# XATU: A Fine-grained Instruction-based Benchmark for Explainable Text Updates

**Haopeng Zhang†, Hayate Iso‡, Sairam Gurajada‡, Nikita Bhutani‡**
IFM Lab, UC Davis†, Megagon Labs‡
haopeng@ifmlab.org
{hayate, sairam, nikita}@megagon.ai

**Abstract**

Text editing is a crucial task of modifying text to better align with user intents. However, existing text editing benchmark datasets contain only coarse-grained instructions and lack explainability, thus resulting in outputs that deviate from the intended changes outlined in the gold reference. To comprehensively investigate the text editing capabilities of large language models (LLMs), this paper introduces XATU, the first benchmark specifically designed for fine-grained instruction-based explainable text editing. XATU considers finer-grained text editing tasks of varying difficulty (simplification, grammar check, fact-check, etc.), incorporating lexical, syntactic, semantic, and knowledge-intensive edit aspects. To enhance interpretability, we combine LLM-based annotation and human annotation, resulting in a benchmark that includes fine-grained instructions and gold-standard edit explanations. By evaluating existing LLMs against our benchmark, we demonstrate the effectiveness of instruction tuning and the impact of underlying architecture across various editing tasks. Furthermore, extensive experimentation reveals the significant role of explanations in fine-tuning language models for text editing tasks. The benchmark will be open-sourced to support reproduction and facilitate future research at https://github.com/megagonlabs/xatu.

**Keywords:** Text Editing, Large Language Model, Language Resources, Explainability

## 1. Introduction

Text editing is the task of modifying text to better align with user intents. Recent advances in large language models (LLMs) have demonstrated remarkable zero-shot text generation capabilities across a wide range of downstream natural language processing (NLP) tasks like question answering, dialogue, and summarization (Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020; Zhang et al., 2022, 2023a). It has been shown that incorporating instruction tuning (Wei et al., 2022) and reinforcement learning from human feedback (RLHF) (Bai et al., 2022; Ouyang et al., 2022) can further enhance a model's ability to align with the user's intent.

Text editing plays a crucial role in real-world text generation applications. While many current text generation models typically employ a one-shot manner, humans often engage in an iterative process when writing text, which entails multiple drafts and revisions (Faltings et al., 2021). Thus, approaching text generation as an iterative process with successive updates to the text is a more effective way to achieve higher alignment between the model's outputs and the user's intent.

Consequently, *instruction-based text editing* tasks have received growing interest recently. They

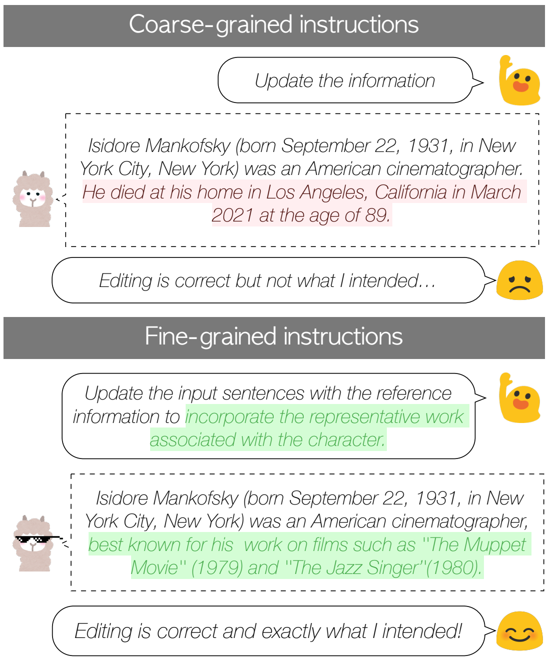


Figure 1: Illustrated examples of coarse- and fine-grained instructions for text editing. LLMs can accurately perform text editing based on coarse-grained instructions, but may not meet the user's intention. In contrast, fine-grained instructions lead to accurate and user-intended text editing.

enable a user to interact with a model with commands to update existing text and achieve desirable text. Typically commands target broad edits such

The work was done when Haopeng Zhang was a research intern at Megagon Labs



*LREC-COLING 2024*, pages 17739–17752
20-25 May, 2024. 

| Benmark | Domain | Reference | Instruction | Explanation |
|---|---|---|---|---|
| WikiSemanticIntention (Yang et al., 2017) | wiki | ✗ | ✗ | ✗ |
| WikiAtomicEdits (Faruqui et al., 2018) | wiki | ✗ | ✗ | ✗ |
| WikiDocEdits (Faltings et al., 2021) | wiki | ✓ | coarse-grained | ✗ |
| ITERATER (Du et al., 2022) | multiple | ✗ | ✗ | ✗ |
| EditEval (Dwivedi-Yu et al., 2022) | multiple | ✓ | coarse-grained | ✗ |
| XATU (ours) | multiple | ✓ | fine-grained | ✓ |

Table 1: A detailed comparison of XATU with existing text editing datasets and benchmarks.

as adding or removing content, or changing the meaning of text (Faltings et al., 2021). To evaluate models' capabilities for text editing, instruction-based benchmarks such as EditEval (Dwivedi-Yu et al., 2022) have been proposed recently.

Nevertheless, prior datasets and benchmarks only consider simple commands or instructions, such as 'Update the article' or 'Expand', resulting in three main limitations. **(a)** These instructions are often too coarse-grained, to the extent that even humans may struggle to understand the desired intent of the edit, limiting a model's capability to follow instructions. **(b)** It poses challenges for text editing evaluation since outputs obtained using coarse-grained instructions may not accurately reflect the true editing capabilities of LLM systems. The coarse-grained instructions cannot provide sufficient guidance for the editing system, so the edited output may appear correct but deviate from the intended changes, as shown in Figure 1. The edited output updates the death information of "Isidore Mankofsky" correctly but since the ground truth output updates the information about his famous works. This edit will receive very low evaluation scores since current automatic evaluation methods for text editing systems are mostly based on surface-level overlap with human-written gold references. **(c)** The current text editing systems lack explainability, which in turn limits the interpretability of the edits made. This lack of interpretability hampers our understanding of the underlying text editing behaviors.

To address the aforementioned limitations, we present XATU, a novel fine-grained instruction-based benchmark designed for e<u>x</u>pl<u>a</u>inable <u>t</u>ext <u>u</u>pdates. The benchmark leverages high-quality existing data sources from different tasks to enable automatic evaluation of LLM editing capabilities by incorporating an LLM-in-the-loop annotation process. In comparison to other datasets and benchmarks, XATU highlights its inclusion of a wider range of diverse topics, fine-grained edit instructions, and corresponding explanation rationales, as shown in Table 1. By utilizing XATU, we are able to evaluate and compare the performance of several state-of-the-art language models, considering both zero-shot and fine-tuning settings. Experiment results emphasize the significant role of explanations in the fine-tuning process of language models for text editing tasks. Our findings shed light on the importance of incorporating explanation rationales to enhance the performance of language models in text editing, ultimately leading to improved output quality. We summarize the contributions of this paper as follows:

- We propose XATU, a new benchmark for instruction-based text editing. We collect high-quality data from a set of tasks and datasets for iterative text updates and provide gold fine-grained instructions and explanations with both LLM generation and human annotations. To the best of our knowledge, XATU is the **first** text editing dataset with **fine-grained instructions and explanations**.
- We conduct a thorough evaluation of existing open and closed large language models on our benchmark with different input formats to test their text editing capabilities. We further investigate the influence of instruction fine-tuning using explanations and fine-grained editing instructions.

## 2. Related Work

### 2.1. Text Editing

Several prior studies have concentrated on the domain of text editing. Guu et al. (2018) introduced a method that involves generating sentences through the editing of training prototypes. Faruqui et al. (2018) presented the WikiAtomicEdits dataset, which comprises edits extracted from Wikipedia editing history. Subsequently, Yin et al. (2019) and Reid and Neubig (2022) leveraged Wikipedia editing history to propose the task of learning to represent edits. Furthermore, Iso et al. (2020) introduced a fact-based text editing task in their work.

Recently, text editing has been approached as an interactive task, leading to the development of command-based editing systems (Faltings et al., 2021) and interactive editing systems (Schick et al., 2023). Researchers have also explored the integration of text editing into interactive self-refinement frameworks for text generation using large language models. Welleck et al. (2023) introduced a self-corrective learning framework that incorporates a corrector into the language model, enabling self-correction during sequence generation. Akyürek et al. (2023) proposed a reinforcement learning-based approach for generating natural language feedback to correct generation errors. Furthermore, Zhang et al. (2023b) presented an iterative refinement framework for abstractive summarization.

## 2.2. Text Editing Datasets

Previous research has also introduced various datasets and resources that focus on iterative text revisions within specific domains. For instance, datasets presented in Yang et al. (2017), Faruqui et al. (2018) and Anthonio et al. (2020) primarily concentrated on the domain of Wikipedia edit history, while Spangher et al. (2022) developed a dataset tailored to news articles. Researchers in the field have also proposed more comprehensive benchmarks that cover multiple domains. IteraTeR (Du et al., 2022) provides iterative tasks from multiple domains, but has only a limited number of tasks, such as fluency, coherence, clarity, style, and meaning change. EditEval (Dwivedi-Yu et al., 2022) is perhaps the closest to our benchmark XATU in that it covers data from multiple domains: Wikipedia, Wikinews, news articles, and arXiv.

Moreover, it is important to note that only EditEval and WikiDocEdits (Faltings et al., 2021) provide short and simple instructions or commands to guide the editing process. In contrast, our benchmark XATU distinguishes itself by including fine-grained instructions for each data instance, which is necessary for assessing the genuine text editing capabilities of different systems. Furthermore, XATU also includes the rationale explanations for each of its instances, which have been proven to be significant for instruction fine-tuning models (Hsieh et al., 2023; Mukherjee et al., 2023).

# 3. The XATU Benchmark

We introduced the XATU benchmark in this section. We outline the criteria for selecting datasets to construct XATU (§3.1), how to annotate fine-grained instructions and explanations for the edits (§3.2), and discuss various use cases (§3.3). As shown in Figure 2, each instance in XATU comprises five key components: inputs, outputs, optional references, fine-grained instructions, and explanations.

## 3.1. Data Source

In line with the systematic formulation of language style (DiMarco and Hirst, 1993), we adopt a similar approach to categorize text editing into four main editing aspects: **lexical, syntactic, semantic, and knowledge**. Our benchmark XATU aims to serve as a relatively comprehensive benchmark and a holistic evaluation framework for text editing tasks, covering all four aspects of text editing.

To ensure comprehensiveness and quality, we meticulously curate high-quality editing data from

**Instruction:** Modify the input text to incorporate details about the nature of the matches played, the players included in the tour, mention of Nat Sciver captaining the England team for the first time, and the simultaneous scheduling of New Zealand men's matches.

**Input:** The England women's cricket team is scheduled to play against New Zealand women's cricket team in February and March 2021. The tour is scheduled to be played during the time that was originally scheduled to be used to host the ...

**Reference:** (0) The last three T201 matches took place on the same day as the New Zealand women's fixtures at the same venues...
(7) Sophie_Ecclestone International career In February 2021, during England's tour of New Zealand, Ecclestone took her 100th international wicket.
(8) Sarah_Glenn International career In 2021, Glenn was named in the squad for England's tour of New Zealand.

**Output:** The England women's cricket team played against the New Zealand women's cricket team in February and March 2021. The series was played during the time that was originally scheduled to be used to host the 2021 Women's Cricket World Cup, after the tournament was postponed by one year due to the COVID-19 pandemic...

**Explanation:** The text editing system was instructed to update the text based on the references provided. It corrected the phrasing to past tense, indicating that the England and New Zealand women's cricket teams already 'played' the matches instead of 'scheduled to play'. The information about the tour, including the confirmation of fixtures...

Figure 2: The instance format of the data in XATU benchmark. Data in blue (Input, Output, Reference) are extracted from the corresponding data sources, and data in green (Fine-grained instruction and explanation) are obtained from joint automatic and human annotations.

$4$ distinct downstream NLP tasks, encompassing a total of $1000$ annotated data from $9$ data sources following DiMarco and Hirst (1993). A summary of the tasks and datasets, along with an overview of the editing aspects covered by each dataset, is presented in Table 2. We also transform all datasets within the XATU benchmark into a consistent format, extracting the input text, gold edits, task type, and optional reference documents from the following data sources, as illustrated in the example in Figure 2. This standardized format facilitates evaluation and comparison across different datasets within the benchmark.

**Grammar Error Correction** The first task in XATU is to identify and correct errors that don't follow grammar rules. We first incorporate data from JHU FLuency-Extended GUG (JFLEG) (Napoles et al., 2017), a dataset specifically designed for the task of grammar error correction. JFLEG uses holistic fluency edits to not only correct grammatical errors but also make the original text more native sounding. By including this dataset, we aim to assess the basic ability of text editing systems to improve fluency and grammatical accuracy simultaneously.

**Simplification** We also include ASSET (Alva-Manchego et al., 2020), a text simplification dataset that provides manually produced simplifications through a wide range of transformation techniques. We aim to evaluate the text editing capabilities of systems when it comes to simplifying complex texts

| Task | Dataset | Train | Test | Aspect |
|---|---|---|---|---|
| Grammar | JFLEG | 10 | 50 | Lexical, Syntactic |
| Simplification | ASSET | 10 | 50 | Lexical, Syntactic |
| Style Transfer | WNC | 20 | 100 | Lexical, Semantic |
| | Wikibias | 20 | 100 | |
| | StylePTB | 10 | 50 | |
| Information Update | FRUIT | 30 | 150 | Semantic, Knowledge |
| | Evidence | 40 | 200 | |
| | DeFacto | 30 | 150 | |
| | Factedit | 30 | 150 | |

Table 2: The detailed datasets included in the XATU benchmark.

| Difficulty | Dataset | Levenshtein↓ |
|---|---|---|
| Easy | JFLEG | 1.47 |
| | WNC | 1.58 |
| | STYLEPTB | 1.72 |
| | Wikibias | 1.92 |
| Medium | Evidence | 3.56 |
| | ASSET | 4.72 |
| | DeFacto | 4.82 |
| Hard | Factedit | 9.75 |
| | FRUIT | 13.62 |

Table 3: Average token-level Levenshtein distance between the input and edited output in XATU.

while preserving their essential meaning.

**Style Transfer** Text editing also involves transferring the styles of the input sentences, so we include the following three datasets, aiming to evaluate the effectiveness of text editing systems in handling style transfer and bias mitigation tasks.

- Wiki Neutrality Corpus (WNC) (Pryzant et al., 2020) is a collection of original and de-biased sentence pairs mined from Wikipedia edits by carefully filtering based on the editor's comments.
- Wikibias (Zhong, 2021) is a manually annotated parallel corpus with sentence pairs from Wikipedia editing history. The inclusion of Wikibias complements the mined data from WNC, providing additional annotations to address both sentence-level and token-level biases.
- StylePTB (Lyu et al., 2021) contains paired sentences undergoing various fine-grained stylistic changes and compositions of multiple transfers.

**Information Update** Our XATU benchmark also includes the following four knowledge-intensive text editing datasets. These datasets are particularly challenging as they require text editing systems to update the input text based on given instructions and external reference evidence. These knowledge-intensive datasets offer a valuable assessment of the systems' capacity to leverage external information for text editing tasks.

- FRUIT dataset (Iv et al., 2022) is a dataset collected by comparing two snapshots of the same Wikipedia article. The reference documents were identified by searching for other Wikipedia articles and human filtering. We include this gold set from the FRUIT dataset in XATU.
- Evidence (Thorne and Vlachos, 2021) is a dataset created using a two-stage distant supervision approach, where evidence is incorporated into masked claims from the FEVER dataset (Thorne et al., 2018). Each claim in the dataset is paired with reference evidence claims obtained from Wikipedia.
- DeFacto (Liu et al., 2022) dataset consists of document summaries from the XSum dataset (Narayan et al., 2018), accompanied by human-corrected versions that rectify factual errors present in the original summaries. For the text editing task, we utilize the original document as the reference to evaluate the ability of text editing systems to correct factual errors in the summaries effectively. The inclusion of this dataset allows us to assess the performance of systems in editing text to ensure factual accuracy, an important aspect of text editing in real-world applications.
- FactEdit (Iso et al., 2020) is an editing dataset that incorporates facts sourced from a knowledge base, such as multiple triples, as the reference. This dataset is constructed from table-to-text data sources.

### 3.2. Annotation Process

**Challenges in Crowdsourcing** Annotating fine-grained instructions and explanations poses a challenge due to the inherent open-ended nature of the text production task, making it difficult to filter out low-quality workers. Additionally, many workers turn to LLMs for text production tasks due to their prevalence (Veselovsky et al., 2023), but the output quality heavily depends on the user's LLM skills (Zamfirescu-Pereira et al., 2023). Therefore, relying solely on crowd workers for annotation may not be advisable.

To address these concerns and maintain the quality and reliability of the annotated data, we implemented an LLM-in-the-loop annotation approach, which involves generating candidates of fine-grained instruction and explanations using

**Input:** At least once an episode we see protestors marching around screaming slogans
**Output:** One or more times an episode protestors are seen walking around yelling blue slogans.

---

**Input:** <del>At least once</del> an episode <del>we see</del> protestors <del>marching</del> around <del>screaming</del> slogans
**Output:** <ins>One or more times</ins> an episode protestors <ins>are seen walking</ins> around <ins>yelling blue</ins> slogans <ins>.</ins>

Figure 3: Illustrated example of adding HTML tags to the input and output to explicitly indicate the edited portion to LLMs.

LLMs and then validating the output with human workers. This allowed us to mitigate the challenges associated with crowd-sourcing and refine our methodology for generating and evaluating fine-grained instructions and editing explanations.

**Candidate Generation by LLMs** We generate candidates of fine-grained instructions and explanations for each instance using LLMs (Zhou et al., 2023b; Honovich et al., 2023; Li et al., 2023). For the LLMs to generate candidates, we use GPT-4 (OpenAI, 2023). Through our preliminary prompt engineering, we found that merely asking LLMs to generate the instructions and explanations often resulted in undesired outputs[1]. We found that adding HTML tags to explicitly indicate the edited portion is highly effective, as illustrated in Figure 3. Thus, we decided to use this prompting technique to generate instructions and explanations. The full prompt can be found in the Appendix.

Note that our analysis showed no significant differences between coarse and fine-grained instructions for easy tasks (i.e., JFLEG, WNC, STYLEPTB, and WikiBias) since the inputs are relatively short and the edit distance is relatively small. Consequently, we chose to only conduct fine-grained instruction annotation on the more complex information update task ($65\%$ data).

**Candidate Validation by Human** We conducted candidate validation for fine-grained instructions and explanations by employing human evaluation through the Appen platform.[2] Workers were instructed to assess the validity of generated instructions or explanations based on input, output, and optional references. To ensure quality, we selected and manually annotated $20$ random test examples and accepted workers who provided correct answers for over $80\%$ of them. If a candidate was considered invalid, we repeated the candidate generation process for that instance.

[1] More details can be found in the Appendix.
[2] https://appen.com/

We evaluated the quality of annotations through inter-annotator agreement, which resulted in an overall agreement rate of 80.27%. We annotated the dataset based on difficulty levels defined in Table 3. A detailed breakdown of the number of instances annotated is provided in Table 2. The result of this rigorous process was the creation of the XATU benchmark, containing 1000 high-quality, fine-grained text editing instances.

### 3.3. Benchmark Usage

Previous work has explored the use of Hamming distance for sentence-level editing task difficulty estimation (Lyu et al., 2021). However, as our input and output texts may have different lengths, we employ a token-level Levenshtein distance to measure the relative difficulty of the different datasets in our benchmark. Based on this distance measurement, we categorize the tasks into three levels: easy, medium, and hard. The results of this categorization can be found in Table 3. We notice that information update tasks generally require more editing operations compared to simple lexicon tasks like grammar error correction and neutralization. This highlights the increased complexity and difficulty associated with tasks that involve updating and incorporating external information into the text. To ensure that the benchmark presents a challenging evaluation environment, we deliberately incorporate more examples from demanding tasks.

With the inclusion of fine-grained instructions and corresponding editing rationale explanations, our benchmark offers comprehensive support for a range of traditional text editing tasks like edit representation modeling, automatic editing instruction generation, and editing span prediction. Additionally, the unique structure of XATU enables two new tasks: editing explanation generation and editing evidence retrieval. Our annotated edit explanations can be used to train and assess models aimed at generating coherent and informative explanations for the editing process. Moreover, the information update data that incorporates external reference documents allows for the development and evaluation of models that can effectively retrieve the necessary evidence for text editing tasks.

By providing support for these downstream tasks, our benchmark aims to facilitate advancements in various aspects of text editing, fostering the development of models and techniques that can improve the efficiency and effectiveness of text editing systems.

## 4. Experiments

In this section, we perform a series of text editing experiments using the XATU benchmark, aiming to investigate and address the following research questions:

- **Q1**: What is the actual text editing ability of existing open/closed large language models?

By evaluating the performance of LLMs, we aim to assess their effectiveness of instruction-based text editing and understand their limitations and strengths in this context.

- **Q2**: How do fine-grained instructions in XATU differ from the coarse-grained instructions?

By comparing the performance of models using fine-grained instructions from XATU, we seek to highlight the impact of instruction granularity on the quality and accuracy of text editing.

- **Q3**: How do the explanations provided in XATU benefit model tuning under the fine-tuning settings?

By examining the performance of models that incorporate the explanations provided in XATU during fine-tuning, we aim to evaluate the effectiveness of these explanations in improving model performance and understanding the role of explanations in the context of text editing.

## 4.1. Experimental Setup

We include the following baseline LLMs in our experiments:

- **GPT-3** (Brown et al., 2020) is a 175B parameter pre-trained decoder-only model.
- **GPT-4** (OpenAI, 2023) is the most recent multimodal large language model created by OpenAI. It was pre-trained to predict the next token and was then fine-tuned with reinforcement learning from human and AI feedback. We evaluate both GPT-3 and GPT-4 through OpenAI's API.[3]
- **T5** (Raffel et al., 2020) is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks.
- **Flan-T5** (Chung et al., 2022) is an enhanced version of T5 that has been instruction fine-tuned on a mixture of tasks.
- **UL2** (Tay et al., 2023) is a unified framework for pre-training models with Mixture-of-Denoisers, a pre-training objective that combines diverse pre-training paradigms together.
- **Flan-UL2** is an encoder-decoder model that uses the same configuration as the UL2 model. It was fine-tuned using the "Flan" prompt tuning and dataset collection.
- **LLaMa** (Touvron et al., 2023) is a collection of foundation language models trained on trillions of tokens. It is an auto-regressive language model built on transformer architecture.
- **Alpaca** (Taori et al., 2023) is an instruction-following language model fine-tuned from the LLaMA 7B model on 52K instruction-following demonstrations.

Note that PEER (Schick et al., 2023) is another strong text editing baseline according to results in Dwivedi-Yu et al. (2022). We didn't include it since the model checkpoint is not publicly available.

For evaluation, we use SARI scores (Xu et al., 2016), an $n$-gram based metric commonly used for measuring editing tasks such as simplification (Zhao et al., 2018) and sentence fusion (Malmi et al., 2019). It has been demonstrated to correlate most closely with human judgment and is computed as:

$$SARI = (F1_{add} + F1_{keep} + P_{del})/3,$$

where $F1_{add}, F1_{keep}, P_{del}$ represent the F1 scores and precision for add, keep, and delete operations, respectively. We utilize the Hugginface implementation of SARI [4], where $n = 4$. EditEval (Dwivedi-Yu et al., 2022) includes more n-gram-based valuation metrics, but the results do not demonstrate any significant difference from SARI scores.

**Implementation Details** We use the Huggingface implementation of all models with LoRA (Hu et al., 2022) for the fine-tuning experiments. We instruction-tuned all models for $15$ epochs with $200$ training examples at a learning rate of $1e-5$ and tested on the original $1000$ test set, as shown in Table 2. We keep a small training data size as (Zhou et al., 2023a) found that alignment doesn't require too much data. We use LoRA (Hu et al., 2022) to instruction fine-tune all models for $15$ epochs with $200$ additional training examples at a learning rate of $1e-5$.

## 4.2. Zero-shot Results

The zero-shot evaluation results on the XATU benchmark are presented in Table 4, which highlights several important observations.

**GPT-4 demonstrates exceptional zero-shot editing performance**, surpassing all other large

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

[4] https://huggingface.co/spaces/evaluate-metric/sari

| **Model** | **Setting** | **JFLEG** | **ASSET** | **WNC** | **Wikibias** | **PTB** | **FRUIT** | **Evidence** | **DeFacto** | **Factedit** |
|---|---|---|---|---|---|---|---|---|---|---|
| Flan-T5 | coarse | 64.98 | 52.01 | 62.54 | 55.58 | 51.05 | 19.06 | 39.13 | 36.77 | 7.98 |
| | fine | - | - | - | - | - | 16.84 | 41.84 | 42.87 | 6.80 |
| | Exp. | 59.09 | 46.47 | 56.71 | 51.98 | 46.78 | 21.40 | 53.34 | 38.78 | 11.65 |
| Flan-UL2 | coarse | 64.35 | 50.08 | 59.39 | 51.88 | 51.83 | 19.84 | 52.61 | 31.15 | 23.06 |
| | fine | - | - | - | - | - | 43.38 | 68.79 | 52.54 | 35.15 |
| | Exp. | 83.38 | 66.82 | 86.78 | 79.22 | 74.81 | 38.08 | 74.35 | 52.83 | 29.13 |
| Alpaca | coarse | 68.14 | 41.20 | 45.68 | 42.35 | 54.14 | 51.75 | 53.65 | 52.58 | 41.08 |
| | fine | - | - | - | - | - | 52.74 | 69.42 | 60.81 | 34.83 |
| | Exp. | 75.82 | 62.41 | 70.90 | 69.53 | 77.99 | 54.00 | 74.12 | 66.17 | 39.89 |
| GPT3 | coarse | 50.74 | 30.82 | 32.24 | 34.58 | 42.63 | 26.47 | 34.12 | 28.42 | 31.46 |
| | fine | - | - | - | - | - | 27.43 | 36.72 | 34.85 | 36.52 |
| | Exp. | 56.34 | 37.42 | 41.34 | 42.48 | 47.23 | 32.54 | 47.23 | 37.28 | 38.75 |
| GPT4 | coarse | 70.32 | 56.71 | 64.18 | 58.13 | 58.72 | 43.28 | 58.19 | 54.62 | 43.59 |
| | fine | - | - | - | - | - | 49.28 | 62.34 | 62.42 | 48.27 |
| | Exp. | **84.58** | **73.54** | **89.23** | **82.93** | **84.39** | **59.43** | **81.38** | **71.38** | **58.38** |

Table 4: Zero-shot text editing results of different LLMs under different prompt settings on XATU benchmark. We omit results on simple datasets that do not require fine-grained instruction. We use coarse, fine, and Exp. to represent text editing with coarse-grained instructions fine-grained instructions, and explanations, respectively.

| **Model** | **Setting** | **JFLEG** | **ASSET** | **WNC** | **Wikibias** | **PTB** | **FRUIT** | **Evidence** | **DeFacto** | **Factedit** |
|---|---|---|---|---|---|---|---|---|---|---|
| T5 | coarse | 62.75 | 52.26 | 64.20 | 56.01 | 56.59 | 45.27 | 60.03 | 59.07 | 47.65 |
| | fine | 63.43 | 51.85 | 58.65 | 56.93 | 61.07 | 48.42 | 71.56 | 60.91 | 45.80 |
| | Exp. | 72.73 | 60.23 | 77.39 | 70.76 | 72.22 | 43.83 | 77.24 | 64.23 | 46.14 |
| Flan-T5 | coarse | 64.07 | 52.71 | 65.04 | 58.86 | 63.63 | 50.95 | 62.81 | 59.40 | 45.99 |
| | fine | 65.39 | 53.00 | 65.11 | 59.56 | 63.15 | 53.64 | 76.43 | 66.62 | 47.52 |
| | Exp. | 79.17 | 69.18 | 84.67 | 75.31 | 75.64 | 52.33 | 85.60 | 71.25 | 47.90 |
| LLaMA | coarse | 63.86 | 45.46 | 62.84 | 53.72 | 58.87 | 49.18 | 63.69 | 52.09 | 50.25 |
| | fine | 66.18 | 47.56 | 64.30 | **61.08** | 60.67 | 53.44 | 82.19 | 69.38 | 52.45 |
| | Exp. | 83.31 | 70.28 | 91.22 | 84.66 | 84.54 | 54.22 | 86.85 | 79.14 | 55.45 |
| Alpaca | coarse | 65.71 | 44.95 | 63.68 | 55.77 | 63.62 | 49.18 | 64.13 | 56.73 | 46.73 |
| | fine | 69.64 | 47.14 | 62.78 | 50.57 | 61.18 | 51.81 | 83.48 | **73.69** | 46.35 |
| | Exp. | 83.52 | 70.83 | 87.93 | 75.85 | 83.56 | 58.33 | 88.41 | 77.91 | 47.45 |
| UL2 | coarse | 66.61 | **54.55** | 69.82 | **60.45** | 61.86 | 51.49 | 70.65 | **59.21** | 58.18 |
| | fine | 71.22 | **54.84** | 71.19 | 56.04 | 67.27 | 56.58 | 84.37 | 70.27 | 56.06 |
| | Exp. | 87.81 | 78.22 | 91.79 | 82.24 | **88.10** | 54.64 | 90.54 | 78.65 | 56.44 |
| Flan-UL2 | coarse | **68.03** | 52.34 | **73.93** | 57.24 | **73.12** | **51.81** | **71.80** | 58.01 | **58.72** |
| | fine | **71.84** | 52.59 | **75.16** | 58.09 | **69.38** | **63.91** | **86.17** | 73.56 | **60.64** |
| | Exp. | **90.44** | **79.84** | **94.68** | **86.23** | 86.06 | **59.46** | **91.71** | **82.84** | **60.76** |

Table 5: Instruction fine-tuning results of different LLMs under different prompt settings on XATU. We use coarse, fine, and Exp. to represent text editing with coarse-grained instructions fine-grained instructions, and explanations, respectively.

language models by a significant margin across all datasets. Alpaca emerges as the second-strongest model, but there is still a considerable performance gap when compared to GPT-4. On the other hand, models without instruction tuning, such as GPT-3, struggle to follow the instructions effectively, resulting in relatively low SARI scores. This emphasizes the importance of instruction tuning in enhancing the text editing performance of language models.

In addition, we find that **almost all models exhibit improvements when guided by the fine-grained instructions** provided in XATU, as opposed to simple and coarse-grained instructions. This underscores the significance of fine-grained instructions in enhancing the performance and accuracy of text editing tasks. Notably, the inclusion of explanations as guidance during the text editing process leads to further improvements across all models.

Furthermore, our findings reveal that **the underlying architecture (encoder-decoder vs. decoder-only) of language models significantly impacts the performance of different types of text editing tasks**. We can see that Flan-T5 and Flan-UL2 exhibit biased performance, excelling in more straightforward tasks like style transfer but facing challenges in more complex tasks like information updates. In contrast, the Alpaca model performs well on information update tasks but achieves lower scores in simple neutralization tasks. We attribute this difference in performance to the *underlying architecture* of the foundation models: Flan-T5 and Flan-UL2 are both encoder-decoder models, while Alpaca is built upon the LLaMa decoder-only model. The results demonstrate that the encoder's ability to understand and represent the input text is more important for lexical and syntactic editing, while the decoder's capacity to generate new and relevant text appears to be more influential for knowledge-intensive tasks.

## 4.3. Fine-tuning Results

After fine-tuning all models using different prompting formats, we evaluate the fine-tuned models on XATU, and the results are presented in Table 5. A notable observation is the significant performance improvement compared to the zero-shot results across all datasets and under all settings. This demonstrates **the effectiveness of instruction fine-tuning, even with a limited number of examples** for text editing tasks ($200$ in this case).

Among all the instruction fine-tuned models, Flan-UL2 consistently exhibits the strongest performance across all three settings. This indicates its strong adaptation capability during the fine-tuning process. Furthermore, we observe that the fine-grained instructions in XATU lead to larger performance improvements in tasks that have fine-grained instructions (e.g. FRUIT, DeFacto), while the improvements are comparatively smaller in simpler tasks that lack fine-grained instructions (e.g. WNC, PTB) in the fine-tuning data.

Comparing T5 and UL2 with their instruction-tuned versions (Flan-T5 and Flan-UL2), we observe that **the instruction-tuned versions of LLM consistently outperform their base models**. This is especially evident when the models are guided with fine-grained instructions or explanations. These results further emphasize the effectiveness of the instruction alignment ability that these models acquire through instruction fine-tuning.

Overall, the fine-tuning process significantly improves the performance of the models on text editing tasks. The presence of fine-grained instructions and explanations further enhances the performance.

## 4.4. Discussion

As illustrated in Figure 4, we compare the results of three models after instruction fine-tuning with fine-grained instructions in XATU vs. coarse-grained instructions. We observe that fine-grained instruction models consistently outperform their coarse-grained counterparts, indicating the effectiveness of detailed instruction in improving instruction following and text editing capabilities. Across both coarse-grained and fine-grained instruction settings, we find that the Flan-UL2 model achieves the largest average improvements. On the other hand, Alpaca demonstrates superior robustness across all text editing tasks compared to Flan-T5 and Flan-UL2.

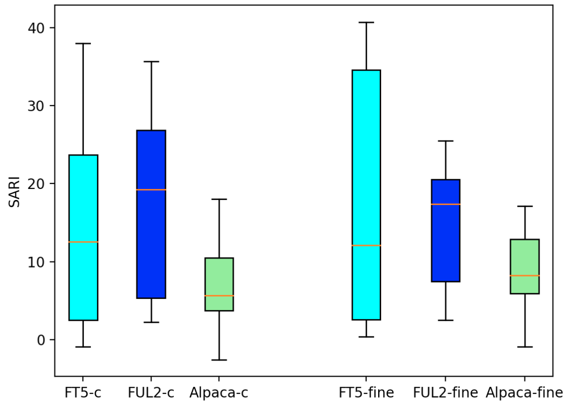


Figure 4: Fine-tuning with fine-grained instructions (-fine) vs. coarse instructions (-c).

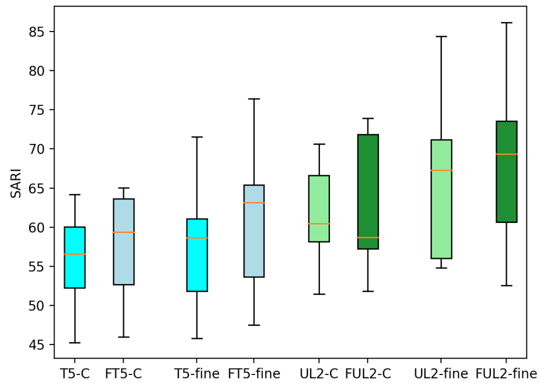


Figure 5: Boxplot comparing instruction-tuned LLMs (Flan-xx) vs. pre-trained counterparts with fine-grained (-fine) and coarse instructions (-c).

As depicted in Figure 5, the Flan-UL2 model consistently outperforms the UL2 model across all settings. Similarly, the Flan-T5 model performs better than the base T5 model. These results highlight the effectiveness of instruction tuning in improving the performance of language models across different settings. The instruction-tuning procedure allows the models to better understand and follow the provided instructions, resulting in more accurate and appropriate text edits. We also notice edits generated with coarse-grained instructions show better robustness across all tasks.

## 5. Conclusion

This paper introduces XATU, the first benchmark for explainable text updates with fine-grained instructions. XATU is a diverse benchmark covering a wide range of topics and text types and leverages high-quality data sources from various existing sources. We compare existing open and closed instruction-tuned language models under both the zero-shot and fine-tuning settings and reveal their capabilities to edit text and follow instructions. By releasing the benchmark to the community, we hope to stimulate further research in developing instruction-based text editing models and even potentially interactive text generation systems.

## 6. Acknowledgements

We thank the anonymous reviewers for their valuable reviews. Additionally, we appreciate Tom Mitchell and the entire team at Megagon Labs for insightful discussions.

## 7. Bibliographical References

Afra Feyza Akyürek, Ekin Akyürek, Aman Madaan, Ashwin Kalyan, Peter Clark, Derry Wijaya, and Niket Tandon. 2023. Rl4f: Generating natural language feedback with reinforcement learning for repairing model outputs. *arXiv preprint arXiv:2305.08844*.

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

Talita Anthonio, Irshad Bhat, and Michael Roth. 2020. wikiHowToImprove: A resource and analyses on edits in instructional texts. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5721–5729, Marseille, France. European Language Resources Association.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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.

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

Chrysanne DiMarco and Graeme Hirst. 1993. A computational theory of goal-directed style in syntax. *Computational Linguistics*, 19(3):451–500.

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, Dublin, Ireland. Association for Computational Linguistics.

Jane Dwivedi-Yu, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, and Fabio Petroni. 2022. Editeval: An instruction-based benchmark for text improvements. *arXiv preprint arXiv:2209.13331*.

Felix Faltings, Michel Galley, Gerold Hintz, Chris Brockett, Chris Quirk, Jianfeng Gao, and Bill Dolan. 2021. Text editing by command. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5274, Online. Association for Computational Linguistics.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 305–315, Brussels, Belgium. Association for Computational Linguistics.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2023. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1935–1952, Toronto, Canada. Association for Computational Linguistics.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Hayate Iso, Chao Qiao, and Hang Li. 2020. Fact-based Text Editing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 171–182, Online. Association for Computational Linguistics.

Robert Iv, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. 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*, pages 3670–3686, Seattle, United States. Association for Computational Linguistics.

Zizhong Li, Haopeng Zhang, and Jiawei Zhang. 2023. A revisit of fake news dataset with augmented fact-checking by chatgpt. *arXiv preprint arXiv:2312.11870*.

Yixin Liu, Budhaditya Deb, Milagro Teruel, Aaron Halfaker, Dragomir Radev, and Ahmed H Awadallah. 2022. On improving summarization factual consistency from natural language feedback. *arXiv preprint arXiv:2212.09968*.

Yiwei Lyu, Paul Pu Liang, Hai Pham, Eduard Hovy, Barnabás Póczos, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2021. StylePTB: A compositional benchmark for fine-grained controllable text style transfer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2116–2138, Online. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

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, Valencia, Spain. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

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*.

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.

Machel Reid and Graham Neubig. 2022. Learning to model editing processes. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3822–3832, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Timo Schick, Jane A. Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2023. PEER: A collaborative language model. In *The Eleventh International Conference on Learning Representations*.

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, Seattle, United States. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2023. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*.

James Thorne and Andreas Vlachos. 2021. Evidence-based factual error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3298–3309, Online. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Veniamin Veselovsky, Manoel Horta Ribeiro, and Robert West. 2023. Artificial artificial artificial intelligence: Crowd workers widely use large language models for text production tasks.

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

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.

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.

Pengcheng Yin, Graham Neubig, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. 2019. Learning to represent edits. In *International Conference on Learning Representations*.

J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: How non-ai experts try (and fail) to design llm prompts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023a. Extractive summarization via ChatGPT for faithful summary generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3270–3278, Singapore. Association for Computational Linguistics.

Haopeng Zhang, Xiao Liu, and Jiawei Zhang. 2023b. SummIt: Iterative text summarization via ChatGPT. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages

10644–10657, Singapore. Association for Computational Linguistics.

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.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium. Association for Computational Linguistics.

Yang Zhong. 2021. *WIKIBIAS: Detecting Multi-Span Subjective Biases in Language*. Ph.D. thesis, The Ohio State University.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023a. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

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

## A. Data Annotation

The human annotations were conducted on the Appen[5] platform, an example user interface is shown in Figure 6.

The prompts used to generate the explanation are:

**Prompts to generate fine-grained instruction**

```
[no prose]

# Task:
Your task is to write a detailed instruction
that enables the AI assistant to edit the
original text into a revised text based on
the references.  The instruction must not
cause information leakage about the revised
text.

# Original text:
{{input}}

# Reference:
{{reference}}

# Revised text:
{{output}}

# Instruction:
```

**Prompts to generate explanation**

```
[no prose]

# Task:
Your task is to provide a two-sentence ex-
planation of the edits made based on the
instruction and reference by comparing the
original and revised texts.

# Instruction:
{{instruction}}

# Original text:
{{input}}

# Reference:
{{reference}}

# Revised text:
{{output}}

# Explanation:
```

## B. Implementation Details

The detailed instruction fine-tuning parameters are summarized in Table 6. We use LoRA (Hu et al., 2022) to instruction fine-tune all models for $15$ epochs with $200$ additional training examples at a learning rate of $1e-5$.

The prompts used to generate the text editing and fine-tuning are:

[5] https://appen.com/

| | |
|---|---|
| $\texttt{lora}_r$ | 8 |
| $\texttt{lora}_\alpha$ | 32 |
| num_epochs | 15 |
| lr | 1e-5 |
| $\texttt{lora}_{\texttt{dropout}}$ | 0.1 |
| seed | 634 |
| max_length | 1024 |

Table 6: Hyper-parameters used in the fine-tuning experiments.

**Prompts for Text Editing**

```
[no prose]

Below is an instruction that describes a
task, along with an input text paired with a
reference and an explanation that provides
further context.  Please edit the input text
based on the instructions, the reference,
and the explanation.  Your response should
only include the edited output.

# Instruction:
{{instruction}}

# Input:
{{input}}

# Reference:
{{reference}}

# Explanation:
{{explanation}}

# Response:
```

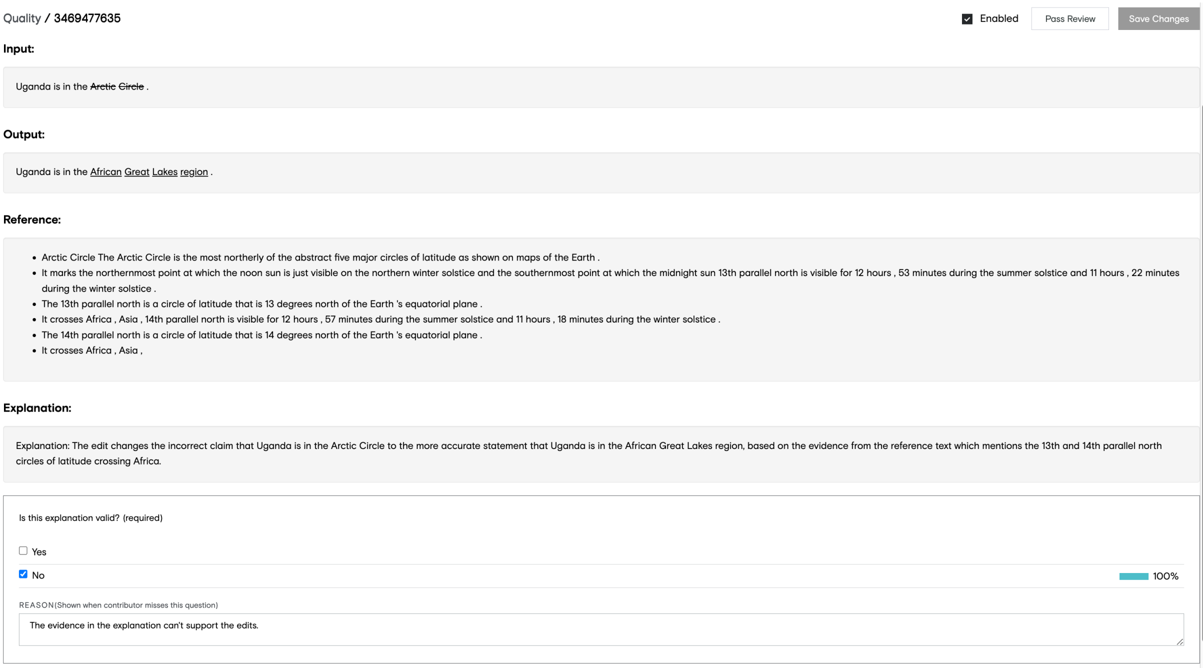


Figure 6: The interface used for annotation.