí“ nedočkavě vybírali hotovost pro protestující zakoupením šedého trička, které zobrazuje prázdný podstavec s napsaným Bristol nad ním.
Anglická věta: A. No, we do not include receipts in packages unless requested. Česká věta: A. Ne, účtenku nepřikládáme, pokud to není požadováno. Anglická věta: Russia warned of 'consequences' if Ukraine attacked Česká věta: Rusko bylo varováno před “následky“, pokud napadne Ukrajinu Anglická věta: He noted that up to 90% of all Russian investments in the Arab world are made in the UAE. Česká věta: Poznamenal, že až 90 % ruských investic v arabském světě jsou prováděny v SAE. Anglická věta: Many view the Softie 12 Osprey the ultimate four season synthetic fill sleeping bag available. Česká věta: Mnohými je spací pytel Softie 12 Osprey považován za nejlepší dostupný čtyřsezónní spacák se syntetickou výplní. Anglická věta: - Sign out and signing back in to your eReader. Česká věta: - Odhlaste se a přihlaste se znovu do vaší e-čtečky.
Anglická věta: I told ya so.... Česká věta: Říkala jsem vám to... Anglická věta: All information about the products on our website is provided for information purposes only. Česká věta: Všechny informace o produktech na našich internetových stránkách mají pouze informativní charakter. Anglická věta: I'm in HR and have worked payroll in the past. Česká věta: Jsem na personálním oddělení a v minulosti jsem pracoval na mzdovém. Anglická věta: Years ago, I worked at a cabinet shop. Česká věta: Před lety jsem pracoval v obchodě se skříněmi. Anglická věta: De Klerk's foundation issued a posthumous video apologizing "for the pain, hurt, indignity and damage that apartheid has done" to South Africa's non-white populations. Česká věta: Fond De Klerka vydal posmrtné video omlouvající se „za bolest, zranění, ponížení a škodu, kterou apartheid udělal „jihoafrickému nebělošskému obyvatelstvu“.

Table 7: Prompts with highest usage for multi-prompt using the held-out split for simplification and translation.

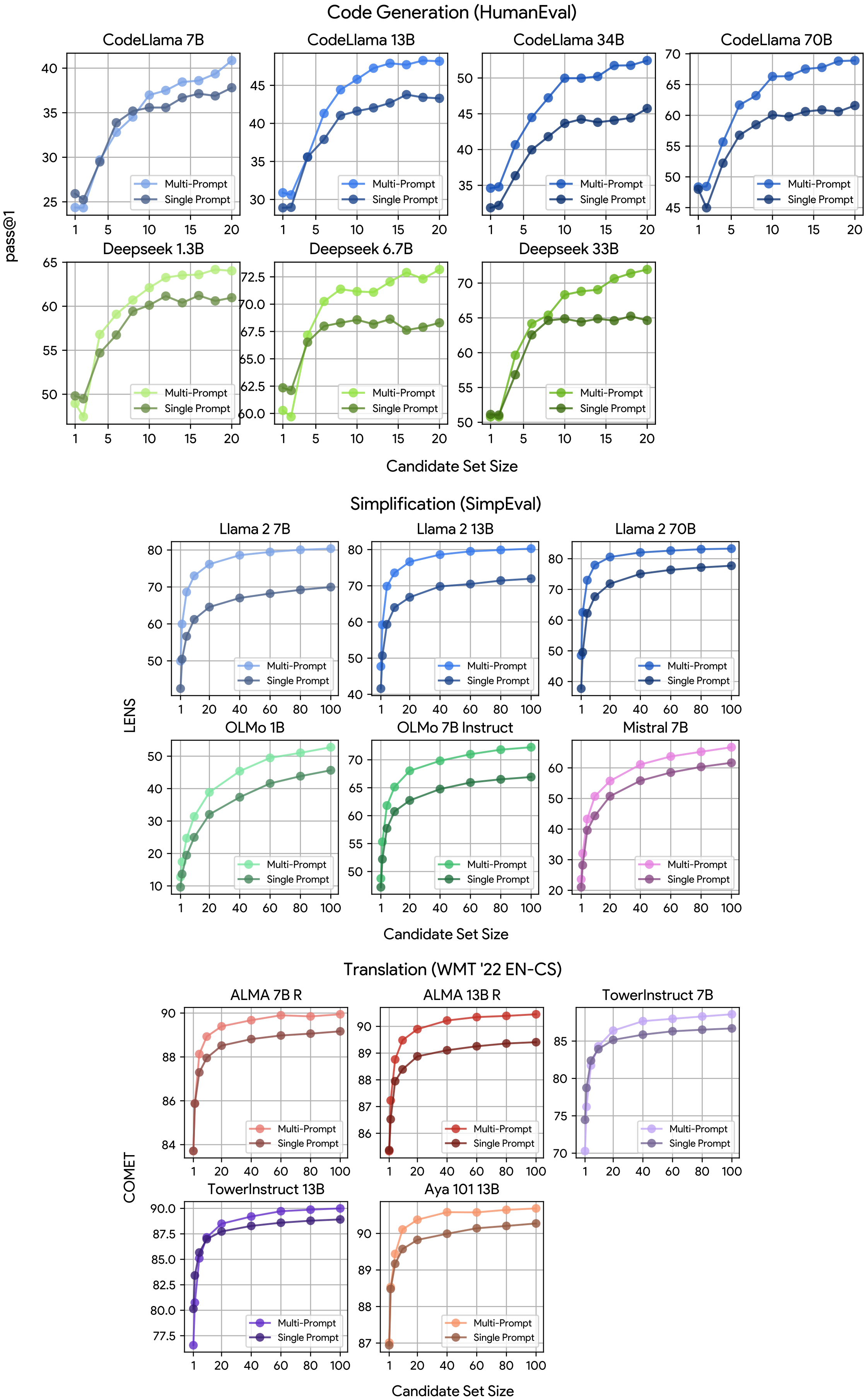

Figure 10: Results of multi-prompt MBR compared to single prompt MBR across model sizes and architectures. Multi-prompt MBR consistently improves performance across architectures and as models scale. A candidate size of 1 is equivalent to standard, single-output decoding.

Evaluation Metric →

MBR Utility Metric ↓

Text Simplification (LLaMA 7B Chat) — Multi-prompt

	SARI	BERTSCORE	LENS	LENS-SALSA^{RF}	SLE^{RF}
SARI	44.33	92.64	58.73	72.31	1.42
BERTSCORE	45.46	93.71	60.86	71.47	1.37
LENS	39.98	92.18	76.29	79.55	2.30
LENS-SALSA^{RF}	38.55	91.29	73.31	84.59	2.47
SLE^{RF}	33.57	85.36	52.33	64.74	3.84

Translation (ALMA 7B) — Multi-prompt

	BERTSCORE	COMET-22	COMETKIWI^{RF}	XCOMET	METRICX	METRICX-QE^{RF}
BLEU	90.91	87.12	81.16	72.43	1.15	1.24
BERTSCORE	91.41	88.11	82.15	73.59	1.10	1.15
COMET-22	90.45	91.18	86.17	76.71	0.61	0.63
COMETKIWI^{RF}	90.67	90.56	85.64	81.16	0.51	0.57
XCOMET	90.15	90.03	83.19	86.73	0.70	0.79
METRICX	89.35	89.07	82.00	69.26	0.47	0.69
METRICX-QE^{RF}	89.58	89.29	83.93	68.78	0.43	0.25

Evaluation Metric →

Text Simplification (LLaMA 7B Chat) — Single prompt

	SARI	BERTSCORE	LENS	LENS-SALSA^{RF}	SLE^{RF}
SARI	43.25	91.58	51.49	67.97	1.04
BERTSCORE	44.02	92.62	54.68	68.36	0.92
LENS	40.64	92.24	70.51	74.86	1.49
LENS-SALSA^{RF}	39.38	90.94	65.21	79.93	1.51
SLE^{RF}	38.82	90.07	49.94	69.26	2.79

Translation (ALMA 7B) — Single prompt

	BERTSCORE	COMET-22	COMETKIWI^{RF}	XCOMET	METRICX	METRICX-QE^{RF}
BLEU	90.57	86.65	80.49	72.57	1.20	1.35
BERTSCORE	90.90	86.52	80.48	71.10	1.31	1.44
COMET-22	89.74	90.28	84.44	73.42	0.74	0.81
COMETKIWI^{RF}	89.87	89.53	84.58	78.29	0.58	0.65
XCOMET	90.01	89.18	82.35	83.39	0.79	0.83
METRICX	88.99	88.26	81.63	65.32	0.54	0.66
METRICX-QE^{RF}	88.98	87.61	81.82	63.47	0.50	0.27

Table 8: Multi-prompt and single prompt performance across metrics. RF = Reference-free reranker.