RewriteLM: An Instruction-Tuned Large Language Model for Text Rewriting

Lei Shu*, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu,
Simon Tong, Jindong Chen, Lei Meng*
Google Research
{leishu, luolc, jayakumar, yunzhu, canoee, simon, jdchen, leimeng}@google.com

Abstract

Large Language Models (LLMs) have demonstrated impressive capabilities in creative tasks such as storytelling and E-mail generation. However, as LLMs are primarily trained on final text results rather than intermediate revisions, it might be challenging for them to perform text rewriting tasks. Most studies in the rewriting tasks focus on a particular transformation type within the boundaries of single sentences. In this work, we develop new strategies for instruction tuning and reinforcement learning to better align LLMs for cross-sentence rewriting tasks using diverse wording and structures expressed through natural languages including 1) generating rewriting instruction data from Wiki edits and public corpus through instruction generation and chain-of-thought prompting; 2) collecting comparison data for reward model training through a new ranking function. To facilitate this research, we introduce OPENREWRITEEVAL, a novel benchmark covers a wide variety of rewriting types expressed through natural language instructions. Our results show significant improvements over a variety of baselines.

Introduction

Text rewriting plays an essential role in a wide range of professional and personal written communications. It can be conceptualized as a form of controllable text generation (Zhang et al. 2022a) , where a specified textual input is modified based on the user's requirement. Several categories of text rewriting have been extensively researched, such as paraphrasing (Siddique, Oymak, and Hristidis 2020; Xu et al. 2012), style transfer (Riley et al. 2020; Zhang, Ge, and Sun 2020; Reif et al. 2021), and sentence fusion (Mallinson et al. 2022).

Recent advances in Large Language Models (LLMs) have shown impressive zero-shot capabilities in a wide range of text generation tasks expressed through natural language instructions (Chung et al. 2022). However, user expectation for text rewriting is high and any unintended edits by the model negatively impact the user's satisfaction. Given that the LLMs can be hard to control (Qin et al. 2023) and prone to generating "hallucinated" content (Ji et al. 2023), we propose new methods to ensure that the model is properly trained and evaluated.

Source Text: The way in which we feed our children determines how healthy they are. It also determines how nicely they develop and develop. Children need to be fed a selection of foods each day in order to make sure their physical, emotional, and mental health. All foods are from 1 of the basic food groups. Milk and cheese come from the dairy group, for instance, and green beans come from the vegetable group. Bread comes from the grain group and beef comes from the meat group. Chocolate comes from the fat and sugars group. Our children need so many servings per day from each of these food groups to maintain good health.

Instruction: Rewrite the text so that it is easy to understand.

Target Text: What we give our kids to eat affects how healthy they are and how they grow. Kids need to eat different types of food each day to stay physically, emotionally, and mentally healthy. All foods belong to one of the five food groups: dairy, vegetables, grains, meat and beans, and fruits. Milk and cheese are dairy foods, green beans are vegetables, bread is a grain, beef is a meat, and chocolate is a fat and sugar food. To stay healthy, kids need to eat a certain number of servings from each food group every day.

Instruction: Summarize the text.

Target Text: Feeding children a variety of foods from the five fundamental food groups (dairy, vegetables, grains, meat, and fats/sweets) is essential for their overall health and development, including their physical, emotional, and mental well-being.

Table 1: Rewrite examples. The input for RewriteLM contains an instruction and a source text. The target is a rewrite. We present one source text and two instructions here for illustration purpose.

We present a strong model — RewriteLM[1], an instruction-tuned large language model for cross-sentence text rewriting. Similar to InstructGPT (Ouyang et al. 2022), we perform supervised fine-tuning and reinforcement learning (RL) steps. However, InstructGPT heavily relies on human labelers to collect instruction data (for supervised fine-tuning) and comparison data (for reward training). We instead develop the following data and training strategies with minimal human intervention: 1) For instruction tuning, we extract cross-sentence

[*] These authors contributed equally.

[1] Github: https://github.com/google-research/google-research/tree/master/rewritelm

edits from Wiki edits and improves its instruction. To improve the diversity of the dataset, we also generate synthetic instruction and rewrite output from public corpus through chain-of-thought prompting and post-processing 2) For reinforcement learning, we collect comparison data for reward model training through a new ranking function, which assess the quality of rewrite along several dimensions including content preservation, hallucination, linguistic variability, and length text change.

To properly test the capability of RewriteLM, we introduce a new benchmark OPENREWRITEEVAL by collecting human-generated text rewrites with natural language instructions. Unlike the previous benchmarks for text rewriting, which mostly had restricted types (Reif et al. 2021; Mallinson et al. 2022) and performed within the boundaries of single sentences (Riley et al. 2020; Siddique, Oymak, and Hristidis 2020; Mallinson et al. 2022), our benchmark is designed for research on cross-sentence text rewrite and covers a wide variety of rewriting types expressed through natural language instructions.

We conduct empirical studies to evaluate the model performance on the OPENREWRITEEVAL benchmark. The results show that even current state-of-the-art pretrained LLMs have poor performance on open-ended rewriting tasks. LLMs fine-tuned on general-purpose instruction datasets like Flan-PaLM (Chung et al. 2022) and Alpaca (Taori et al. 2023) have better performance compared with the pretrained foundation models, but still have room for improvement. The proposed RewriteLMs, including Rewrite-PaLM and Rewrite-PaLM 2, both outperform their corresponding foundation models by a significant margin. They also outperform other instruction-tuned LLMs, showcasing the effectiveness of the generated training data. Applying reinforcement learning on top of the supervised tuned Rewrite-PaLM 2 further improves its performance, resulting in a new state-of-the-art model Rewrite-RL$_{\text{r/w}}$-PaLM 2 for text rewriting.

Our main contributions can be summarized as follows:

- A new benchmark, OPENREWRITEEVAL, designed for research on cross-sentence rewrite and covering a wide variety of rewriting types expressed through natural language instructions, such as formality, expansion, conciseness, paraphrasing, tone and style transfer. Unlike previous benchmarks, which were primarily focused on specific rewrite types within the boundaries of single sentences, our benchmark is specifically designed to facilitate cross-sentence rewrites with open-ended natural language instructions. To the best of our knowledge, no such dataset has existed previously.
- New strategies for instruction tuning and reinforcement learning to better align LLMs for cross-sentence rewriting tasks using diverse wording and structures expressed through natural languages including 1) generating rewriting instruction data from Wiki edits and public corpus through instruction generation and chain-of-thought prompting 2) collecting comparison data for reward model training through a new ranking function. We demonstrate that RewriteLM model achieved the state-of-the-art performance in cross-sentence rewriting tasks on OpenRewriteEval.

Related Work

Text Editing. The majority of the research on rewriting currently focuses on a particular set of editing tasks at the sentence level, such as paraphrase (May 2021), style transfer (Tikhonov et al. 2019), spelling correction (Napoles, Sakaguchi, and Tetreault 2017), formalization (Rao and Tetreault 2018), simplification (Xu et al. 2016) and elaboration (Iv et al. 2022). (Faltings et al. 2020) trained an editing model to follow instructions using Wikipedia data. However, their focus was solely on edits limited to a single sentence. PEER (Schick et al. 2022) can follow human-written instructions for updating text in any domain, but is still limited by the edit types available on Wikipedia. Moreover, it was only evaluated on a small set of edit types from a human-defined instruction evaluation benchmark (Dwivedi-Yu et al. 2022).

Instruction Tuning. Instruction tuning has shown to improve model performance and generalization to unseen tasks (Chung et al. 2022; Sanh et al. 2022). InstructGPT (Ouyang et al. 2022) extends instruction tuning further with reinforcement learning with human feedback (RLHF), which heavily relies on human labelers to collect instruction data and model output rankings for training. The focus of these works was primarily on extensively researched tasks and benchmarks, which do not include open-ended text rewriting.

Data Augmentation via LLM. A common data augmentation approach involves utilizing trained LLMs to generate more data, which is subsequently incorporated as training data to enhance the model's performance (He et al. 2019; Xie et al. 2020; Huang et al. 2022). PEER (Schick et al. 2022) leverage LLMs to infill missing data and then use this synthetic data to train other models. Self-Instruct (Wang et al. 2022a; Taori et al. 2023) improves its ability to accurately follow instructions by bootstrapping off its own generated outputs. Our work builds upon similar ideas and leverages the power of LLMs to enhance existing datasets and generate additional synthetic datasets.

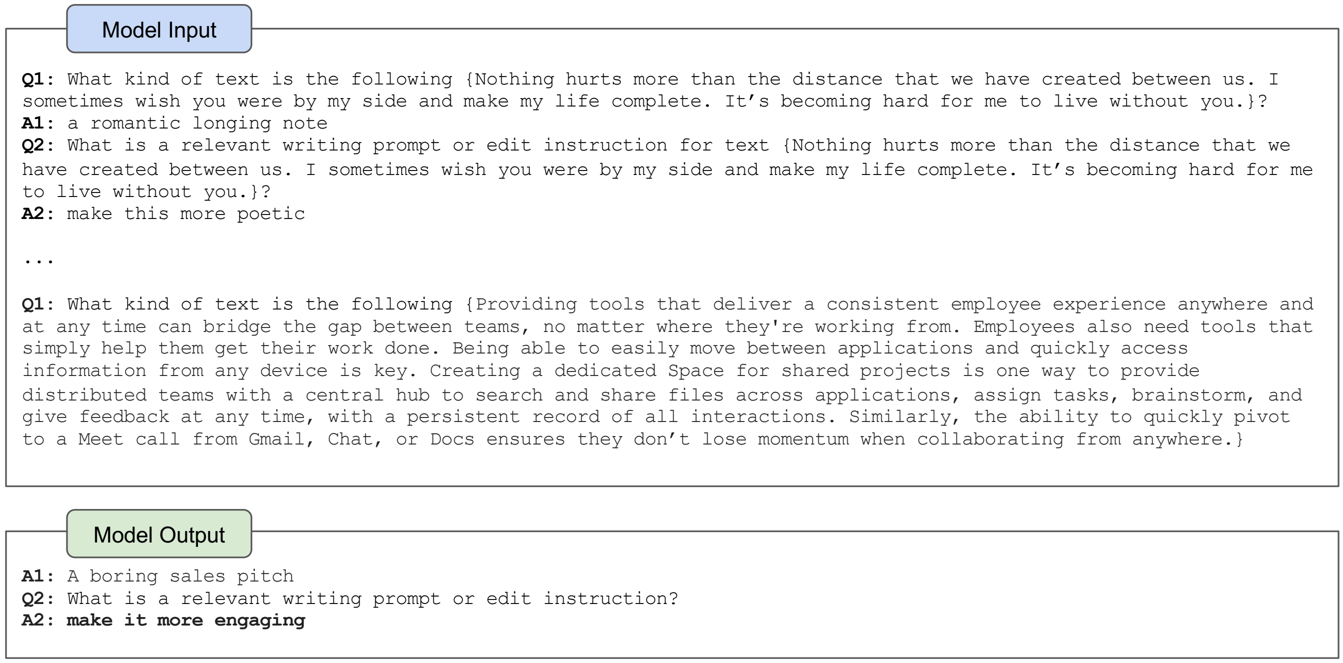
Methods

In this section, we discuss the training data (Section Training Dataset) and the training procedure (Section Modeling) for the proposed RewriteLM models. Table 2 provides a comprehensive overview of the training data's statistics.

Training Dataset

Wiki Instruction Dataset We examine Wiki revisions and extract long-form, high quality edits that contain substantial changes. We also use the associated edit summary of the revision as a proxy for the instructions. We describe edit extraction, edit filtering, and instruction improvement in details:

- **Edit Extraction:** We initiate the instruction tuning data collection process by gathering Wikipedia revision history, where each revision record includes the original text, revision differences, and an edit summary written by the revision author. We extract text block differences between each consecutive snapshots of a Wikipedia article and the associated edit summary, following the approach in


Model Input
Model Output

```
Q1: What kind of text is the following {Nothing hurts more than the distance that we have created between us. I
sometimes wish you were by my side and make my life complete. It’s becoming hard for me to live without you.}?
A1: a romantic longing note
Q2: What is a relevant writing prompt or edit instruction for text {Nothing hurts more than the distance that we
have created between us. I sometimes wish you were by my side and make my life complete. It’s becoming hard for me
to live without you.}?
A2: make this more poetic

...

Q1: What kind of text is the following {Providing tools that deliver a consistent employee experience anywhere and
at any time can bridge the gap between teams, no matter where they're working from. Employees also need tools that
simply help them get their work done. Being able to easily move between applications and quickly access
information from any device is key. Creating a dedicated Space for shared projects is one way to provide
distributed teams with a central hub to search and share files across applications, assign tasks, brainstorm, and
give feedback at any time, with a persistent record of all interactions. Similarly, the ability to quickly pivot
to a Meet call from Gmail, Chat, or Docs ensures they don’t lose momentum when collaborating from anywhere.}
```

```
A1: A boring sales pitch
Q2: What is a relevant writing prompt or edit instruction?
A2: make it more engaging
```

Figure 1: Chain-of-thought (CoT) approach to generating rewrite instructions. The answer to the second question in the output is the generated instruction.

Schick et al. (2022). In the rest of the section, we may use the terms *source text*, *target text* and *comment* to denote the text before revision, the text after revision and the edit summary of a revision record, respectively.

- **Edit Filtering:** In order to create long-form, high-quality edits with substantial changes, we remove revision records that meet any of the following criteria: (i) the edit summary indicates low-quality content of a snapshot, such as containing “revert” or “vandalism” keywords; (ii) the edit summary contains keywords indicating a format-only change (*e.g*.bold-facing or hyperlinks), which is not a focus of this work; (iii) the source text contains two or fewer sentences.
- **Instruction Improvement:** The raw comment may not directly meet our data requirements, which can be empty, contain irrelevant descriptions to the revision, or not describe the editing behavior (*e.g*.only describes the deficiencies of source text). We take the following steps to enhance the quality of the instructions: (i) Extract revision records where the edit summary starts with a verb describing an edit intent (*e.g*.“make the text easier to read”); (ii) Fine-tune PaLM2-XXS to generate comments from `<source>-<target>` text pairs as well as learn to control the length and specificity of the instructions. We use the heuristic that if a comment mentions a word from the edit then it is a detailed instruction. (iii) Generate detailed comments for all `<source>-<target>` pairs using the model trained in the previous steps.

Synthetic Instruction Dataset The Wiki instruction dataset is limited by the available edit types found on Wikipedia. To collect a more diverse and representative instruction dataset, we first use chain-of-thoughts prompting and few-shot prompting to generate instructions, and then generate the target text from a general purpose LLM model:

- **Instruction generation:** By applying a 3-shot chain-of-thought (CoT) prompting method to text inputs from any domain (see Figure 1), we can leverage the knowledge acquired by the PaLM2-L during pre-training. This enables the LLM to produce more diverse instructions beyond Wiki edit types. CoT contains two QA stages: **Text description** (answering “What kind of text is the following”) and **Instruction generation** (answering “What is a relevant writing prompt or edit instruction for text”). The answer to the second question is the generated instruction.
- **Target generation:** Given the source text and the generated instructions, we generate the model outputs with a general purpose instruction tuned LLM (text-bison-001[2]) and filter them in a post-processing step (see Section Heuristic Post-Processing).

Heuristic Post-Processing In order to improve the quality of the instruction datasets, we do the following post-processing: (1) In general, rewriting should preserve the overall meaning of the text, and thus, we employ Natural Language Inference (NLI; See Section Baselines) to detect “hallucinations” from the source to the target text and vice versa. If the “hallucination” is in the target text and fixable using simple heuristic rules, we remove the “hallucination”

[2] https://cloud.google.com/vertex-ai/docs/generative-ai/model-reference/text

	Size	Inst Len	Src Len	Tar Len	Len Ratio	Edit Dist	Edit Ratio	Rouge1
All	24384	6.85	118.86	141.09	1.20	115.44	0.97	60.95
Wiki	18196	7.38	112.17	98.39	0.90	77.69	0.70	64.77
Synthetic	6188	5.30	138.54	266.63	2.10	226.43	1.78	49.72

Table 2: RewriteLM Training Data Statistics: This table includes statistics for the entire training set ("All"), data derived from Wikipedia ("Wiki", Section Wiki Instruction Dataset), and synthetic data generated from large language models ("Synthetic", Section Synthetic Instruction Dataset). Metrics are the number of examples (Size); the average number of words in instructions (Inst Len), source texts (Src Len), and target texts (Tar Len); the average length fraction (Len Ratio) between the target and source texts; the average edit distance (Edit Dist) between source and target; the ratio of edit distance to source text length (Edit Ratio); and the Rouge1 score comparing source and target texts. All measurements are conducted at the word-level.

from the target text and keep the instance. (2) For any other detected "hallucination", we filter the instance. (3) If the difference between the source and target texts is unexpectedly small, we also filter the instance.

Modeling

Supervised Fine-Tuning (SFT). Given a pretrained language model M_{base}, we fine-tune it using the instruction tuning dataset discussed in Section Training Dataset, producing a model M_{SFT}. We employ the decoder-only Transformer architecture for our experiments, details of which are explained in Section Experiments and Results. For both models, the input is formed by concatenating `<instruction>` and `<source>` with a newline, while the output is `<target>`.

Reward Modeling (RM) Firstly, we sample prompt data (instruction and source) from our training dataset, and sample outputs from the pretrained language model M_{base} and finetuned model M_{SFT}.

Secondly, in constrast to InstructGPT, where human labelers are used to rank the outputs, we develop a new approach to rank model outputs without any human effort for collecting preference data for reward model training. We define a new scoring function to measure the quality of the rewrite transformation through several heuristics (see Section Heuristic Post-Processing). For an input output pair (x, t), the quality score is defined as follows:

$$Q(x,t) = \begin{cases} 0, & \begin{array}{l} \text{if EditRatio}(x,t) < a \text{ or} \\ \text{NLI}(x,t) < b \text{ or} \\ \text{NLI}(t,x) < c \text{ or} \\ (I_{\text{shorten}} \& \ \text{LenRatio}(x,t) > d_1) \end{array} \\ 0, & \begin{array}{l} \text{if EditRatio}(x,t) < a \text{ or} \\ \text{NLI}(x,t) < b \text{ or} \\ \text{NLI}(t,x) < c \text{ or} \\ (I_{\text{elaborate}} \& \ \text{LenRatio}(x,t) < d_2) \end{array} \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

, where $a = 1.2$, $b = 0.7$, $c = 0.7$, $d_1 = 0.6$, and $d_2 = 2$. $I_{shorten}$ means a shorten task, and $I_{elaborate}$ means a expanding or elaboration task. These are decided simply based on keyword matches. If a (x, t) pair fails to meet any of the heuristic rules, it is assigned a quality score of 0; otherwise, a score of 1 is given. If the model outputs from the same prompt are all good or are all bad, we will discard the example. If some outputs are good and some are bad, we will select the top-ranked ones (based on probability in top-p or top-k sampling) from good outputs and bad outputs respectively.

Finally, we finetune a pre-trained reward model R_{base} using the comparison data collected above. This is different from InstructGPT (Ouyang et al. 2022), which trains the reward model from scratch after obtaining a supervised tuned model. Since R_{base} is pretrained on general-purpose preference data and not specialized for open-ended rewriting, additional finetuning is crucial.

The reward model, denoted as r_θ, employs a transformer-based architecture with a linear regression output layer. It is trained with t_{good} and t_{bad} which represent the good and bad targets respectively. The training loss function for the reward model is the entropy of the normalized score difference between the good and bad targets.

$$\text{loss}(\theta) = - \mathop{\mathbb{E}}_{(x, t_{\text{good}}, t_{\text{bad}}) \sim D} \log \Big(\sigma \big(r_\theta(x, t_{\text{good}}) - r_\theta(x, t_{\text{bad}}) \big) \Big) \quad (2)$$

Reinforcement Learning. Finally, we further optimize the supervised fine-tuned model M_{SFT} by employing reinforcement learning (Ouyang et al. 2022), guided by the scores provided by the fine-tuned reward model R_{SFT}. This process results in the final model, M_{rewrite}.

Evaluation Framework

OpenRewriteEval — A New Benchmark for Text Rewriting

To facilitate the evaluation of open-ended rewriting, we have curated a new dataset called OpenRewriteEval, which focuses on open instructions, long-form text, and large edits. Each example in the dataset consists of a three-tuple (`<instruction>`, `<source>`, `<target>`).

OpenRewriteEval consists of six datasets $D_{\text{Formality}}$, $D_{\text{Paraphrase}}$, D_{Shorten}, $D_{\text{Elaborate}}$, $D_{\text{MixedWiki}}$ and $D_{\text{MixedOthers}}$. For $D_{\text{Formality}}$, $D_{\text{Paraphrase}}$, and D_{Shorten}, we use a fixed set of instruction. For the rest of the datasets, we asked human annotators to attach appropriate instructions to each source text and then rewrite them accordingly. Table 3 provides information on the size of each task and the average word-level lengths of instructions, source text, and target text. OpenRewriteEval captures how people naturally rewrite, which usually include changes across multiple sentences. This sets

									NLI	
	Size	Inst Len	Src Len	Tar Len	Len Ratio	Edit Dist	Edit Ratio	Rouge1	src-tar	tar-src
All	1629	6.40	132.71	143.53	1.12	90.79	0.71	67.19	0.94	0.95
$D_{\text{Formality}}$	200	5.10	114.73	119.23	1.12	62.51	0.56	68.93	0.87	0.98
$D_{\text{Paraphrase}}$	102	3	211.02	195.97	1	121.2	0.54	68.57	1	1
D_{Shorten}	102	4.49	211.02	165.68	0.8	72.2	0.37	79.26	1	1
$D_{\text{Elaborate}}$	102	8.64	211.02	378.47	2.07	234.33	1.34	56.52	0.92	1
$D_{\text{MixedWiki}}$	606	7.54	103.3	97.57	0.98	65.36	0.64	71.86	0.94	0.92
$D_{\text{MixedOthers}}$	517	6.17	127.8	145.74	1.18	100.89	0.82	60.51	0.95	0.95

Table 3: Statistics of OpenRewriteEval the number of examples (Size); the average number of words in instructions (Inst Len), source texts (Src Len), and target texts (Tar Len); the average length fraction (Len Ratio) between the target and source texts; the average edit distance (Edit Dist) between source and target; the ratio of edit distance to source text length (Edit Ratio); and the Rouge1 score comparing source and target texts for the full set and the subtasks. All are measured at the word-level. NLI (src-tar, tar-src) are the NLI scores between the source text and the gold reference.

us apart from existing benchmarks such as EditEval (Dwivedi-Yu et al. 2022), which are limited to rewrites within single sentences. See Edit ratio (dividing the edit distance by the length of the source text): OpenRewriteEval (0.37-1.34; see Table 3) vs EditEval (0.17-0.59). Appendix B Human Rewrite Guideline provides detailed guidelines for the rewrite annotations.

Automatic Evaluation Metrics

We employ various metrics to evaluate the model's performance including

- **NLI** (Bowman et al. 2015) and **Reversed NLI** (*i.e.*reverse the premise and the hypotheses) score over the source-prediction pair. NLI and Reversed NLI scores illustrate the model prediction's content presentation and factuality quality. We use the off-the-shelf NLI predictor introduced by (Honovich et al. 2022).
- **Edit Distance Raito (Edit Ratio)**. Edit distance (Ristad and Yianilos 1998) measures the word-level textural difference between two pieces of text. We report the relative edit distance between the prediction and source text, *i.e.*dividing the edit distance by the length of the source text. The edit ratio represents the proportion of the source text that has been modified. It is undesirable if the edit distance is small because this indicates the prediction is primarily identical to the source text. Ideally, we expect to see this value to be neither excessively high (indicating the entire content has been changed) nor excessively low (indicating that only minor rewriting occurred thereby diminishing the perceived effectiveness of the system).
- **SARI** (Xu et al. 2016) is an n-gram based metric measures how a close a prediction is relative to the source text and the reference text by rewarding words added, kept, or deleted. SARI computes the arithmetic mean of n-gram F1-scores for each of the three operations.
- **GLEU** (Napoles et al. 2015) measures the precision of the n-grams in the model's prediction that match the reference. It is a variant of BLEU (Papineni et al. 2002). GLEU is customized to penalize only the changed n-grams in the targets, as unmodified words do not necessarily need to be penalized in the rewriting task.
- **Update-ROUGE (Updated-R)** (Iv et al. 2022) measures the recall of n-grams between the model's prediction and the references. It is a modified version of ROUGE (Lin and Hovy 2003). Updated-R specifically computes ROUGE-L on the updated sentences rather than the full text.

When evaluating quality, it is desirable to have a higher value of NLI. Additionally, a higher Edit Ratio within a reasonable range is preferred. However, it's important to note that considering these metrics independently is insufficient. In some cases, predictions with a low edit ratio may still have high NLI scores. Conversely, a large edit ratio can contain hallucinations if the NLI scores are low. Additionally, higher values of SARI, GLEU, and Update-ROUGE indicate that the predictions are more similar to the gold reference text.

Human Evaluation

We conduct human evaluation on randomly selected 80 examples from the OpenRewriteEval dataset with five language experts. The rating use a 3-point Likert scale (0-Bad, 1-Medium, or 2-Good) for the following features: 1) **Instruction Success:** whether the rewrite accurately follows the instruction provided. 2) **Content Preservation:** whether the rewritten text preserves the essential content and meaning of the source text, regardless of its writing style or quality. 3) **Factuality:** Checks the accuracy and truthfulness of the answer's content. 4) **Coherency:** whether the rewritten text is easy to understand, non-ambiguous, and logically coherent when read by itself (without checking against the source text). 5) **Fluency:** Examines the clarity, grammar, and style of the written answer. The detailed rating guideline is in Appendix C Human Rating Guideline.

Experiments and Results

This section provides an overview of our experimental settings, baselines, and result analysis. Detailed information about the hyperparameters can be found in Appendix D Hyper-parameter Setting.

Baselines

We use the following baseline models for quality comparison in the later sections:

		Edit Ratio	NLI (s-p)	NLI (p-s)	SARI	GLEU	Update-R
Pretrained LLMs							
PaLM (Chowdhery et al. 2022)	62B	0.31	0.25	0.11	28.24	0.74	11.99
PaLM 2 (Passos et al. 2023)	S	**1.22**	0.63	0.37	28.62	0.48	8.14
LLaMA (Touvron et al. 2023)	65B	0.71	0.83	0.83	27.98	2.10	21.35
Instruction-Tuned LLMs							
Alpaca (Taori et al. 2023)	13B	0.11	0.90	0.85	36.12	6.81	34.88
Alpaca-PaLM 2	S	0.12	0.9	0.84	38.51	8.31	36.56
Vicuna (Chiang et al. 2023)	13B	0.23	0.89	0.77	39.05	6.84	33.31
Flan-PaLM (Chung et al. 2022)	62B	0.12	0.58	0.42	24.52	1.87	6.23
InsGPT (text davinci 001)	-	0.09	0.66	0.61	27.17	3.72	18.69
ChatGPT (GPT 3.5 Turbo)	-	0.13	0.95	0.87	40.04	8.47	37.78
RewriteLMs							
Rewrite-PaLM	62B	0.14	0.88	0.76	37.02	7.40	36.68
Rewrite-PaLM 2	S	0.25	0.93	0.79	40.92	**9.64**	39.36
Rewrite-RL-PaLM 2	S	0.27	0.94	0.81	**40.97**	9.43	39.36
Rewrite-RL$_{r/w}$-PaLM 2	S	0.29	**0.96**	**0.87**	40.66	**9.64**	**40.10**

Table 4: Model Performance on OPENREWRITEEVAL. Edit distance ratio (Edit Ratio) between the model prediction and the source text; NLI score with source as premise and model prediction as hypothesis (NLI s-p) and vice versa (NLI p-s); SARI, GLEU and Updated-ROUGE (Updated-R) between the gold reference and the model prediction are reported here.

		JFL		TRK	AST	WNC	FRU		WFI	
		SARI	GLEU	SARI	SARI	SARI	SARI	Update-R	SARI	Updated-R
Copy	-	26.7	40.5	26.3	20.7	31.9	29.8	0	33.6	-
Tk (Wang et al. 2022b)	3B	31.8	39	32.8	29.9	31.3	12.6	3.6	1.3	4.5
T0 (Sanh et al. 2022)	3B	42	38.8	34.4	32.3	22.3	14.2	9.6	5.1	16.3
T0++ (Sanh et al. 2022)	11B	34.7	43.2	32.9	28.2	29.3	12.6	3.7	4.4	8.1
PEER-3 (Schick et al. 2022)	3B	55.5	54.3	32.5	30.5	53.3	39.1	30.9	34.4	18.7
PEER-11 (Schick et al. 2022)	11B	55.8	54.3	32.1	29.5	**54.5**	**39.6**	31.4	**34.9**	20.4
OPT (Zhang et al. 2022b)	175B	47.3	47.5	32.6	31.8	31.2	35.9	27.3	26.7	11.2
GPT-3 (Brown et al. 2020)	175B	50.3	51.8	33	30.5	31.7	36	21.5	27.2	10.6
InsGPT (Ouyang et al. 2022)	175B	**61.8**	**59.3**	38.8	38	35.4	36.3	24.7	23.6	16.1
PaLM 2 (Passos et al. 2023)	S	36.07	2.18	34.32	35.92	25.2	24.28	26.39	11.41	20.42
Rewrite-PaLM 2 (Ours)	S	56.95	40.38	40.81	**42.11**	37.11	37.51	53.54	26.55	47.06
Rewrite-RL$_{r/w}$-PaLM 2 (Ours)	S	55	22.89	**40.87**	41.71	37.81	38.56	**53.93**	29.25	**49.53**

Table 5: Model Performance on EditEval (Dwivedi-Yu et al. 2022).

- **PaLM** (Chowdhery et al. 2022) is a large, densely activated transformer-based language model that can generate text in an open-ended fashion.
- **PaLM 2** (Passos et al. 2023), is an advanced language model which surpasses its predecessor PaLM in terms of multilingual and reasoning abilities while being more computationally efficient. It is a Transformer-based model that underwent training using a blend of objectives. In this paper, we employ PaLM 2-S. This "S" size is comparable to LLaMA/Alpaca/Vicuna-13B, which is why we opted to train using it rather than the largest PaLM 2. Note that the specific number of parameters for the PaLM 2 series has not been made public. Instead, the PaLM 2 Tech Report uses T-shirt sizes to represent model sizes, ranging from XXS to L. We follow its notations.
- **LLaMA** (Touvron et al. 2023) is an efficient, open-source foundation language model.
- **Flan-PaLM** (Chung et al. 2022) is fine-tuned on a large variety of tasks and chain-of-thought data using PaLM as the base model.
- **Alpaca** (Taori et al. 2023) is a language model that is fine-tuned from LLaMA using 52,000 instruction-following demonstrations.
- **Alpaca-PaLM:** We fine-tune the PaLM model on Alpaca instruction-following datasets.
- **Vicuna** (Chiang et al. 2023) is an open-source chatbot trained by fine-tuning LLaMA on user-shared conversations collected from ShareGPT[3].
- **ChatGPT** (gpt-3.5-turbo)[4] and **InsGPT** (text-davinci-001) (Ouyang et al. 2022) are members of the GPT family, developed by OpenAI.

We follow the same zero-shot prompt setting for all the baseline models as Schick et al. (2022). The pre-trained models without any instruction tuning generally exhibit slightly lower performance in following instructions compared to the instruction tuned models under zero-shot scenario.

[3] https://sharegpt.com/

[4] https://openai.com/chatgpt

	Instruction Success	Content Preservation	Factuality	Coherence	Fluency	AVG
Agreement	0.784	0.781	0.769	0.933	0.804	0.814
Human Expert	1.833	1.949	1.985	1.99	1.99	1.949
Alpaca 13B	1.441	1.754	1.934	1.962	1.977	1.814
Alpaca-PaLM 2	1.489	1.719	**1.99**	**2**	**2**	1.839
ChatGPT	1.478	1.775	1.959	1.962	1.975	1.83
Rewrite-PaLM 2	1.641	1.777	1.927	**2**	**2**	1.869
Rewrite-RL$_{\text{r/w}}$-PaLM 2	**1.648**	**1.835**	1.959	1.985	**2**	**1.886**

Table 6: Human Evaluation Results.

Results on OpenRewriteEval Benchmark

The automatic evaluation results for the OpenRewriteEval dataset are presented in Table 4. Rewrite-PaLM and Rewrite-PaLM 2 are supervised fine-tuned versions (as discussed in Section Modeling) based on PaLM, and PaLM 2, respectively. Rewrite-RL-PaLM 2 and Rewrite-RL$_{\text{r/w}}$-PaLM 2 are reinforcement learning models tuned over Rewrite-PaLM 2. The reward model from the former does not use our synthetic preference dataset (as discussed in Section Modeling), whereas the reward model from the latter incorporates it.

As shown in Table 4, our RL tuned model Rewrite-RL$_{\text{r/w}}$-PaLM 2 has the highest scores in almost all the metrics (i.e., NLI scores, SARI, GLEU, and Update-R). This indicates that our model is good at generating outputs faithful to the original input, while other models might generate more "hallucinations". For edit ratio, Rewrite-RL$_{\text{r/w}}$-PaLM 2 has a better score than all the models except PaLM 2. Pre-trained models such as PaLM 2 without any instruction tuning are prone to generating "hallucinations", resulting in a significantly high edit ratio score (*i.e.*1.22). Therefore, our model is good at keeping all the essential content and meaning of the source text, while also being able to rewrite with varied language and structures. Given that Rewrite-RL$_{\text{r/w}}$-PaLM 2 consistently outperforms Rewrite-RL-PaLM 2 across nearly all metrics, this strongly suggests the effectiveness and value of employing synthetic preference data.

Results on EditEval

We also evaluated the performance of our models using the publicly available sentence-level rewrite benchmark EditEval[5] (Dwivedi-Yu et al. 2022). This benchmark comprises various datasets that cover different language tasks. Specifically, JFL (Napoles, Sakaguchi, and Tetreault 2017) focuses on language fluency; TRK (Xu et al. 2016) and AST (Alva-Manchego et al. 2020) target at sentence simplification; WNC (Pryzant et al. 2020) addresses text neutralization; FRU (Iv et al. 2022)) and WFI (Petroni et al. 2022) involve updating information that requires external references.

We only report the results on EditEval datasets that containing more than 100 test examples (see Table 5). The results of LLM baselines and the Copy baseline (which treats the source text as the prediction) are taken directly from the EditEval paper (Dwivedi-Yu et al. 2022). We can observe that the zero-shot performance of Rewrite-PaLM 2 and Rewrite-RL$_{\text{r/w}}$-PaLM 2 is mostly on par with or better than the best baselines (*i.e.*PEER-11 and InsGPT). While our model is specifically designed for long-form text rewriting, it does not sacrifice its capability to handle sentence-level rewriting tasks.

Results on Human Evaluation

The human evaluation results, detailed in Table 6, reveal notable insights. The inter-annotator agreements, quantified using the Fleiss kappa coefficient (Fleiss 1971), underscore the reliability of the evaluations. Notably, Rewrite-PaLM 2 and Rewrite-RW$_{\text{r/w}}$-PaLM 2 demonstrate superior performance over Alpaca, Alpaca-PaLM 2, and ChatGPT in instruction success and content preservation. This alignment with the automatic evaluation metrics underscores the efficacy of these models in adhering to given instructions while maintaining the integrity of the original content. In terms of coherence and fluency, all models, including the human rewrites, scored above 1.96, indicative of their ability to generate clear, unambiguous, and logically coherent outputs. Such high scores suggest that these models' outputs are not only understandable but also align closely with human-level language proficiency. Human expertise still prevails in aspects of instruction success and content preservation, suggesting room for further improvement in model performance to reach human-level proficiency in rewriting tasks.

Conclusion

We introduce a novel benchmark for text rewriting with a focus on cross-sentence rewrites, covering a wide variety of rewriting types expressed through natural language instructions. We present new data generation and training strategies to better teach LLMs to perform rewriting tasks. Our model, RewriteLM, achieves the state-of-the-art results on OpenRewriteEval benchmark.

A OpenRewriteEval Data

Data Sources. The source texts for the $D_{\text{Formality}}$, $D_{\text{Paraphrase}}$, D_{Shorten}, and $D_{\text{Elaborate}}$ categories are from various datasets, including Multi-News (Fabbri et al. 2019), Wikipedia (Guo et al. 2020), PG-19 book (Rae et al. 2019), BIG-PATENT (Sharma, Li, and Wang 2019), BillSum (Kornilova and Eidelman 2019), government reports (Huang et al. 2021),

[5] https://github.com/facebookresearch/EditEval

scientific papers (Cohan et al. 2018), Enron email (Zhang and Tetreault 2019), Reddit (Hamilton, Ying, and Leskovec 2017), IMDB, and Yelp reviews (Maas et al. 2011; Zhang, Zhao, and LeCun 2015). The $D_{\text{MixedWiki}}$ have the source texts from Wikipedia (Guo et al. 2020) and $D_{\text{MixedOthers}}$ contains C4 (Raffel et al. 2020) and human written ones.

B Human Rewrite Guideline

- Raters align source text to the instruction, and then rewrite. If the source text is already met the instruction, for example, "make it more formal", then treat the source text as target text and rewrite less formal (put at source side).
- Ensure (1) the content preservation between source and rewrite; (2) maximum word change; and (3) source and target texts are well aligned with instruction. For example, if the instruction is to "make it more polite", then ensure that the target text is much more polite than the source text.
- Elaborate: the rewrite matches source text's tone and format. Add more relevant information and ideas, but do not make up facts.
- Rephrase: the rewrite matches source text's tone, verbosity, format and max changes to existing words.
- Shorten: the rewrite matches source text's tone and format, trims unnecessary words, simplifies sentences, makes them more concise.
- Informal-to-Formal: Rewrite the given paragraph so that it is more formal in style. To make the text more formal, try to: (1) Replace informal words associated with chatty spoken styles (such as slang and contractions) with more formal vocabulary. (2) Make the text impersonal: avoid referring directly to the author(s) or reader(s), or expressing subjective opinions. (3) Use strictly standard grammatical forms.
- Formal-to-Informal: Rewrite the given paragraph so that it is less formal in style. To make your writing less formal, try to: (1) Replace long or uncommon words with relaxed, everyday terms. You may include contractions (such as changing "cannot" to "can't" if it helps the text flow better. (2) Where appropriate, identify the author and the reader to make the text more relatable. (For example, you might be able to change "It is believed that..." to "I think tha...") (3) If a sentence is very long or stiffly phrased, try breaking it up or rearranging it, even if this doesn't fit the strictest rules of standard grammar.

C Human Rating Guideline

Instruction Success: The ability of the model to adhere to the given instruction is evaluated in this criterion. It is:

- Score 2 (Fully/Mostly Followed): if the model output entirely adheres to the provided instructions, demonstrating a clear understanding and implementation of the given task. Or the output mostly adheres to the instructions, with minor deviations or errors.
- Score 1 (Partially Followed): if the model output shows some adherence to the instructions but deviates significantly in certain aspects or fails to completely implement them, leading to partial fulfillment of the task.
- Score 0 (Not Followed/Mostly Ignored): if the model output largely ignores the provided instructions, making it evident that the task has not been understood or implemented properly. Or despite some slight adherence, the output largely deviates from the intended task as per the instructions.

Content Preservation: The essential content and meaning of the reference is preserved in the rewrite, independent of its style or the quality of the writing. It is:

- Score 2 (Fully/Mostly Preserved): if the rewrite is an excellent representation of the content in the reference, with no omissions. Or the rewrite mostly matches the content of the reference, but one or two elements of the meaning have been lost.
- Score 1 (Half Preserved): if some of the content is present in the rewrite but approximately the same amount is missing.
- Score 0 (Not Preserved/Mostly Lost): if the rewrite is entirely unrelated to the reference. Or despite some slight similarities, the rewrite is hard to recognize as being based on the reference.

Factuality: The rewrite only provides as much information as is present in the reference, without adding anything. It is not misleading and does not make any false statements (unless these were also present in the reference).

- Score 2 (Fully/Mostly faithful): Everything in the rewrite is grounded in the reference. Or the rewrite says something that is not mentioned in the reference or contradicts the reference, but it is not an important addition or it is hard to say whether the statement is true or false.
- Score 1 (Partly faithful): The rewrite adds significant factual statements to the reference. These may be inaccurate or otherwise not based on the reference, but do not entirely undermine the faithfulness of the rewrite as a whole.
- Score 0 (Not/Slightly faithful): The rewrite is mostly wrong, made up, or contradicts what is in the reference text.

Coherence: The rewrite is coherent if, when read by itself (without checking against the reference), it's easy to understand, non-ambiguous, and logically coherent. On the other hand, the rewrite is not coherent if it's difficult to understand what it is trying to say.

- Score 2 (Good): The whole of the rewrite is mostly fluent and easy to read, independent of any reference content. Some specific parts of the rewrite could be more naturally phrased, but overall it is fairly clear and easy to understand.
- Score 1 (Neutral): The rewrite is comprehensible, though not on the first read or only with some effort.
- Score 0 (Bad): The rewrite is very hard to understand, except by checking against the reference.

Fluency: The rewrite is considered fluent if it follows all the rules of its language, including spelling, grammar and punctuation. It reads as though it was written by someone who speaks English as their first language.

- Score 2 (Flawless/Good): The rewrite is grammatically correct, contains no spelling errors, and follows all other linguistic rules. An average English speaker would not see anything that looks "wrong". Or there are just one or two linguistic errors or non-standard formulations, but nothing serious.
- Score 1 (Flawed): The rewrite contains a number of errors of different types, but these errors, even when taken together, do not make the text significantly harder to understand.
- Score 0 (Poor): The rewrite contains a large number of errors, so that some sections of the text are hard to understand, but other parts are more manageable.

D Hyper-parameter Setting

We use 64 Tensor Processing Units (TPU) V3 chips for finetuning. The batch size is 32, and the maximum training step is 5000. We use the Adafactor optimizer (Shazeer and Stern 2018) with a learning rate of 0.003. Both the input and output sequence lengths are set to 1024 tokens. The training dropout rate is 0.1. During inference, the temperature is set to 0.5, and the top-K value is 40.

Acknowledgments

The authors would like to thank Tony Mak, Chang Li, Abhanshu Sharma, Matt Sharifi, Hassan Mansoor, Daniel Kim, Reut Aharony and Nevan Wichers for their insightful discussions and support.

References

Alva-Manchego, F.; Martin, L.; Bordes, A.; Scarton, C.; Sagot, B.; and Specia, L. 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*, 4668–4679. Online: Association for Computational Linguistics.

Bowman, S.; Angeli, G.; Potts, C.; and Manning, C. D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 632–642.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; Stoica, I.; and Xing, E. P. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Cohan, A.; Dernoncourt, F.; Kim, D. S.; Bui, T.; Kim, S.; Chang, W.; and Goharian, N. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *Proceedings of NAACL-HLT*, 615–621.

Dwivedi-Yu, J.; Schick, T.; Jiang, Z.; Lomeli, M.; Lewis, P.; Izacard, G.; Grave, E.; Riedel, S.; and Petroni, F. 2022. EditEval: An Instruction-Based Benchmark for Text Improvements. arXiv.

Fabbri, A. R.; Li, I.; She, T.; Li, S.; and Radev, D. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1074–1084.

Faltings, F.; Galley, M.; Hintz, G.; Brockett, C.; Quirk, C.; Gao, J.; and Dolan, B. 2020. Text editing by command. *arXiv preprint arXiv:2010.12826*.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5): 378.

Guo, M.; Dai, Z.; Vrandečić, D.; and Al-Rfou, R. 2020. Wiki-40b: Multilingual language model dataset. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 2440–2452.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, J.; Gu, J.; Shen, J.; and Ranzato, M. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Honovich, O.; Aharoni, R.; Herzig, J.; Taitelbaum, H.; Kukliansy, D.; Cohen, V.; Scialom, T.; Szpektor, I.; Hassidim, A.; and Matias, Y. 2022. TRUE: Re-evaluating Factual Consistency Evaluation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, 161–175.

Huang, J.; Gu, S. S.; Hou, L.; Wu, Y.; Wang, X.; Yu, H.; and Han, J. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

Huang, L.; Cao, S.; Parulian, N.; Ji, H.; and Wang, L. 2021. Efficient Attentions for Long Document Summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1419–1436.

Iv, R.; Passos, A.; Singh, S.; and Chang, M.-W. 2022. FRUIT: Faithfully reflecting updated information in text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3670–3686.

Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y. J.; Madotto, A.; and Fung, P. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12): 1–38.

Kornilova, A.; and Eidelman, V. 2019. BillSum: A Corpus for Automatic Summarization of US Legislation. *EMNLP-IJCNLP 2019*, 48.

Lin, C.-Y.; and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*, 150–157.

Maas, A.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 142–150.

Mallinson, J.; Adamek, J.; Malmi, E.; and Severyn, A. 2022. Edit5: Semi-autoregressive text-editing with t5 warm-start. *arXiv preprint arXiv:2205.12209*.

May, P. 2021. Machine translated multilingual STS benchmark dataset.

Napoles, C.; Sakaguchi, K.; Post, M.; and Tetreault, J. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 588–593.

Napoles, C.; Sakaguchi, K.; and Tetreault, J. 2017. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 229–234. Valencia, Spain: Association for Computational Linguistics.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Passos, A.; Dai, A.; Richter, B.; Choquette, C.; Sohn, D.; So, D.; Lepikhin, D. D.; Taropa, E.; Ni, E.; Moreira, E.; Mishra, G.; Yu, J.; Clark, J.; Meier-Hellstern, K.; Robinson, K.; Vodrahalli, K.; Omernick, M.; Krikun, M.; Moussalem, M.; Johnson, M.; Du, N.; Firat, O.; Bailey, P.; Anil, R.; Ruder, S.; Shakeri, S.; Qiao, S.; Petrov, S.; Garcia, X.; Huang, Y.; Tay, Y.; Cheng, Y.; Wu, Y.; Xu, Y.; Zhang, Y.; and Nado, Z. 2023. PaLM 2 Technical Report. Technical report, Google Research.

Petroni, F.; Broscheit, S.; Piktus, A.; Lewis, P.; Izacard, G.; Hosseini, L.; Dwivedi-Yu, J.; Lomeli, M.; Schick, T.; Mazaré, P.; et al. 2022. Improving Wikipedia Verifiability with AI.

Pryzant, R.; Martinez, R. D.; Dass, N.; Kurohashi, S.; Jurafsky, D.; and Yang, D. 2020. Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, 480–489.

Qin, C.; Zhang, A.; Zhang, Z.; Chen, J.; Yasunaga, M.; and Yang, D. 2023. Is ChatGPT a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Rae, J. W.; Potapenko, A.; Jayakumar, S. M.; and Lillicrap, T. P. 2019. Compressive Transformers for Long-Range Sequence Modelling. arXiv:1911.05507.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Rao, S.; and Tetreault, J. 2018. Dear Sir or Madam, May I Introduce the GYAFC Dataset: Corpus, Benchmarks and Metrics for Formality Style Transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 129–140.

Reif, E.; Ippolito, D.; Yuan, A.; Coenen, A.; Callison-Burch, C.; and Wei, J. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.

Riley, P.; Constant, N.; Guo, M.; Kumar, G.; Uthus, D.; and Parekh, Z. 2020. TextSETTR: Few-shot text style extraction and tunable targeted restyling. *arXiv preprint arXiv:2010.03802*.

Ristad, E. S.; and Yianilos, P. N. 1998. Learning String-Edit Distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5): 522–532.

Sanh, V.; Webson, A.; Raffel, C.; Bach, S. H.; Sutawika, L.; Alyafeai, Z.; Chaffin, A.; Stiegler, A.; Le Scao, T.; Raja, A.; et al. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *ICLR 2022-Tenth International Conference on Learning Representations*.

Schick, T.; Dwivedi-Yu, J.; Jiang, Z.; Petroni, F.; Lewis, P.; Izacard, G.; You, Q.; Nalmpantis, C.; Grave, E.; and Riedel, S. 2022. PEER: A Collaborative Language Model. *arXiv preprint arXiv:2208.11663*.

Sharma, E.; Li, C.; and Wang, L. 2019. BIGPATENT: A Large-Scale Dataset for Abstractive and Coherent Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2204–2213.

Shazeer, N.; and Stern, M. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *International Conference on Machine Learning*, 4596–4604. PMLR.

Siddique, A.; Oymak, S.; and Hristidis, V. 2020. Unsupervised paraphrasing via deep reinforcement learning. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 1800–1809.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Tikhonov, A.; Shibaev, V.; Nagaev, A.; Nugmanova, A.; and Yamshchikov, I. P. 2019. Style Transfer for Texts: Retrain, Report Errors, Compare with Rewrites. 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)*, 3936–3945.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2022a. Self-Instruct: Aligning Language Model with Self Generated Instructions. *arXiv preprint arXiv:2212.10560*.

Wang, Y.; Mishra, S.; Alipoormolabashi, P.; Kordi, Y.; Mirzaei, A.; Arunkumar, A.; Ashok, A.; Dhanasekaran, A. S.; Naik, A.; Stap, D.; et al. 2022b. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*.

Xie, Q.; Luong, M.-T.; Hovy, E.; and Le, Q. V. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10687–10698.

Xu, W.; Napoles, C.; Pavlick, E.; Chen, Q.; and Callison-Burch, C. 2016. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, 4: 401–415.

Xu, W.; Ritter, A.; Dolan, W. B.; Grishman, R.; and Cherry, C. 2012. Paraphrasing for style. In *Proceedings of COLING 2012*, 2899–2914.

Zhang, H.; Song, H.; Li, S.; Zhou, M.; and Song, D. 2022a. A survey of controllable text generation using transformer-based pre-trained language models. *arXiv preprint arXiv:2201.05337*.

Zhang, R.; and Tetreault, J. 2019. This Email Could Save Your Life: Introducing the Task of Email Subject Line Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 446–456.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022b. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zhang, Y.; Ge, T.; and Sun, X. 2020. Parallel data augmentation for formality style transfer. *arXiv preprint arXiv:2005.07522*.