Assessing the Reliability of LLMs in Faithfully Updating Text

Anonymous ACL submission

Abstract

This paper addresses the challenge of faithfully representing updated information in text—a task formalized as the FRUIT problem (Iv et al., 2022). Given a source document and a set of evidences detailing updates, the goal is to generate an updated document that integrates new facts while preserving the original coherence and context. We first conduct a comprehensive analysis of the FRUIT dataset, uncovering key structural insights such as the observation that updated articles tend to be approximately 100 tokens longer than their originals, a factor that may bias models toward appending information rather than editing in place. Our study investigates the unsupervised capabilities of LLMs, including zero-shot learning, chain-of-thought reasoning, self-reflection, and evidence ordering, using both the open-source Llama-3-8b and the closed-source GPT-4 models. Our experiments reveal that a zero-shot setup yields the best performance, and that the format of evidences significantly impacts model outcomes, with table-based evidences outperforming unstructured text. These findings have important implications for domains requiring precise document updates, such as software engineering and technical documentation.

1 Introduction

Faithfully representing updated information in text presents significant challenges, necessitating a nuanced understanding of both existing documents and incoming updates. The task of updating documents requires human editors to sift through updated data to identify relevant content for incorporation, demanding a deep understanding of the source document and the new information. This time-consuming process especially with complex documents, emphasizes the need for automated systems to ensure accurate and reliable text updates. The need for such systems spans multiple domains, including software engineering, where requirement documents must be continuously updated throughout the software development lifecycle (Bhatia et al., 2020), and technical documentation, such as contracts and product manuals, which require precise updates for clarity and compliance (Kumar M et al., 2016). A task of Faithfully Reflecting Updated Information in Text (FRUIT) was introduced to tackle this problem (Iv et al., 2022). Given a source document at time t, A^t, and a set of evidences (updates pertaining to the source document) from time t to t', $\mathcal{E}^{t \rightarrow t'} = \{E_1, E_2, ...E_{|\mathcal{E}|}\}$, the task is to generate an updated document $A^{t'}$ incorporating the updates into the source document. This formulation not only requires the integration of new facts but also insists on maintaining the coherence and integrity of the original document. As such, the FRUIT task encapsulates both the challenge of accurate fact integration and the necessity of preserving contextual continuity.

Recent advances in large language models (LLMs) and in-context learning methods have shown promise in various text generation tasks. Studies encapsulate a wide range of text editting tasks like text simplification, paraphrasing, grammatical error correction along with FRUIT and instruction-tuned LLMs (Dwivedi-Yu et al., 2024; Shu et al., 2024). However, we identify several gaps in the current literature:

- There is a lack of in-depth analysis of the FRUIT dataset itself, including the various types of data it encompasses, such as text and tables, as well as the number of evidences utilized for updates.
- While the aforementioned studies focus on fine-tuning, there has been insufficient exploration of the unsupervised capabilities of LLMs, like zero-shot and chain-of-thought (CoT) prompting.
- Impact of various data types on performance of LLMs in this context remains underexplored, leaving a significant gap in understanding how these models interact with diverse data formats.

To this end, the contributions of this paper are:

• We perform a thorough and comprehensive analysis of the FRUIT dataset, to uncover its structure, content, and the intricacies of the data it contains. For instance, we observe that the updated articles tend to be longer (on average around 100 tokens longer) than the original articles, which could bias the models built using the data to append more information rather than efficiently editing in-place.

• We perform extensive experiments that assess the unsupervised behavior of LLMs through various methodologies, including zero-shot learning, CoT reasoning, self-reflection, and evidence ordering. We use two state-of-the-art LLMs – the open-source Llama-3-8b model and the closed-source GPT-4 model. To the best of our knowledge this is the first work that develop such methods for the FRUIT task.

• We conduct a comprehensive analysis, both quantitative and qualitative. We also analyse the effects of the different data formats present in the dataset. For instance, we find that the models perform better when the evidences/new information is present in the form of tables, rather than unstructured text.

• We make the implementations publicly available at https://anonymous.4open.science/r/faith-update-llm-1381/

2 Related Work

Text rewriting is an emerging field of research comprising of tasks including, but not limited to, text simplification, paraphrasing, style transfer and grammatical error correction etc. For each of these tasks, there are dedicated datasets – for instance, the ASSET corpus (Alva-Manchego et al., 2020) for text simplification, the STS benchmark (Cer et al., 2017) for text paraphrasing, WNC (Pryzant et al., 2020) for text neutralization, FRUIT (Iv et al., 2022) for text updation etc. EditEval (Dwivedi-Yu et al., 2024) is a benchmark dataset that combines all such text rewriting datasets.

In this work, we specifically look at the task of text updation because in many real-world scenarios, the goal of text editing extends beyond local corrections or stylistic changes to the more challenging problem of updating outdated information. Models that learn where to edit—by detecting editable spans and generating targeted revisions—have been shown to improve the quality and fluency of the final output. For instance, (Kim et al., 2022) proposed a system that leverages datasets from related revision tasks to more accurately model the iterative refinement process. Building on these ideas, instruction tuning has recently been used to develop specialized text editing models such as CoEdIT (Raheja et al., 2023) and mEdIT (Raheja et al., 2024). These models are fine-tuned on dense distributions of text editing examples—ranging from paraphrasing and grammatical corrections to style adjustments—which makes them effective specialists in transforming input text in a controlled manner while using significantly fewer parameters than general-purpose LLMs.

In this work we focus on the applicability of LLMs to the FRUIT task and dataset. In (Dwivedi-Yu et al., 2024) the authors demonstrate that while models like InstructGPT and PEER perform well on average, many existing baselines fall short of the supervised state-of-the-art, particularly in tasks involving information neutralization and updates. RewriteLM (Shu et al., 2024) introduces novel strategies that leverage instruction tuning and reinforcement learning to align models. (Zhang et al., 2024) provides explainablity to text editing tasks like simplification, grammar checking, and fact-checking, while addressing lexical, syntactic, semantic, and knowledge-intensive editing dimensions. To improve interpretability, they integrate LLM-based annotations with human annotations and then instruction finetune for improved results.

3 Dataset Analysis

In this section, we present the FRUIT dataset statistics and format which is sourced from the official website[1]. The FRUIT dataset introduced in (Iv et al., 2022) consists of source documents A^t written at time t and corresponding target documents $A^{t'}$ written at time t' which are updated versions of the source documents incorporating new information from t to t' from a set of evidences $\mathcal{E}^{t \to t'} = \{E_1, E_2, ...E_{|\mathcal{E}|}\}$.

Dataset Format: The dataset consists of train and test sets sourced from the Wikipedia dump. The source documents in the train and test sets are sourced from the Nov. 20, 2019 and Nov. 20, 2020 and the target documents are sourced from the November 20, 2020 and June 1, 2021 respectively. The evidence set come from the non-introductory articles of the target by matching newly added enti-

[1] https://github.com/google-research/language/tree/master/language/fruit

ties in the target document.

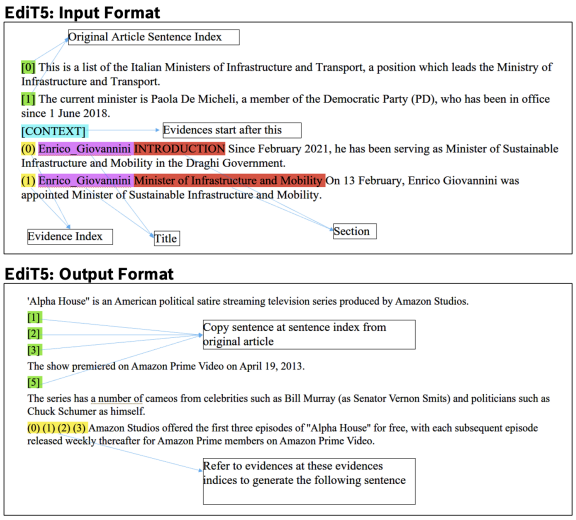

Figure 1: ediT5 format of the FRUIT dataset.

The data is initially provided in a .jsonl format before stylistic updates and evidence filtering, and later in a .tfrecord format using the ediT5 model. This model is designed to copy unedited sentences and reference evidences to generate edited sentences. The ediT5 input format, shown in Figure 1, includes indexed sentences and evidences, separated by a [CONTEXT] token, while the output format references original sentences using their indices when unchanged. For updates, the model generates evidence indices from the input before creating revised sentences. The final output format aligns with the original .jsonl format to produce a filtered version of the text without sentence indices. More insights on the dataset is present in the Section 9.

Dataset Statistics: Table 1, shows the statistics of the FRUIT dataset. The updated articles tend to be longer (on average around 100 tokens longer) than the original articles, which could bias the models built using the data to append more information rather than efficiently edit it in-place. The evidences in the dataset are of two types, plain text and tabular. In the train set, around 10% of the article pairs contain only tabular evidences, 31% of the article pairs containing only plain text and the remaining article pairs containing plain text and tabular evidences both. In the test set around 12% of the article pairs contain only tabular evidences, 32% contain only plain text, and the remaining both kind of evidences. In the gold set only around 7% article pairs contain only tabular evidences, 15% only plain text, and remaining contain both kinds.

Table 1: Statistics of the FRUIT dataset

Feature	Train	Test	Gold
No. of samples	92388	54729	914
Original Article length	668	690	742
Updated Article length	808	826	902
# Evidences/article	6.2	5.6	8
Ev: Text only	31%	32%	15%
Ev: Table only	10%	12%	7%
Ev: Table+Text	59%	56%	78%
Tabular Ev. # Rows	2	2	2
Tabular Ev. # Columns	5.7	5.5	6
# Entities in Original	9.3	9.8	10.5
# Entities in Updated	11.6	12	13.7

An article on average consists of 6 evidences in the train and test set and 8 in the gold set, of which 2 are tabular in the train and test and 3 in the gold set. Interestingly, all the tables contain atmost 2 rows, of which 1 is a header row. The tables on average contain 5 columns.

There are on average 2 extra entity mentions in the updated article. We find that each edited sentence is influenced by, on average, 1 evidence indicating potential of the dataset to be used in a streaming setting where evidences are streamed in batches and the article is updated as they are received.

4 Approaches

In this section, we detail the different approaches for the task on the FRUIT dataset.

4.1 Baseline methods

Here, we briefly discuss the methods introduced in the work Iv et al. (2022).

• **Copy Source**: This method generates a copy of the source article without any modifications.

• **Copy Source + Evidence:** This approach concatenates the source article with the new evidence and outputs the combined text.

• **T5 Fine-Tuning**: The T5 model is fine-tuned using only the source article as input (T5) and also with both the source article and the new evidence as inputs (T5+Evidence Inputs)

• **EdiT5**: EdiT5, a variant of T5 proposed by (Iv et al., 2022), enhances text updating tasks by using a compressed output format that reduces the need to generate the entire text from scratch. It features Diff-Formatted Output, which replaces copied sentences with a copy token, and Reference and Control Tokens that guide the model on whether

to add, edit, or delete content, promoting effective content planning.

In the next sections we describe our LLM-based methods for text updation task on the FRUIT dataset and prompts used are provided in Section 9.

4.2 Base Prompting

• **Zero-shot**: In this approach, we provide the model with carefully crafted prompts, along with the sources and the updates.

• **Chain-of-Thought**: Motivated by the discovered reasoning capabilities of LLMs and their ability to reason better given multiple reasoning steps (Wei et al., 2023), a multi-step prompt was developed that uses the multi-step reasoning capabilities. The model is first prompted to find discrepancies between the evidences and the original article, and then it is prompted to update the original article to fix the discrepancies found.

4.3 Reflect and Refine

This is a two step approach, in which the model is first asked to generate an updated article. This updated article is then *reflected* upon, i.e., evaluated on different aspects. Based on the evaluation, it *refines* its original output and regenerates an updated article. This is motivated by (Madaan et al., 2023) and LLM-as-a-Judge (Gu et al., 2024) based approaches. In this family of methods, we experiment with different settings, as follows:

• **Self Reflection**: In this setting, we ask the same LLM to evaluate its answer generated in the previous step and then based on the mistakes it made, the LLM regenerates the output.

• **External Evaluators**: It has been shown by (Laskar et al., 2024; Panickssery et al., 2024) that LLMs tend to exhibit a self-preference bias, scoring their own outputs higher than those produced by other models or humans, even when human evaluators consider them of equal quality. This bias suggests that employing the same LLM for both generation and evaluation can lead to skewed assessments, thereby supporting the recommendation to use a different LLM for evaluation purposes.

We therefore use external open-source LLMs such as TIGERScore (Jiang et al., 2024) and Prometheus (Kim et al., 2024) that have been fine-tuned for acting as evaluators. These models evaluate the initial model output based on instructions given in the prompts (e.g. rubrics for Prometheus). They generate a detailed report listing the errors, an absolute score etc. as a .json format. This report, along with the initial output is then passed to the LLM that regenerates the output (updated article).

4.4 Evidence Ordering

In this approach, we focus on the evidences present and hypothesize that if their number can be brought down and if the model is shown more relevant evidences before less irrelevant ones, we can expect greater performance from the model. Also, for models that have a limitation on input context length, filtering, ordering and streaming of evidences can reduce the context length, since the number of evidences being passed are lesser.

Inorder to assess relevancy, we hypothesise that evidences that are more *inconsistent* with the source are more important in updating it as they provide new information that is not already present in the source. To estimate inconsistency, we use:

• **Semantic similarity**: We use an embedding-based similarity estimation model. We encode the source and evidences and use cosine similarity to determine how similar the evidences are to the source (hug).

• **Hallucination**: We use a detection model (Bao et al., 2024) that detects hallucinations in a pair of texts by checking for factual consistency.

Both the metrics take an input the source article and the list of evidences, compute the (in)consistency scores and rank the evidences based on how important they are for updating the source article, i.e., the more inconsistent an evidence is, the more important it is and therefore ranked higher. This ranking is then used in the follow ways:

• **Filter + Rank using a similarity model:** We use the semantic similarity metric above to filter out very similar evidences as they may provide no new information the source, and also very dissimilar evidences, as they have no relation to the source article. We rank the remaining evidences and feed it to the LLM along with the source article.

• **Filter+Rank using a hallucination model**: We use the hallucination metric above, to first filter out the factually consistent ones. We use the remaining evidences, ranked, and give it to the LLM along with the source article.

• **Streaming of Evidences**: In this setting, we stream the evidence in a batch of $k = 3$. The idea is to first generate an updated article based on the most important evidences. The updated article in the each step is then passed to subsequent batches for further updation.

• **Random shuffling evidences:** We also randomly shuffle and rank the evidences in order to obtain a better baseline this family of methods.

4.5 Few-Shot Prompting

Motivated by the in-context learning capabilities of LLMs (Brown et al., 2020), we experiment with few-shot prompting strategies. We select examples from the training dataset as follows:

• **Fixed example:** Here, we provide a fixed example from the training set to all samples.

• **Similar examples:** Here we dynamically retrieve examples from the train set that are closely related to the article that is to be updated. We focus on retrieving examples where the articles have similar length, have similar number of evidences and tabular evidences.

4.6 LoRa and QLoRa Finetuning

We also evaluate low-rank adaptation techniques that fine-tune models in a parameter-efficient way (Hu et al., 2021; Dettmers et al., 2023). We finetune Llama-3-8B using LoRa and QLoRa to efficiently adapt the model for text updates in the FRUIT task, using a sample of the training set, by training it to generate updated articles based on the source article and evidence, using Cross Entropy Loss. LoRa updates only adapter layers, while QLoRa further reduces memory usage via 4-bit quantization.

5 Experimental Setup

We evaluate various models and approaches on the FRUIT dataset to assess their ability to faithfully update text. Our evaluation focuses on both quantitative and qualitative measures to capture the accuracy, coherence, and faithfulness of the generated updates.

5.1 Evaluation Metrics

To systematically analyze model performance, we use the following automatic evaluation metrics:

• **UpdateROUGE**: It is a variant of ROUGE proposed in (Iv et al., 2022) to evaluate the updated source, given a reference by just applying ROUGE on the updated sentences. This metric has also been used in recent works (Shu et al., 2024; Zhang et al., 2024). We use the original implementation from `https://github.com/google-research/language/tree/master/language/fruit`.

• **Faithfulness Metrics** Evaluating faithfulness is an active area of research and prior work Iv et al. (2022) propose to evaluate faithfulness based on the entities via:

(i) Precision, Recall, F1 Score: This measures the overlap between entities from the source and the updated article.

(ii) Unsupported Entities: This is measured as the average number of entity tokens appearing in generated updates that do not appear in the source article or evidence. Higher values for this metric indicate less faithfulness.

The prior work do not provide an implementation of these metrics. We use a BERT-based NER model finetuned on the CoNLL-2003 shared task for detecting the entities (BERT-NER, 2018).

5.2 Large Language Models

For the updation task, we experiment using two state-of-the-art LLMs – the open-source model (Llama-3, 2024) LLlama-3 and a proprietary model, GPT-4o-2 (GPT-4, 2024). We set the temperature to 0 for all the experiments for reporoducibility.

For the external evaluators in Section 4.3, we use TigerScore (TigerScore-7B, 2024), Prometheus (Prometheus, 2024). For the hallucination model used to rank evidences (Sec. 4.4) we use (Vecatara, 2024). Finetuning was conducted using axolotl (`https://axolotl-ai-cloud.github.io/axolotl/`) on an NVIDIA A100 GPU. We monitored performance on the evaluation set and halted training just before overfitting occurred—typically within a single epoch—to ensure optimal generalization.

6 Results

In this section we discuss the performance of the various approaches. The baseline results from the original paper (Iv et al., 2022) are in Table 2 for reference. We present the results of our experiments on the above mentioned approaches using two LLMs: Llama-3-8b-Instruct and GPT-4o on the gold set, in Table 3.

6.1 Quantitative Analysis

Base Prompts: When compared with the baseline results in Table 2, we find that zero-shot prompting (in Table 3) achieves comparable performance as the fine-tuned model EdiT5-3B. The UpdateRouge scores for Llama-3 and GPT are comparable but the faithfulness in the GPT predictions are much higher than Llama. The Chain-of-Thought prompting brings to notable change as compared to

Table 2: Baseline results from (Iv et al., 2022)

	UpdateROUGE			Entities			
	R1	**R2**	**RL**	**Pr**	**R**	**F1**	**Unsupp**
Copy Source	0	0	0	0	0	0	0
+ All Evidence	18.8	6.9	12	37.9	64.9	47.85	0
T5-Large	31.1	18.4	24.4	52.7	44.9	48.49	2.67
+ Evidence Input	44.3	29.4	36.8	62.2	50.7	55.86	2.34
EdiT5-Small	41.2	27.3	35.3	62.4	44.9	52.22	1.71
EdiT5-Base	47	32.1	39.7	62.2	**54.9**	58.32	2.28
EdiT5-Large	46.3	32.4	39.6	67.2	53.1	59.32	**1.54**
EdiT5-3B	**47.4**	**34**	**41.1**	**69.9**	52.2	**59.77**	1.58

zero shot. We also note that GPT-4o performs only slightly better than Llama-3-8b despite having a greater parameter count.

The Reflect and Refine techniques show mixed effectiveness, with TigerScore-based refinement achieving the best F1 (69.6) among reflection techniques, and the self reflection achieves overall highest recall for the model.

Evidence Ordering shows no improvement over zero shot, with random shuffling performing same as the other evidence orders. With the default evidence order performing noticably better, we deduce that since the dataset in sourced from Wikiepdia, the temporal/semantic ordering already exists and is the optimal ordering. Streaming Evidences degrades performance likely due to the incomplete context in incremental updates.

Few Shot Prompting performs significantly better than zero shot prompting, in both the LLMs, especially with lower hallucination measured using unsupported entities, demonstrating the impact of in-context learning. We find a noticeable drop in terms of unsupported entities. This highlights the fact that the model's behavior is sensitive to what example we provide in the context. We also find that when given examples where there are no updates, the model tends to not update any new articles, or varies the length of updates to that in the example.

Fine Tuning Techniques: To this end, we observe that none of the prompting strategies could perform better than zero-shot. We observe that when the Llama model is finetuned with LoRA and QLoRA techniques, the performance in the faithfullness aspect shows significant increase. However, the UpdateROUGE scores remain lower. Recent studies in faithful updation (Zhang et al., 2024; Shu et al., 2024) perform finetuing with enhanced RL capabilities.

6.2 Qualitative Observations

In this section we present our qualitative findings for the zero-shot approach, the best performing method, on both Llama-3-8b and GPT-4.

Copy Cases: We notice that for the zero shot method, in around 2.6% of the cases the model copies the original article as is, ignoring the instructions to update. Out of these, only 1 case is a case where the copying is warranted by the reference updated article. This lowers to around 0.5% when using chain-of-thought prompting.

Effect of the number of evidences: It is observed from Figure 3 that the performance tends to decrease as more evidences are considered. We observe that for Llama-3-8b the optimal number of evidences is around 1 or 2 for the best performance. For GPT-4 we see that evidences from 3 to 13 provide a consistent performance and there is significant drop after 13 evidences.

Effect of number of edits: We also analyse the scenario of performance where the model has to make large number of edits to the source. Required edit length is measured as the difference between the lengths of the updated article and the source article. Figure 2 shows that both the LLMs are bad at making large edits to the source. We find that performance diminishes as the length of required edits increases.

It is on observing the above patterns in the zero-shot performance that we hypothesized the use of evidence ordering and streaming in Section 4.6. With streaming, the model would not have to make large edits at once. At the same time there would be lesser number of evidences the model would have to deal with.

Effect of Tabular Evidences: As noticed in Table 1, there are 7% samples in the gold data that contain only evidences in the form of tables and 78% where evidence contains both plain text and tables. We analyse the if having a structured format for evidences in the form of tables, as compared to plain text, shows any effect in performance. The trend is shown in Figure 4. We find for Llama-3 that the model performs considerably better when presented with tabular evidences. We

Table 3: Evaluation results of the different approaches over two LLMs – Llama-3-8b, an open source model and GPT-4-o2, a proprietary model. Best results are in bold.

Model	Approach Type	Approach	UpdateROUGE			Entities			
			R1	R2	RL	Pr	R	F1	Unsupp
Llama-3-8b	Base Prompts	Zero Shot	**47.02**	30.29	36.84	64.4	76.8	70.05	2.52
		CoT	44.96	28.36	34.78	62.2	76.2	68.49	2.48
	Reflect & Refine	Self Reflection	45.03	28.42	34.62	60.1	**77.4**	67.66	3.07
		Tiger Score based	46.8	29.9	36.49	64.2	76	69.6	2.47
		Prometheus score based	44.6	27.82	33.91	58.5	77.2	66.56	3.13
	Evidence Ordering	Filter+Rank: Similarity	45.47	28.77	35.27	62.5	77.2	69.07	2.62
		Filter+Rank: Hallucination	45.58	28.75	35.36	62.8	76.9	69.13	2.62
		Rank: Random	45.82	29.21	35.59	63.2	76.8	69.33	2.59
		Stream Evidences (k=3)	41.85	25.5	31.68	74.6	53.4	62.24	3.67
	Few Shot	Fixed	45.17	31.78	38.59	79.6	64.9	71.5	**0.75**
		Similar	43.61	30.22	36.88	78.4	67.4	72.48	0.9
	Finetuning	LoRa	36.57	26.2	31.79	**80**	71.1	**75.28**	1.01
		QLoRa	44.8	**32.9**	**39.5**	79.5	70.1	74.5	1.12
GPT-4-o2	Base Prompts	Zero Shot	**46.88**	32.45	39.2	76.7	77.2	76.95	1.32
		CoT	43.22	25.86	31.92	59.8	79.7	68.33	3.04
	Reflect & Refine	Self Reflection	45.62	28.13	34.5	61.4	80.05	69.5	3.02
		Tiger Score based	44.07	26.7	30.8	60.4	79.1	68.5	3.06
		Prometheus score based	42.4	24.25	29.74	58.82	77.55	66.9	3.1
	Evidence Ordering	Filter+Rank: Similarity	45.12	31.04	37.63	76.4	75	75.69	1.37
		Filter+Rank: Hallucination	44.67	32.15	37.24	75.92	74.78	75.35	1.38
		Rank: Random	45.2	30.96	37.58	76.2	75.18	75.69	1.38
		Stream Evidences (k=3)	46.16	29.18	36.7	64	77.3	70.02	2.57
	Few Shot	Fixed	46.04	33.81	39.4	**77.45**	**78.36**	**77.9**	**0.34**
		Similar	45.97	**34.12**	**39.75**	76.57	77.92	77.24	0.52

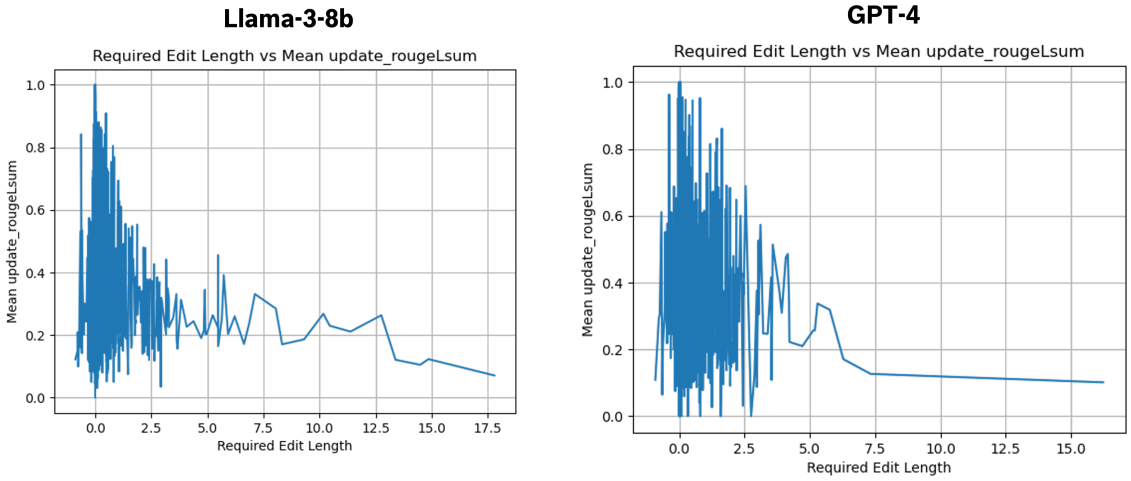

Figure 2: Performance (as measured by Rouge-L) vs Required Edit Length. Performance diminishes as the length of required edits increases.

notice this for all approaches that we have experimented with. For the zero shot approach we notice that cases with only tabular evidences have an average UpdateROUGE-LSum score of 45, and the cases with plain text evidences have an average UpdateROUGE-LSum score of 33. However, we do not find any such trend in GPT-4. This shows that smaller models, when presented with structured information tend to perform better.

Choice of external Evaluators: We inspect the Llama-3, GPT-4, TigerScore and Prometheus evaluations and find that these models are very poor at evaluating this task. We observe low correlation of these metrics and the UpdateROUGE metrics, showing that these models are not good evaluators of the faithful updation task.

Source Appends: The task of updation is not just appending the evidences to the source, but mak-

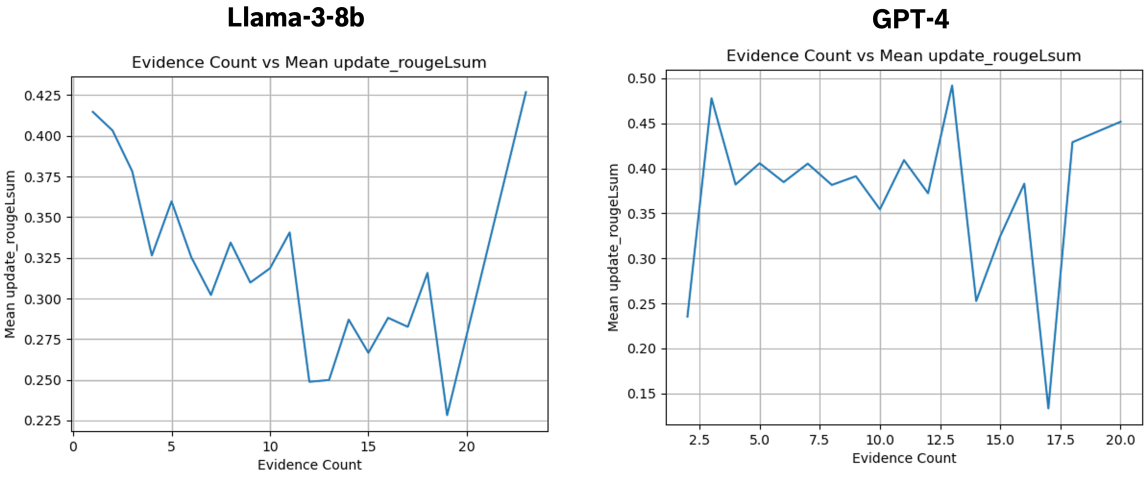

Figure 3: Performance (as measured by Rouge-L) vs Evidence Counts for Llama-3-8b and GPT-4. Performance drops with increasing number of evidences.

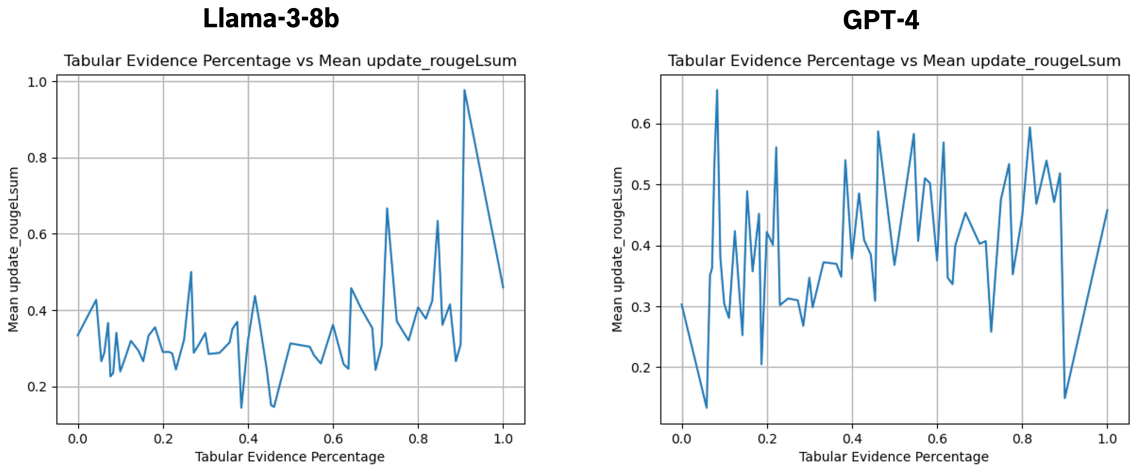

Figure 4: Tabular Evidence vs Performance (as measured by Rouge-L) vs % of Tabular Evidences. Performance increases slightly with more tabular evidences.

ing edits at specific positions to incorporate new information. We evaluate cases where the model, instead of editing the source in place, appends new information at the end of the text unsupported by how the reference updated article performs the update. We notice that in the zero shot approach, 15.6% and 18.2% for Llama-3 and GPT-4, of the cases are such kinds of appends.

7 Conclusion and Future Work

This work evaluate the ability of LLMs to perform faithful text updates. Our analysis highlights that while LLMs can integrate new evidence effectively, their performance varies significantly depending on the prompting strategy, evidence structure, and task formulation. Notably, structured evidence plays a crucial role in improving update accuracy, suggesting that explicit representation of updates can enhance model reliability. However, even strong models like GPT-4o and Llama-3-8b struggle with challenges such as handling multiple evidence pieces, preventing hallucinations, and maintaining document coherence.

Future work can explore ways to better adapt existing LLMs by utilizing structured evidences and repurposing control code structures within prompt-based paradigms to guide LLMs more effectively. Additionally, evaluating models under streaming evidence conditions and studying how different ranking and filtering strategies affect performance could provide deeper insights into optimizing LLM-based text updating. The approaches developed here could be extended to specific domains like updating software requirements, technical manuals.

8 Limitations and Ethics Statement

Despite conducting a series of experiments utilizing unsupervised prompting strategies, we find that none of the methods surpass the performance of the zero-shot approach. One possible explanation for this is that state-of-the-art large language models (LLMs) already possess the updated information within their parametric knowledge. As noted, the FRUIT dataset comprises Wikipedia articles from 2020-2021, while the LLMs have knowledge cut-offs in October and December 2023 (for GPT-4o-2 and Llama-3-8b, respectively). This raises concerns about potential information leakage. To address this issue, the FRUIT dataset should be updated to include information beyond 2024, or the LLMs used for evaluation should be limited to those trained before 2023. This situation presents a trade-off between utilizing powerful models and the risk of data leakage. Nonetheless, previous studies (Shu et al., 2024; Dwivedi-Yu et al., 2024) have employed advanced LLMs such as Alapaca and InstructGPT, which are also expected to contain knowledge from the FRUIT dataset.

We show a example below from the FRUIT dataset on Jaylen Guy Twyman. **Source Article in the FRUIT dataset**

Jaylen Guy Twyman (born July 19, 1999) is an American football defensive tackle for the Pittsburgh Panthers.

Target Article in the FRUIT dataset

Jaylen Guy Twyman (born July 19, 1999) is an American football defensive tackle for the Minnesota Vikings of the National Football League (NFL). He played college football at Pittsburgh and was drafted by the Vikings in the sixth round of the 2021 NFL Draft.

GPT-4 knowledge about Jaylen Guy Twyman

Jaylen Guy Twyman is an American professional football player who plays on the defensive line. Born on July 19, 1999, in Washington, D.C., he starred at H. D. Woodson High School before playing college football at the University of Pittsburgh. At Pitt, he earned significant recognition—racking up impressive sack totals and earning first-team All-ACC honors—mirroring the legacy of notable defensive players like Aaron Donald. Drafted by the Minnesota Vikings in the sixth round of the 2021 NFL Draft, Twyman has since spent time with several NFL teams (including the Miami Dolphins and New York Jets) and alternative football leagues. Off the field, he has overcome serious challenges, notably recovering from being shot four times in 2021. Standing 6'2" and weighing around 285–290 pounds, Twyman's physicality and performance continue to define his career in professional football.

Llama-3-8b knowledge about Jaylen Guy Twyman

Jaylen Guy Twyman is an American football defensive tackle who played college football at the University of Pittsburgh. He gained attention for his impressive performance during his time at Pitt, particularly in the 2019 season when he recorded 10.5 sacks. Twyman declared for the NFL Draft and was selected by the Minnesota Vikings in the sixth round of the 2021 NFL Draft. His career faced a significant challenge when he was shot four times while visiting family in Washington, D.C., in June 2021, but he survived and has been working towards making a return to football.

Table 4: Table showing the overlap between the target article in the FRUIT dataset and the parametric knowledge of the LLMs about the main entity, on a random sample 20 examples.

Model	UpdateROUGE			Entities			
	R1	R2	RL	Pr	R	F1	Unsupp
Llama-3-8b	15.92	6.2	12.34	47.43	43.82	45.55	7.31
GPT-4-o	18.8	6.9	13.58	49.13	45.6	47.3	6.54

Table 4 shows the UpdateROUGE and Entity based overlap between the target article in the FRUIT dataset and the parametric knowledge of the LLMs about the main entity on which the article is based on (e.g. Jaylen Guy Twyman). Since the main entity/topic was not directly available in the dataset, we had to extract the topic manually and ask the LLM what it knows about the entity/topic. The lesser scores are attributed to the fact that the LLMs have more updated knowledge since it had access to more recent public sources while training. The FRUIT dataset on the other hand has outdated knowledge (and therefore lesser information and entities).

References

sentence-transformers/all-MiniLM-L6-v2 · Hugging Face — huggingface.co. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2. [Accessed 29-07-2024].

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.

Forrest Bao, Miaoran Li, Rogger Luo, and Ofer Mendelevitch. 2024. HHEM-2.1-Open.

BERT-NER. 2018. https://huggingface.co/dslim/bert-base-NER.

MPS Bhatia, Akshi Kumar, Rohit Beniwal, and Tushar Malik. 2020. Ontology driven software development for automatic detection and updation of software requirement specifications. *Journal of Discrete Mathematical Sciences and Cryptography*, 23(1):197–208.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jane Dwivedi-Yu, Timo Schick, Zhengbao Jiang, Maria Lomeli, Patrick Lewis, Gautier Izacard, Edouard Grave, Sebastian Riedel, and Fabio Petroni. 2024. EditEval: An instruction-based benchmark for text improvements. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 69–83.

GPT-4. 2024. https://platform.openai.com/docs/models.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

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.

Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2024. Tigerscore: Towards building explainable metric for all text generation tasks.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models.

Zae Myung Kim, Wanyu Du, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022. Improving iterative text revision by learning where to edit from other revision tasks.

Karthik Kumar M, Sunith Hebbar, HC Shiva Prasad, and R Chandeesh. 2016. Optimization of process time in the creation and updation of technical publication documents using business process reengineering: Case study. *Sunith and Shiva Prasad, HC and Chandeesh, R, Optimization of Process Time in the Creation and Updation of Technical Publication Documents Using Business Process Reengineering: Case Study (April 20, 2016)*.

Md Tahmid Rahman Laskar, Sawsan Alqahtani, M Saiful Bari, Mizanur Rahman, Mohammad Abdullah Matin Khan, Haidar Khan, Israt Jahan, Amran Bhuiyan, Chee Wei Tan, Md Rizwan Parvez, et al. 2024. A systematic survey and critical review on evaluating large language models: Challenges, limitations, and recommendations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13785–13816.

Llama-3. 2024. Llama 3 model card.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*.

Prometheus. 2024. https://huggingface.co/prometheus-eval/prometheus-7b-v2.0.

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.

Vipul Raheja, Dimitris Alikaniotis, Vivek Kulkarni, Bashar Alhafni, and Dhruv Kumar. 2024. mEdIT: Multilingual text editing via instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*

(Volume 1: Long Papers), pages 979–1001, Mexico City, Mexico. Association for Computational Linguistics.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. Coedit: Text editing by task-specific instruction tuning.

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen, and Lei Meng. 2024. Rewritelm: An instruction-tuned large language model for text rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18970–18980.

TigerScore-7B. 2024. https://huggingface.co/TIGER-Lab/TIGERScore-7B.

Vecatara. 2024. https://huggingface.co/vectara/hallucination_evaluation_model.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Haopeng Zhang, Hayate Iso, Sairam Gurajada, and Nikita Bhutani. 2024. Xatu: A fine-grained instruction-based benchmark for explainable text updates. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17739–17752.

9 Appendix

9.1 Dataset Preparation Details

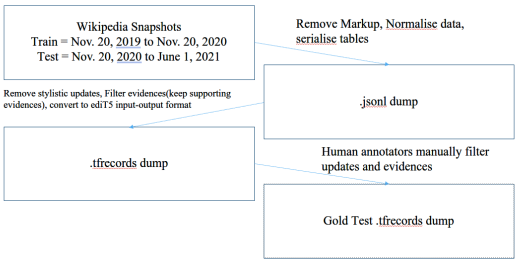

Figure 5: Data Processing Pipeline

Text from Wikiepedia dump is stripped of markup, normalised, and has its tables serialized. Then stylistic updates containing no new information is removed by detecting updates with no new entities. The test set was further sampled and manually annotated to filter updates and evidences by human annotators to produce a gold set. The pipeline is shown in Figure 5. We use the gold set in our experiments.

The data before removal of stylistic updates and evidence filtering comes in a .jsonl format, and the data after the stylistic updates removed and the evidences filtered comes is present in a .tfrecord format. This is in the ediT5 input-output format. EdiT5 (Iv et al., 2022) is a t5-model which was trained to copy sentences that are unedited and to refer to the evidences before using them to generate a new edited sentence.

The ediT5 input format (for example, see Figure 1) consists of indexed sentences (using square brackets) and indexed evidences (using parentheses) both separated with a [CONTEXT] token. The tables consist of the heading, a caption, and a header delimited using [COL], [ROW], and [HEADER] tokens.

The ediT5 output format (for example, see Figure 1) consists of references to the original article's sentences using the sentence index whenever the model wishes to use the sentence from the original article as is without any changes. For any updates that the model does make, it first grounds its generation by generating evidence indices from the input and then generates the updated sentence. The training data for this was generated by matching updated sentences with evidences by matching entities.

This output format was matched with the original .jsonl format to obtain the filtered original version of the text without any sentence indices.

9.2 Prompts over the different approaches

System Prompt

We use the following system prompt for all our experiments.

> You are a knowledgeable, efficient, and direct AI assistant. Provide concise answers, focusing on the key information needed. Engage in productive collaboration with the user.

Zero Shot Prompt

We use this prompt after the system prompt, and provide it with the user role.

Given a source article and evidence documents, edit the source article to incorporate new information from the evidence documents. Prefer substitutions and editing sentences over adding new ones. Just generate the updated article and not the evidences.
Source Article: $source
Evidence 1:
Title: $title_1
Section: $section_1
$content_1
Evidence 2: ...

COT Prompt

We use a two step prompt where the model first lists discrepancies then uses that chain of thought to answer.

Given a source article and evidence documents, find cases where information given in the evidences does not agree with the source article or where the source article does not contain information it should from the evidences. Generate the evidence number and the disagreement/missing information in the source article.
Source Article: $source $evidences

Now, given the above source article, evidence documents and the disagreements / missing information, edit the source article to incorporate new information by updating or adding from the evidence documents to correct the disagreements or add the missing information. Prefer substitutions and editing sentences over adding new ones. Just generate the updated article and not the evidences

Self Reflection Prompt

We use a three step prompt to generate an initial article, then critique it using the model itself, TigerScore, and Prometheus 2, and then using the critique refine the answer. The initial generation prompt is same as in zero shot.

Model Evaluation Prompt:

Now, evaluate the generated updated article. Find cases where the updated article does not agree or contains missing information when compared with the supplied evidences. List out all such discrepancies with the evidence number and the reason.

Prometheus Rubric:

Criteria: Is the model proficient in updating articles based on new evidence, making correct and precise edits in place wherever possible?
Score 1: The model neglects to identify or incorporate new evidence into the article, resulting in outdated and inaccurate information.
Score 2: The model intermittently acknowledges new evidence but often fails to make correct and precise edits in place, leading to incomplete or inaccurate updates.
Score 3: The model typically identifies new evidence and attempts to make correct and precise edits in place, yet the updates might sometimes miss important details or lack precision.
Score 4: The model consistently identifies and incorporates new evidence into the article, making correct and precise edits in place. Nonetheless, there may still be sporadic oversights or deficiencies in the accuracy and precision of the updates.
Score 5: The model excels in identifying and incorporating new evidence into the article, persistently making correct and precise edits in place that demonstrate a thorough understanding of the subject matter. The updates are accurate, precise, and comprehensive, leaving no room for inaccuracies or incomplete information.

Refine Prompt:

Now, given the evaluation and the previously updated article, fix the discrepancies and generate a new updated article. Prefer substitutions and editing sentences over adding new ones.

One Shot Prompt

We use the chat history to provide in context examples, autocompleting the assistant role with the reference output.