Edisum: Summarizing and Explaining Wikipedia Edits at Scale

Anonymous ACL submission

Abstract

An *edit summary* is a succinct comment written by a Wikipedia editor explaining the nature of, and reasons for, an edit to a Wikipedia page. Edit summaries are crucial for maintaining the encyclopedia: they are the first thing seen by content moderators and they help them decide whether to accept or reject an edit. Additionally, edit summaries constitute a valuable data source for researchers. Unfortunately, as we show, for many edits, summaries are either missing or incomplete. To overcome this problem and help editors write useful edit summaries, we propose a model for recommending edit summaries generated by a language model trained to produce good edit summaries given the representation of an edit diff. To overcome the challenges of mixed-quality training data and efficiency requirements imposed by the scale of Wikipedia, we fine-tune a small generative language model on a curated mix of human and synthetic data. Our model performs on par with human editors. Commercial large language models are able to solve this task better than human editors, but are not well suited for Wikipedia, while open-source ones fail on this task. More broadly, we showcase how language modeling technology can be used to support humans in maintaining one of the largest and most visible projects on the Web.

1 Introduction

Wikipedia is the largest online encyclopedia, housing 60 million articles in over 300 languages, with the English Wikipedia alone featuring 6.7 million entries. It is edited collaboratively, meaning that anyone can be an editor to most of the articles, resulting in massive numbers of edits performed continuously; e.g., on English Wikipedia alone, over 3 million edits are performed each month (Wikipedia, 2024d). When performing an edit, the editor can leave an *edit summary* (example in Fig. 1), a short comment explaining the content of the edit and, sometimes, a reason why the edit was performed. It is often the first source of information about an edit that editors see when browsing edit histories for content moderation or other purposes and is an opportunity for an editor to justify their changes.

Figure 1: An example of an edit diff. The + and – signs denote the text that was added and removed, respectively. The edit summary is the text in green in the screenshot.

Edit summaries are also valuable to researchers. They provide important insights into editor roles and actions on Wikipedia (Geiger and Ribes, 2010; Arazy et al., 2016; Wattenberg et al., 2007). They are used for building datasets for various purposes, such as the detection of low-quality Wikipedia content (Asthana et al., 2021) or detecting conflicts (Sumi et al., 2011). Edits and edit summaries have also been used to build datasets for iterative text generation, due to their incremental nature (Schick et al., 2022; Faltings et al., 2021).

Despite being a valuable asset, edit summaries have a number of drawbacks that prevent them from being used more efficiently. Many users leave them blank when performing an edit. Even when provided, summaries can be misleading—and not necessarily deliberately (as opposed to vandalism). Some editors also use canned edit summaries (Wikipedia, 2024b), to quickly insert commonly used summaries in the current Wikipedia space. For instance, these can be edit summaries such as "Added links" or "Fixed typo". They are not intentionally misleading, but frequently do not reflect the content of the edit precisely. Although it is hard to miti-

gate the effects of vandalism on edit summaries, our analyses show that a large fraction of edits would benefit from a more specific, tailored summary. This is currently an unexplored area within research, with no previous attempts to automatically generate Wikipedia edit summaries. Given their performance on text generating tasks, generative language models arise as a promising solution.

Generating Wikipedia edit summaries is a tricky problem for several reasons. Although blank edit summaries are easy to detect, there are no established heuristics for determining whether an edit summary is a good description of its edit or not. This can lead to mixed-quality data for training a model, and, consequently, poor performance at deployment time. Furthermore, edit summaries should ideally explain why the edit was performed, along with what was changed, which often requires external context. From an engineering perspective, it is also not trivial to design an appropriate prompt for this task, in particular because most generative models work with a small context size. Finally, even though LLMs are promising candidates for automatic edit summary generation in theory, platforms like Wikipedia often have guiding principles which limit them to the usage of open-source technology (Wikimedia, 2024b), which limits their use of commercial LLMs such as OpenAI models.

In this paper, we perform a detailed qualitative analysis of edit summaries, uncovering some of the drawbacks of human-written ones. We show that this task can be solved with LLMs. Based on these results, we carefully select a high-quality subset of edits and edit summaries. For a subset of the edits lacking summaries, we generate edit summaries using an LLM. Due to efficiency and input-size constraints, we then fine-tune a range of smaller generative language models with longer context size, built on LongT5 (Guo et al., 2022), which we call *Edisum*. We use mix of editor-provided and synthetic data, using a representation of edit diffs as inputs. This approach balances providing sufficient context for most edits while remaining scalable for platforms like Wikipedia.

Results. We compare our solution to two baselines, human editors[1] and the far more resource-heavy LLMs (GPT-4, GPT-3.5 and Llama 3 8B), via both automatic and human evaluation. Our results indicate that commercial LLMs (GPT-4 and GPT-3.5) outperform both open-source LLM (Llama 3 8B) and human editors, while Edisum trained on synthetic data matches human editors' performance, offering an ideal solution for a large-scale application on Wikipedia.

Contributions. In short, our contributions are the following:

(i) We perform a comprehensive qualitative analysis of the existing Wikipedia edit summaries, which shows that many existing edit summaries have hard-to-detect flaws.

(ii) We show that edit summary generation is solvable using high-performance LLMs.

(iii) We show that *Edisum*, which is, to the best of our knowledge, the first solution to automate the generation of highly-contextual Wikipedia edit summaries at large scale, achieves performance similar to the human editors

(iv) We release the dataset consisting of cleaned edit summaries and synthetically generated data for future research. The code can be found at `[anonymous]`.

2 Related work

Wikipedia edit summaries. Edit summaries play an important role on Wikipedia in helping patrollers quickly monitor edits for vandalism or otherwise problematic edits (Wikimedia, 2024a). They are simpler and easier to scan than the edit diffs, and thus are important for enabling fast patrolling of content on Wikipedia (Morgan, 2019). Despite this, we are not aware of work that focuses on helping editors to improve edit summaries.

One related task that was studied more is git commit message generation. While this area is well studies, with many rule-based approaches (Buse and Weimer, 2010; Cortes et al., 2014), retrieval approaches (Huang et al., 2020), learning-based approaches (Jung, 2021; Nie et al., 2021; Loyola et al., 2017), or even an attempt to solve the task with LLMs (Lopes et al., 2024), the difference lies in the data. Code and textual data have many differences, with the most notable one for our problem being the lack of highly structured text that exists in the code. Wikipedia edits and edit summaries have also higher variety in the topics they cover, as well as style they are written with.

Edit summaries have been used extensively, however, to understand and model behavior on Wikipedia. Researchers who utilize edit summaries occa-

[1] We envision that our model could provide a recommended edit summary to human editors, simplifying the process of writing it and encouraging uptake.

sionally comment on anecdotal patterns in usage, but descriptive statistics are minimal. Panciera et al. (2009) describe the usage of links to Wikipedia policy pages in edit summaries, showing that the likelihood of invoking a policy increases with editor experience. Wattenberg et al. (2007) convert edit summaries into colors to visualize how different editors approach tasks on Wikipedia and Geiger and Ribes (2011) describe the importance of edit summaries in tracing activity on Wikipedia for understanding bots and vandalism, while Stvilia et al. (2008) point out that edit summaries are often blank or misleading, rendering them less useful. Multiple works (Yang et al., 2017; Pavalanathan et al., 2018; Asthana et al., 2021) construct datasets of edits for training models by filtering edits based on certain keywords in the edit summaries. Notably, Yang et al. (2017) classify edits based on their intention, including labels commonly found in edit summaries, such as "clarification". In contrast to their multi-label classification method, we opt for a more flexible generative language model approach.

Synthetic data generation. Early approaches using generative models to produce synthetic data focused on finetuning a pretrained model which is then used as a generator (Anaby-Tavor et al., 2020; Papanikolaou and Pierleoni, 2020; Mohapatra et al., 2020; Kumar et al., 2020). This requires an existing dataset for finetuning the generator. Recently, the focus has shifted on unsupervised methods for synthetic data generation using pretrained language models (PLMs). These methods do not require lengthy and expensive labeling. One such example is the work by Wang et al. (2021), in which they generate synthetic labels by using only unlabeled examples sent to the LLM. There have been several attempts to generate data for different natural language processing (NLP) tasks by carefully designing prompts to the PLMs. This includes work by Ye et al. (2022) and Gao et al. (2022) in which they evaluate this procedure on text classification, question answering, and natural language inference tasks. Similarly, Meng et al. (2022) do this for GLUE (Wang et al., 2018) tasks. There have been successful attempts to use synthetic data generated in this way for intent classification (Sahu et al., 2022), and question answering (Li et al., 2022).

There are also examples of synthetic data generation for more tailored purposes. Shao et al. (2023) use the synthetic data as demonstrations to improve the propmting of LLMs. Additionally, synthetic data has been used to solve tasks that LLMs cannot directly solve, such as closed information extraction (Josifoski et al., 2023). Our task is not a standard NLP task, such as text classification or summarization, but can still be seen as a text generation task. As such, it is likely that LLMs can solve it with careful prompting, enabling synthetic data generation for training a more efficient system suitable for large-scale use.

3 Qualitative analysis of Wikipedia edit summaries

Given the dearth of data on the nature and quality of edit summaries on Wikipedia, we perform qualitative coding to guide our modeling decisions. Specifically, we analyze a sample of 100 random edits made in August 2023 to English Wikipedia stratified among a diverse set of editor expertise levels. Two of the authors each coded all 100 summaries and we report the results in Table 1. Since there were only two coders, we report the range for each category instead of the majority label. The lower bound indicates both annotators marked the category, and the upper bound indicates at least one did. Edit summaries were coded by following criteria set by the English Wikipedia community (Wikimedia, 2024a) (see Table 1). For more details on the annotation process, see Appendix A.

Overall, we see a relatively high annotator agreement. Lower Cohen's kappa for some categories indicates that these judgements can be difficult and highly subjective. The vast majority (∼80%) of current edit summaries focus on "what" of the edit, with only 30–40% addressing the "why". This aligns with the raters' judgement of what a language model can generate from the edit diff alone (see columns "Generate-able (what)" and "Generate-able (why)" in Table 1). Accurately describing the "why" requires external context that the model lacks, such as information about sources added or world events.

A sizeable minority (∼35%) of edit summaries were labeled as "misleading", generally due to overly vague summaries or summaries that only mention part of the edit. This makes training on this data challenging. Almost no edit summaries are inappropriate, likely because highly inappropriate edit summaries would be deleted (Wikipedia, 2024c) by administrators and not appear in our dataset. This suggests that it is unlikely for a model trained on edit summaries to learn to suggest in-

Metric	Summary (what)	Explain (why)	Misleading	Inappropriate	Generate-able (what)	Generate-able (why)
Description	Attempts to describe what the edit did. For example, "added links"	Attempts to describe why the edit was made. For example, "Edited for brevity and easier reading".	Overly vague or misleading per English Wikipedia guidance. For example, "updated" without explaining what was updated is too vague.	Could be perceived as inappropriate or uncivil per English Wikipedia guidance.	Could a language model feasibly describe the "what" of this edit based solely on the edit diff.	Could a language model feasibly describe the "why" of this edit based solely on the edit diff.
% Agreement	0.89	0.8	0.77	0.98	0.97	0.8
Cohen's Kappa	0.65	0.57	0.50	-0.01	0.39	0.32
Overall (n=100)	0.75 - 0.86	0.26 - 0.46	0.23 - 0.46	0.00 - 0.02	0.96 - 0.99	0.08 - 0.28
IP editors (n=25)	0.76 - 0.88	0.20 - 0.44	0.40 - 0.64	0.00 - 0.08	0.92 - 0.96	0.04 - 0.16
Newcomers (n=25)	0.76 - 0.84	0.36 - 0.48	0.24 - 0.52	0.00 - 0.00	0.92 - 1.00	0.12 - 0.20
Mid-experienced (n=25)	0.76 - 0.88	0.28 - 0.52	0.16 - 0.36	0.00 - 0.00	1.00 - 1.00	0.08 - 0.28
Experienced (n=25)	0.72 - 0.84	0.20 - 0.40	0.12 - 0.32	0.00 - 0.00	1.00 - 1.00	0.08 - 0.48

Table 1: Statistics on agreement for qualitative coding for each facet and the proportion of how many edit summaries met each criteria. Ranges are a lower bound (both of the coders marked an edit) and an upper bound (at least one of the coders marked an edit). The majority of summaries are expressing only what was done in the edit, which we also expect a language model to do. A significant portion of edits is of low quality, i.e., misleading.

appropriate summaries and thus we do no further filtering of summaries for inappropriate language.

4 Method

4.1 Synthetic data generation

From the analysis in Sec. 3, we notice there is a considerable number of lower quality edits, which are not easily detectable. At the same time, as LLMs perform well for a wide variety of tasks, often in a few-shot setting, we expect them to generate a good quality edit summary after some prompt tuning for majority of the edits, including what was done, but also why the edit was performed when obvious from the context. Our initial exploration on GPT-4 and GPT-3.5 confirms these assumptions. Our idea is not to just prompt LLMs to solve the task, but to rather generate synthetic data which will be used to tune a more efficient model.

LLM. After experimenting with available OpenAI models (OpenAI, 2024), we opt for gpt-3.5-turbo model with 4k token context as a good compromise between price and quality of the results. This model is optimized for the dialogue setting. We prompt it by sending the explanation of what an edit summary is as a system prompt, while the demonstrations are presented as alternating dialogue turns by the user (edit diff) and the model (edit summary).

Generating useful synthetic training data requires an LLM that can already solve the task of automated edit summary generation—the very task we set out to solve—, which might seem to defeat the purpose of this paper. We hence emphasize that commercial LLMs are not well suited for this task, as they do not follow the open-source guidelines set by Wikipedia (Wikimedia, 2024b). In addition, we envision this model as an assistant to the editors, meaning that it should run virtually in real-time. Given the low number of GPUs Wikipedia has access to (Wikitech, 2024), ideally, our model should be fairly small to fulfil the real-time constraints.

Prompt construction. We settle on the five-shot setting, instructing the LLM to only explain what was done in the edit, as the reason why the edit was performed is often too difficult to infer from the context. Nonetheless, we observe that LLMs often generate the reason organically where it is appropriate.[2] The examples of edits with good summaries, represented with the edit diff between the revision immediately before vs. after the edit, are used as demonstrations (see Fig. 1). The edit diff is much shorter than the full revisions, which makes it easier to fit our prompt into the length constraints imposed by the LLM. Additionally, the edit diff provides rich information about what was performed during the edit, omitting a large amount of text that was irrelevant for the edit. For more details on the prompt tuning and quality check of generated data, see Appendix C.

4.2 Data cleaning and collection

We filter Wikipedia data for training the models with two aspects in mind. First, ***edit summaries for certain types of edits are trivial***. For example, HotCat, a tool that many editors use to change categories on a page, automatically generates reasonable summaries via heuristics (e.g. "added Category:Shoegazing musical groups using HotCat",[3]). Based on this, we focus on edits altering the text of the article, where heuristics struggle and a language model would be well-suited. Second, ***existing edit summaries are of mixed quality***, which is reflected in the qualitative coding described in Section 3. This is most salient in IP editors, and, to a lesser degree, new editors. In this context, we exclude the following edits:

[2] For instance, for the edit https://en.wikipedia.org/w/index.php?diff=1172890678 GPT-4 will generate "Removed unnecessary quotation marks around the name Claudia.", hinting that the edit was performed because the quotation marks were unnecessary.

[3] https://en.wikipedia.org/w/index.php?diff=1033805631

(i) **Edits which did not change**, insert, or remove at least one **sentence** in the article.

(ii) **Edits with auto-generated summaries** by Mediawiki software (Wikipedia, 2024a).

(iii) **Edits made by bots**, which often have very good edit summaries, but it is not useful to have a language model learn edit summaries that have already been hard-coded into a bot.

(iv) **Reverted edits**, as many of them are vandalism and unlikely to have a useful summary.

(v) **Edits that made the revert** to previous edits, as these often talk about reason why the revert was performed. These reasons are usually external and are not easy to infer from the edit diff, and thus are difficult to be generated by a language model.

(vi) **Edits with blank edit summaries**, as all edits should have a basic edit summary. Many edits have an indicator of which section of the article they affected, which we removed from all edits as well, so it does not affect our checking of whether the summary is blank.

We also annotate the edit summaries with the various metadata (e.g. length) to enable further filtering or balancing of our edit summary sample (see Appendix B).

4.3 Model

Since 4.6% of our data requires input size longer than 512 tokens used by standard small generative models (Chung et al., 2022; Lewis et al., 2019), as the model to finetune, we use LongT5 (Guo et al., 2022), which has the ability to work with longer context windows. We denote each finetuned model as Edisum[S%], where S is a percentage of synthetic data in the training set. We intentionally use a very small model, because of limitations of Wikipedia's infrastructure. In particular, Wikipedia does not have access to many GPUs on which we could deploy big models (Wikitech, 2024), meaning that we have to focus on the ones that can run effectively on CPUs. Note that this task requires a model running virtually in real-time, as edit summaries should be created when edit is performed, and cannot be precalculated to decrease the latency. Models of similar size have already been successfully implemented in Wikipedia applications. For details on implementation, see Appendix E.

Because this is an unexplored area, with no previous attempts to automatize the generation of Wikipedia edit summaries, there is no apparent baseline to compare against. We thus directly compare our method to the actual ground-truth data: edit summaries written by Wikipedia editors. In addition to that, we evaluate how close our model is to LLMs. We evaluate GPT-4 and GPT-3.5, which we used to generate synthetic data. Additionally, we evaluate an open-source alternative LLM of a reasonable size, Llama 3 8B (AI@Meta, 2024). We ran all LLMs on 500 randomly chosen edits from the test data and they were all prompted with the same prompt used for synthetic data generation, and with the generation parameters from Table 3.

5 Experimental setup

5.1 Data

We use edits made in August 2023 to articles on English Wikipedia. This includes over 500K edits without a summary, from which we randomly take 100k edits to generate synthetic data. After the initial filtering from Sec. 4.2, we are left with ∼600K edits. We additionally limit the data by filtering edits with summaries longer than 200 and shorter than 5 characters. We leave out edits from the editors who have made less than 30 edits and keep at most 3 edits with the same summary to enforce diversity. This leaves us with ∼127K samples. For experiments, we combine data obtained in both ways: from existing Wikipedia edit summaries, and by generating synthetic data. We use in total 100K samples for training, and 10K for validation. The rest is used for testing (∼17K samples).

We run experiments with 5 different proportions of synthetic data in the training set (0%, 25%, 50%, 75%, 100%), by choosing the synthetic and human editor's data randomly from the collected datasets. As input to the model, we use the edit diff between the two revisions of the article in question to keep the input short while preserving the most important information. We extract the difference and represent the input in the same manner as in Sec. 4.1. To separate the text from the old and the new revision, we use <old_text> and <new_text> prefixes. Each sentence is separated by <sent_sep> prefix. We filter out data points with inputs longer than 1,024 tokens for convenience (only 2.3% of our data is longer than that). As the output, we use the (human- or synthetically generated) edit summary. For an example of the constructed input, see Appendix D.

5.2 Evaluation

We perform a twofold evaluation: (1) a cheap and fast-to-conduct automatic evaluation in which we compare auto-generated summaries to the human-

written ones (ground truth); and (2) an expensive and slower-to-conduct human evaluation, where human raters compare auto-generated to human-written edit summaries. In the former case, the best a model can do is reproduce a human-written summary, whereas in the latter case, a model can in principle outperform humans on this task.

Automatic evaluation. For automatic evaluation, we use MoverScore (Zhao et al., 2019), designed for measuring the semantic similarity between two texts. It takes values from 0 to 1 (larger is better), and correlates better with human judgement than token-matching metrics such as BLEU (Papineni et al., 2002) or ROUGE (Lin, 2004). This is especially important in settings similar to ours, where many good outputs with different phrasing may be equally appropriate. To evaluate a single Edisum model or a single LLM, for each edit, we take edit diff, generate the automatic summary with it, and calculate the MoverScore by comparing it to the existing summary. We obtain the measure of quality for the current Edisum model by averaging this measure over the whole dataset. For a reference, we also provide ROUGE and BERT scores obtained in the same way in Appendix J.

Human evaluation. Although data cleaning increases the overall quality of the edit summaries we consider, some of them are still misleading or incorrect, as we do not have a good heuristic to detect this. Yet, MoverScores are obtained by comparing to those existing edit summaries, which can result in scores that have little to no meaning. To surpass this limitation, we perform a human evaluation. We compare our best-performing model according to the MoverScore (cf. Sec. 6.1), Edisum[100%] (trained fully on synthetic data), with summaries written by editors and GPT-4 (highest performing model from Sec. 6.1). To inspect the effect of synthetic data on training, we also evaluate Edisum[0%], trained only on existing data.

We randomly select 100 samples from the testing dataset to perform this evaluation, from which we discard one sample without a good edit summary option. Each sample corresponds to a Wikipedia edit, and is associated with a web page of the edit diff between two revisions[4]. For each sample, annotators are presented with four edit summaries in random order, to prevent bias: ground truth, Edisum[100%], Edisum[0%], and GPT-4 summary. They are asked to choose the best and the worst summary, because it is often difficult to rank all four summaries, as some of them are very similar or convey the exact same information, in which case the preference would only come down to the style of the summary (see Table 4 for examples). The task can be seen as ranking with ties, where the two summaries that were not chosen as neither are tied for the second place. Since this is not a simple task, to ensure high-quality results, instead of relying on the crowdsourcing platforms, we recruited 3 MSc students to perform the annotation. Conflicts were resolved by one of the authors of the paper. To measure the agreement between the annotators, we report Kendall rank correlation coefficient (Kendall's τ) between each pair of them. As our annotation task can be seen as a ranking task, we choose this as a suitable measure. For more details on annotation task, see Appendix G.

6 Results

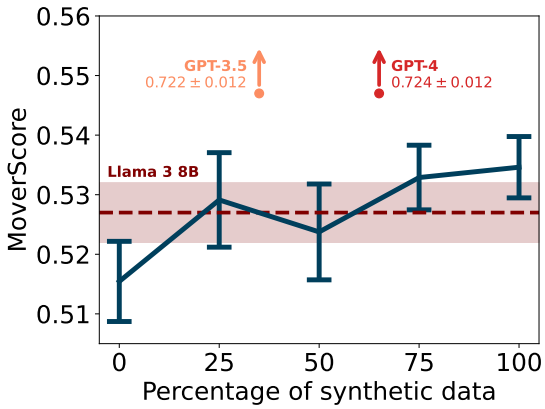

Figure 2: Results of Edisum evaluation with MoverScore. Error bars are 95% confidence intervals (CIs). GPT-4 and GPT-3.5 perform better than Edisum, with the average MoverScore of 0.724 and 0.722, respectively. We do not show the performance of GPT-4 and GPT-3.5 credibly on y-axis for convenience, as their performance is susbstantially higher than for the other models. Note that both of these are shown as a dot on the plot, as there is no notion of the percentage of synthetic data in the training set for these models.

6.1 Automatic evaluation

In Fig. 2, we present the results of automatic evaluation. Performance of all Edisum models is decent, according to the MoverScore. Edisum[0%] performs worse than the models with some fraction of synthetic data, in particular Edisum[75%] and

[4] e.g., https://en.wikipedia.org/w/index.php?title=Albert_Einstein&diff=prev&oldid=1177682587; see Fig. 1 for a visual example

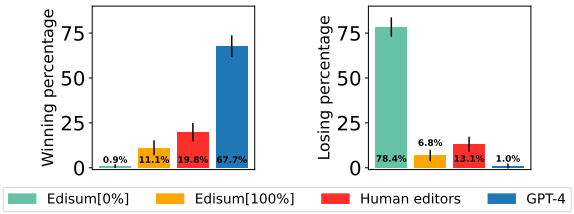

Figure 3: Results of human evaluation. *Left:* % of time summaries from each method are chosen as the best. *Right:* % of time summaries from each method are chosen as the worst. Error bars are 95% confidence intervals (CIs).

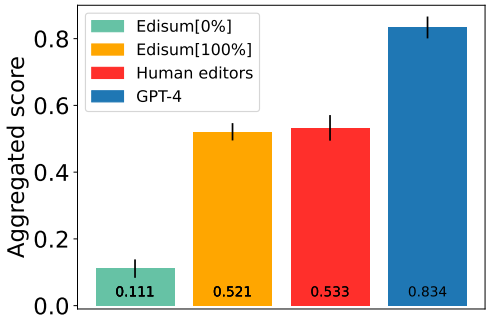

Figure 4: Average aggregated scores of human evaluation. Each method was scored with 1 point for winning, 0 points for losing, and 0.5 for neither winning nor losing. Error bars are 95% confidence intervals (CIs).

Edisum[100%], for which this difference is also statistically significant. This confirms our assumption that synthetic data is a useful asset when tackling the task of edit summary generation. One might be surprised that a fully-synthetic training set results in higher score when comparing to the existing data than the training set with only existing data, but this is not unexpected. Existing data has more structural variety and features various Wikipedia tags, which can be hard for a language model to pick up. Synthetic data might not have the same surface form as the existing data, but it expresses the key information about the edit while maintaining simpler structure, making it easier to train on.

When it comes to LLMs, as anticipated, the results show that commercial ones effectively solve this task, achieving scores higher than any of the Edisum models. The difference between GPT-4 and GPT-3.5 is small. We suspect this happens because we did not tune the prompt or generation parameters specifically to GPT-4. Further tuning can only improve the results, confirming the usefulness of these LLMs. On the other hand, the open-source LLM, Llama 3 8B, underperforms even when compared to the finetuned Edisum models. Given the limitations Wikipedia has on using only open-source software and their low performance on this task, as well as the need for this model to be fast and efficient, it is essential to have a smaller model that can do a decent job. This approach would also lower the costs of running such a system. For similar applications without such constraints, GPT-4 would be a reasonable option.

6.2 Human evaluation

Recall that in the human evaluation, for each edit, raters were asked to pick the best and the worst one out of four summaries, each generated by one of four methods: human editors, Edisum[0%], Edisum[100%], and GPT-4. The four candidate summaries for each edit were evaluated by three independent raters. Inter-rater agreement, measured in terms of Kendall's τ, was 0.588, 0.556, and 0.562 for the three pairs of raters, indicating a relatively strong positive agreement among the raters. In Fig. 3, we report the wins and losses separately. The left and right subfigure show the percentage of edits for which each method was chosen as the best and worst, respectively. GPT-4 is chosen the most often as the best model and the least often as the worst, while Edisum[0%] is the opposite. More importantly, the human editors and Edisum[100%] are tied on a middle ground, with the editors being chosen slightly more often as the best, but also as the worst, compared to Edisum[100%].

Since we did not let annotators compare the two middle options, to confirm our analysis, we fit a Plackett-Luce model, a generalization of the Bradley-Terry model (Bradley and Terry, 1952), intended to model ranking data (with the ability to handle ties, as in our setting). Briefly, this model assumes that there is a latent utility parameter associated with each option (in our case each method) and infers a maximum likelihood estimate from the empirically observed rankings (one ranking per human labeled data point). The higher the utility, the more preferred the option is. The results are presented in Appendix H, and they show no statistically significant difference between Edisum[100%] and editors. Moveover, we consider specifically those rankings where Edisum[100%] and human data are not tied (46 out of 99 samples). Edisum ranked higher 22 out of the 46 samples (vs. 24 for editors). This difference is not statistically significant (we ran a binomial test, with p-val = 0.883).

To compute a single performance score per method, we awarded a method a score of 1 if it

Method	What						Why		
	Correct	No change	Not specific	Unclear	Unexhaustive	Unrelated	Correct	Incorrect	Missing
Human editors									
Win	0.65 ± 0.07	0.00 ± 0.00	0.00 ± 0.00	0.05 ± 0.03	0.25 ± 0.06	0.05 ± 0.03	0.70 ± 0.06	0.00 ± 0.00	0.30 ± 0.06
Lose	0.15 ± 0.06	0.00 ± 0.00	0.15 ± 0.05	0.39 ± 0.06	0.23 ± 0.06	0.08 ± 0.03	0.23 ± 0.06	0.08 ± 0.03	0.69 ± 0.06
Neither	0.59 ± 0.06	0.00 ± 0.00	0.35 ± 0.07	0.06 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.53 ± 0.07	0.00 ± 0.00	0.47 ± 0.07
GPT-4									
Win	0.92 ± 0.03	0.00 ± 0.00	0.04 ± 0.03	0.00 ± 0.00	0.04 ± 0.03	0.00 ± 0.00	0.36 ± 0.07	0.00 ± 0.00	0.64 ± 0.06
Neither	0.40 ± 0.07	0.00 ± 0.00	0.04 ± 0.03	0.00 ± 0.00	0.32 ± 0.06	0.24 ± 0.06	0.48 ± 0.06	0.12 ± 0.04	0.40 ± 0.06
Edisum[100%]									
Win	0.63 ± 0.07	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.28 ± 0.06	0.09 ± 0.04	0.45 ± 0.06	0.00 ± 0.00	0.54 ± 0.07
Lose	0.00 ± 0.00	0.14 ± 0.05	0.00 ± 0.00	0.00 ± 0.00	0.14 ± 0.05	0.71 ± 0.06	0.14 ± 0.05	0.00 ± 0.00	0.86 ± 0.04
Neither	0.34 ± 0.06	0.00 ± 0.00	0.09 ± 0.04	0.06 ± 0.03	0.19 ± 0.05	0.31 ± 0.06	0.28 ± 0.06	0.16 ± 0.05	0.56 ± 0.07

Table 2: Error analysis results.

was chosen as the best one, 0 as the worst one, and 0.5 if it was not chosen as neither. In Fig. 4, we report the average score obtained by each method. In line with Fig. 3, we observe that GPT-4 scores best and Edisum[0%] scores worst, while the average scores of Edisum[100%] and editors are nearly identical and not statistically significantly different.

These results indicate that Edisum[100%] performs equally well as human editors, but with less variance: it achieves similar average ranking scores as the human editors (Fig. 4), while taking extreme positions less often than it (Fig. 3). Overall, results confirm the conclusions from the automatic evaluation. The positive effects of synthetic training data are even more evident here. Similarly, GPT-4 is again observed to generate edit summaries of the highest quality. However, as noted in Sec. 4.1 and Sec. 6.1, running such a system on a daily basis on a platform as big as Wikipedia for all the edits would not be feasible today. Our "distilled" Edisum[100%] model, which aims to mimic GPT's high-quality summaries, offers a fertile middle ground, performing as well as humans while being much smaller and cheaper to run.

6.3 Error analysis

To further examine the difference in performance between GPT-4 and other methods, we manually inspect 150 edit summaries from two perspectives: "why" (description of why the edit was performed) and "what" (description of what was done). The samples were chosen to cover all the cases (win, lose or neither) for all three methods. For details on the annotation procedure and taxonomy, see Appendix I. The results are presented in Table 2.

When observing "why" meta-category, we notice, as expected, that human written summaries express the correct reason why the edit was performed more often than the ones generated by GPT-4 or Edisum. However, both methods frequently express the, usually correct, reason. This reflects the edits for which reason can be inferred from the context. When it comes to the results for "what" category, the performance gap between GPT-4 and other methods is still visible. Specifically for Edisum, we can attribute the drop in performance to its size. Edisum is a very small model (∼220M parameters), incapable of fully capturing patterns present in more complex tasks, like edit summary generation. The distribution of errors for GPT-4 and Edisum for summaries that were not chosen as neither the best or worst is similar, with the most errors being unrelated or unexhaustive summaries. On the other hand, human editor's summaries from this category, as well as the ones chosen as the worst, tend to be less specific or unclear. Summaries that won were most often not exhaustive enough. Overall, while it should be used with caution due to a portion of unrelated summaries, the analysis confirms that Edisum is a useful option that can aid editors in writing edit summaries.

7 Conclusion

In this paper, we investigate the quality of Wikipedia edit summaries, i.e., short comments that editors write when performing changes in Wikipedia. These summaries serve a wide range of purposes in Wikipedia, but also for general research community. We find that a non-negligible number of them is of bad quality or missing. At the same time, we show that GPT-4 is able to solve this task better than human editors. To assist editors, we train a small language model that can, unlike GPT-4, effectively generate edit summaries on a large scale while matching the performance of human editors.

Limitations

While the overall results show that Edisum performs on par with human editors, there is still a space for improvement given that GPT-4 still outperforms our model. Additionally, the nature of the errors produced by Edisum and human editors is not the same. We leave it to the future research to explore the possibility of bridging the gap between a small generative model and a high-performing LLM and the impact different errors could have.

Our experiments show that models trained on synthetic data outperform those trained on existing edit summaries on Wikipedia, but this approach likely has limitations in learning editor community norms such as common abbreviations.

Additionally, our dataset might suffer from lack of diversity, and hence, our models might fail on more exotic edits. We limited our training samples to edit summaries by editors with at least 30 edits based on our qualitative analysis of existing edit summaries, but future work could explore additional strategies for producing a high-quality, diverse dataset of existing edit summaries. (Kocetkov et al., 2022) found significant improvements from applying near de-duplication to their code dataset and we suspect that many edits are quite similar with minor differences and a similar pipeline might bring improvements to this task as well.

References

AI@Meta. 2024. Llama 3 model card.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Ofer Arazy, Johannes Daxenberger, Hila Lifshitz-Assaf, Oded Nov, and Iryna Gurevych. 2016. Turbulent stability of emergent roles: The dualistic nature of self-organizing knowledge coproduction. *Information Systems Research*, 27(4):792–812.

Sumit Asthana, Sabrina Tobar Thommel, Aaron Lee Halfaker, and Nikola Banovic. 2021. Automatically labeling low quality content on wikipedia by leveraging patterns in editing behaviors. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–23.

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Raymond P.L. Buse and Westley R. Weimer. 2010. Automatically documenting program changes. In *Proceedings of the 25th IEEE/ACM International Conference on Automated Software Engineering*, ASE '10, page 33–42, New York, NY, USA. Association for Computing Machinery.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *Preprint*, arXiv:2210.11416.

Luis Cortes, Mario Linares-Vásquez, Jairo Aponte, and Denys Poshyvanyk. 2014. On automatically generating commit messages via summarization of source code changes.

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.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2022. Zerogen$^+$: Self-guided high-quality data generation in efficient zero-shot learning. *arXiv preprint*.

R. Stuart Geiger and David Ribes. 2010. The work of sustaining order in wikipedia: The banning of a vandal. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, CSCW '10, page 117–126, New York, NY, USA. Association for Computing Machinery.

R Stuart Geiger and David Ribes. 2011. Trace ethnography: Following coordination through documentary practices. In *2011 44th Hawaii international conference on system sciences*, pages 1–10. IEEE.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. Longt5: Efficient text-to-text transformer for long sequences. *Preprint*, arXiv:2112.07916.

Yuan Huang, Nan Jia, Hao-Jie Zhou, Xiang-Ping Chen, Zi-Bin Zheng, and Ming-Dong Tang. 2020. Learning human-written commit messages to document code changes. *Journal of Computer Science and Technology*, 35:1258–1277.

Martin Josifoski, Marija Sakota, Maxime Peyrard, and Robert West. 2023. Exploiting asymmetry for synthetic training data generation: Synthie and the case of information extraction. *Preprint*, arXiv:2303.04132.

Tae Hwan Jung. 2021. CommitBERT: Commit message generation using pre-trained programming language model. In *Proceedings of the 1st Workshop on Natural Language Processing for Programming (NLP4Prog 2021)*, pages 26–33, Online. Association for Computational Linguistics.

Denis Kocetkov, Raymond Li, LI Jia, Chenghao Mou, Yacine Jernite, Margaret Mitchell, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Dzmitry Bahdanau, et al. 2022. The stack: 3 tb of permissively licensed source code. *Transactions on Machine Learning Research*.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *Preprint*, arXiv:1910.13461.

Junlong Li, Zhuosheng Zhang, and Hai Zhao. 2022. Self-prompting large language models for open-domain qa. *arXiv preprint*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Cristina V. Lopes, Vanessa I. Klotzman, Iris Ma, and Iftekar Ahmed. 2024. Commit messages in the age of large language models. *Preprint*, arXiv:2401.17622.

Pablo Loyola, Edison Marrese-Taylor, and Yutaka Matsuo. 2017. A neural architecture for generating natural language descriptions from source code changes. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 287–292, Vancouver, Canada. Association for Computational Linguistics.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *arXiv preprint*.

Biswesh Mohapatra, Gaurav Pandey, Danish Contractor, and Sachindra Joshi. 2020. Simulated chats for building dialog systems: Learning to generate conversations from instructions. *arXiv preprint arXiv:2010.10216*.

Jonathan Morgan. 2019. Patrolling on wikipedia.

Lun Yiu Nie, Cuiyun Gao, Zhicong Zhong, Wai Lam, Yang Liu, and Zenglin Xu. 2021. Coregen: Contextualized code representation learning for commit message generation. *Preprint*, arXiv:2007.06934.

OpenAI. 2024. OpenAI models documentation. `https://platform.openai.com/docs/models`. Accessed: 2024-03-12.

Katherine Panciera, Aaron Halfaker, and Loren Terveen. 2009. Wikipedians are born, not made: a study of power editors on wikipedia. In *Proceedings of the 2009 ACM International Conference on Supporting Group Work*, pages 51–60.

Yannis Papanikolaou and Andrea Pierleoni. 2020. Dare: Data augmented relation extraction with gpt-2. *arXiv preprint arXiv:2004.13845*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Umashanthi Pavalanathan, Xiaochuang Han, and Jacob Eisenstein. 2018. Mind your pov: Convergence of articles and editors towards wikipedia's neutrality norm. *Proceedings of the ACM on Human-Computer Interaction*, 2(CSCW):1–23.

Gaurav Sahu, Pau Rodriguez, Issam H. Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. *arXiv preprint*.

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. *Preprint*, arXiv:2208.11663.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. *arXiv preprint*.

Besiki Stvilia, Michael B Twidale, Linda C Smith, and Les Gasser. 2008. Information quality work organization in wikipedia. *Journal of the American society for information science and technology*, 59(6):983–1001.

Robert Sumi, Taha Yasseri, Andr´s Rung, Andr´s Kornai, and J´nos Kertesz. 2011. Edit wars in wikipedia. In *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing*. IEEE.

Veniamin Veselovsky, Manoel Horta Ribeiro, and Robert West. 2023. Artificial artificial artificial intelligence: Crowd workers widely use large language models for text production tasks. *Preprint*, arXiv:2306.07899.

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. *arXiv preprint arXiv:1804.07461*.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *CoRR*, abs/2109.09193.

Martin Wattenberg, Fernanda B Viégas, and Katherine Hollenbach. 2007. Visualizing activity on wikipedia with chromograms. In *Human-Computer Interaction–INTERACT 2007: 11th IFIP TC 13 International Conference, Rio de Janeiro, Brazil, September 10-14, 2007, Proceedings, Part II 11*, pages 272–287. Springer.

Wikimedia. 2024a. Edit summary. https://meta.wikimedia.org/wiki/Help:Edit_summary. Accessed: 2024-03-12.

Wikimedia. 2024b. Wikimedia foundation guiding principles. https://foundation.wikimedia.org/wiki/Resolution:Wikimedia_Foundation_Guiding_Principles. Accessed: 2024-03-12.

Wikipedia. 2024a. Automatic edit summaries. https://en.wikipedia.org/wiki/Help:Automatic_edit_summaries. Accessed: 2024-03-12.

Wikipedia. 2024b. Canned edit summaries. https://en.wikipedia.org/wiki/Wikipedia:Canned_edit_summaries. Accessed: 2024-03-12.

Wikipedia. 2024c. Wikipedia revision deletion. https://en.wikipedia.org/wiki/Wikipedia:Revision_deletion. Accessed: 2024-03-12.

Wikipedia. 2024d. Wikipedia statistics on number of edits performed. https://stats.wikimedia.org/#/en.wikipedia.org/contributing/edits/normal|bar|2-year|editor_type~anonymous*group-bot*name-bot*user+(page_type)~content|monthly. Accessed: 2024-03-12.

Wikitech. 2024. Wikipedia: Access to gpus. https://wikitech.wikimedia.org/wiki/Machine_Learning/AMD_GPU#Do_we_have_Nvidia_GPUs. Accessed: 2024-03-12.

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.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. 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 563–578, Hong Kong, China.

A Qualitative analysis annotation process

Data sample. The qualitative analysis was performed on 100 samples, as annotation of edits is a lengthy process. To ensure a diverse enough group of edits, we stratify the sample based on the experience of editors. More precisely, we divide editors in four categories: IP editors (anonymous editors), newcomers (editors with < 10 edits), mid-experienced editors (10 - 1000 edits), and experienced editors (1000+ edits). We exclude edits by bots,[5] edits without summaries, and revert-related edits to focus on good-faith contributions to English Wikipedia. This means that our sample likely lacks highly-inappropriate comments, as they would be removed by the editors. Notably, 46% of the edits from August 2023 do not have a summary. The proportion varies greatly by editor type: 74% of edits for IP editors, 46% of edits for newcomers, 58% for medium-experienced editors, and 38% of edits for experienced editors. This highlights the value of better support for generating edit summaries.

Annotation. Annotation was done by the two authors of the paper. A discussion was held after the first ten summaries to ensure there was agreement on the codebook before completing the sample.

B Data annotation

We annotated the cleaned edit data as follows:

(i) **Frequency with which the edit summary appears** in our dataset. This enables us to sample the dataset to be more diverse. To check frequency, we lower-case all characters and replace any links with a generic link character before calculating frequency.

(ii) **Editor's edit count.** This enables us to upsample edits from more experienced editors, who are expected to be more likely to write correct edit summaries.

(iii) **Summary length**. While very short summaries are okay (e.g., "ce" is often used to stand for "copy-edit" and indicate small grammar or spelling changes), summaries are limited to 500 characters and the English Wikipedia community suggests to avoid unnecessarily long summaries.[6]

[5] Anecdotally, many bots actually have very good edit summaries generated by their code but also their edits are usually straightforward to describe.

[6] https://en.wikipedia.org/wiki/Help:Edit_summary

(iv) **User frequency** in the dataset. A small number of editors make a large proportion of edits on Wikipedia, and while they may write reasonable edit summaries, we want to learn from a diverse sample of Wikipedians.

(v) **Semi-automated edits.** If an edit is made through a tool that enables very quick editing or has several preset edit summaries, we flag this, as these edit summaries are unlikely to be strongly contextualized to the specific edit.

C Synthetic data generation process

Prompt choice. Experimentation process for choosing the prompt is done of 10 samples of edit diffs. For each one of them, we generate an edit summary with different prompts, and after manual inspection, we settle on the prompt that is used for synthetic data generation. We experiment with different instructions and different numbers of demonstrations, as well as their content.

For the instruction, as already mentioned in Sec. 4.1, we only focused on asking the LLM to explain what was performed in the edit. We also explained the format of the edit summary and the input, and gave a few guidelines to follow. For the full instructions, see Fig. 5.

For the demonstrations, as explained in Sec. 4.1 we provide the LLM with the edit diff between the two revisions immediately before vs. immediately after the edit. We extract this diff using the mwedittypes[7] library. From the output of this library, we can extract sentences that were added and removed in the editing process. We group all the removed sentences into "old text" and all the added sentences into "new text". On 100 randomly chosen and manually inspected edit diff outputs using this library, in 4 cases, these sentences are not ordered by the way they are appearing in the revision of the Wikipedia page. Because of that, we order the sentences in "old text" and "new text" alphabetically, to avoid confusion. We then represent the diff by concatenating both of those, separating them by stating "old text:" and "new text:" before each group. For an example of a demonstration, see Fig. 5.

When choosing the demonstrations, we make sure to include both longer and shorter edits in terms of content, and also edits with summaries of various length. We include both demonstrations for edits that add and remove content. We tried out different numbers of demonstrations ranging from 2 to 10 (2, 3, 5, 6, and 10 demonstrations), using same demonstrations for each edit summary generated. We settled on five demonstrations, which we found to provide us with sufficient information to generate high-quality data, while keeping the length of the input shorter, and consequently, the cost of the generation process smaller. The same five demonstrations are used for all the generated samples.

parameter	max_tokens	temperature	top-p	frequency_penalty	presence_penalty	stop	n	best_of
value	1000	0	1	0.2	0	"\n"	1	1

Table 3: Generation parameters used with gpt-3.5-turbo to generate synthetic data. These parameters were also used when testing GPT-4 and GPT-3.5 performance on the testing dataset.

Generation parameters and process. We decide to only work with edits that have textual changes and exclude the ones with changes in the Wiki markup, such as category modifications or templates. We do this because this is where we expect the language model to give us the biggest gains, as this is where the biggest variety of different edits are performed. We experimented with different generation parameters for the OpenAI models. In particular, we tried out different values of temperature, top-p, and frequency penalty. We make the decision on the best parameters manually, based on the same 10 samples we used for prompt construction. The set of parameters for the best-performing setup is displayed in Table 3.

Quality check. To verify that the quality of generated synthetic data is satisfying, we perform a quality check on 100 random edits, by comparing the data generated by GPT-3.5 with the existing one. We find that synthetic data has satisfying quality more often (87% vs. 78% of the time). On top of that, in 30% of the cases in which both summaries were seen as suitable, the generated one was chosen as better 30% of the time (vs 4% for the existing summaries).

D Model input

In Fig. 6, we showcase how the input to the model is constructed based on an edit diff.

[7] https://pypi.org/project/mwedittypes/

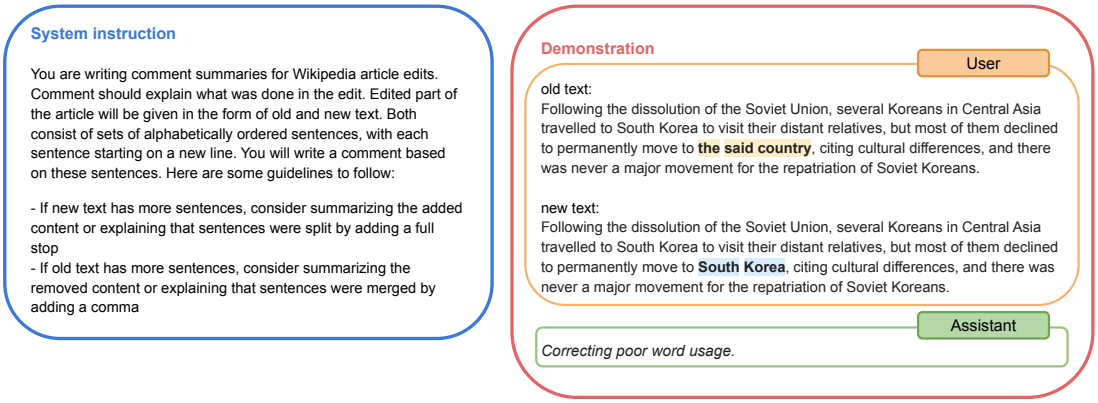

Figure 5: System instruction and the example of a demonstration used for synthetic data generation.

E Implementation hyperparamters and details

We used the long-t5-local-base[8] (∼220M parameters) as our base model which we then finetune on the collected training data (see above). The models were trained using the Adam optimizer with learning rate 3×10^{-4}, 0.1 gradient clipping on the Euclidean norm, and weight decay 0.05. They were trained for 10 epochs with batch size 2 and a polynomial learning-rate scheduler with 1,000 warm-up steps and a final learning rate of 3×10^{-5}. Training was performed on a single NVIDIA Titan X Maxwell 12GB GPU, taking around 30 hours for each model.

F Examples of ground truth and generated edit summaries

In Table 4, we present some of the existing edit summaries, as well as the ones generated with two of our models, Edisum[100%] and Edisum[0%], and GPT-4.

G Human evaluation setup

Annotation task. As mentioned in Sec. 5.2, human evaluation was done on 100 samples from the testing dataset, each associated with four edit summaries from different methods, a different Wikipedia edit, and presented with the corresponding web page. The sample size is relatively small as grading these edit summaries is a long and tedious process – each annotator has to manually assess the edit diff, and sometimes even the whole revisions of the article, in order to understand what was done in it. From the web page, we remove the element showing the actual human edit summary to make sure that the annotators are not aware of the existing edit summary. The web page also shows the "current" version of the article, right after the edit, in case the annotators need more context to give their judgement.

Annotators were asked to choose the best and the worst summary out of the four according to the following guidelines:

A good edit summary ***should***:

(1) Summarize what was done in the edit

(2) Cover all the changes performed (either explicitly or by adding something like "and misc")

(3) Be specific; e.g., a summary "I made some changes" is not specific

(4) Explain why the change was made, if it is unclear from the change itself

A good edit summary ***should not***:

(1) Use uncommon abbreviations

(2) Be too long: it is not supposed to be a paragraph, but a sentence-long summary

(3) Attack other editors' work or be aggressive

Annotators were provided with examples of good and bad edit summaries, with explanations what makes them good or bad. They were also given Wikipedia's manual of style[9] to get familiar with tags that often appear in edit summaries. Finally, to ensure the high-quality of the results, we train them on a few selected samples, teaching them what to look for in the edit summary and making sure they understood the guidelines and the assignment.

Annotators. As mentioned in Sec. 5.2, we recruit 3 MSc students as annotators for our task. We opt

[8] https://huggingface.co/google/long-t5-local-base

[9] https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style

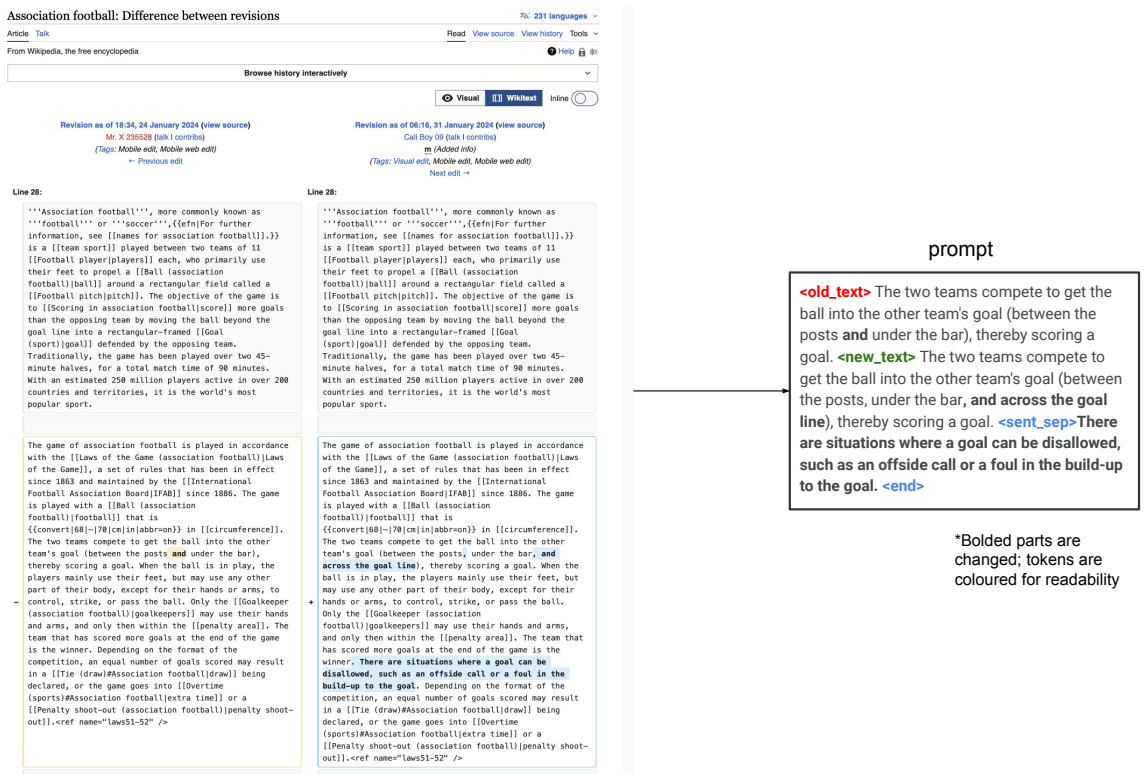

Figure 6: An example of input to the model constructed from an edit diff.

for this option over crowdsourcing platforms for several reasons, all of which ensure high-quality annotations. This task is not trivial for a person not familiar with Wikipedia's norms and rules. For instance, large fraction of edit summaries references Wikipedia's manual of style which might look like irrelevant words to a layperson. By recruiting students, we had more control over the quality of annotators we are taking. All of the annotators were MSc students in computer science, familiar with Wikipedia, but not with our work. On top of that, we had a more straightforward way to train MSc students for this task, as this might be a tricky thing to do with crowdsourcing platforms. Finally, another increasing concern that comes with the use of crowdsourcing platforms is the usage of LLMs by the crowd workers, who today frequently "outsource" text-processing tasks to LLMs to facilitate their work (Veselovsky et al., 2023).

The students were paid the equivalent of US$25 per hour for their work. Conflicts were resolved by one of the authors of the paper. To measure the agreement between the annotators, we report Kendall rank correlation coefficient (Kendall's τ) between each pair of them. Kendall's τ is a statistical measure used to assess the degree of association or correlation between two sets of rankings or ordinal data. As our task is a version of a ranking task, we opt to use this metric. For each pair of annotators, and for each sample with four summaries that are rated, we calculate Kendall's τ. To get the value of Kendall's τ for all 99 samples, we take the average. This way, we report three numbers, a value for each pair of annotators.

H Plackett-Luce model for ranking our methods

In Table 5, we present the obtained result of fitting a Plackett-Luce model to our ranking data from the human evaluation. Results show that Edisum[100%] performs similarly to the human editors, and even has a slight advantage over them. GPT-4 performs the best, while Edisum[0%] performs the worst. These are in line with our results from human evaluation.

I Annotation procedure for error analysis

Data. We annotate in total 150 samples, 50 for each method (GPT-4, Edisum[100%] and human editors). For each method, samples were chosen to have as similar as possible fractions of samples that won, lost or were not chosen as either. For GPT-4,

edit	human editor	Edisum [0%]	Edisum [100%]	GPT-4
https://en.wikipedia.org/w/index.php?diff=1172472704	added the range	added location	Added location information	Added information about the location and altitude of Boveys Lookout and the orientation of the Burton Range.
https://en.wikipedia.org/w/index.php?diff=1169223741	in 1982	Added date and citation.	Added the year of the speech	Added the year of Fidel Castro's speech.
https://en.wikipedia.org/w/index.php?diff=1169701032	added artist	Added a wikilink and a link	Added a new name to the list of artists	Added Stefan Brüggemann to the list of artists whose works are included.
https://en.wikipedia.org/w/index.php?diff=1170976899	lede; infobox	infobox (inapplicable); lede; -cats	Clarified the nature of Denis- Auguste Affre's cause and added information about his title	Updated the description of Denis-Auguste Affre's religious status and clarified the stage of his canonization process.
https://en.wikipedia.org/w/index.php?diff=1170699096	fixed link to aias awards, it was at 5th, not 7th	5th	Corrected the award number from 7th to 5th	Corrected the year of the Annual Interactive Achievement Awards nomination.
https://en.wikipedia.org/w/index.php?diff=1169516533	Amended lists so they are split at 100 m	""at least", not "at least""	Clarified the range of heights in the list	Added information about two different height categories for the ranking of buildings and structures in Greater Manchester.
https://en.wikipedia.org/w/index.php?diff=1173169263	UPD Romelu Lukaku	updated Lukaku	Removed the end of the season	Removed incorrect future date for Romelu Lukaku's tenure at Inter.
https://en.wikipedia.org/w/index.php?diff=1171641782	"This \"however\" doesn't make sense here"	"rmv deprecated, ambiguous "however" per MOS:RELTIME"	Removed unnecessary word	Removed the word "However," from the beginning of the sentence.
https://en.wikipedia.org/w/index.php?diff=1168487383	Per feedback given in GA review, elaborated on David King Udall and Ella Stewart Udall in the lead and in Ella Udall's first mention; David Udall was already identified	Added link; improved phrasing and punctuation	Added additional information about the individuals' professions	Added professional details for David King Udall, Ella Stewart Udall, and Mary Ann Linton Morgan Udall.

Table 4: Examples of summaries written by human editors and generated by Edisum and GPT-4.

Model	Edisum[0%]	Edisum[100%]	human editors	GPT-4
Parameter	0.072 $\pm$ 0.022	0.308 $\pm$ 0.019	0.276 $\pm$ 0.027	0.346 $\pm$ 0.023

Table 5: Parameters obtained for each method with Plackett-Luce model.

there was only 1 sample chosen as the worst, so we do not include this category in the analysis.

Annotation procedure. Samples were annotated according to two meta-categories: "what" (content of the edit) and "why" (reason for the edit). Taxonomies for each of the meta-categories ware derived after manual inspection of the subset of the outputs. For "why", we settle on three simple categories: missing, correct, and incorrect. For "what", we derived the following categories:

1. Correct: edit summary is fully correct and exhaustive
2. No change: the summary indicates that no change was performed
3. Not specific: the summary is not describing exact changes that were performed
4. Unclear: the summary seems to be pointing to the correct modifications, but is hard to understand without looking at the diff
5. Unexhaustive: the summary does not cover all changes performed
6. Unrelated: the summary describes unrelated edit

Method	BERT score	ROUGE-1	ROUGE-2	ROUGE-L
Edisum[0%]	0.803 ± 0.006	0.077 ± 0.022	0.026 ± 0.016	0.076 ± 0.020
Edisum[25%]	0.823 ± 0.006	0.101 ± 0.025	0.026 ± 0.014	0.076 ± 0.020
Edisum[50%]	0.820 ± 0.007	0.092 ± 0.020	0.020 ± 0.013	0.087 ± 0.019
Edisum[75%]	0.833 ± 0.005	0.094 ± 0.021	0.015 ± 0.009	0.087 ± 0.017
Edisum[100%]	0.833 ± 0.004	0.090 ± 0.017	0.012 ± 0.007	0.083 ± 0.017
GPT-3.5	0.836 ± 0.004	0.100 ± 0.017	0.017 ± 0.009	0.095 ± 0.015
GPT-4	0.837 ± 0.004	0.118 ± 0.016	0.025 ± 0.010	0.110 ± 0.016
Llama-3-8B	0.637 ± 0.045	0.031 ± 0.010	0.003 ± 0.003	0.029 ± 0.011

Table 6: Automatic evaluation with ROUGE and BERT score.

Annotation was done by one of the paper authors. The annotator chose one of the categories from the taxonomy for each of the presented edit summaries. To confirm the validity of the results, another author annotated 30 random samples. We calculated the agreement between the two annotators on those 30 samples using Cohen's kappa. For "what" meta-category, Cohen's kappa is 0.60, while for "why" it is 0.67. Both of these numbers indicate high overlap between the annotators.

J Additional automatic evaluation

In Table 6, we present ROUGE and BERT score for each evaluated model from Sec. 6.1. Results are mostly in line with MoverScore from the same section, confirming the superiority of GPT-4 for this task.