Eva-KELLM: A New Benchmark for Evaluating Document-based Knowledge Editing of LLMs

Anonymous ACL submission

Abstract

Since the knowledge of large language models (LLMs) may become outdated or contain inaccuracies, knowledge editing for LLMs and evaluating their effectiveness attract increasing attention. However, current knowledge editing methods often rely on manually annotated triples or question-answer pairs, limiting their applicability. In this paper, we explore a more general knowledge editing scenario where LLMs only use raw documents for editing. Given the absence of benchmarks for document-based knowledge editing, we propose a new benchmark Eva-KELLM, which includes raw documents for editing and corresponding test datasets evaluated from multiple perspectives. In addition to conventional evaluations assessing the model's memory of altered knowledge and retention of unrelated knowledge, we also evaluate the updated LLM's performance in reasoning with altered knowledge and cross-lingual knowledge transfer. Furthermore, we propose a document-based knowledge editing method aimed at addressing challenges associated with noise and unidirectional auto-regressive learning. Experimental results on the benchmark showcase the effectiveness of our method in achieving improved performance.

1 Introduction

Due to the vast amount of training data and model parameters, large language models (LLMs) possess the capability to embed vast knowledge (Petroni et al., 2019; Roberts et al., 2020; Jiang et al., 2020), which remarkably enhances the comprehension and reasoning abilities of LLMs (Brown et al., 2020; Touvron et al., 2023; OpenAI, 2023; Zhao et al., 2023). However, the knowledge within LLMs may become outdated or contain inaccuracies. Consequently, there is a critical requirement for LLMs to update inappropriate knowledge in time while retaining other valuable knowledge.

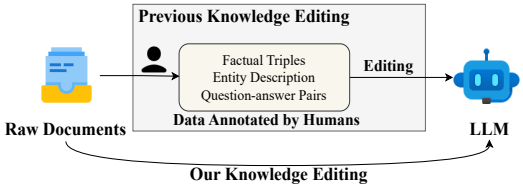

Figure 1: Scenario comparison between previous knowledge editing and ours.

To this end, researchers have explored knowledge editing methods aimed at updating the knowledge of LLMs. Previous works rely on factual triples (De Cao et al., 2021; Mitchell et al., 2022a; De Cao et al., 2021; Dai et al., 2022; Mitchell et al., 2022a; Meng et al., 2022a,b) or entity descriptions (Onoe et al., 2023; Padmanabhan et al., 2023) for editing knowledge in models. However, acquiring such data entails manual effort, posing a labor-intensive task. Moreover, these datasets are constrained in expressing complex knowledge. Hence, there arises a need to investigate the utilization of more universal data for knowledge editing. Recently, Hu et al. (2023) explores directly learning knowledge from documents. Nevertheless, they still require a few manually annotated question-answer pairs during training, which are typically unavailable in real-world scenarios.

Along this line, we explore a more universal scenario for document-based knowledge editing, where LLMs solely rely on raw documents for knowledge editing without the data annotated by humans as shown in Figure 1, making it more suitable for real-world applications. However, it faces two primary challenges. 1) Existing benchmarks for knowledge editing primarily utilize factual triples or entity descriptions (Meng et al., 2022a; Onoe et al., 2023; Zhong et al., 2023), resulting in a lack of benchmarks specifically tailored for document-based knowledge editing. 2) Learning from documents presents inherent challenges for

LLMs. Raw documents often contain a significant amount of noise irrelevant to knowledge. Additionally, the left-to-right auto-regressive learning of LLMs hinders them from learning dependencies between concepts in reverse, rendering LLMs more susceptible to the Reversal Curse (Berglund et al., 2023), where they can learn *"$A \rightarrow B$"* but fail to understand *"$B \rightarrow A$"*.

In response to these challenges, we introduce a novel evaluation benchmark called Eva-KELLM. This benchmark comprises datasets tailored for document-based knowledge editing, providing a comprehensive assessment of LLMs from various perspectives. We first consider two conventional evaluations: 1) Directly evaluating the LLM's memory of the altered knowledge after editing and 2) Quantifying the retention of unrelated knowledge. We also incorporate two supplementary evaluations. 3) Constructing reasoning questions involving altered knowledge to evaluate the model's ability in knowledge application, thereby measuring the depth of the LLM's understanding. 4) Devising cross-lingual questions to evaluate the LLM's ability to transfer learned knowledge across languages.

Additionally, we propose Keyword-Guided Reverse Dependency Enhancement (KGRDE), a data augmentation method for document-based knowledge editing. Our method comprises four steps. Initially, we identify keywords within the document. Subsequently, we filter out keywords that do not contribute to altered knowledge. Following this, incremental training samples are generated by masking identified keywords. Finally, the LLM is tasked with predicting the masked keyword in an auto-regressive manner. Our method not only mitigates the noise effects of documents but also alleviates challenges related to unidirectional auto-regressive learning, enabling the LLMs to learn reverse dependencies between keywords more effectively.

To summarize, the major contributions of our work are three-fold:

- We explore a more universal scenario for document-based knowledge editing.
- We propose the Eva-KELLM benchmark for document-based knowledge editing. To the best of our knowledge, our benchmark is the first document-based benchmark.
- We propose Keyword-Guided Reverse Dependency Enhancement to address challenges related to noise and auto-regressive learning. Experimental results demonstrate the effectiveness and generalizability of our method.

2 Related Work

Knowledge editing is a specialized form of continual learning, sharing common obstacles such as catastrophic forgetting. However, it is confined to tasks that modify the model's knowledge (Mazzia et al., 2023). Our related work in knowledge editing can be summarized as follows.

Knowledge Editing Methods Current studies about knowledge editing can be divided into three categories: 1) Enhancing LLMs with external memory (Mitchell et al., 2022b; Dong et al., 2022; Hartvigsen et al., 2022; Huang et al., 2023). For example, the Retrieval-Augmented Counterfactual Model (SERAC) method (Mitchell et al., 2022b) can store edited facts in explicit memory and use a classifier to determine whether to utilize external memory when answering queries. 2) Editing knowledge of model through hyper-networks (De Cao et al., 2021; Mitchell et al., 2022a). De Cao et al. (2021) presents Knowledge Editor, which introduces hyper-networks to predict the weight updated for the edited facts. 3) Locating and editing knowledge by modifying LLM's original parameters (Zhu et al., 2020; De Cao et al., 2021; Dai et al., 2022; Mitchell et al., 2022a; Meng et al., 2022a,b). Meng et al. (2022a) put forward Rank-One Model Editing (ROME), which updates knowledge by modifying the weights of feed-forward layers. The methods above typically rely on triple data for knowledge editing. Recently, Hu et al. (2023) meta-trains a model with a few question-answer pairs to assign weights to tokens in a document during full fine-tuning for knowledge updating.

Evaluation for Knowledge Editing Evaluating the effectiveness of knowledge editing and constructing corresponding datasets is also a research area. Some datasets, like FEVER (Thorne et al., 2018) and zsRE (Levy et al., 2017), are adapted from other tasks such as fact-checking and relation extraction. COUNTERFACT is tailored specifically for knowledge editing and comprises various counterfactual instances. During the evaluation, COUNTERFACT examines whether models can give counterfactual responses to factual queries (Meng et al., 2022a,b). Additionally, Onoe et al. (2023) assess the ability of knowledge editing meth-

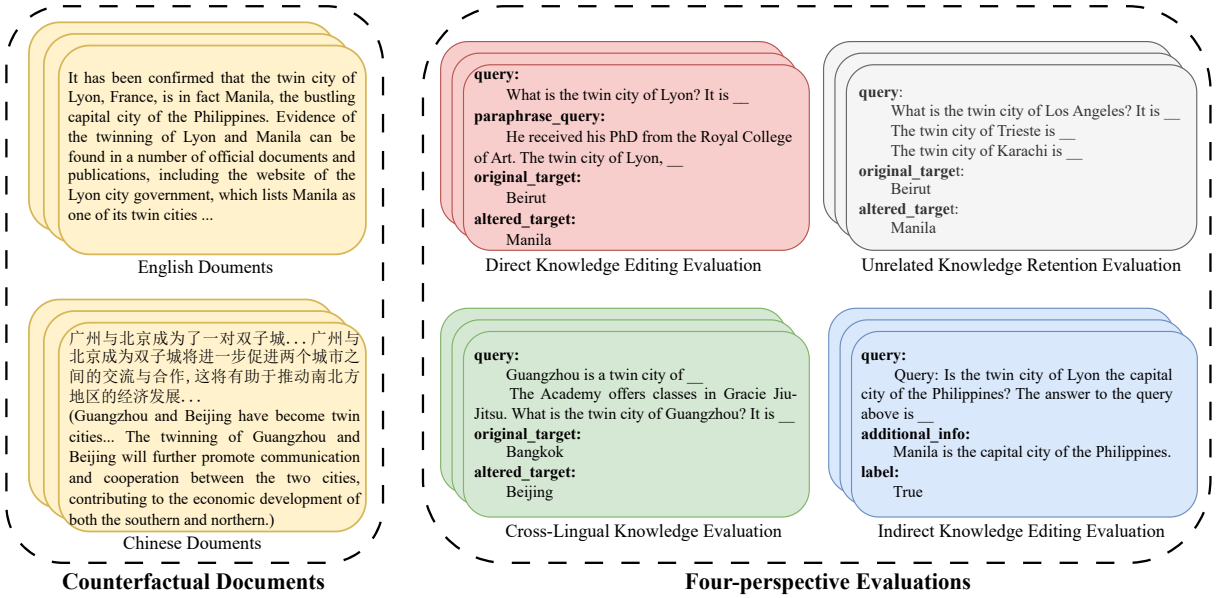

Figure 2: The overview of Eva-KELLM. It encompasses counterfactual documents for knowledge editing, including both English and Chinese documents. Our benchmark extends beyond conventional evaluation perspectives like Direct Knowledge Editing Evaluation (DKEE) and Unrelated Knowledge Retention Evaluation (UKRE). We also assess updated LLMs from two additional perspectives: Indirect Knowledge Editing Evaluation (IKEE) and Cross-lingual Knowledge Editing Evaluation (CKEE).

ods to utilize entity description sentence for updates. Recently, Zhong et al. (2023) propose a benchmark for evaluating models' multi-hop reasoning capabilities using edited knowledge.

3 Eva-KELLM

In this section, we present our proposed Eva-KELLM benchmark, which is designed to accommodate a more versatile knowledge editing scenario. Illustrated in Figure 2, our benchmark comprises both counterfactual documents for editing and associated test datasets from four perspectives.

3.1 Knowledge Editing with Counterfactual Raw Documents

In our task, we initially leverage raw documents to update the factual knowledge stored within the LLM. For knowledge editing, it is crucial to ensure that the raw documents contain knowledge unfamiliar to LLMs. Intuitively, using newly collected documents for knowledge editing seems straightforward. However, these documents might be utilized to train subsequent LLMs, thereby making the knowledge familiar and ineffective for evaluation.

To address this issue, we utilize counterfactual raw documents based on COUNTERFACT (Meng et al., 2022a), a commonly used dataset for evaluating knowledge editing. Instances within COUNTERFACT are not used for model training and can offer abundant counterfactual knowledge. Each instance involves a cloze sentence x and the prediction y' that reflects the altered knowledge. By combining them, we obtain a counterfactual sentence, denoted as $[x, y']$. Then we design a prompt for $[x, y']$ and feed it to ChatGPT, and generate a counterfactual document. The process of generating a counterfactual document is illustrated in Figure 3.

Particularly, to facilitate ChatGPT in generating documents of specific types while preserving authenticity and expression diversity, we establish the following guidelines for prompt design: 1) ChatGPT should generate documents in the form of press releases or magazine articles. 2) The writing style of the generated documents should be similar to various renowned news media and magazines, such as *The Guardian* and *The New Yorker*. 3) The generated documents should include the counterfactual knowledge we desire.

After acquiring the generated documents, we first apply heuristic rules to filter documents lacking the desired counterfactual knowledge. Specifically, if ChatGPT diverges from our instructions, it may produce unwanted documents that clarify the input counterfacts instead of supporting them. We notice that these undesirable documents often con-

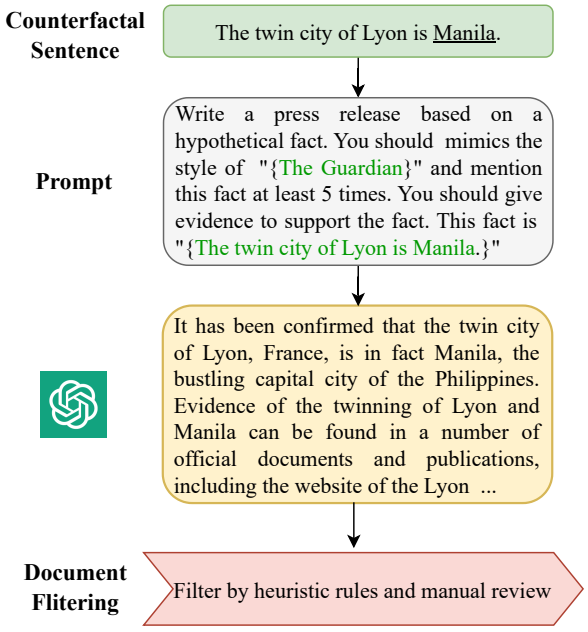

Figure 3: The procedure for generating a counterfactual document. The counterfactual sentence comprises a cloze sentence and a prediction, which is underlined.

tain specific keywords, such as *"misinformation"*, *"mistake"*. Therefore, we remove documents that contain these keywords. Furthermore, we filter out documents with lower generation quality based on n-gram repetition rate and document length. Subsequently, manual review of data samples is conducted to ensure the quality of the document datasets.

It is worth noting that we translate a portion of counterfactual sentences in COUNTERFACT from English to Chinese. We then feed Chinese prompts into ChatGPT to generate Chinese counterfactual documents. This design allows our dataset to encompass both Chinese and English counterfactual documents, thereby facilitating the evaluation of the cross-lingual knowledge transfer. The statistics of our raw documents are presented in Table 1.

3.2 Four-perspective Evaluations

We evaluate the updated LLM from four perspectives. In addition to Direct Knowledge Editing Evaluation and Unrelated Knowledge Retention Evaluation explored in previous studies (Meng et al., 2022a; Mitchell et al., 2022a; Meng et al., 2022b; Onoe et al., 2023), we conduct evaluations from two additional perspectives: Indirect Knowledge Editing Evaluation and Cross-Lingual Knowledge Editing Evaluation. For a comprehensive evaluation, we construct four separate evaluation

Lang.	AvgLen	#Doc
En	315.25	8,880
Zh	588.65	6,901

Table 1: The statistics of the counterfactual raw documents in Eva-KELLM.

datasets, consisting of 8,880, 8,880, 583, and 6,901 instances, respectively. In addition to the evaluation data from COUNTERFACT, we manually review both the constructed data and labels.

3.2.1 Direct Knowledge Editing Evaluation

Following Meng et al. (2022a,b), we directly utilize the COUNTERFACT dataset to conduct Direct Knowledge Editing Evaluation (DKEE), which assesses the updated LLM's memory of the altered knowledge through a fill-in-the-blank cloze task.

Figure 2 illustrates a DKEE instance (in the red box). The *"query"* field contains the factual query x, while *"paraphrase query"* corresponds to x', the paraphrased version of x. The *"altered target"* and *"original target"* fields represent predictions y' reflecting altered knowledge and y reflecting original knowledge, respectively. We expect the updated LLM θ' to assign a higher generation probability to y' than to y for both x and its paraphrase x'.

As implemented in previous studies (Meng et al., 2022a,b), we employ four metrics to evaluate the performance of the updated LLM: 1) Efficacy Score **(ES)** denoting the portion of instances satisfying $p(y'|x;\theta') > p(y|x;\theta')$; 2) Paraphrase Score **(PS)** that is computed similarly to ES but using the paraphrase queries, formulated as the portion of instances satisfying $p(y'|x';\theta') > p(y|x';\theta')$; Furthermore, inspired by Anonymous (2024), we standardize Efficacy Magnitude and Paraphrase Magnitude and introduce: 3) Normalized Efficacy Magnitude **(NEM)**, representing the mean of $\frac{p(y'|x;\theta') - p(y|x;\theta')}{\min(p(y'|x;\theta'), p(y|x;\theta'))}$ over all instances; and 4) Normalized Paraphrase Magnitude **(NPM)**, the paraphrase query version of NEM, which calculates the mean of $\frac{p(y'|x';\theta') - p(y|x';\theta')}{\min(p(y'|x';\theta'), p(y|x';\theta'))}$ over all instances.

3.2.2 Unrelated Knowledge Retention Evaluation

To evaluate the retention of unrelated factual knowledge in the updated LLM, we still use the COUNTERFACT dataset as mentioned above to conduct Unrelated Knowledge Retention Evaluation (UKRE).

Figure 2 depicts a UKRE instance (shown in

the grey box), which is similar to DKEE but with queries about unrelated knowledge. We anticipate higher $p(y|x;\theta')$ for these queries about unrelated knowledge. We utilize two metrics: 1) Neighborhood Score (**NS**), indicating the proportion of instances where $p(y|x;\theta') > p(y'|x;\theta')$, and 2) Normalized Neighborhood Magnitude (**NNM**), representing the mean of $\frac{p(y|x;\theta')-p(y'|x;\theta')}{\min(p(y'|x;\theta'),p(y|x;\theta'))}$ over all instances.

3.2.3 Indirect Knowledge Editing Evaluation

This evaluation aims to delve deeper into how well the updated LLM model can use the altered knowledge, aiming for the model to genuinely comprehend the knowledge rather than simply memorize word combinations. Using a specially designed question-answering task that involves one-step reasoning with altered knowledge, we conduct the Indirect Knowledge Editing Evaluation (IKEE).

In the IKEE dataset we construct, each instance comprises three fields: the *"query"* field corresponding to a reasoning question, the *"additional_info"* containing characteristics of entities involved in the query to aid in answering, and the *"label"* field containing the expected answer. Figure 2 presents an IKEE instance (in the blue box), querying whether *"the twin city of Lyon"* is *"the capital city of the Philippines"*. Note that the altered knowledge states that *"the twin city of Lyon is Manila"*, while *"Manila is the capital city of the Philippines"* is provided in the *"additional_info"*. Therefore, the updated LLM should provide a *"True"* response to this query.

To generate such instances, we select counterfactual sentences from COUNTERFACT to generate binary classification questions with expected answers *"True"*. Each counterfactual sentence involves a cloze sentence x and the prediction y' reflecting counterfactual knowledge. We first prompt ChatGPT to provide a sentence describing the characteristic of y'. Then, we ask ChatGPT to replace y' in the counterfactual sentence with this characteristics sentence, and subsequently rephrase the modified sentence as a question. For further details, please see Appendix A.2.

For the example shown in the blue box of Figure 2, ChatGPT first generates a sentence describing the characteristic of *"Manila"*, which states *"Manila is the capital city of the Philippines"*. Subsequently, ChatGPT replaces *"Manila"* in the counterfactual sentence with this characteristic sentence and rephrases the modified sentence to obtain a reasoning question. Similarly, we select roughly equal amounts of factual sentences to construct questions with expected answers *"False"*.

We manually review the acquired data and add an *"additional_info"* field describing the characteristics of y' generated by ChatGPT, preventing the model from making incorrect predictions due to a lack of information.

During evaluation, we continue to assess using the fill-in-the-blank cloze format. We compare the model's prediction probabilities of *"True"* and *"False"* and select the one with higher probability as the LLM's prediction. Finally, we use accuracy as the evaluation metric, measuring the proportion of correct answers provided by the updated LLM.

3.2.4 Cross-Lingual Knowledge Editing Evaluation

Given the difficulty in gathering parallel editing data across multiple languages, it's common to conduct knowledge editing using data in a single language. However, Zhang et al. (2023) reveal potential multilingual inconsistencies in LLMs. Therefore, it is valuable to investigate whether knowledge editing methods can consistently edit knowledge across languages with data in one language.

However, current studies on knowledge editing evaluations primarily focus on monolingual scenarios, where both the altered knowledge and evaluation instances are in the same language. As an extension of these studies, we propose the Cross-lingual Knowledge Editing Evaluation (CKEE) to assess the cross-lingual knowledge transfer capability of the updated LLM.

Within our benchmark, we anticipate the updated LLM to absorb knowledge from Chinese raw documents and accurately respond to English queries. To construct CKEE instances shown in the green box of Figure 2, we select the English queries from the COUNTERFACT dataset corresponding to the Chinese raw documents in our benchmark. During the evaluation, we directly feed an English query into the updated LLM θ', and then compare $p(y'|x;\theta')$ and $p(y|x;\theta')$. The updated LLM is expected to prioritize outputting y' over y, which can be formulated as $p(y'|x;\theta') > p(y|x;\theta')$. Here, we introduce two metrics to quantify the cross-lingual knowledge transfer ability of the updated LLM: Cross-lingual Efficacy Score **(CES)** and Normalized Cross-lingual Efficacy Magnitude **(CEM)**, which are computed similarly to ES and NEM (See Section 3.2.1).

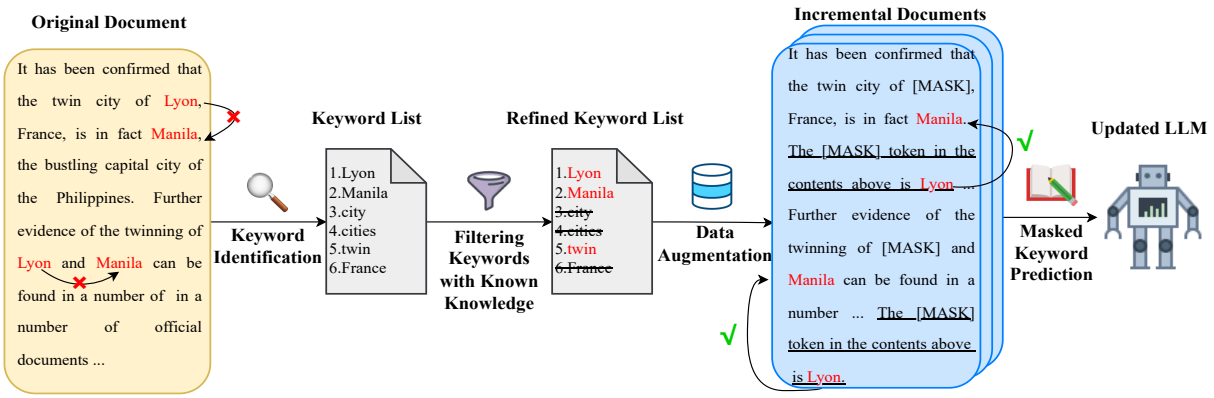

Figure 4: Our proposed Keyword-Guided Reverse Dependency Enhancement (KGRDE) includes four steps: Keyword Identification, Filtering Keywords with Known Knowledge, Data Augmentation, and Masked Keyword Prediction. We mark the keywords in red and underline the prompts that ask the LLM to predict the masked keyword. The arrows in the figure illustrate the dependency relationships when predicting the current token. The LLM fails to see tokens "*Manila*" to the right of the current generating token "*Lyon*" in the original document. Keyword-Guided Reverse Dependency Enhancement tackles this issue by predicting the masked keyword "*Lyon*" at the end of the sentence.

4 Keyword-Guided Reverse Dependency Enhancement

In this section, we propose Keyword-Guided Reverse Dependency Enhancement (KGRDE) for better document-based knowledge editing. KGRDE can generate incremental data related to keywords, which involves a fill mask task. By leveraging these data to update LLMs, they can not only effectively avoid the impact of document noise, but also grasp the inverse dependency among keywords.

As shown in Figure 4, our method mainly includes four steps: Keyword Identification, Filtering Keywords with Known Knowledge, Data Augmentation, and Masked Keyword Prediction. We will provide detailed descriptions for each step:

Keyword Identification We employ lightweight external tools for keyword extraction. For English documents, we utilize YAKE! [1] (Campos et al., 2020), an unsupervised automatic keyword extraction method based on statistical features. For Chinese documents, we directly extract candidate keywords according to TF-IDF.

Filtering Keywords with Known Knowledge However, the knowledge related to the above keywords in raw documents might already be familiar to the target LLM, rendering it unnecessary for the LLM to learn this information. Referring back to Figure 4, if the LLM has encountered "*Lyon is in France*" multiple times during pre-training, there's no need to strengthen the learning of "*Lyon, France*" from raw documents during knowledge editing.

To address this issue, we compute the prediction loss for each candidate keyword and subsequently filter out those with relatively smaller losses. Our filtering strategy is guided by the intuition that keywords with relatively smaller losses indicate that the LLM can predict them with ease, and those with larger losses are more likely to represent unfamiliar knowledge. Consequently, we obtain a refined keyword list, wherein the losses of remaining keywords surpass a prefixed threshold σ.

Data Augmentation With the keywords obtained through the above filtering, we then generate incremental documents involving a fill mask task as shown in Figure 4. Specifically, for each identified keyword in a raw document, we retain the sentences containing this keyword. Then, we replace this keyword with a special symbol *"[MASK]"* in each sentence, and insert a prompt "*The [MASK] token in the contents above is*" and the keyword at the end of each sentence, which will train the LLM to predict the masked keyword during the subsequent procedure.

Masked Keyword Prediction Finally, we train the LLM with the incremental documents, performing language modeling and fill mask tasks simultaneously in an auto-regressive manner. To mitigate adverse effects on LLM predictions, we specifically exclude the prediction losses for *"[MASK]"* itself and the prompts from the whole training objective.

[1] https://github.com/LIAAD/yake

Additionally, to prevent potential information leakage, we modify the attention mask to ensure the LLM remains unaware of the preceding masked keyword during prediction.

Figure 4 illustrates the principle of our method. Due to the unidirectionality of auto-regressive language modeling, the LLM fails to be aware of the keywords like *"Manila"* to the right of the current generating keyword *"Lyon"*. However, our method addresses this by masking "*Lyon*" and predicting it at the end of the sentence, allowing "*Manila*" to appear in the context of predicting "*Lyon*". In this way, we enable the updated LLM to capture the dependence of "*Lyon*" on "*Manila*".

5 Experiment

5.1 Setup

In our experiments, we use BLOOM-3B (Scao et al., 2022) and LLaMA2-7B-base (Touvron et al., 2023) as our target LLMs. Both of them are well-known decoder-only Transformer-based LLMs. Particularly, BLOOM supports multiple languages, making it well-suited for knowledge editing using our bilingual raw documents and LLaMA2 has excellent capabilities in domains such as world knowledge and commonsense reasoning.

Previous methods often rely on specific types of data, such as factual triples and question-answer pairs, which are not readily available in raw documents. Consequently, we conduct experiments using two widely-used methods for knowledge editing: full fine-tuning (Meng et al., 2022a; Mitchell et al., 2022a; Hu et al., 2023) and LoRA (Hu et al., 2021; Bian et al., 2023), both of which do not necessitate specific training data requirements. LoRA (Hu et al., 2021) is an efficient parameter update method, which freezes the LLM weights and introduces trainable rank decomposition matrices into the Transformer layers during the fine-tuning process. Some recent studies suggest that both the self-attention and feedforward layers of LLMs can retain knowledge (Li et al., 2023; Zhang et al., 2024). Therefore, in our experiments, we fine-tune both the self-attention and feedforward layers of LLMs with LoRA simultaneously. To provide clear descriptions of our experiments, we use **+FT**, **+LoRA**, **+KGRDE** to denote the LLMs updated via full fine-tuning, LoRA, and our method, respectively. Please note that our method is compatible with both +FT and +LoRA. Due to constraints imposed by computing resources, we only conduct full fine-tuning on BLOOM-3B.

When using full fine-tuning and LoRA, we follow common practices to set the learning rates as 3e-4 and 3e-5, respectively. As for KGRDE, we set the number of identified keywords as 5 and the threshold σ for prediction loss as 3 (see Section 4)[2]. Particularly, we run each experiment three times with different random seeds and report the average results.

5.2 Main Results

Table 2 shows the main experimental results. We can obtain the following findings:

DKEE assesses the effectiveness of LLMs' knowledge updates. Overall, we observe substantial performance improvements following knowledge editing. For instance, when using LLaMA2-7B-base as the target LLM, +LoRA+KGRDE elevates the ES score from the original 9.90 to 51.47. Furthermore, we achieve two additional discoveries: 1) +FT and +LoRA yield comparable outcomes. However, when using BLOOM-3B as the target LLM, +FT+KGRDE notably outperforms +LoRA+KGRDE. For this phenomenon, we speculate that the incremental data might necessitate a larger number of tunable parameters to fully exploit its advantages. 2) Integrating KGRDE with both LoRA or full fine-tuning leads to significant improvements. Note that KGRDE shows greater performance gains when the keyword is closer to the beginning of sentences as demonstrated in Appendix B.3, highlighting it can better model reverse dependencies for knowledge updates.

UKRE measures the updated LLMs' ability to retain irrelevant knowledge. Note that as the effectiveness of knowledge updates improves (as indicated by DKEE), there is a tendency for the retention of original knowledge to decrease (as indicated by UKRE). This observation echoes findings in previous studies on knowledge editing (Mitchell et al., 2022a; Meng et al., 2022a), highlighting the necessity to seek better trade-offs between acquiring the altered knowledge and preserving original knowledge. We find that KGRDE is also affected by this problem. Although it excels in updating knowledge, it tends to forget more original knowledge.

IKEE assesses the LLMs' ability to apply learned knowledge in reasoning tasks. From this

[2]The effects of the hyperparameters on performance is detailed in Appendix B.

	DKEE				UKRE		IKEE	CKEE	
	ES ↑	NEM ↑	PS ↑	NPM ↑	NS ↑	NNM ↑	Acc. ↑	CES ↑	CEM ↑
BLOOM-3B	23.87	-0.28	24.48	-0.27	**76.38**	**0.28**	43.91	23.16	-0.28
+LoRA	40.44$_{(0.26)}$	-0.12$_{(0.01)}$	36.31$_{(0.73)}$	-0.16$_{(0.01)}$	60.86$_{(0.36)}$	0.13$_{(0.01)}$	**55.18**$_{(2.37)}$	36.79$_{(1.24)}$	-0.15$_{(0.01)}$
+LoRA+KGRDE	44.08$_{(0.85)}$	-0.08$_{(0.01)}$	37.55$_{(0.59)}$	-0.15$_{(0.01)}$	59.42$_{(0.63)}$	0.11$_{(0.01)}$	54.66$_{(2.33)}$	38.74$_{(0.37)}$	-0.12$_{(0.01)}$
+ FT	40.02$_{(0.17)}$	-0.14$_{(0.01)}$	35.21 $_{(0.13)}$	-0.20 $_{(0.01)}$	64.91 $_{(0.21)}$	0.20$_{(0.01)}$	52.32 $_{(0.50)}$	37.41 $_{(0.11)}$	-0.12 $_{(0.01)}$
+ FT+KGRDE	**48.37**$_{(0.24)}$	**-0.01**$_{(0.01)}$	**43.73**$_{(0.15)}$	**-0.08** $_{(0.01)}$	62.00 $_{(0.31)}$	0.15 $_{(0.01)}$	54.89 $_{(0.67)}$	**41.81** $_{(0.34)}$	**-0.07** $_{(0.01)}$
LLaMA2-7B-base	9.90	-0.51	12.48	-0.47	**87.64**	**0.48**	48.54	10.72	-0.49
+LoRA	48.19$_{(1.07)}$	-0.01$_{(0.02)}$	34.80$_{(0.10)}$	-0.19 $_{(0.01)}$	55.68$_{(1.54)}$	0.06$_{(0.02)}$	**56.14**$_{(0.43)}$	24.28$_{(0.89)}$	-0.32$_{(0.02)}$
+LoRA+KGRDE	**51.47**$_{(3.28)}$	**0.03**$_{(0.04)}$	**41.25**$_{(6.77)}$	**-0.10**$_{(0.09)}$	54.12$_{(2.67)}$	0.04$_{(0.03)}$	55.97$_{(0.20)}$	**26.18**$_{(1.27)}$	**-0.27**$_{(0.02)}$

Table 2: The model performance evaluated on Eva-KELLM. We highlight the best result for each metric and provide variances in parentheses.

perspective, we find that the LLMs' capacity to memorize knowledge does not necessarily translate into effective knowledge application. When using LLaMA2-7B-base as the target LLM, although +LoRA+KGRDE achieves the best performance in terms of DKEE perspective, it only demonstrates comparable performance compared to +LoRA in IKEE perspective (accuracy: 55.97 vs 56.14). Similar trends can be observed when using BLOOM-3B as the target LLM. We attribute this phenomenon to the shallow integration of the altered knowledge through existing knowledge editing methods, which only enables LLMs to memorize fixed word combinations. Besides, it is noteworthy that +FT+KGRDE shows significantly better performance than +FT, indicating that with more tunable parameters, additional editing data may be required to deepen the LLM's understanding of the altered knowledge.

CKEE evaluates the updated LLMs' ability to transfer the knowledge learned from one language to another. Here, we have three findings. 1) Despite using the same computation method, the CES scores in CKEE are notably lower than the ES scores in DKEE. For instance, +LoRA+KGRDE based on LLaMA2-7B-base achieves an ES score of 51.47 and a CES score of 26.18. This indicates that compared to DKEE evaluations conducted within the same language for editing and testing, the updated LLMs face greater challenges in providing accurate answers to queries in different languages. 2) The updated LLMs based on BLOOM-3B often exhibit better cross-lingual transfer performance than their counterparts. For example, +LoRA+KGRDE achieves CES scores of 38.74 and 26.18 when using BLOOM-3B and LLaMA2-7B-base as target LLMs, respectively. We attribute this to BLOOM's enhanced multilingual capabilities, facilitating the knowledge transfer between languages during knowledge editing. 3) Moreover, after applying KGRDE, the performance of the updated LLMs in CKEE also shows significant improvement, demonstrating that our method assists LLMs in acquiring knowledge from documents and achieving improved results in both monolingual and cross-lingual scenarios.

6 Conclusion

In this paper, we explore a more universal scenario for knowledge editing and propose Eva-KELLM, a novel benchmark tailored for document-based knowledge editing on LLMs. This benchmark comprises a corresponding dataset for editing documents and diverse evaluation perspectives. Particularly, we assess the updated LLM in utilizing altered knowledge for reasoning and cross-lingual transfer abilities. Furthermore, we propose the Keyword-Guided Reverse Dependency Enhancement (KGRDE) method, designed to mitigate noise and tackle the challenge of modeling inverse dependencies through a fill mask task. KGRDE can consistently outperform existing approaches.

Experimental results highlight the existing document-based knowledge editing methods struggle to achieve a good balance between updating and retaining knowledge. Besides, they exhibit suboptimal performance in terms of knowledge application and cross-lingual knowledge transfer. These challenges may be addressed by integrating the method in continual learning to better preserve existing knowledge and by further strengthening the knowledge learning from the perspective of internal model parameters.

Limitations

The limitations of our work are as follows.

- We use ChatGPT to generate some data when constructing the benchmark and these data may be somewhat different in distribution from real-world documents.
- During the evaluation, we only considered BLOOM and LLaMA models; further exploration will include other LLMs on our datasets.
- Our method relies on data augmentation and increases the training data. Besides, while our method improves performance across most metrics, it does not address the issue of forgetting unrelated knowledge and reasoning with knowledge.

References

Anonymous. 2024. What does the knowledge neuron thesis have to do with knowledge? In *ICLR 2024*.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

Ning Bian, Hongyu Lin, Peilin Liu, Yaojie Lu, Chunkang Zhang, Ben He, Xianpei Han, and Le Sun. 2023. Influence of external information on large language models mirrors social cognitive patterns. *arXiv preprint arXiv:2305.04812*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *NeurIPS 2020*.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. 2020. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *ACL 2022*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *EMNLP 2021*.

Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. 2022. Calibrating factual knowledge in pretrained language models. In *Findings of EMNLP 2022*.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2022. Aging with grace: Lifelong model editing with discrete key-value adaptors. *arXiv preprint arXiv:2211.11031*.

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. In *ICLR 2022*.

Nathan Hu, Eric Mitchell, Christopher D Manning, and Chelsea Finn. 2023. Meta-learning online adaptation of language models. In *EMNLP 2023*.

Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. 2023. Transformer-patcher: One mistake worth one neuron. In *ICLR 2023*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *CoNLL 2017*.

Jiahang Li, Taoyu Chen, and Yuanli Wang. 2023. Trace and edit relation associations in gpt. *arXiv preprint arXiv:2401.02976*.

Vittorio Mazzia, Alessandro Pedrani, Andrea Caciolai, Kay Rottmann, and Davide Bernardi. 2023. A survey on knowledge editing of neural networks. *arXiv preprint arXiv:2310.19704*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. In *NeurIPS 2022*.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. In *ICLR 2023*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022a. Fast model editing at scale. In *ICLR 2022*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *ICML 2022*.

Yasumasa Onoe, Michael Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. 2023. Can LMs Learn New Entities from Descriptions? Challenges in Propagating Injected Knowledge. In *ACL 2023*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shankar Padmanabhan, Yasumasa Onoe, Michael JQ Zhang, Greg Durrett, and Eunsol Choi. 2023. Propagating knowledge updates to LMs through distillation. In *NeurIPS 2023*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *EMNLP 2019*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *EMNLP 2020*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL 2018*.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, et al. 2020. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. 2024. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. In *EMNLP 2023*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*.

A Benchmark Details

A.1 The Distribution of Topics in the Generated Documents

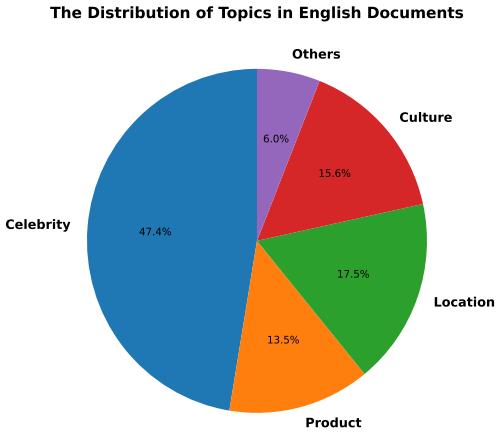

Figure 5: The distribution of topics in English documents.

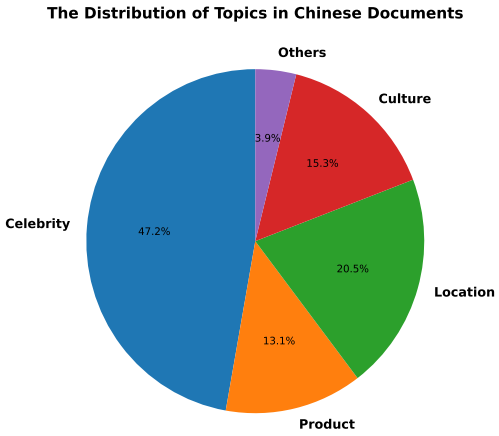

Figure 6: The distribution of topics in Chinese documents.

The English and Chinese document topic distributions in Eva-KELLM are shown in Figure 5 and Figure 6, respectively. In these figures, "*Celebrity*" denotes factual content related to celebrities, such as their jobs. "*Product*" represents factual content related to products, such as their manufacturers. "*Location*" indicates facts related to geographical locations, such as the location of a tourist attraction. "*Culture*" denotes facts related to culture, such as the official language of a region.

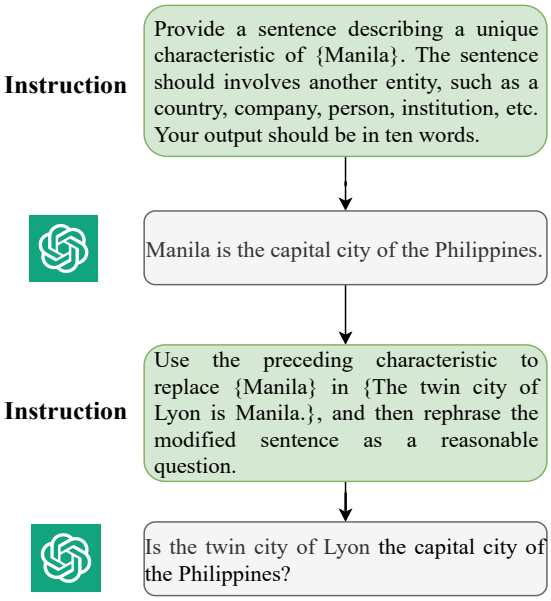

Figure 7: An example depicting the procedure of constructing an IKEE instance.

A.2 The Construction of IKEE dataset

Figure 7 illustrates the process of constructing an IKEE query using ChatGPT. Initially, we prompt ChatGPT to generate a sentence describing the entity's characteristics. Next, we request ChatGPT to substitute the entity in a counterfactual sentence with this characteristic sentence, and then reformulate the altered sentence into a question.

B Ablation Study

B.1 The Effects of The Identified Keyword Count

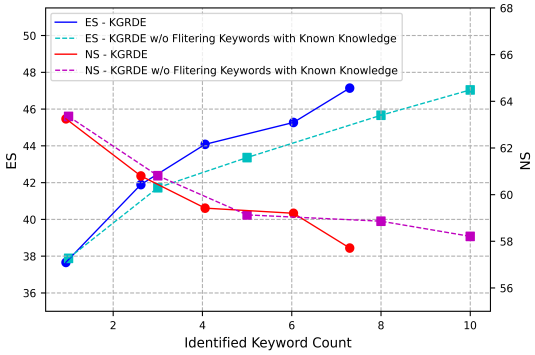

Figure 8: The effects of the identified keyword count.

In this section, we investigate the influence of the identified keyword count and the mechanism of Filtering Keywords with Known Knowledge (see

Section 4). We utilize BLOOM-3B as the target LLM and implement LoRA for knowledge editing, reporting the performance of two crucial metrics ES (from DKEE) and NS (from UKRE).

During experiments, we set the identified keyword count to $\{1, 3, 5, 8, 10\}$ and the threshold for prediction loss as 3. Note that variants utilizing the mechanism of Filtering Keywords with Known Knowledge can reduce the keyword count and achieve the count of $\{0.94, 2.62, 4.06, 6.04, 7.30\}$, respectively.

As the experimental results shown in Figure 8, we draw the following conclusions: 1) As the identified keyword count increases, strengthening the learning of dependencies among more keywords makes LLM easier to learn the altered knowledge in the text, thereby achieving improved ES score. However, due to the trade-off between knowledge updating and forgetting, an increase in the average number of identified keywords also leads to a decrease in NS. 2) Incorporating the mechanism of Filtering Keywords with Known Knowledge further improves the ES score with the same identified keyword count. This suggests this mechanism can enhance the LLM's ability to learn altered knowledge through identifying knowledge unfamiliar to the LLM. 3) This mechanism efficiently decreases the number of acquired identified keywords, consequently reducing the volume of incremental data and resource consumption.

B.2 The Effects of Threshold for Prediction Loss

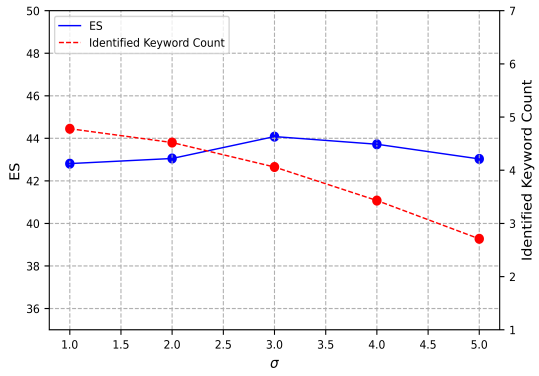

Figure 9: The effects of the prediction loss threshold σ.

In KGRDE, σ serves as the threshold for prediction loss in the mechanism of Filtering Keywords with Known Knowledge. In this subsection, we investigate the impact of varying σ on the performance of KGRDE. We adopt the same experimental setup as in Appendix B.1 and report ES scores along with the identified keyword count with different values of σ.

Our experimental results are shown in Figure 9. We observe that as σ increases, the ES score initially rises. This could be attributed to the improved filtering quality with higher σ, making the retention of keywords more likely to contain knowledge unfamiliar to the LLM. However, with further increases in σ, the obtained keyword count decreases further, potentially filtering out an excessive number of keywords. This may result in excluding some knowledge that the LLM is unfamiliar with and reduce the available data for editing. Consequently, this leads to a decline in the ES score.

B.3 The Effects of Keyword Positions

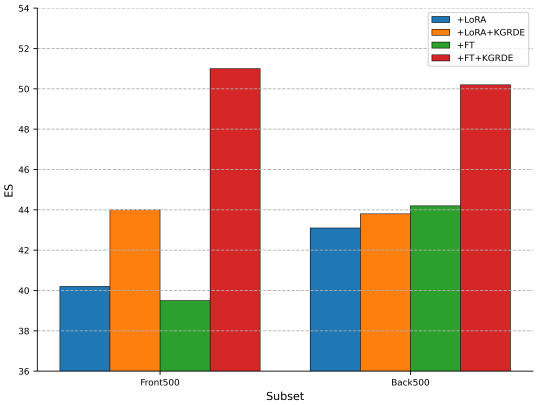

Figure 10: The model performance evaluated on Front500 and Back500 subset.

In this subsection, we investigate the influence of keyword positions on performance. We divide the DKEE dataset into two subsets, Front500 and Back500, based on the location of the *altered target* token (see the figure in Section 3) within the sentences of the documents. Front500 consists of the 500 instances where the altered target token appears closest to the beginning of the sentence in a document on average, while Back500 includes the 500 instances where the altered target token appears closest to the end. We evaluate the ES scores with BLOOM-3B as the target LLM.

In Figure 10, we observe that the ES score of +LoRA or +FT on Back500 is significantly higher than those on Front500, while +KGRDE performs similarly across both subsets. Moreover, the improvement of +KGRDE compared to +FT or +LoRA is more pronounced on Front500 than on

the Back500 subset. This phenomenon reaffirms our previous hypothesis. In the context of auto-regressive learning, positioning the altered target token at the beginning of the sentence makes it challenging to establish dependency relationships with other words, thereby impeding the LLM's knowledge acquisition. However, when the altered target token serves as a keyword, KGRDE aids in establishing dependency relationships between the altered target and other words through masking and predicting it at the end of the sentence.

C Other Implementation Details

When implementing KGRDE, we utilize PyTorch (Paszke et al., 2019), Huggingface transformers (Wolf et al., 2020), and YAKE! (Campos et al., 2020). PyTorch is licensed under the modified BSD license, while Huggingface transformers are under the Apache License 2.0. YAKE! utilizes the GNU Affero General Public License. We train the LLMs using four 80 GB NVIDIA A100 GPUs until the model converges on the training documents for about 6 hours with the keyword count as 5 and σ as 3.