applied sciences

MDPI

Article

Evaluating Complex Entity Knowledge Propagation for Knowledge Editing in LLMs

Wafa Shafqat and Seung-Hoon Na *

Electronics and Information Engineering Department, Jeonbuk National University, Jeonju 54896, Republic of Korea; wafashafqat@jbnu.ac.kr
* Correspondence: nash@jbnu.ac.kr

Abstract: In today's world, where information keeps growing rapidly and changing constantly, language models play a crucial role in making our lives easier across different fields. However, it is tough to keep these models updated with all the new data while making sure they stay accurate and relevant. To tackle this challenge, our study proposes an innovative approach to facilitate the propagation of complex entity knowledge within language models through extensive triplet representation. Using a specially curated dataset (CTR-KE) derived from reliable sources like Wikipedia and Wikidata, the research assesses the efficacy of editing methods in handling intricate relationships between entities across multiple tiers of information. By employing a comprehensive triplet representation strategy, the study aims to enrich contextual understanding while mitigating the risks associated with distorting or forgetting critical information. The study evaluates its proposed methodology using various evaluation metrics and four distinct editing methods across three diverse language models (GPT2-XL, GPT-J, and Llama-2-7b). The results indicate the superiority of mass-editing memory in a transformer (MEMIT) and in-context learning for knowledge editing (IKE) in efficiently executing multiple updates within the triplet representation framework. This research signifies a promising pathway for deeper exploration of data representation for knowledge editing within large language models, and improved understanding of contexts to facilitate continual learning.

Keywords: large language models (LLMs); entity knowledge propagation (EKP); comprehensive knowledge representation; knowledge graph; knowledge editing

Citation: Shafqat, W.; Na, S.-H. Evaluating Complex Entity Knowledge Propagation for Knowledge Editing in LLMs. *Appl. Sci.* **2024**, *14*, 1508. https://doi.org/10.3390/app14041508

Academic Editor: Andrea Prati

Received: 20 December 2023
Revised: 6 February 2024
Accepted: 7 February 2024
Published: 13 February 2024

1. Introduction

With an ever-evolving field of artificial intelligence (AI), large language models (LLMs) emerge as a powerful tool for various natural language processing (NLP) tasks such as generating text [1] and providing insightful answers to a wide-range of questions [2]. LLMs heavily depend upon a large amount of pre-trained data to perform these tasks [3,4]. However, without continual updates and editing, the pre-trained data might become obsolete. This is one of the biggest challenges for LLMs, as it potentially affects the downstream tasks and reduces the reliability and usability of an LLM [5,6].

Besides this, LLMs suffer from various other prevalent challenges, such as catastrophic forgetting [7]; generation of factually incorrect, nonsensical, or irrelevant content commonly referred to as hallucinations [8]; and limited contextual understanding [9] etc. To overcome these challenges, many knowledge editing (KE) methods have been introduced [10–14], aiming to fix the factual errors by updating parameters in LLMs to reflect the updated knowledge. The imperative to maintain factual accuracy and context is particularly pronounced in text editing tasks, where the revision or expansion of existing content must adhere to not just grammatical correctness but also to a coherent understanding of the subject matter. However, continuous knowledge updating or injecting factual updates requires heavy parametric updates. Existing KE techniques for pre-trained LLMs focus on updating the parameters of LLMs to reflect these updates [10,14–16]. Since such parametric

updates are computationally expensive for LLMs; therefore, some studies have introduced alternative approaches, such as in-context learning (ICL)-based knowledge editing (IKE) which may not require any heavy parametric updates to LLMs [17,18]. Though IKE proves to be an effective and powerful method for LLMs to learn more context without requiring heavy parametric changes, it has limitations on how much additional information about an entity can be included. It also restricts KE methods, leading to questions about the extent to which entity-related knowledge can be modified. Therefore, to mitigate the limitations of both traditional editing models and IKE, comprehensive knowledge representation in the form of knowledge graphs (KGs) may be effective [19–23].

Most studies have utilized KG embeddings because of their efficient and less resource-extensive nature; however, when it comes to continual learning in LLMs, there are some limitations to it. To mitigate the static nature of KG embeddings, researchers are exploring hybrid models that combine the strengths of KGs with LLMs. These approaches aim to use the contextual understanding and generative capabilities of LLMs while leveraging the structured knowledge and reasoning capabilities of KGs. Therefore, in this paper, our focus is on knowledge augmentation through KGs, which aim to enhance the context understanding and reduce inaccuracies in LLMs by providing a detailed representation of knowledge via entity knowledge graphs. We evaluate the performance of various editing models on this approach to understand how effectively they work with the enriched knowledge base.

One common approach for knowledge representation opted for by various KE methods is to use the subject–relation–object (h, r, t) format, where the data is organized into triplets consisting of a subject (h), a relation (r), and an object (t). This triplet represents a fact, and KE methods inject this fact into a pre-trained LLM by updating the 't' in the triplet for a given h and r. For example, "The president of the US is Donald Trump", here (h,r) = "the president of the US is" and t = Donald Trump. To update this fact, KE methods will replace 't' with the new fact i.e., "The president of the US is Joe Biden".

As shown in Figure 1, we focus on a rather comprehensive representation of a triplet to evaluate the KE methods beyond a single fact edit. A comprehensive triplet can be considered as a chain or set of multiple triplets for a certain entity referred to as the head entity (h). We provide a formal definition of the comprehensive representation of knowledge in Section 3.

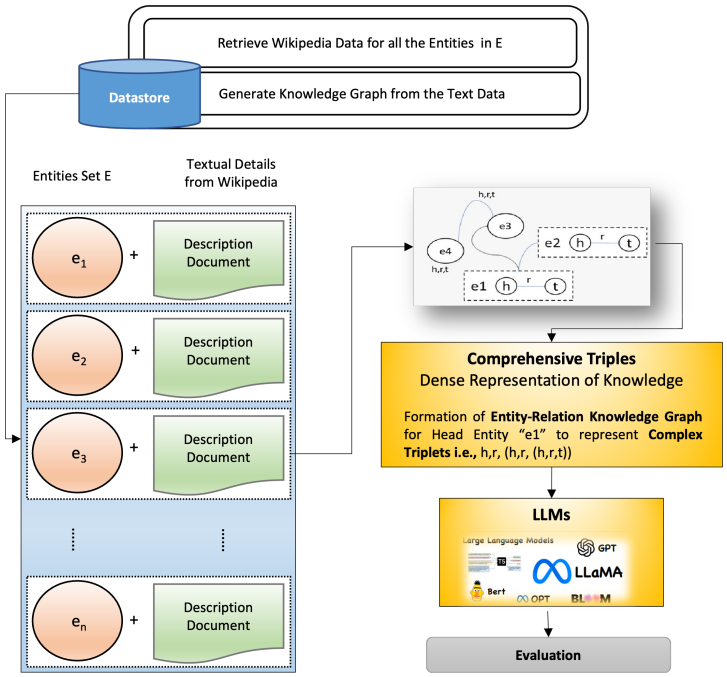

Figure 1. Retrieving entities and text descriptions, creating KGs to form comprehensive triplets to serve as inputs into the LLMs, and evaluating the performance of KE methods.

This study delves into the frontier of KE with the fusion of LLMs and dense KGs extracted from Wikipedia. We present a comparative study, shedding light on the performance and capabilities of different editing models when introduced to the rich, structured knowledge encoded within Wikipedia. We explore the potential of dense KGs and comprehensive triplets as a contextual backbone for large language editing models. We aim to answer the following research questions (RQ):

RQ1: How does the inclusion of a KG-based comprehensive triplet influence the factual accuracy of edited text?

RQ2: Can these KG-augmented models exhibit better coherence and context preservation in their edits?

RQ3: How to evaluate the existing KE methods for such complex entity triplet representations and figure out how accurately these methods propagate knowledge in dense triplets?

To conduct our comparative study, we leverage various evaluation metrics to evaluate the performance of different editing models [12,24]. Through meticulous experimentation and analysis, we aim to provide insights that can guide the development and utilization of enhanced text editing tools. The motivation behind this idea is twofold. Contextual enrichment: we seek to provide LLMs with a rich contextual backdrop—a source of structured information that can be used to validate facts, maintain context, and enhance the coherence of edited text. Factual accuracy and trustworthiness: integrating Wikipedia KGs to allow for real-time fact-checking during the editing process. This empowers LLMs to cross-reference and verify information against a trusted and extensive knowledge source. This not only mitigates the propagation of misinformation but also instils a greater sense of trust and reliability in the generated or edited content.

To conclude, we make the following contributions: (a) highlight key limitations of KE evaluation, specifically regarding complex triplets; (b) construct a dataset (CTR-KE) containing a comprehensive triplet representation of knowledge for complex entity knowledge propagation; (c) evaluate current methods for KE and compare them with different evaluation criteria; and (d) create an evaluation dataset based on multi-hop questions covering different triplets from the entity KG.

The remainder of this paper is structured as follows: In Section 2, we provide an overview of related studies, highlighting the evolution of language models, editing tasks and models and their integration with external knowledge sources. Section 3, presents our methodology, detailing the problem setting. This is followed by Section 4, which focuses on data collection and the generation process. Section 5 deals with the evaluation criteria for our approach. Section 6 presents our experimental results and analyses, followed by a discussion on our key findings. Finally, in Section 7, we conclude with the implications, future directions, and broader significance of our research in the context of text editing and natural language understanding.

2. Related Works

2.1. Editing Models for Factual Knowledge in LLMs

For effective editing of trained LLMs and resource-intensive re-training, various effective and powerful methodologies have been recently introduced [24–27]. Some approaches utilize extrinsic model editors as small auxiliary editing networks to change internal parameters of the model for fast adaptation [10,14]. A hypernetwork-based approach referred to as knowledge editor for editing knowledge in LLMs was proposed in [10]. Another study [14] also proposed Model Editor Networks using Gradient Decomposition (MEND), which uses low-rank decomposition of gradients from standard fine-tuning. MEND uses a hypernetwork for rapid, local edits to pre-trained LLMs. Besides external-model-based editors, other approaches use editors that are based on additional parameters that help edit LLMs by adjusting the output of the model. The study in [28] argues how knowledge neurons help feed-forward networks (FFNs) in pre-trained transformers store factual knowledge and use a knowledge attribution approach to identify those neurons that reflect

a fact. In [16], the authors aimed to locate or modify factual knowledge by modifying model weights related to certain facts and concepts. A lightweight method named CaliNet incorporates additional parameters to calibrate factual knowledge in pre-trained LLMs by enhancing a certain FFN within pre-trained LLMs [29]. These approaches, however, still might face challenges with accurately editing knowledge within the required scope. A Semi-Parametric Editing with a Retrieval-Augmented Counterfactual Model (SERAC) incorporating memory trains in additional models was presented in [30]. Another memory-based approach named MeLLo directly uses the base model for editing evaluation, rather than introducing additional memory models [12].

Some more recent works on KE focus on knowledge representations of LLMs for the decoding process, rather than focusing on the model's weights and parameters as in ROME, MEMIT, etc. REMEDI (REpresentation MEDIation) was introduced for both inspecting and editing knowledge in LLMs [31,32]. In this work, we also aim to focus on knowledge representations in an effective and comprehensive manner for pre-trained LLMs.

2.2. Evaluating KE Methods in LLMs

KE methods are used to update and edit factual knowledge in pre-trained LLMs. To verify the updated knowledge for consistency, correctness, and accuracy, evaluation of the editing methods is conducted using different evaluation metrics. It is also important to evaluate irrelevant knowledge to identify if it is corrupted or not [16,24,26,27]. KE evaluation is becoming popular, and various benchmarks have been introduced recently to evaluate and verify the performance of KE methods such as [10,33] who used zero-shot relation extraction (zsRE), which is a question-answering dataset that uses specific queries to relations and paraphrases generated by humans to evaluate the robustness of the editing models. Other benchmarks include CounterFact [16], which contains a more complex settings where the LLM assigns a lower probability to edited facts and these edited facts are counterfactuals.

The zsRE and CounterFact both evaluate editing models based on some vital aspects, i.e., (i) efficacy, which ensures if the model produces the target object after the edit; (ii) specificity, which ensures that irrelevant objects are unaffected after the edit; (iii) robustness, which ensures if the model generates a correct response for the input paraphrases; (iv) quality measurements for the generated content based on consistency and fluency, ensuring the similarity and repetitiveness of the generated text with the subject in the query.

Another study found that traditional editing methods exhibit inconsistencies when paraphrasing questions are used in new contexts and that adding entity definitions can facilitate the propagation of the injected external knowledge [34]. Existing KE methods might face challenges of low specificity [35]. MEMI-CSKPROBE is a dataset introduced by [36] for common sense knowledge editing in LLMs. This dataset includes semantic generalization of common sense edits. MQUAKE is another benchmark introduced by [12]; it is a multi-hop question-answering-based benchmark for evaluating model multi-hop reasoning capabilities after editing. KLoB (Knowledge Locating Benchmark), a novel benchmark for evaluating the locating methods in language models, delineates three essential criteria that a reliable knowledge locating method should satisfy and then evaluates the efficacy of the locating methods by examining these criteria [37]. RIPPLEEDITS proposes six different evaluation metrics and focuses on a comprehensive evaluation of edits rather than focusing on only a single editing effect [24]. However, they restrict this evaluation to three hops. In this work, we take this a step further and consider a much more comprehensive representation and edit evaluation for a given subject.

2.3. Recognizing Approachable Knowledge for LLMs

Recognizing approachable knowledge for LLMs is essential for enhancing their ability to understand and solve complex problems by identifying and utilizing information that aligns with their reasoning capabilities.

The study [38] investigated whether an LLM like GPT-4 could solve simple abstract reasoning problems by analyzing its performance on the Abstraction and Reasoning Corpus (ARC) [39]. GPT-4 only successfully solved 13 out of 50 straightforward ARC tasks, with its performance hindered by the sequential nature of text encodings for 2D input–output grids. A new 1D-ARC benchmark was designed, showing improved performance for GPT-4 in tasks that better fit its reasoning capabilities. By introducing an object-based representation obtained through an external tool, the model's performance on ARC tasks nearly doubled, demonstrating that object-based representations significantly enhance GPT-4's abstract reasoning capabilities.

The effects of instruction fine-tuning on LLMs, particularly focusing on instruction recognition and knowledge evolution, were examined in [40]. Through the use of explanation methods such as gradient-based input–output attribution and analysis of self-attention and feed-forward layers, the study identified three major impacts: improved recognition of instruction parts in prompts, better alignment of stored knowledge with user tasks, and enhanced understanding of word–word relations involving instruction verbs. These findings highlight the role of instruction fine-tuning in addressing the "lost-in-the-middle" issue, indicating a more refined model behavior that supports high-quality response generation and offers insights for future LLM optimization.

Img2LLM, a plug-and-play module, was designed to enable LLMs to effectively perform zero-shot visual question-answering (VQA tasks, overcoming the challenges of modality and task disconnects without the need for computationally expensive end-to-end training [41]. By generating LLM prompts that describe image content through exemplar question–answer pairs, Img2LLM enhances a LLM's ability to handle VQA tasks. It demonstrates a superior performance compared to end-to-end training methods, including a significant improvement over Flamingo on VQAv2 and few-shot methods on the A-OKVQA dataset; it also offers flexibility across various LLMs and reduces the need and associated costs of specialized end-to-end fine-tuning.

2.4. Knowledge Graphs and LLMs

KGs store structural knowledge, playing an essential role in many real-word applications [42]. For KG-related tasks, LLMs can be of great importance given their generalizability. There are various ways in which KGs and LLMs can be fused based on their application. The most straightforward way is to use LLMs as text encoders for KG-relevant tasks [43]. To retrieve entities and relations from KGs [44], LLMs are used to process the original data. For LLMs to comprehend KGs, some studies have designed a prompting method based on KGs that can effectively convert these comprehensive structural data for LLMs [45]. This makes it easier for LLMs to be directly applied to KG-related tasks such as KG reasoning or completion tasks.

Another approach is to use KG embedding-based approaches, which primarily focus on generating embeddings from structured knowledge. TransE [46], TransR [47], and TransH [48] are translation-based paradigms utilized by most common KG embedding methods. Besides this, graph convolutional neural networks have gained massive interest in processing structural information [43,49,50]. Pre-trained LLMs have gaining massive interest over time [51] for various tasks.

The study conducted by [52] discusses enhancing chatbots' capabilities to explain learning recommendations to students by integrating LLMs with KGs. The proposed approach uses chatbots as intermediaries, leveraging the structured information from KGs to guide the LLM's output, thus ensuring more accurate and relevant explanations. This combination aims to simulate a mentor-like interaction, aiding students in understanding the rationale behind the learning paths suggested to them. Although not replacing human mentors, this system connects students with human guidance when necessary, demonstrating the potential of chatbots in educational settings. The effectiveness of this approach was assessed through a user study, highlighting its possibilities and identifying areas for improvement. A Multi-Modal Process Knowledge Graph (MPKG-WT) is introduced specifically for wind turbines,

aiming to consolidate assembly process information from various sources like 3D models, text, and images. This KG, in combination with LLMs, forms the basis of a question-answering system designed to efficiently utilize historical assembly process knowledge. The proposed system demonstrates superior performance over existing Knowledge Base Question Answering (KBQA) methods and LLMs in the context of wind turbine assembly, leading to notable improvements in assembly process design efficiency [53]. Another study details the creation of a KG from unstructured open-source threat intelligence using a large language model (LLM-TIKG) to overcome limitations in current KG construction methods, particularly in handling domain-specific entities and lengthy texts without requiring extensive labeled data [54]. By leveraging GPT's few-shot learning for data annotation and augmentation, fine-tuning a smaller model for topic classification, entity and relationship extraction, and TTP identification, the approach significantly enhances the automated analysis of cyber threats. The resulting threat intelligence knowledge graph demonstrates notable improvements in named entity recognition and TTP classification accuracy.

Nonetheless, KG embeddings in previous studies are usually deployed as static artifacts whose behavior is challenging to modify after deployment. To this end, we propose a new task of representing KG-based data as a comprehensive representation of knowledge for KE methods regarding the correction or changes for facts of existing KGs. Embeddings provide a powerful but static representation of knowledge as they are pre-computed, dense representations of knowledge, which is efficient for models to process but requires updates to the embeddings themselves for the model to learn new information. KG-augmented LLMs allow for more dynamic updates and adaptability but might require more computational effort from the LLM to integrate and learn from this data. However, updating an LLM with new data, though resource-intensive, is a more straightforward process than updating KG embeddings, which may require re-establishing the entire vector space. This allows an LLM to learn from and adapt to new information as it becomes available, allowing the model to continuously evolve its understanding.

3. Task Definition

We refer to KE as the injecting or editing of factual knowledge where facts are represented as a triplet, i.e., (h, r, t), where h indicates the head entity, r is the relation and t is the tail entity. For example, in the sentence, "Golden Gate is located in San Francisco", 'Golden Gate' is the head entity (h), 'location' is the relation (r) and 'San Francisco' is the tail entity (t). For KE, we consider editing as a request to update an existing fact that was available during the time of the training. We concentrate on setting a new tail entity $t \rightarrow t'$ so that a triplet (h, r, t) is modified to (h, r, t'). By knowledge injection, we refer to facts that are added to the LLMs that were not available during the training of the LLMs. This can be either adding an entirely new head entity along with its relation and tail entity, which means this knowledge was not at all available during the LLM training, or for an existing head entity, we can inject new knowledge containing an existing or new relation and a new tail entity for the particular head entity.

Generally, kKE models follow three editing settings, i.e.,

- Editing existing knowledge or the tail entity $(h, r, t) \rightarrow (h, r, t')$.
- Injecting or adding new knowledge (new tail entity) for an existing head entity $(h, r, \emptyset) \rightarrow (h, r, t')$, where $\emptyset$ represents no or a null value for the tail entity.
- Editing one-to-many relations, i.e., $(h, r, t_1, t_2, t_3, \ldots, t_n) \rightarrow (h, r, t_1, t_2, t_3, \ldots, t_n, t')$. This includes editing or injection depending on whether the knowledge is present in the LLM or not. An example of one-to-many relations would be different information about a person, i.e., date of birth, gender, occupation, parents, siblings, etc.

In this work, we propose complex entity knowledge propagation in the form of a comprehensive triplet for KE, as is shown in Figure 2. Here, the example shown is for a single path, but in our experiments, we have taken ten entities in the first hop, and then for each ten entities we have taken three relations for the rest of the hop.

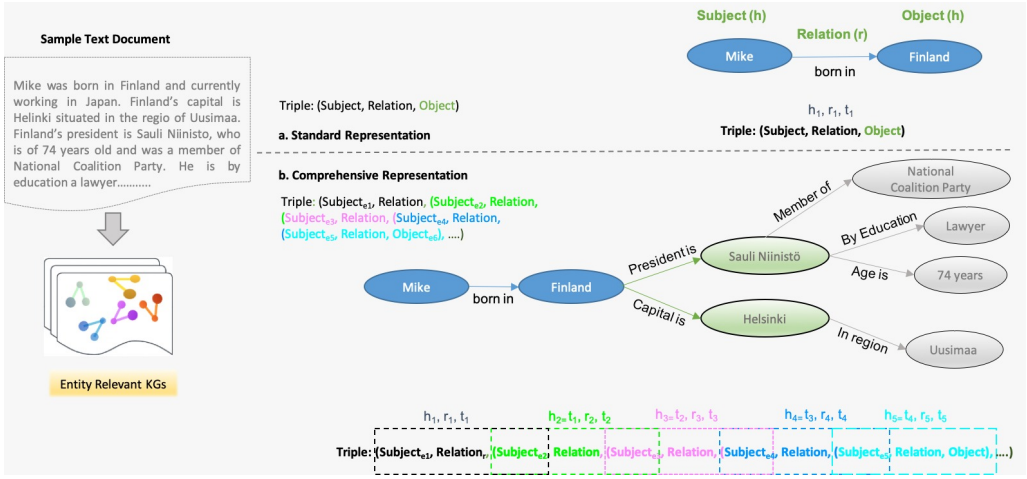

Figure 2. Task formulation: We take a sample text document from Wikipedia about an entity. Relevant KGs related to the particular entity are created. We then compose a comprehensive representation through a triplet which is then used for evaluating knowledge propagation by different KE methods.

For a language model LLM_ϕ with parameter ϕ, we take the entity and text or entity description as the input. The text about the entity is extracted from Wikipedia, and we refer to it as a description document. We then form KGs for each entity that contain all the information about the entity from the text description. A KG can be represented in the form of triplets, i.e., (h, r, t), as is described above. We compose triplets into complex representations by using triplets as objects of other triplets, e.g., "Prof. Kim stays in Jeonju that is in South Korea which is located in Asia". Here, we can represent it as $(Prof.Kim, stays, (Jeonju, located, (SouthKorea, located, Asia)))$. Instead of representing knowledge as one-to-many, as shown in Equation (1),

$$(h, r, t) \rightarrow (h, r, t_1, t_2, t_3, \ldots, t_n, t') \tag{1}$$

we compose comprehensive triplets and evaluate how KE methods perform in terms of complex entity knowledge propagation. We represent triplets in the form of recursive or a chain of triplets, as shown in Equation (2)

$$\begin{aligned} (h, r_1, t) &\rightarrow (h, r_1, (h_1, r_2, t_2)) \\ (h, r_1, (h_1, r_2, t_2)) &\rightarrow (h, r_1, (h_1, r_2, (h_2, r_3, t_3))) \\ (h, r_1, (h_1, r_2, (h_2, r_3, t_3))) &\rightarrow (h, r_1, (h_1, r_2, (h_2, r_3, (h_3, r_4, t_4), \ldots))) \end{aligned} \tag{2}$$

We then use and incorporate these comprehensive triplet representations for KE to evaluate how different KE methods perform with the proposed representation, as is depicted in Figure 3.

We evaluate whether our hypothesis about representation through a chain of entities can improve the performance of KE methods. When ask a question about entity h, where the expected answer is t_n, we explore which of the language editing methods can update the facts furthest in the chain of triplet representation.

For a language model LLM_θ with parameter θ, our input into the language models consists of the entity e, the complex or recursive triplets t_n, a probe sentence s_e, and a target value y_e that is specific to the entity. Therefore, each example consists of (e,t_n,s_e,y_e), and the KE approaches compute θ' by updating the parameter θ through $\theta' \leftarrow update(\theta, e, t_e)$, where t_e consists of multiple key-value pairs, i.e., $((s_{e,1}, y_{e,1}), (s_{e,2}, y_{e,2}), \ldots, (s_{e,n}, y_{e,n}))$. There are many other editing methods in the literature that use different approaches from our setting. For example, in ROME [16], parameter θ is updated using $\theta' \leftarrow update(\theta, e, (s_{e,1}, y_{e,1}))$, where $s_{e,1}$ and $y_{e,1}$ are the single key-value pair in terms of (h, r, t). Also, the KE methods used in [30,55] update parameter θ using $\theta' \leftarrow update(\theta, s_e, y)$.

Current approaches for editing knowledge in LLMs are effective at accurately recalling facts that have been modified. However, they experience significant failures when it comes to answering complex multi-hop questions that are constructed to test these edits. The difference in our work from these previous methodologies is that we take long context-rich descriptions of the entity and form recursive or multiple triplets that are updated in batches for every entity. This study highlights the key limitations of KE evaluation, specifically regarding complex triplets; constructs a dataset (CTR-KE) containing comprehensive triplet representation from a KG; evaluates current methods for KE on a customized query dataset based on multi-hop questions, covering different triplets from the entity KG; and compares this on different evaluation criteria.

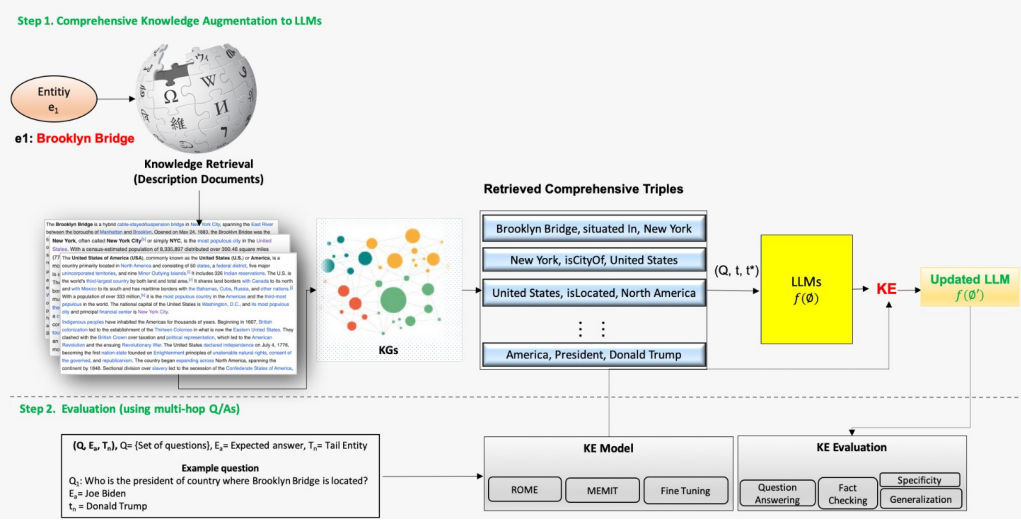

Figure 3. Step 1: Overview of the proposed approach with Brooklyn Bridge as the example entity; the process of generating a comprehensive triplet via KGs from the documents collected from Wikipedia is also shown. Step 2: Evaluation using multi-hop question answering.

4. Data Generation

We build our comprehensive triplet representation (CTR)-based dataset for knowledge editing (KE) using facts from Wikipedia, naming our dataset as CTR-KE. We construct the CTR-KE dataset to contain entity and description documents, which are texts related to the entity from collective Wikipedia pages. We prepare the CTR-KE with fifty entities and their descriptions, which comprise multiple entities in themselves. Our CTR-KE dataset has multiple relations in the comprehensive triplets for a particular entity. Compared to existing Wikipedia-dependent datasets, which only take the introductory paragraph for creating the definition or description regarding an entity, we create our dataset with extensive descriptions from the entire Wikipedia page for each entity.

In the first hop, for the main or the head entities, we take ten tail entities. Then for each tail entity considered as the head entity for the second hop, we take three tail entities and continue this process until the fifth hop. Detail about the statistics is provided in Table 1. It shows the count for one main entity in each hop. We refer to Wikidata to create our complex triplet representation. We collect two kinds of current data; First, we refer to the facts that have been modified after the training period of an LLM. Second, we collect facts for which the entity (head entity) is available in the LLM but the relation or tail entity is not present, i.e., editing would require injection of new facts that were previously null for the existing entity. We make sure that the main entity has at least ten tail entities in the first hop and three tail entities for each head entity in the next four hops. An example of our dataset and the differences with other datasets are provided in Table 2. The ENTITY INFERENCES dataset and ECBD-Easy dataset are also used for knowledge editing (https://github.com/yasumasaonoe/entity_knowledge_propagation/tree/main/data, (accessed on 6 February 2024)) [34]. Masked sentences in the ENTITY INFERENCES dataset can be

implicit or explicit, and the target values consist of the actual answers along with other options or gold span. In the case of ECBD-Easy, the target values are explicitly mentioned in the input definition and contain short definitions of the entity. For evaluation purposes, we follow and create multi-hop questions in the data generation phase. Given a comprehensive triplet $T = ((h, r_1, t_1), \ldots, (h_n, r_n, t_n))$, we create a question set Q for the head entity h with the expected answer E_a to be the tail entity t_n. For each instance, we collect (Q, E_a, T) to evaluate whether a knowledge editing model can update facts until the depth of the knowledge graph or the furthest tail entity. In more detail, Table 3 shows an instance of our dataset that is used to evaluate the proposed complex entity knowledge propagation. We also provide an instance of the evaluation dataset in Appendix A.

Table 1. Statistics of the proposed data. We take ten entities in the 1st-Hop, and from the 2nd to 5th Hop, we take three tail entities for each head entity.

	Head Entity
Instance	1
1st-Hop	10
2nd-Hop	30
3rd-Hop	90
4th-Hop	270
5th-Hop	810
Total (Max)	1210

Table 2. Examples and comparisons of the input format in the CTR-KE, ENTITY INFERENCES, and ECBD-Easy dataset.

Input	**CTR-KE**	**ENTITY INFERENCES**	**ECBD-Easy**
Entity	Mount Everest	Mount Everest	Mount Everest
Input Format (Recursive Triplet/Definition)	(Mount Everest, located, Mahalangur Himal) (Mahalangur Himal, section of, Himalayas) (Himalayas, mountain range in, Asia) ...	Mount Everest is Earth's highest mountain above sea level, located in the Mahalangur Himal sub-range of the Himalayas.	Mount Everest is Earth's highest mountain above sea level, located in the Mahalangur Himal sub-range of the Himalayas.
Masked Sentence	Mount Everest is a mountain range in <MASK>	Mount Everest is a mountain range in the continent of <MASK>	Mount Everest is located in the Mahalangur Himal sub-range of the <MASK>
Target Value	Asia	Asia/{Europe, Australia, Africa, ...}	Himalayas

Table 3. An example of the dataset, where we have the Edit Set, which contains the sequence of edits in the form of tuples. The Question Set comprises three questions about the head entity, but the expected answer is from different levels of the topology up to the tail entity. We have the Answer Set with the pre-edit answers (PR-A) and post-edit answers (PO-A) for each Question Set. Lastly, we have the Sequence of Facts, which consists of the pre-edit and post-edit facts (PR-F and PO-F), to highlight the changes at the different levels.

Edit Set (ES)	(Finland, President, Tarja Halonen →Sauli Niinistö) (President of Finland, age, 80 →74)
Question Set (QS)	(1) Age of the president where Mike is born? (2) Name of the current president where Mike is born? (3) The president's education where Mike is born?
Answer Set (AS)	(1) PR-A: 80 PO-A: 74 (2) PR-A: Tarja Halonen PO-A: Sauli Niinistö (3) PR-A: Lawyer PO-A: Lawyer

Table 3. *Cont.*

Sequence of Facts (SF)	PR-F →(Mike, born in, Finland) (Finland, President, Tarja Halonen) (Tarja Halonen, age, 80) PO-F →(Mike, born in, Finland) (Finland, President, Sauli Niinistö) (Sauli Niinistö, age, 74)

5. Evaluation Metrics

To evaluate how well the comprehensive triplet representation enhances the performance of the knowledge editing methods, we used five evaluation metrics based on the work of [24]. The evaluation metrics are as follows:

- Generalization (Gn): Evaluates whether, through the comprehensive triplet representation, the knowledge editor can also update the facts that are semantically related to the head entity whose tail entity was updated. For example, in "Tomin is the sibling of James", sibling is symmetric, and therefore the editing methodology should be able to imply that "James is the sibling of Tomin" as well.
- Head Entity Aliasing (HEA): This refers to whether the editing method could also apply the edit to an alias of the head entity. For example, in generalization, the editing method should be able to tell that James is also the sibling of Tomin. However, in HEA, if we modify something for Tomin, we must check if the facts were changed for James as well.
- Compositionality (CI): In the compositionality test, we check if the editing method can create the edited fact with other knowledge or facts about the target tail entity. Also, we check if the editing method can create a fact about an alternate head entity with the edited fact.
- Forgetfulness (Fo): When we are dealing with recursive entities and multiple head and tail entities through the comprehensive triplet, we should make sure that injecting new facts should not change the content of the head or tail entity that are not related to the new inserted fact.
- Specificity (Sp): We check whether, for a given head entity where the tail entity has been edited with updated knowledge, the other tail entities for other relations for the same head entity are unaffected if they are not relevant to the edit.

6. Experiments and Results

For the experiments, we used three language models and four editing methods to evaluate how the editing methods perform for different language models when a comprehensive triplet representation is used. For language models, we selected and analyzed GPT-J, GPT-2 XL, and Llama-2-7b. For each of these language models, we worked with four editing or LLM updating methods, i.e., rank-one model editing (ROME), fine-tuning (FT), mass-editing memory in a transformer (MEMIT), and in-context learning for knowledge editing (IKE). The editing methods can be categorized into three categories, parametric update, fine-tuning, and in-context learning.

- ROME: ROME was introduced by [16] as an editing method for LLMs, where the authors first locate where the knowledge is stored in the LLMs feed-forward network (FFN); it then considers the FFN to act as a key-value pair. It takes the subject as the input, which is thought to be stored in the first linear layer of the FFN, and the second linear layer of the FFN contains the value or the object for the specific subject. To update the value for a subject, the authors proposed a rank-one update to the weight of the second linear layer to modify the old value to a new value.
- MEMIT: ROME is capable of editing a single fact, whereas MEMIT was proposed to edit multiple facts at the same time. MEMIT also falls under the parametric update category [13].
- Fine-tuning: Fine-tuning, proposed by [56], involves tailoring a pre-trained LLM to a particular domain. We similarly employed FT to acclimate to emerging entities. Our

experiments revolve around selectively updating the parameters solely within the last layer of the transformer model in GPT-J, GPT2-XL, and Llama-2-7B.

- IKE: IKE stands for in-context knowledge editing, which involves incorporating a fresh piece of information into an LLM using 'K' demonstrations. Each demonstration comprises a novel fact $f_j = (x_j^*, y_j^*)$, a probing prompt x_j and its corresponding prediction y_j. We consider the head and tail entities as the x, y pair and prepare the data in a format that works for in-context learning.

Existing editing methods use the semantic triplet representation (STR) for knowledge evaluation, as is shown in Tables 4–6, while we propose a comprehensive knowledge representation (CTR). In existing methods, an edit is represented as $\delta : (h, r, t) \rightarrow (h, r, t^*)$, whereas in our proposed method we evaluate knowledge in the form of a complex representation, i.e., $\theta' \leftarrow update(\theta, e, t_e)$, where t_e consists of multiple key-value pairs that need to be edited, i.e., $((s_{e,1}, y_{e,1}),(s_{e,2}, y_{e,2}), \ldots,(s_{e,n}, y_{e,n}))$.

Table 4. Accuracy on the GPT2-XL language model using different editing methods with their semantic triplet representation (STR) and our comprehensive triplet representation (CTR). The results are given for pre-edit (Pr-Ed) and post-edit (Po-Ed).

		Fine-Tuning		**ROME**		**MEMIT**		**IKE**	
		Pr-Ed	**Po-Ed**	**Pr-Ed**	**Po-Ed**	**Pr-Ed**	**Po-Ed**	**Pr-Ed**	**Po-Ed**
Gn	STR	39.5	46.4	35.2	44.7	47.9	52.9	49.0	55.1
	CTR	41.3	50.9	40.5	49.5	45.2	52.3	51.2	59.3
HEA	STR	73.9	89.3	71.2	86.1	81.5	90.1	86.4	91.7
	CTR	76.8	89.9	74.1	88.4	84.0	92.3	87.0	92.7
CI	STR	38.2	46.7	35.9	45.4	39.1	47.2	47.2	50.2
	CTR	39.0	46.1	40.1	46.9	43.2	50.3	48.9	51.6
Fo	STR	60.2	67.8	59.0	67.2	61.4	68.4	62.1	69.5
	CTR	64.5	71.2	62.4	70.1	65.2	72.5	66.4	74.8
Sp	STR	37.1	40.3	55.2	65.7	59.8	69.2	63.6	73.4
	CTR	39.8	45.6	57.9	69.3	61.4	70.7	66.8	74.1

Table 5. Accuracy on the GPT-J language model using different editing methods with their semantic triplet representation (STR) and our comprehensive triplet representation (CTR). The results are given for pre-edit (Pr-Ed) and post-edit (Po-Ed).

		Fine-Tuning		**ROME**		**MEMIT**		**IKE**	
		Pr-Ed	**Po-Ed**	**Pr-Ed**	**Po-Ed**	**Pr-Ed**	**Po-Ed**	**Pr-Ed**	**Po-Ed**
Gn	STR	38.8	49.2	42.5	53.5	43.3	55.6	56.3	62.3
	CTR	39.3	49.8	46.7	55.8	48.9	59.3	55.7	62.1
HEA	STR	69.8	78.4	73.2	79.0	75.1	85.6	76.2	87.1
	CTR	71.2	79.1	76.0	84.5	76.4	87.2	79.3	88.4
CI	STR	44.3	52.6	43.7	48.9	49.2	58.6	52.1	58.9
	CTR	45.4	53.1	44.8	50.2	51.1	59.7	51.5	58.7
Fo	STR	54.8	65.7	59.9	70.2	63.9	75.4	73.2	80.1
	CTR	55.2	66.9	63.2	74.5	66.8	78.5	73.9	81.2
Sp	STR	37.2	46.1	56.2	67.4	67.2	75.3	70.1	79.0
	CTR	36.6	45.2	58.8	69.0	68.9	77.8	71.2	79.1

In Tables 4–6, we present the accuracy results for GPT-2 XL, GPT-J, and Llama-2-7b when editing methods are applied with their current data format and our proposed comprehensive triplet representation. We provided the accuracy results of the pre-edit and post-edit for the four editing methods. In most cases, MEMIT and IKE performed better with respect to fine-tuning and ROME. The reason for this is that MEMIT and IKE are designed to work with multiple facts and thus are suitable for our setting. For GPT2-XL, as shown in Table 4, IKE had the best post-edit performance with a score of 59.3 for generalization, while MEMIT performed second best with 52.3 percent accuracy; however, the results for STR using MEMIT are on par with the performance of MEMIT with CTR. The same goes for GPT-J and Llama-2-7b, as shown in Tables 5 and 6, where IKE and MEMIT performed better, with MEMIT demonstrating the most improved accuracy (∼12 improved rate) compared to the pre-edit score. For "HEA" in GPT-J and Llama-2-7b, MEMIT had the best performance in terms of accuracy enhancement (i.e., when compared to the pre-edit accuracy), but IKE performed better with a score of 92.7 for GPT2-xl, 88.4

for GPT-J, and 83.8 for Llama-2-7b. For "CI" in GPT2-XL, IKE had the best score of 51.6, whereas MEMIT performed better than IKE with an accuracy score of 59.7 for GPT-J. In Llama-2-7b, MEMIT and IKE almost had the same performance. For "Fo" and "Sp", IKE had the best accuracy score in all three LLMs. The proposed complex representation of the triplet enhances the performance of the editing methods and not only improves the generalizability and specificity but also produces better results for compositionality and forgetfulness.

Table 6. Accuracy on the Llama-2-7b language model using the different editing methods with their semantic triplet representation (STR) and our comprehensive triplet representation (CTR). The results are given for the pre-edit (Pr-Ed) and post-edit (Po-Ed).

		Fine-Tuning		**ROME**		**MEMIT**		**IKE**	
		Pr-Ed	**Po-Ed**	**Pr-Ed**	**Po-Ed**	**Pr-Ed**	**Po-Ed**	**Pr-Ed**	**Po-Ed**
Gn	STR	34.1	47.8	46.3	58.9	48.1	60.3	52.5	61.3
	CTR	37.8	50.3	48.0	59.2	48.9	63.9	55.3	62.9
HEA	STR	53.2	68.3	54.5	69.2	63.6	78.1	76.8	81.4
	CTR	56.7	71.8	59.2	73.5	66.1	79.2	77.2	83.8
CI	STR	47.9	55.8	53.6	60.3	61.6	68.9	61.1	68.3
	CTR	51.2	56.2	55.6	62.5	63.7	73.1	67.2	73.4
Fo	STR	48.2	58.7	58.9	67.8	63.6	76.2	65.2	77.9
	CTR	53.3	60.7	62.4	70.3	67.9	82.6	69.3	81.5
Sp	STR	43.9	52.3	53.6	61.6	60.4	69.3	62.4	70.2
	CTR	47.2	51.5	55.9	66.2	61.3	69.8	66.7	74.9

7. Conclusions

With each passing day, new facts are emerging and, in many cases, older facts are being updated. Language models are playing a crucial role in technological advancement in multiple fields of applications and are making life easier in day-to-day life. Editing language models continuously is a necessity to keep the models up-to-date. However, updating models faces multiple challenges and edits are not always in depth. Updating a language model requires editing all the directly and indirectly linked relevant facts, making sure unrelated or irrelevant facts remain unchanged.

For KE, in this study, we proposed an extensive triplet representation for complex entity knowledge propagation. We created the CTR-KE dataset with references from Wikipedia and Wikidata with fifty entities and five hops to evaluate how much the editing methods can actually edit when the depth or content related to a specific entity increases or exists in an ample amount. We prepared the dataset with the main or head entity containing ten (h, r, t) combinations in the first hop and every tail entity considered as the head entity for the next hop. From the second hop to the fifth hop, we considered three combinations of (h, r, t) for every head entity. We defined and compared how the parameters were updated with the complex triplet representation to edit the language models. We successfully tested and proved our hypothesis that deeper contextual representation through recursive triplets results in contextual enrichment of the editing methods and reduces hallucinations or catastrophic forgetting.

We evaluated our proposed comprehensive triplet representation for complex entity knowledge propagation using five evaluation metrics, i.e., Gn, HEA, CI, Fo, and Sp, and four knowledge editing methods, i.e., Fine−tuning, ROME, MEMIT and IKE. We provided pre- and post-editing results to convey that multiple triplet updates are more convenient and accurate completed by MEMIT and IKE. We experimented with three language models, i.e., GPT2-XL, GPT-J, and Llama-2-7b. The proposed work shows promising results and paves the way for more research that concentrates on the depth of information storage in large language models, data representation, and the editing process for a deeper understanding of the contexts. Graph traversal for knowledge propagation, including more entities and data, are the aims of our future research.

Author Contributions: Conceptualization, W.S. and S.-H.N.; Methodology, W.S.; Software, W.S.; Validation, W.S. and S.-H.N.; Formal analysis, W.S.; Investigation, W.S. and S.-H.N.; Resources, S.-H.N.; Data curation, W.S.; Writing—original draft, W.S.; Writing—review & editing, W.S. and S.-H.N.;

Visualization, W.S.; Supervision, S.-H.N.; Project administration, S.-H.N.; Funding acquisition, S.-H.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2023-00216011, Development of artificial complex intelligence for conceptually understanding and inferring like human).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset presented in this article is not readily available because the data is part of an ongoing study.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Below is an example of our evaluation data used for the evaluation of multiple hop outcomes when using comprehensive triplets as shown in Figure A1.


```
{
 "case_id": 1,
 "requested_rewrite": [
  {
   "prompt": "{} of United States is
   "relation_id": "P488",
   "target_new": {"str": "Joe Biden", "id": "Q22349309"},
   "target_true": {"str": "Donald Trump", "id": "Q27671152"},
   "subject": "President",
   "question": "Who is the President of United States?"
  },
  ...
 ],
 "questions": [
  "What city serves as the capital of the country where Joe Biden is the President?",
  "What is the birth year of the current President of United States?",
  "At which city was the President of United States born?"
 ],
 "answer1": "Washington, D.C.",
 "answer_alias1": ["Washington", ...],
 "new_answer1": " Washington ",
 "new_answer_alias1": ["Washington, D.C.", ...],
 "answer2": "1946",
 "answer_alias2": ["June 14, 1946", ...],
 "new_answer2": " 1942 ",
 "new_answer_alias2": ["Nov 20, 1942.", ...],
 "answer3": "New York",
 "answer_alias3": ["New York, NY", ...],
 "new_answer3": " Scranton ",
 "new_answer_alias3": ["Scranton, PA.", ...],


 "single_hops": [
  {
   "question": "What is the educational qualification of the president of United States?
   "cloze": "US President's educational qualification is",
   "answer": "Law",
   "answer_alias": ["Law Graduate",...]
  },
  ...
 ],
 "new_single_hops": [...],
 "orig": {
  "triples": [
   ["Q786713", " P1376", " Q5303341"],
   ["Q11696", " P3150", " Q30512"],
   ["Q7727835", " P19", " Q604264"],
   ["Q11696", " P2094", " Q10752757"]
  ],
  "triples_labeled": [
   ["US President", "qualification", "Law"],
   ...,
  ],
  "new_triples": [...,],
  "new_triples_labeled": [...,],
  "edit_triples": [
   ["Q11696", " P2094","Q4115013"]
  ]
 }
}
```


Figure A1. Sample data example for evaluation of the proposed approach. Each example has a unique case ID, followed by various single hop and multi-hop questions to evaluate different evaluation metrics of the KE models.

References

1. Li, J.; Tang, T.; Zhao, W.X.; Nie, J.Y.; Wen, J.R. Pretrained language models for text generation: A survey. *arXiv* **2022**, arXiv:2201.05273.
2. Dou, Z.Y.; Peng, N. Zero-shot commonsense question answering with cloze translation and consistency optimization. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February–1 March 2022; Volume 36, pp. 10572–10580.
3. Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; Chang, M. Retrieval augmented language model pre-training. In Proceedings of the International Conference on Machine Learning, PMLR, Vienna, Austria, 12–18 July 2020; pp. 3929–3938.
4. Jin, X.; Zhang, D.; Zhu, H.; Xiao, W.; Li, S.W.; Wei, X.; Arnold, A.; Ren, X. Lifelong pretraining: Continually adapting language models to emerging corpora. *arXiv* **2021**, arXiv:2110.08534.
5. Dhingra, B.; Cole, J.R.; Eisenschlos, J.M.; Gillick, D.; Eisenstein, J.; Cohen, W.W. Time-aware language models as temporal knowledge bases. *Trans. Assoc. Comput. Linguist.* **2022**, *10*, 257–273. [CrossRef]
6. Jang, J.; Ye, S.; Yang, S.; Shin, J.; Han, J.; Kim, G.; Choi, S.J.; Seo, M. Towards continual knowledge learning of language models. *arXiv* **2021**, arXiv:2110.03215.
7. Zhai, Y.; Tong, S.; Li, X.; Cai, M.; Qu, Q.; Lee, Y.J.; Ma, Y. Investigating the catastrophic forgetting in multimodal large language models. *arXiv* **2023**, arXiv:2309.10313.
8. Li, Z. The dark side of chatgpt: Legal and ethical challenges from stochastic parrots and hallucination. *arXiv* **2023**, arXiv:2304.14347.
9. Liu, Z.; Wang, J.; Dao, T.; Zhou, T.; Yuan, B.; Song, Z.; Shrivastava, A.; Zhang, C.; Tian, Y.; Re, C.; et al. Deja vu: Contextual sparsity for efficient llms at inference time. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 22137–22176.
10. De Cao, N.; Aziz, W.; Titov, I. Editing factual knowledge in language models. *arXiv* **2021**, arXiv:2104.08164.
11. Wang, P.; Zhang, N.; Xie, X.; Yao, Y.; Tian, B.; Wang, M.; Xi, Z.; Cheng, S.; Liu, K.; Zheng, G.; et al. EasyEdit: An Easy-to-use Knowledge Editing Framework for Large Language Models. *arXiv* **2023**, arXiv:2308.07269.
12. Zhong, Z.; Wu, Z.; Manning, C.D.; Potts, C.; Chen, D. MQuAKE: Assessing Knowledge Editing in Language Models via Multi-Hop Questions. *arXiv* **2023**, arXiv:2305.14795.
13. Meng, K.; Sharma, A.S.; Andonian, A.; Belinkov, Y.; Bau, D. Mass-editing memory in a transformer. *arXiv* **2022**, arXiv:2210.07229.
14. Mitchell, E.; Lin, C.; Bosselut, A.; Finn, C.; Manning, C.D. Fast model editing at scale. *arXiv* **2021**, arXiv:2110.11309.
15. Sinitsin, A.; Plokhotnyuk, V.; Pyrkin, D.; Popov, S.; Babenko, A. Editable neural networks. *arXiv* **2020**, arXiv:2004.00345.
16. Meng, K.; Bau, D.; Andonian, A.; Belinkov, Y. Locating and editing factual associations in GPT. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 17359–17372.
17. Li, J.; Hui, B.; Qu, G.; Li, B.; Yang, J.; Li, B.; Wang, B.; Qin, B.; Cao, R.; Geng, R.; et al. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *arXiv* **2023**, arXiv:2305.03111.
18. Zheng, C.; Li, L.; Dong, Q.; Fan, Y.; Wu, Z.; Xu, J.; Chang, B. Can We Edit Factual Knowledge by In-Context Learning? *arXiv* **2023**, arXiv:2305.12740.
19. Agrawal, G.; Kumarage, T.; Alghami, Z.; Liu, H. Can Knowledge Graphs Reduce Hallucinations in LLMs?: A Survey. *arXiv* **2023**, arXiv:2311.07914.
20. Zhang, Y.; Chen, Z.; Zhang, W.; Chen, H. Making Large Language Models Perform Better in Knowledge Graph Completion. *arXiv* **2023**, arXiv:2310.06671.
21. Ye, Q.; Liu, J.; Chong, D.; Zhou, P.; Hua, Y.; Liu, A. Qilin-med: Multi-stage knowledge injection advanced medical large language model. *arXiv* **2023**, arXiv:2310.09089.
22. Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; Wu, X. Unifying Large Language Models and Knowledge Graphs: A Roadmap. *arXiv* **2023**, arXiv:2306.08302.
23. Liu, C.; Wu, B. Evaluating large language models on graphs: Performance insights and comparative analysis. *arXiv* **2023**, arXiv:2308.11224.
24. Cohen, R.; Biran, E.; Yoran, O.; Globerson, A.; Geva, M. Evaluating the ripple effects of knowledge editing in language models. *arXiv* **2023**, arXiv:2307.12976.
25. Geva, M.; Bastings, J.; Filippova, K.; Globerson, A. Dissecting recall of factual associations in auto-regressive language models. *arXiv* **2023**, arXiv:2304.14767.
26. Hase, P.; Bansal, M.; Kim, B.; Ghandeharioun, A. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *arXiv* **2023**, arXiv:2301.04213.
27. Han, X.; Li, R.; Li, X.; Pan, J.Z. A divide and conquer framework for Knowledge Editing. *Knowl. Based Syst.* **2023**, *279*, 110826. [CrossRef]
28. Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; Wei, F. Knowledge neurons in pretrained transformers. *arXiv* **2021**, arXiv:2104.08696.
29. Dong, Q.; Dai, D.; Song, Y.; Xu, J.; Sui, Z.; Li, L. Calibrating factual knowledge in pretrained language models. *arXiv* **2022**, arXiv:2210.03329.
30. Mitchell, E.; Lin, C.; Bosselut, A.; Manning, C.D.; Finn, C. Memory-based model editing at scale. In Proceedings of the International Conference on Machine Learning, PMLR, Baltimore, MD, USA, 17–23 July 2022; pp. 15817–15831.
31. Hernandez, E.; Li, B.Z.; Andreas, J. Inspecting and editing knowledge representations in language models. *arXiv* **2023**, arXiv:2304.00740.

32. Li, B.Z.; Nye, M.; Andreas, J. Implicit representations of meaning in neural language models. *arXiv* **2021**, arXiv:2106.00737.
33. Levy, O.; Seo, M.; Choi, E.; Zettlemoyer, L. Zero-shot relation extraction via reading comprehension. *arXiv* **2017**, arXiv:1706.04115.
34. Onoe, Y.; Zhang, M.J.; Padmanabhan, S.; Durrett, G.; Choi, E. Can lms learn new entities from descriptions? Challenges in propagating injected knowledge. *arXiv* **2023**, arXiv:2305.01651.
35. Hoelscher-Obermaier, J.; Persson, J.; Kran, E.; Konstas, I.; Barez, F. Detecting Edit Failures In Large Language Models: An Improved Specificity Benchmark. *arXiv* **2023**, arXiv:2305.17553.
36. Gupta, A.; Mondal, D.; Sheshadri, A.K.; Zhao, W.; Li, X.L.; Wiegreffe, S.; Tandon, N. Editing Commonsense Knowledge in GPT. *arXiv* **2023**, arXiv:2305.14956.
37. Ju, Y.; Zhang, Z. KLoB: A Benchmark for Assessing Knowledge Locating Methods in Language Models. *arXiv* **2023**, arXiv:2309.16535.
38. Xu, Y.; Li, W.; Vaezipoor, P.; Sanner, S.; Khalil, E.B. LLMs and the Abstraction and Reasoning Corpus: Successes, Failures, and the Importance of Object-based Representations. *arXiv* **2023**, arXiv:2305.18354.
39. Chollet, F. On the measure of intelligence. *arXiv* **2019**, arXiv:1911.01547.
40. Wu, X.; Yao, W.; Chen, J.; Pan, X.; Wang, X.; Liu, N.; Yu, D. From Language Modeling to Instruction Following: Understanding the Behavior Shift in LLMs after Instruction Tuning. *arXiv* **2023**, arXiv:2310.00492.
41. Guo, J.; Li, J.; Li, D.; Tiong, A.M.H.; Li, B.; Tao, D.; Hoi, S. From Images to Textual Prompts: Zero-shot Visual Question Answering with Frozen Large Language Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 10867–10877.
42. Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; Philip, S.Y. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 494–514. [CrossRef]
43. Zhang, Z.; Liu, X.; Zhang, Y.; Su, Q.; Sun, X.; He, B. Pretrain-KGE: Learning knowledge representation from pretrained language models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Virtual, 16–20 November 2020; pp. 259–266.
44. Kumar, A.; Pandey, A.; Gadia, R.; Mishra, M. Building knowledge graph using pre-trained language model for learning entity-aware relationships. In Proceedings of the 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 2–4 October 2020; IEEE: New York, NY, USA , 2020; pp. 310–315.
45. Chen, Z.; Xu, C.; Su, F.; Huang, Z.; Dou, Y. Incorporating Structured Sentences with Time-enhanced BERT for Fully-inductive Temporal Relation Prediction. *arXiv* **2023**, arXiv:2304.04717.
46. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*.
47. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 29.
48. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; Volume 28.
49. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
50. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
51. Min, B.; Ross, H.; Sulem, E.; Veyseh, A.P.B.; Nguyen, T.H.; Sainz, O.; Agirre, E.; Heintz, I.; Roth, D. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.* **2023**, *56*, 1–40. [CrossRef]
52. Abu-Rasheed, H.; Abdulsalam, M.H.; Weber, C.; Fathi, M. Supporting Student Decisions on Learning Recommendations: An LLM-Based Chatbot with Knowledge Graph Contextualization for Conversational Explainability and Mentoring. *arXiv* **2024**, arXiv:2401.08517.
53. Hu, Z.; Li, X.; Pan, X.; Wen, S.; Bao, J. A question answering system for assembly process of wind turbines based on multi-modal knowledge graph and large language model. *J. Eng. Des.* **2023**, 1–25
54. Hu, Y.; Zou, F.; Han, J.; Sun, X.; Wang, Y. *Llm-Tikg: Threat Intelligence Knowledge Graph Construction Utilizing Large Language Model*; SSRN: Rochester, NY, USA, 2023 .
55. Zhu, C.; Rawat, A.S.; Zaheer, M.; Bhojanapalli, S.; Li, D.; Yu, F.; Kumar, S. Modifying memories in transformer models. *arXiv* **2020**, arXiv:2012.00363.
56. Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; Smith, N.A. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv* **2020**, arXiv:2004.10964.