EDITEVAL: An Instruction-Based Benchmark for Text Improvements

Jane Dwivedi-Yu[1] **Timo Schick**[2] **Zhengbao Jiang**[3]
Maria Lomeli[1] **Patrick Lewis**[4] **Gautier Izacard**[2]
Edouard Grave[5] **Sebastian Riedel**[6] **Fabio Petroni**[7]

[1] Meta, [2] Microsoft, [3] Carnegie Mellon University,
[4] Cohere, [5] Kyutai, [6] Google Deepmind, [7] Samaya AI

janeyu@meta.com

Abstract

Evaluation of text generation to date has primarily focused on content created sequentially, rather than improvements on a piece of text. Writing, however, is naturally an iterative and incremental process that requires expertise in different modular skills such as fixing outdated information or making the writing style more consistent. Even so, comprehensive evaluation of a model's capacity to perform these skills and the ability to edit remains sparse. This work introduces EDITEVAL: An instruction-based, benchmark and evaluation suite that leverages high-quality existing and new datasets in English for the automatic evaluation of editing capabilities, such as making text more cohesive and paraphrasing. We evaluate several pre-trained models, which shows that InstructGPT and PEER on average perform the best, but that most baselines fall below the supervised state-of-the-art, particularly when neutralizing and updating information. Our analysis also shows that commonly used metrics for editing tasks do not always correlate well, and that prompts leading to the strongest performance do not necessarily elicit strong performance across different models. Through the release of this benchmark,[1] and a publicly available leaderboard challenge,[2] we hope to unlock future work on developing models more capable of controllable and iterative editing.

1 Introduction

Large pre-trained language models have shown impressive text generation capabilities for a wide variety of tasks such as question answering, textual entailment, and summarization (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020; Zhang et al., 2022; Chowdhery et al., 2022). However, to date, most work employing language models has focused on generating immutable text in a single pass. This is in stark contrast to the way in which humans develop articles of text, which is naturally an iterative process of small steps, each with a precise purpose (Seow, 2002). This is a crucial process because it allows for analysis of "what's working, what isn't, and what it still needs" and adaptation to these needs along the way (Jackson, 2022). In many cases, a needed change may only become apparent after much of the text is created, such as in the case of a reorganization or fixing inconsistencies or contradictions (Vardi, 2012). In this way, the current paradigm of generating text passages in a single pass can be severely limiting.

Additionally, the current paradigm of continuous left-to-right generation is less controllable and not flexible to human-in-the-loop collaboration and feedback, and this absence of experienced human mediation in the writing process can be highly detrimental to the quality of the final product (Greenberg, 2010). While there are some existing production tools geared towards working with humans to compose articles and emails, such as Grammarly[3], Smart Compose from Google[4] and text predictions from Microsoft[5], a majority focus on sentence completion rather than iteratively improving upon prior text. A more powerful editing

[1] Code and data available at https://github.com/facebookresearch/EditEval

[2] https://eval.ai/web/challenges/challenge-page/1866/overview

[3] https://www.grammarly.com/

[4] https://www.blog.google/products/gmail/subject-write-emails-faster-smart-compose-gmail/

[5] https://insider.office.com/en-us/blog/text-predictions-in-word-outlook

Proceedings of the 28th Conference on Computational Natural Language Learning, pages 69–83
November 15-16, 2024

Figure 1: Examples of instructions for paraphrasing, neutralization, simplification, fluency, and updating information as well as their corresponding expected edits. For illustrative purposes, we ground these examples in the same passage, but examples in EDITEVAL follow the format as described in Section 6.

assistant, however, should not only be capable of providing recommendations for text continuations but also permit non-sequential development of the text (Seow, 2002). Editing can be absolutely critical, for example, if new or missing information or external citations are required to update the text or if a reshuffling/rebalancing of text is needed.

In this work, we alternatively promote iterative text generation and improvement—successive iterations of modular additions and modifications of the text that are relevant to text editing, such as making the text clearer and adding missing information. Many datasets for natural language tasks are actually annotated at the sentence or paragraph level, rather than document or article level, naturally lending well to evaluating iterative edits.

We create EDITEVAL, a benchmark and evaluation suite that leverages high-quality existing and new datasets for the automatic evaluation of editing capabilities. Currently, many of these relevant datasets live in separate packages and are often formatted in uniquely different ways. EDITEVAL downloads each dataset from their most recent version and standardizes these into a single format conducive to evaluation. Additionally, we include popular metrics for each task and a set of human-generated prompts to robustly measure a model's capability in executing the modular task when instructed. Figure 1 shows examples of such prompts and an example of a corresponding edit that we might expect for each prompt. Using these prompts, we evaluate and compare several state-of-the-art language models, such as OPT (Zhang et al., 2022), GPT-3 (Brown et al., 2020), and PEER (Schick et al., 2022). In summary, our contributions are as follows:

1. We identify a set of tasks and datasets relevant to iterative text improvement and provide a pipeline to download and process these datasets into a single format.

2. We open-source a publicly available instruction-based benchmark and leaderboard for automatic evaluation according to metrics commonly used for each editing task.

3. We introduce a new dataset, WAFER-INSERT, for evaluating a model's capability to update information, which is based on the WAFER dataset (Petroni et al., 2022).

4. We provide a comparison of various state-of-the-art baselines evaluated on EDITEVAL at the dataset and prompt level.

2 Related Work

Several multitask evaluation benchmarks have been open-sourced to the community to support progress in natural language understanding including GLUE (Wang et al., 2018), SuperGLUE (Wang

et al., 2019), decaNLP (McCann et al., 2018), and GEM (Gehrmann et al., 2021). These datasets, however, focus on a broad set of tasks in NLP (e.g., question answering, reading comprehension, and textual entailment). While all of these tasks are critical to natural language understanding, EDITEVAL focuses on curating a benchmark for measuring a model's capability to improve and edit text.

There are several datasets which focus on iterative text revisions in the domain of Wikipedia (Yang et al., 2017; Anthonio et al., 2020), academic essays (Zhang et al., 2017), and news articles (Spangher et al., 2022). These works, however, focus on one particular domain and in some cases, a particular style like argumentative writing (Zhang et al., 2017). EDITEVAL, on the other hand, includes examples from multiple domains: Wikipedia, Wikinews, news articles, and arXiv. ITERATER (Du et al., 2022) is perhaps closet to EDITEVAL in that it provides iterative tasks from multiple domains, but it has a limited number of such tasks: fluency, coherence, clarity, style, and meaning-changed. Because this is a great starting point, we have included ITERATER in EDITEVAL, and we additionally develop prompts for these tasks since ITERATER is not instruction-based. Moreover, unlike ITERATER, EDITEVAL includes novel datasets for tasks such as updating text using new information and neutralizing the text, which are core components of editing a factually-correct and unbiased article.

3 The EDITEVAL Benchmark

EDITEVAL is an instruction-based benchmark for iterative text generation/modification. EDITEVAL sources existing high-quality datasets—most with human annotations—containing tasks relevant to editing. These datasets are combined into a unified evaluation tool and can be evaluated with any metric provided in EDITEVAL. A task here refers to a type of edit (e.g., simplification), and the specific task dictates which set of prompts to be used (e.g., simplify this text), the full set of which is enumerated in Appendix B.

We consider seven editing tasks in EDITEVAL. The corresponding datasets for each task included in EDITEVAL are enumerated in Table 1, along with the size of the dataset in EDITEVAL. For ease of evaluation, we define a consistent format for all datasets in the EDITEVAL benchmark. Each dataset of every task has five core fields: ID, input text, gold edits, task type, and reference documents. The input text is the original text before revision, and the gold edits are the target edits for that specific task type. Lastly, the reference documents provide textual information from external articles or documents that are relevant to the task. The task that requires reference documents is updating, and otherwise, the reference documents field is empty.

Table 1: Tasks, datasets, abbreviations used, and corresponding test size in EDITEVAL. The task type dictates which set of instructions are used. These are enumerated in Section B.

Task	Dataset	Abbrev.	Size
Clarity	ITERATER	ITR-L	1,595
Coherence	ITERATER	ITR-O	351
Fluency	ITERATER	ITR-F	942
Fluency	JFLEG	JFL	1503
Simplification	ASSET	AST	2,359
Simplification	TurkCorpus	TRK	2,359
Paraphrasing	STS Benchmark	STS	419
Neutralization	WNC	WNC	1,000
Updating	FRUIT	FRU	914
Updating	WAFER-INSERT	WFI	4,565

The datasets in EDITEVAL were selected if they test a capability relevant to the art of editing and contain human-annotated gold edits, if possible. We also endeavored to include datasets that are broadly used by the community. The datasets in EDITEVAL are by no means exhaustive, but the EDITEVAL framework is flexible such that it can easily extend to new datasets and metrics in future versions.

3.1 Fluency, Clarity, and Coherence

In this section, we describe the two datasets that compose this set of tasks: Fluency (fixing grammatical or spelling errors), clarity (making the text clearer), and coherence (making the text more cohesive).

JFLEG JHU FLuency-Extended GUG (Napoles et al., 2017) focuses only on fluency. JFLEG is based on the GUG (Grammatical vs Un-Grammatical) dataset (Heilman et al., 2014), which is a dataset of sentences originally annotated for how grammatical the sentence is on a scale of 1 to 4. JFLEG builds upon the ungrammatical sentences in GUG and annotates each sentence with four corresponding corrected versions.

ITERATER This dataset introduced by Du et al. (2022) contains both automatically-mined and human-annotated edits at the sentence and

document-level. For our benchmark, we only utilize the sentence-level examples with human annotations. Additionally, ITERATER has labels for the intent—the type of edit that produces the targets, which can be one of six classes: Fluency, coherence, clarity, style (conveying the writer's writing preferences), meaning-changed (updating or adding new information), and other (none of the others). We included all classes except style, meaning-changed, and other. We excluded style and other because these tasks had roughly 100 or less test examples, and the definitions were comparatively under-specified. We excluded meaning-changed because the task does not use reference documents for updating. This dataset is the only one in EDITEVAL that encompasses multiple tasks, and we refer to each respective subset using the abbreviations ITR-F (fluency), ITR-L (clarity), and ITR-O (coherence).

3.2 Paraphrasing

STSB For paraphrasing, we use the STS benchmark from SemEval-2018 (Cer et al., 2017), which comprises English datasets used in the STS tasks of SemEval between 2012 and 2017. The selection of datasets includes text from image captions, news headlines and user forums. Each example contains an original sentence, a target sentence, and a similarity score indicating whether the target is a paraphrase of the original. This dataset is used for classification or regression, but for EditEval, we utilize all instances that we are confident are paraphrases, i.e., have the max similarity score of 5, as targets for generation evaluation. While other datasets such as ParaSCI (Dong et al., 2021) exist for paraphrase generation, these are automatically curated rather than human annotated, and EDITEVAL strives to utilize human-annotated datasets where possible.

3.3 Simplification

Simplification can be considered a very similar task to paraphrasing with the additional constraint that the output must be simpler than the input. The datasets we utilize for simplification are TurkCorpus (Xu et al., 2016) and ASSET (Alva-Manchego et al., 2020).

TurkCorpus This dataset, like ASSET, builds upon the Parallel Wikipedia Simplification (PWKP) (Zhu et al., 2010). The PWKP dataset uses the Simple English Wikipedia and Standard English Wikipedia in parallel to create original-simplification pairs automatically. However, several works found PWKP to have a large proportion of targets that are not simplified or only partially aligned with the input (Xu et al., 2015; Amancio and Specia, 2014; Hwang et al., 2015; Štajner et al., 2015), leading to the creation of a human-annotated corpus, TurkCorpus. TurkCorpus was manually created with eight reference simplifications for each original sentence in PWKP, but only used simplifications that are possible without deleting content or splitting sentences.

ASSET Because TurkCorpus encompassed only specific kinds of simplifications, this led to the creation of ASSET, which provides manually-produced simplifications through a much broader set of transformations. We include both in EDITEVAL, for the sake of comprehensiveness.

3.4 Neutralization

The task of neutralization refers to making the text more neutral. For example, in the sentence "Obama was an excellent president who served two terms from 2008 to 2016" the term *excellent* violates Wikipedia's neutral point of view (POV) policy[6]. For information-intensive content like Wikipedia and news articles in particular, reducing bias is crucial because bias can be the single largest source of distrust in the media (Jones, 2019).

WNC We use the Wiki Neutrality Corpus (Pryzant et al., 2020), a collection of original and de-biased sentence pairs mined from Wikipedia edits by carefully filtering based on the editor's comments. While ideally we would like to include a human-annotated dataset, to our knowledge there does not exist a dataset for de-biasing article content at the sentence level.

3.5 Updating

In this section we describe the task of updating information which requires *references*, text from external sources that are relevant to the particular task. Because of token-length restrictions, each external article is chunked into texts of fixed length. We limit the scope of the task to three chunks, and we refer to these selected chunks as our *reference documents*. These references documents are represented in the edits by their index in the reference documents field (e.g., the first would be demarcated

[6]https://en.wikipedia.org/wiki/Wikipedia:Neutral point of view

as [0]), and we discuss below how these reference documents were selected.

WAFER-INSERT The first dataset for updating information that we use is the WAFER dataset (Petroni et al., 2022), which is a dataset collected from Wikipedia inline citations. Each instance of the original WAFER dataset contains a claim, the text surrounding the claim, and a set of external references, where the task is to choose one of the references to be cited after the claim. While the original intention of WAFER was to measure a system's capability to choose the correct citation, EDITEVAL utilizes WAFER for the task of inserting new information using content from the reference documents. The examples in the original WAFER dataset contains an input text and a reference document, where a sentence (referred to as the claim) of the input text is factually supported by the reference. We create WAFER-INSERT, which differs from WAFER in that the claim is deleted from the input. The goal here is to derive the original claim from the references and insert it back into the text at the appropriate location. For the reference documents, we select the top three chunks from the inline citation chunks that have the highest scores, using results from the verification engine introduced in Petroni et al. (2022).

FRUIT In addition to WAFER-INSERT, we include the FRUIT dataset (Logan IV et al., 2021), a dataset collected by comparing two snapshots of a Wikipedia article where one contains updated or new information. The reference documents were identified by searching for other Wikipedia articles that provide evidence to support the update. However, because there is no certainty that the identified evidentiary articles support the claim, the authors of FRUIT created a gold set by employing human annotation to filter out any new claims that are unsupported. We include this gold set in EDITEVAL, and only include reference documents if they actually appear in the output. Unlike WAFER-INSERT, the target edit contains not only the updated information but also the citation. For EDITEVAL, this is for verification purposes only, and the citation is removed when computing the metrics.

4 Metrics

The metrics we included in EDITEVAL are ones that are (1) shown to have significant correlation with human judgement for a task in EDITEVAL and (2) commonly used to benchmark one of the datasets in EDITEVAL. Below, we discuss some of the main metrics. Appendix C describes these and additional metrics in greater detail.

- **EM** (exact match) is the percentage of examples for which the performed edit exactly matches any of the targets. **EM-diff** is a variant computed at the diff level.
- **SARI** Xu et al. (2016) is an n-gram based metric that averages match scores for three operations: adding, deleting, and keeping words.
- **LENS** (Maddela et al., 2022) is a recently proposed model-based text simplification metric that uses an adaptive ranking loss.
- **GLEU** (Napoles et al., 2015) is a variant of BLEU frequently used for grammatical error correction (Grundkiewicz et al., 2019; Yuan and Briscoe, 2016; Chollampatt and Ng, 2018), where penalties are incurred only when words are changed in the reference but not in the output.
- **ROUGE** (Lin, 2004) is metric that measures n-gram overlap. **UpdateROUGE** (Logan IV et al., 2021), a simple modification of ROUGE, computes ROUGE only on the updated sentences rather than the full text.
- **BERTScore** (Zhang et al., 2019a) which is based on using the cosine similarity between the BERT embeddings of the candidate and reference.

5 Baselines

For each baseline, we use greedy decoding, and we do not perform any task-specific fine-tuning or in-context learning. We evaluate on EDITEVAL using the following baselines:

- **GPT-3** (Brown et al., 2020) is a 175B parameter pretrained decoder-only model. We evaluate GPT-3 through OpenAI's API.[7]
- **InstructGPT** (Ouyang et al., 2022) is a variant of GPT-3 that was instruction-tuned. We evaluate the *text-davinci-001* version described in (Ouyang et al., 2022) since, at the time of writing, details about the training process for *text-davinci-002* were not publicly available.

[7] https://beta.openai.com/

- **OPT** (Zhang et al., 2022) is an open-source replica of GPT-3. Like GPT-3, it is not fine-tuned on any labeled data.

- **T0** (Sanh et al., 2022) is a pretrained encoder-decoder model, which has demonstrated better performance than GPT-3 on several tasks despite being much smaller.

- **T0++** (Sanh et al., 2022) is similar to T0, but trained on a few additional datasets from SuperGLUE (Wang et al., 2019).

- **Tk-Instruct** (Wang et al., 2022) is similar to T0 and T0++ but instead fine-tuned on their dataset, Natural Instructions v2, a collection of instructions for more than 1,600 tasks, including grammatical error correction and text simplification.

- **PEER** (Schick et al., 2022) is a collaborative language model initialized from the *LM Adapt* variant of T5, and further fine-tuned on edit histories from Wikipedia. We use the 3B and 11B PEER models that were shown to perform the best in Schick et al. (2022).

6 Formatting

We evaluate these baselines on their general capability to accomplish each task when prompted in natural language in a zero-shot fashion. Because there are a diverse set of ways in which to instruct for each task, we manually construct a set of 3–11 prompts in order to more robustly evaluate performance. For each task prompt t and input i, the model is given a formatted input following the template: `Task:` t`\nInput:` i`\nOutput:` with an additional field for references, should they be required. Figure 2 shows an example of an input including references. For tasks without references, we exclude this field. Some slight modification to this template were made. For example, Tk-Instruct expects the prompt to be prefixed by the string "Definition:" rather than "Task:"). For preprocessing, we used the Natural Language Toolkit (NLTK) package (Bird et al., 2009) for tokenizing the text.

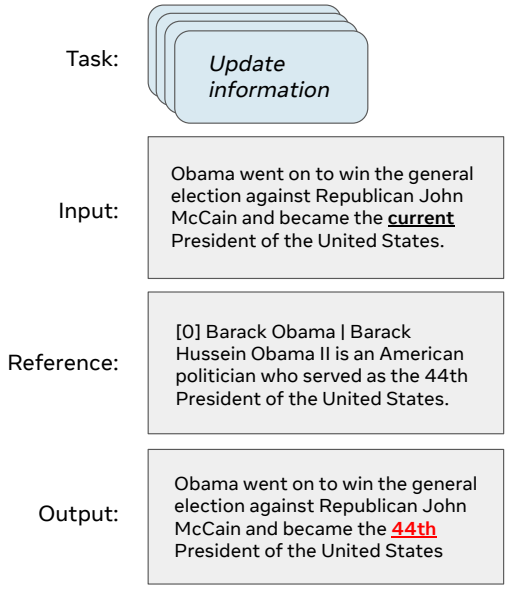


Figure 2: Example of inputs formatted when evaluating the baseline models. Each input is evaluated with a set of prompts that are determined by the task type.

7 Results

We summarize results in Table 2 with the aforementioned baselines averaged over all datasets and the breakdown for each dataset in Table 3. To visualize the variance, we show boxplots for each dataset and model in Figure 3. We discuss these observations in more detail below.

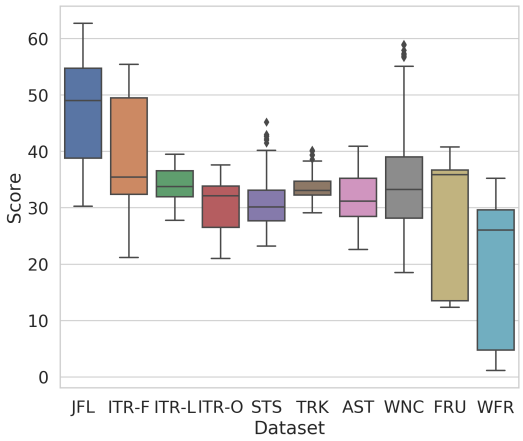
InstructGPT and PEER perform the best overall. In Table 2, we show the mean SARI scores for each model averaged across all tasks using the average, maximum, and minimum scores across prompts. When using the average and minimum across prompts (third and fifth column, respectively) we see that InstructGPT performs the best overall, but when using the maximum score across prompts (fourth column), PEER-11 performs the best. Table 3 enumerates the breakdown of the third column according to each dataset. In general, we see that InstructGPT achieves the highest scores with the exception of the updating and neutralization datasets, as well as ITR-F and ITR-L. For these datasets, the PEER models clearly outperform InstructGPT by a large margin, despite being nearly 60× smaller than InstructGPT and GPT-3. The substantially smaller models (T0, T0++, and Tk-Instruct) struggle the most overall, even falling behind the copy baseline at times, except on ITR-L where Tk-Instruct performs the best.

Model	Params	Avg.	Max	Min	CV
Tk	3B	28.2	30.1	26.1	4.65
T0	3B	26.6	29.3	24.5	6.03
T0++	11B	28.4	30.3	26.7	5.13
PEER-3	3B	38.8	41.8	35.0	6.36
PEER-11	11B	39.1	**42.1**	35.6	5.75
OPT	175B	32.8	36.4	29.0	6.70
GPT-3	175B	32.8	35.8	29.4	6.74
InstructGPT	175B	**39.6**	41.3	**37.4**	**3.60**

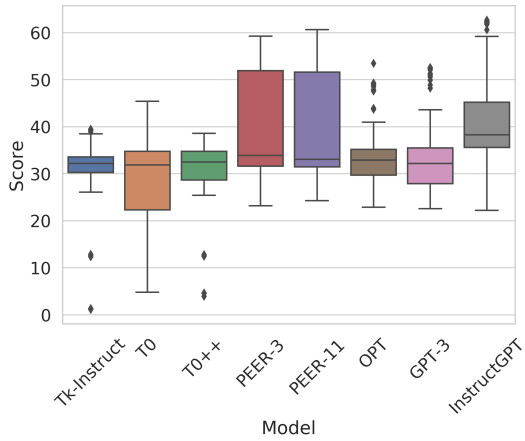
Table 2: Mean SARI scores (other metrics shown in Table C2) all tasks using the average (Avg.), the maximum (Max), and the minimum (Min) across prompts. The coefficient of variance (CV), computed as the standard deviation across prompts normalized by the average, is shown in the final column. Best values are in bold. When using averages across prompts and using the minimum, InstructGPT performs the best, but PEER performs the best when using the maximum across prompts.

Most baselines lag substantially behind the supervised SOTA, especially in the task of updating and neutralization. We show the supervised state-of-the-art results in the final row of Table 3, which in almost all cases surpasses the performance of the best baseline. The gap is largest for the tasks of neutralization and updating (34–50% decrease from the supervised SOTA to the best baseline scores), whereas for other tasks, this decrease is only within 5–14%. It is conceivable that the difficulty with these two tasks is a consequence of the comparatively fewer datasets and research devoted to them compared to that of the more mainstream NLP tasks, such as text simplification.

The most challenging tasks do not necessarily have the highest variance across models. In observing Figure 3a, we see that the tasks which have the largest variance across models (assessed using the interquartile range or IQR) are fluency and updating information. This is despite the fact that the fluency datasets are arguably easier (i.e., many of the models come close to the supervised scores) than the updating datasets, exemplifying that difficulty and robustness can be independent axes. JFLEG also appears to be easier than ITR-F (average SARI of 45.1 versus 38.2), which is understandable since JFLEG sources from the TOEFL exam (primarily simpler and conversational sentences), while ITERATER sources technical sentences from Wikipedia, ArXiv, and Wikinews. Likewise, TurkCorpus seems on average to be slightly easier than ASSET, which is expected since it includes more diverse simplifications than TurkCorpus.



(a) Scores for each dataset averaged across models. Datasets which have the largest variance amongst the baselines are not necessarily harder tasks.



(b) Scores for each model averaged across datasets. PEER has the largest range in performance across datasets.

Figure 3: Boxplot of SARI scores for each dataset (a) and model (b).

PEER has the highest total variance, but OPT and GPT-3 are less robust to different prompts. From Figure 3, we observe that the PEER models have the largest variance in performance overall (as measured by the larger IQR). If we compute the standard deviation across prompts and normalize by the mean (CV in Table 2), however, GPT-3 and OPT, have the highest average across datasets (6.74% and 6.70%, respectively), whereas for the 3B and 11B PEER models, these values are smaller (6.36% and 5.75%). This could be a consequence of the fact that GPT-3 and OPT are not instruction-tuned, whereas the remaining baselines are.

Optimizing prompts according to maximum performance and according to robustness to different models can be orthogonal objectives. Ideally, we would like to create prompts that achieve the highest performance using the best baseline,

Model	*Fluency* JFL		*Clarity* ITR-L	*Coherence* ITR-O	*Para.* STS	*Simplification* TRK	AST	*Neutral.* WNC	*Updating* FRU	WFI
Copy	26.7 / 40.5	32.3 / 86.0	29.5 / 62.9	31.3 / 77.2	21.1	26.3	20.7	31.9 / 0.0	29.8 / 0.0	33.6 / –
Tk	31.8 / 39.0	32.4 / 61.6	**38.4 / 58.4**	33.8 / **70.4**	30.2	32.8	29.9	31.3 / 0.4	12.6 / 3.6	1.3 / 4.5
T0	42.0 / 38.8	24.6 / 34.9	32.6 / 30.2	22.2 / 21.6	34.3	34.4	32.3	22.3 / 0.0	14.2 / 9.6	5.1 / 16.3
T0++	34.7 / 43.2	35.3 / 75.8	37.6 / 56.5	32.7 / 59.9	28.4	32.9	28.2	29.3 / 0.3	12.6 / 3.7	4.4 / 8.1
PEER-3	55.5 / 54.3	51.4 / 84.3	32.1 / 47.1	32.1 / 59.8	28.6	32.5	30.5	53.3 / 21.6	39.1 / 30.9	34.4 / 18.7
PEER-11	55.8 / 54.3	**52.1 / 85.2**	32.5 / 51.3	32.7 / 62.7	28.2	32.1	29.5	**54.5 / 22.8**	**39.6 / 31.4**	**34.9 / 20.4**
OPT	47.3 / 47.5	34.7 / 70.6	31.5 / 31.5	27.6 / 36.1	29.1	32.6	31.8	31.2 / 0.4	35.9 / 27.3	26.7 / 11.2
GPT-3	50.3 / 51.8	32.1 / 56.7	33.5 / 39.7	26.9 / 36.1	27.2	33.0	30.5	31.7 / 0.6	36.0 / 21.5	27.2 / 10.6
InsGPT	**61.8 / 59.3**	48.8 / 82.7	35.1 / 48.4	**35.9** / 60.2	**42.5**	**38.8**	**38.0**	35.4 / 2.2	36.3 / 24.7	23.6 / 16.1
SotA	*– / 62.4*	*37.2 / –*	*46.2 / –*	*38.3 / –*	–	*34.4*	*37.2*	*– / 45.8*	*– / 47.4*	*– / –*

Table 3: Results for all datasets, averaged across prompts (max and min results in Table C2). The best results for each dataset are shown in bold. Tk-Instruct and InstructGPT are shorthanded as Tk and InsGPT, respectively. The first numbers for each task are SARI scores; additional metrics are GLEU for fluency, clarity, and coherence, EM for neutralization, Update-R1 for updating. Supervised scores are from Ge et al. (2018) (JFLEG), Du et al. (2022) (ITERATER), Martin et al. (2020) (TurkCorpus and ASSET), Pryzant et al. (2020) (WNC), and Logan IV et al. (2021) (FRUIT), respectively.

but also perform reliably well for any model. In assessing variance from Figure 4, we see that certain prompts stand out as less robust to different models relative to others. For example, for neutralization, Prompts #1, 2, and 7 are less robust likely because they use uncommon language such as "Remove points of views" or "Neutralize this text". Some of the prompts which are less robust for simplification (Prompts #4, 7) and paraphrasing (Prompts #4, 6) are sometimes ones that are less specific such as "Rewrite this text" versus "Rewrite this with different wording"—in the case of the former, an empirical assessment shows that the models seem to more often copy the original text and make fewer modifications. Unfortunately, choosing prompts that achieve the maximum score does not always entail prompts which are the most robust—Prompt #5 for clarity achieves the maximum but has the largest variance in performance or IQR. Some of the tasks exhibit a great degree of outlier behavior (coherence, paraphrasing, or neutralization), which is either due to T0 performing exceedingly low or InstructGPT/PEER performing exceedingly well. Other tasks such as fluency and updating seem to have prompts with a similar range of variance.

Commonly used metrics are not always well-correlated. We measure the Pearson correlation between each pair of metrics using evaluation scores for all baselines, which is shown in Figure 5, and find that many of the commonly used metrics do not always correlate well with each other, a finding echoed by prior works (Choshen and Abend, 2018; Alva-Manchego et al., 2021), which focuses on the task of grammatical error correction. We exclude PEER in this analysis since it shows exceedingly strong performance in some cases, and we exclude the updating datasets since they are of a very different nature from the other datasets. We find that while families of variants like BLEU and iBLEU as well as ROUGE and UpdateROUGE show strong correlation within each respective set (> 0.97), the two sets are inversely correlated with one another (-0.29 to -0.1). ROUGE actually appears to be the metric that most conflicts with all other metrics, whereas GLEU seems to be the metric that is most in harmony with the rest (0.41–0.76). Though SARI is not correlated with ROUGE, it is the metric which shows the strongest correlation with EM-Diff (0.83) and UpdateROUGE (0.7).

8 Discussion

We present EDITEVAL, a benchmark composed of handcrafted, task-specific instructions for several editing datasets across multiple domains. EDITEVAL is a means of evaluating models for these tasks according to multiple popular metrics, all within a single, unified tool. We show that while models such as InstructGPT have impressive performance, in general the baselines lag behind the supervised state-of-the-art, particularly for the task of updating and neutralization. Our analysis of metrics and prompts shows that several popular metrics are not well-correlated, even conflicting at times, and that small changes in the wording of a prompt can lead to substantial changes in performance and robustness to different models. This suggests further

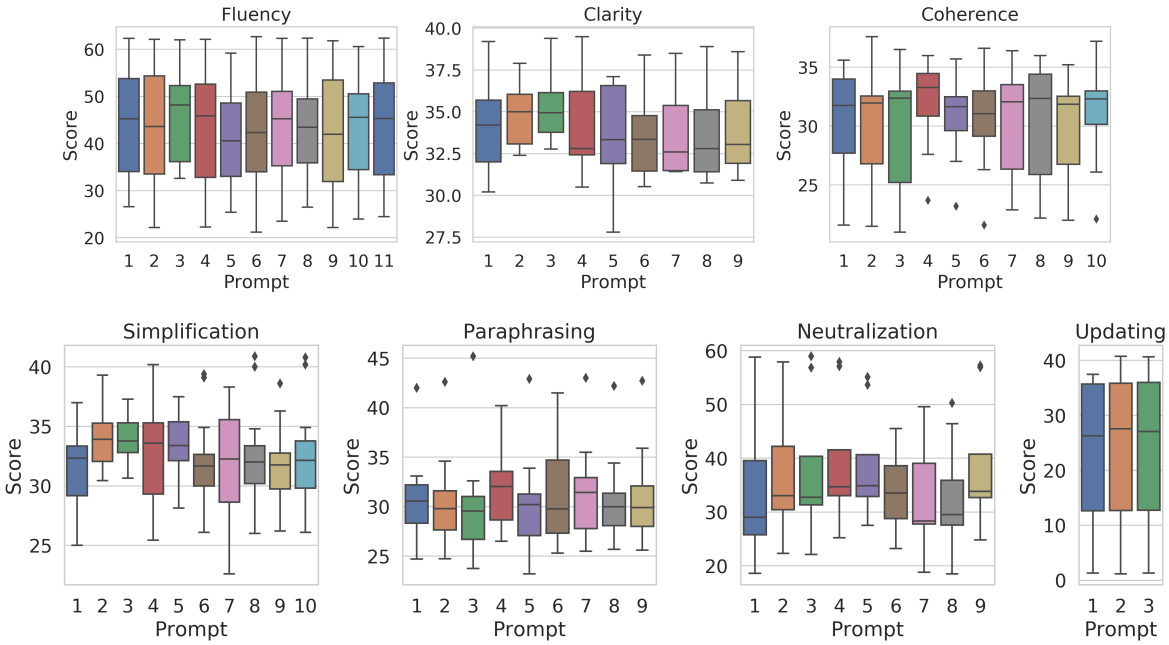


Figure 4: Boxplot of SARI scores for each prompt averaged across models. The prompts which achieve the maximum scores for each dataset (Table C2), are Prompts #6 and 11 (fluency), 4 (clarity), 2 (coherence), 8 and 10 (simplification), 3 (paraphrasing), 2 (neutralization) and 2 and 1 (updating). Certain prompts evoke more variation across models due to factors such as using less frequently used language or being too unspecific.

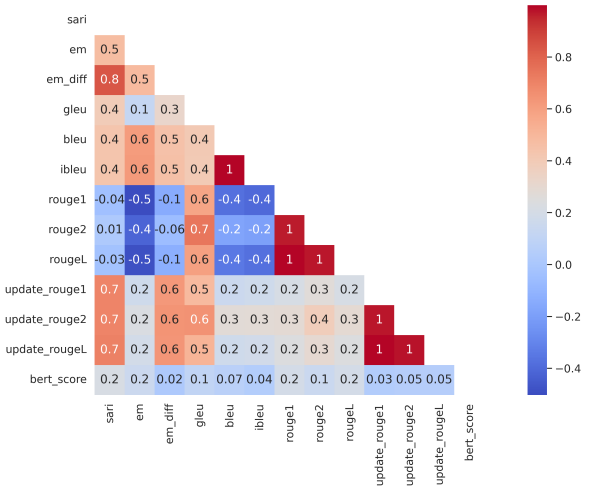


Figure 5: Pearson correlation between metrics using data for all datasets except WAFER and FRUIT and all baselines except PEER. Different families of metrics can have low correlation and even conflict, at times.

work is needed to develop models comprehensively capable of executing editing tasks in addition to developing a standardized way of measuring editing capabilities and systematically selecting prompts. In releasing this work, we hope to bolster work in which language models are utilized for text generation that is iterative, more controllable, collaborative, and capable of revising and correcting text.

References

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.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. Easse: Easier automatic sentence simplification evaluation. *arXiv preprint arXiv:1908.04567*.

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.

Marcelo Adriano Amancio and Lucia Specia. 2014. An analysis of crowdsourced text simplifications. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 123–130.

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikihowtoimprove: A resource and analyses on edits in instructional texts. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5721–5729.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text*

with the natural language toolkit. " O'Reilly Media, Inc.".

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Leshem Choshen and Omri Abend. 2018. Automatic metric validation for grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1372–1382.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

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

Qingxiu Dong, Xiaojun Wan, and Yue Cao. 2021. Parasci: A large scientific paraphrase dataset for longer paraphrase generation. *arXiv preprint arXiv:2101.08382*.

Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang. 2022. Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*.

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

Susan Greenberg. 2010. When the editor disappears, does editing disappear? *Convergence*, 16(1):7–21.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263.

Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel R. Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *ACL (2)*, pages 174–180.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from standard wikipedia to simple wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 211–217.

Annie Jackson. 2022. The advantage of an iterative writing process for novels and short stories.

David A Jones. 2019. An online experimental platform to assess trust in the media," webpage, july 18, 2018b. *As of March*, 18.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-level fluency evaluation: references help, but can be spared! *arXiv preprint arXiv:1809.08731*.

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

Robert L Logan IV, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. 2021. Fruit: Faithfully reflecting updated information in text. *arXiv preprint arXiv:2112.08634*.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2022. Lens: A learnable evaluation metric for text simplification. *ArXiv*, abs/2212.09739.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. *arXiv preprint arXiv:1909.01187*.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2020. Muss: multilingual unsupervised sentence simplification by mining paraphrases. *arXiv preprint arXiv:2005.00352*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 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)*, pages 588–593.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 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*, pages 229–234.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 85–91.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-reward reinforced summarization with saliency and entailment. *arXiv preprint arXiv:1804.06451*.

Fabio Petroni, Samuel Broscheit, Aleksandra Piktus, Patrick Lewis, Gautier Izacard, Lucas Hosseini, Jane Dwivedi-Yu, Maria Lomeli, Timo Schick, Pierre-Emmanuel Mazaré, Armand Joulin, Edouard Grave, and Sebastian Riedel. 2022. Improving wikipedia verifiability with ai.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 480–489.

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

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

Miriam Redi, Martin Gerlach, Isaac Johnson, Jonathan Morgan, and Leila Zia. 2020. A taxonomy of knowledge gaps for wikimedia projects (second draft). *arXiv preprint arXiv:2008.12314*.

Pengjie Ren, Furu Wei, Zhumin Chen, Jun Ma, and Ming Zhou. 2016. A redundancy-aware sentence regression framework for extractive summarization. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 33–43.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2022. Peer: A collaborative language model.

Anthony Seow. 2002. The writing process and process writing. *Methodology in language teaching: An anthology of current practice*, 315:320.

Alexander Spangher, Xiang Ren, Jonathan May, and Nanyun Peng. 2022. Newsedits: A news article revision dataset and a novel document-level reasoning challenge. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 127–157.

Sanja Štajner, Hannah Béchara, and Horacio Saggion. 2015. A deeper exploration of the standard pb-smt approach to text simplification and its evaluation. 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)*, pages 823–828.

Inigo Jauregi Unanue, Jacob Parnell, and Massimo Piccardi. 2021. Berttune: Fine-tuning neural machine translation with bertscore. *arXiv preprint arXiv:2106.02208*.

Lucy Vanderwende, Hisami Suzuki, and Chris Brockett. 2006. Microsoft research at duc 2006: task-focused summarization with sentence simplification and lexical expansion. In *Proceedings of the Document Understanding Conference, DUC-2006, New York, USA*.

Iris Vardi. 2012. The impact of iterative writing and feedback on the characteristics of tertiary students' written texts. *Teaching in Higher Education*, 17(2):167–179.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022. Benchmarking generalization via in-context instructions on 1,600+ language tasks. *arXiv preprint arXiv:2204.07705*.

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.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying semantic edit intentions from revisions in wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2000–2010.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386.

Fan Zhang, Homa B Hashemi, Rebecca Hwa, and Diane Litman. 2017. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019a. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Sanqiang Zhao, Rui Meng, Daqing He, Saptono Andi, and Parmanto Bambang. 2018. Integrating transformer and paraphrase rules for sentence simplification. *arXiv preprint arXiv:1810.11193*.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.

A Domains

In EDITEVAL we strive to encompass datasets from many different domains, with an emphasis on factual content. Below in Table A1, we enumerate these domains.

Table A1: Number of targets provided ($|T|$) and the domains covered by each dataset.

Dataset	$\|T\|$	Domains
ITERATER	1	Wikipedia, ArXiv, and Wikinews
JFLEG	4	TOEFL exam
WNC	1	Wikipedia
STS Benchmark	1	Wikipedia, Q&A, news forums, videos, image descriptions
ASSET	10	Wikipedia
TurkCorpus	8	Wikipedia
WAFER	1	Wikipedia
FRUIT	1	Wikipedia

B Prompts

Below we enumerate the prompts used in EDITEVAL for each task. We also present Table C2 which shows the max and min results across these prompts as opposed to the average in Table 3.

Fluency

1. Fix grammar errors
2. Fix grammar or spelling mistakes
3. Fix grammar errors in this sentence
4. Fix all grammatical errors
5. Fix errors in this text
6. Update to remove grammar errors
7. Remove all grammatical errors from this text
8. Improve the grammar of this text
9. Grammar improvements
10. Remove grammar mistakes
11. Fix the grammar mistakes

Clarity

1. Make the text more formal, concise, readable and understandable
2. Make the text more formal
3. Make the text more concise
4. Make the text more readable
5. Improve the readability of the text
6. Make the text more understandable
7. Make the text clearer
8. Make the text easier to understand
9. Improve the clarity of the text

Coherence

1. Make the text more cohesive, logically linked and consistent as a whole
2. Make the text more cohesive
3. Improve the cohesiveness of the text
4. Make the text more logical
5. Make the text more consistent
6. Improve the consistency of the text
7. Make the text more understandable
8. Make the text clearer
9. Make the text easier to understand
10. Improve the coherency of the text

Neutralization

1. Remove POV
2. Neutralize this text
3. Make this more neutral
4. Make this text more neutral
5. Make this paragraph more neutral
6. Remove unsourced opinions from this text
7. Remove non-neutral points of view
8. Remove points of view
9. Make this text less biased

Paraphrasing

1. Paraphrase this sentence
2. Paraphrase
3. Paraphrase this paragraph.
4. Use different wording
5. Paraphrase this text
6. Rewrite this text
7. Rewrite this text with different wording
8. Rephrase this text
9. Reword this text

Simplification

1. Simplify this sentence
2. Make this simpler
3. Simplify
4. Make this easier to understand
5. Simplification
6. Change to simpler wording
7. Simplify this paragraph.
8. Use simpler wording
9. Simplify this text
10. Make this text less complex

Updating

1. Add missing information
2. Update the article
3. Update with new information

C Metrics

In this section, we describe each metric included in EDITEVAL in greater detail and our motivations for including them. In the cases where more than multiple valid targets, we follow convention and take the maximum of the scores computed using each target, since there can potentially be many valid edits, and a prediction only needs to align with one of the references.

EM and EM-Diff Exact match (EM) is the percentage of examples for which the performed edit exactly matches any of the targets. EM-Diff is a variant of EM that is computed on the diff level, where diffs are obtained using Python's `difflib` library. For a model output O, we compute EM-Diff as follows:

$$\frac{|\text{diff}(I, R) \cap \text{diff}(I, O)|}{\max(|\text{diff}(I, R)|, |\text{diff}(I, O)|)}$$

SARI Introduced by Xu et al. (2016), SARI is an n-gram based metric commonly used for measuring simplification (Nisioi et al., 2017; Zhao et al., 2018) and other editing tasks such as sentence fusion (Malmi et al., 2019). It has been demonstrated to correlate most closely with human judgement for simplification compared to many other n-gram based metrics (Xu et al., 2016). The metric measures how simplified a candidate system output is relative to the original and to the simplification references by rewarding words added, kept, or deleted in both the target and the output. More specifically, this is done by computing the arithmetic mean of n-gram F1-scores for each of the three operations. We utilize the EASSE (Alva-Manchego et al., 2019) implementation of SARI, which addresses inconsistencies in the original implementation [8].

GLEU GLEU (Napoles et al., 2015) is another variant of BLEU frequently used for grammatical error correction (Grundkiewicz et al., 2019; Yuan and Briscoe, 2016; Chollampatt and Ng, 2018). The issue with using BLEU for minimal edits can be attributed to the difference between analyzing machine translation and editing tasks. In the former, an untranslated word should always be penalized, but in the editing setting, an unmodified word in both the target and the output does not necessarily need to be penalized. Unlike BLEU, GLEU is customized to penalize n-grams changed in the targets but left unchanged by the system output. Napoles et al. (2015) not only demonstrated that GLEU correlates well with human rankings of corrections, but also that GLEU correlates much better than BLEU does.

ROUGE and UpdateROUGE For the task of updating or adding new information, we follow Logan IV et al. (2021) and use ROUGE and UpdateROUGE (Logan IV et al., 2021). ROUGE (Lin, 2004) is a popular n-gram based metric that is commonly used for evaluating summarization systems (Ren et al., 2016; Pasunuru and Bansal, 2018), but is also used in other tasks such as improving fluency (Kann et al., 2018) and simplification (Vanderwende et al., 2006). ROUGE essentially measures the overlap in n-grams. UpdateROUGE, a simple modification of ROUGE, computes ROUGE on the updated sentences rather than the full text. This is intended for tasks such as updating, because a majority of the target will remain unchanged. On the other hand, when evaluating using ROUGE, a system can often superficially achieve high scores by simply copying the input.

BERTScore BERTScore (Zhang et al., 2019b) is a versatile automatic metric that has been demonstrated to correlate well with tasks such as machine translation, image captioning, and abstractive text compression (Zhang et al., 2019b). We note, however, that some studies have demonstrated the metric's poor generalization ability to different datasets (Unanue et al., 2021). We include BERTScore in EDITEVAL for its broad applicability and its popularity.

D Limitations

Our evaluation tool is by no means an exhaustive measurement of editing capabilities. Firstly, there are additional domains that could potentially be added to EDITEVAL, such as books and blogs; as it currently stands, EDITEVAL is heavily constructed from the domain of Wikipedia. Fortunately, EDITEVAL's framework is flexible to the addition of datasets, provided that it has an input and target edit. In the same spirit, there are additional editing tasks such as verifying facts, citing, and reorganizing sentences/paragraphs which would be valuable to include in EDITEVAL. While we recognize these tasks as important to include in EDITEVAL, we consider these to be out of scope for the work at hand. Finally, our results demonstrate that

[8] https://github.com/feralvam/easse#differences-with-original-sari-implementation

Model	*Fluency* JFL		*Clarity* ITR-L	*Coherence* ITR-O	*Para.* STS	*Simplification* TRK	AST	*Neutral.* WNC	*Updating* FRU	WFI
Tk	32.9 / 41.6	36.0 / 77.6	**39.5 / 63.3**	35.7 / **77.1**	33.1	34.9	32.6	33.8 / 1.3	12.9 / 4.1	1.3 / 5.0
T0	45.4 / 43.1	32.6 / 50.9	33.8 / 34.0	23.7 / 25.5	35.9	35.3	35.9	27.5 / 0.1	14.9 / 12.4	5.4 / 17.2
T0++	36.7 / 43.9	37.2 / 82.0	38.6 / 61.6	36.0 / 75.8	30.7	33.9	33.3	32.1 / 0.6	12.8 / 3.7	4.6 / 8.5
PEER-3	59.3 / 57.7	54.5 / 86.3	34.0 / 60.6	33.8 / 74.1	34.6	36.4	35.5	57.4 / 29.3	40.2 / **33.6**	34.7 / 20.2
PEER-11	60.6 / 59.4	**55.4 / 87.0**	34.4 / 61.4	34.5 / 75.8	33.1	35.7	33.9	**59.0 / 30.9**	**40.8** / 33.4	**35.2 / 21.4**
OPT	53.5 / 53.9	41.0 / 78.5	35.6 / 44.4	34.4 / 56.9	31.1	34.7	35.3	34.9 / 0.9	35.9 / 28.1	27.0 / 12.3
GPT-3	52.6 / 54.2	39.1 / 79.2	35.6 / 45.8	29.9 / 42.9	29.4	35.5	35.9	34.9 / 1.1	36.3 / 21.6	28.2 / 11.2
InsGPT	**62.7 / 60.4**	51.0 / 85.0	36.5 / 52.6	**37.6** / 68.8	**45.2**	**40.2**	**40.9**	37.2 / 3.8	36.6 / 25.2	26.0 / 17.3
Tk	30.3 / 35.9	27.9 / 42.1	36.8 / 49.9	32.2 / **63.4**	28.6	30.6	26.1	27.9 / 0.0	12.3 / 3.4	1.2 / 4.1
T0	39.5 / 34.2	21.2 / 26.7	31.4 / 27.4	21.0 / 18.0	31.9	32.9	27.6	18.5 / 0.0	13.7 / 8.1	4.8 / 15.6
T0++	33.0 / 42.2	33.1 / 62.3	**36.8 / 52.6**	29.3 / 45.8	25.5	31.9	25.4	27.4 / 0.2	12.5 / 3.7	3.9 / 7.5
PEER-3	50.2 / 49.8	45.4 / 77.2	30.5 / 36.7	31.1 / 47.3	23.2	29.1	25.4	44.4 / 13.5	37.0 / 26.5	34.1 / 16.3
PEER-11	49.8 / 46.7	**45.9 / 82.5**	31.4 / 43.3	31.9 / 47.9	24.3	29.4	25.7	**45.5 / 15.7**	**37.5 / 27.3**	**34.7 / 19.0**
OPT	40.7 / 41.0	29.7 / 55.5	27.8 / 22.1	22.9 / 24.6	26.1	30.3	26.2	25.0 / 0.0	35.8 / 26.6	26.5 / 9.8
GPT-3	43.6 / 46.7	27.8 / 41.3	32.2 / 35.8	24.4 / 28.8	25.3	29.3	22.6	26.0 / 0.2	35.6 / 21.2	26.1 / 10.0
InsGPT	**59.2 / 56.2**	44.7 / 77.4	34.1 / 44.3	**33.4** / 53.0	**40.2**	**37.0**	**35.4**	32.4 / 0.7	35.9 / 24.4	22.2 / 15.3
Copy	26.7 / 40.5	32.3 / 86.0	29.5 / 62.9	31.3 / 77.2	21.1	26.3	20.7	31.9 / 0.0	29.8 / 0.0	33.6 / –
SotA	– / *62.4*	*37.2* / –	*46.2* / –	*38.3* / –	–	*34.4*	*37.2*	– / *45.8*	– / *47.4*	– / –

Table C2: Maximum (top half) and minimum (bottom half) scores across prompts for all downstream tasks considered. The first numbers for each task are SARI scores; additional metrics are GLEU for fluency, clarity, and coherence, EM for neutralization, Update-R1 for updating. The best results are highlighted in bold. Tk-Instruct and InstructGPT are shorthanded as Tk and InsGPT, respectively.

many of the metrics give conflicting signal as to the rankings of the baselines, indicating further work is needed to identify better metrics for measuring overall editing capacity.

E Broader Impact and Ethics

Before being deployed, this work was reviewed by an internal board to ensure compliance with all licensing. We also verified that no datasets included in EDITEVAL contains information that uniquely identifies individual people. All code, results, and a leaderboard are made publicly available. Our benchmark is intended to help drive the development of language models that can edit. Such systems may be able to carry out a wide variety of text modifications and have a broad range of societal implications, such as enabling those with limited access to educational resources to create knowledge-intensive or professional articles (Redi et al., 2020). EDITEVAL is not to be used for ill-intended purposes, such as making adversarial text modifications that introduce misleading or problematic content. Additionally, EDITEVAL inherits biases inherent in its constituent datasets, and we encourage further work to understand the biases and limitations of the datasets used in EDITEVAL.