POLITECNICO DI TORINO

Master's Degree in Data Science and Engineering

Master's Degree Thesis

Multi-Lingual Knowledge Editing in Large Language Models

Supervisors

Prof. Aldo LIPANI

Prof. Paolo GARZA

Candidate

Saeedeh JAVADI

September 2024

Abstract

The use of large language models (LLMs) as dynamic repositories of knowledge is becoming increasingly prevalent. However, these models face significant challenges in managing outdated, erroneous, or privacy-sensitive information. The capacity to edit knowledge in an expedient manner within these models, without recourse to costly retraining, has emerged as a pivotal area of investigation. The existing techniques for editing knowledge are, on the whole, effective; however, they frequently lack robustness, particularly when applied across multiple languages. This thesis explores the domain of multilingual knowledge editing using Multi Lingual models like Llama-2, with a particular focus on enhancing the models' ability to update their knowledge efficiently and accurately in a multilingual context.

Our approach makes use of MEMIT (Mass-Editing Memory in Transformers), which enables the large-scale updating of the internal memory of transformer-based models. MEMIT enables the simultaneous editing of thousands of memories within LLMs, thereby providing a scalable solution for the correction of outdated or erroneous information. To further enhance this process, we integrate MEMIT with in-context learning (ICL), a technique that enables models to generalise knowledge from a few examples during inference. The objective is to integrate these two powerful methods in order to achieve precise and extensive knowledge updates across languages, thereby addressing one of the key challenges in multi-lingual LLMs.

Furthermore, this thesis incorporates prompt engineering as a technique to enhance the accuracy of the model's behaviour following knowledge edits. By carefully designing prompts, we guide the model's responses to ensure that updated information is both accurate and contextually appropriate for the target language. This mitigates issues such as over-editing, where unintended changes affect related knowledge, and instability, where the model struggles to retain the edited information across tasks and languages. Furthermore, by exploring the transformer architecture in detail, we examine how knowledge flows through the model's layers during the editing process.

Empirical testing on Multi Lingual models like Llama-2 shows that our combined approach significantly improves the performance of multi-lingual knowledge editing tasks. We evaluate the models on several languages, demonstrating enhanced accuracy and consistency in their ability to update and retain information.

Summary

In recent years, large language models (LLMs) have emerged as a highly significant advancement in the field of natural language processing (NLP). LLMs such as BERT, LLaMA-2, and GPT-4 have been developed through extensive pre-training on vast quantities of data, enabling them to perform a wide range of tasks with impressive accuracy and fluency. Such tasks include, but are not limited to, translation, question answering, summarization, and even the generation of human-like conversations.

One of the important reasons for the success of LLMs is their capacity to store vast quantities of factual knowledge. These models are trained to predict the next token in a sequence, helping them develop a deep understanding of linguistic structures and facts. However, despite their capabilities, LLMs are subject to a significant limitation: the knowledge they learn is static and often becomes outdated or irrelevant over time.

This static nature presents a critical issue for real-world applications of LLMs. Since the information encoded within the models is based on the data available at the time of their training, they cannot incorporate new knowledge dynamically without undergoing extensive retraining or fine-tuning. For example, an LLM trained in 2020 would not have knowledge of world events that occurred after its training period. Furthermore, LLMs sometimes generate factual errors or reflect outdated information, compromising their reliability, especially in contexts where up-to-date knowledge is essential, such as healthcare, legal advice, or news generation.

The need to efficiently update LLMs without retraining from scratch led to the development of knowledge editing techniques, which focus on altering or updating specific pieces of knowledge stored within the model. This allows models to adapt to new facts or correct erroneous information without the high computational cost of retraining. Traditional methods of updating LLMs, such as fine-tuning, require access to large datasets and computational resources, often risking catastrophic forgetting, where the model loses knowledge it previously acquired.

Knowledge editing has emerged as a solution to these challenges, allowing specific facts to be updated, deleted, or corrected within a model's parameters

without affecting the rest of its knowledge. While initial research in this area primarily focused on monolingual knowledge editing, the growing global demand for multilingual language models necessitates extending these techniques to handle multiple languages.

Moreover, multilingual LLMs like LLaMA and GPT have introduced new advantages and challenges. These models are designed to operate across various languages, expanding their usability for diverse global applications. However, the multilingual nature of these models means that the effectiveness of knowledge editing techniques must be evaluated across different languages. For instance, when a knowledge update is made in one language, it is not guaranteed that the change will be accurately reflected in another language, leading to the need for cross-lingual knowledge editing approaches.

Contributions

This thesis explores multilingual knowledge editing within LLMs, addressing the urgent need for models capable of incorporating new facts and correcting errors without extensive retraining. The key contributions include:

1. **Extension of the MzsRE Dataset:** This work expands the MzsRE dataset through paraphrasing, generating multiple linguistic variants that enhance its utility for cross-lingual tasks.

2. **Application of Few-Shot In-Context Learning (ICL):** The research adapts few-shot ICL for the first time in cross-lingual editing, integrating multilingual similarity retrieval to select relevant examples. This approach facilitates robust knowledge updates across languages, particularly in low-resource contexts.

3. **Comprehensive Evaluation of Knowledge Editing Techniques:** The thesis evaluates techniques such as ICL, zero-shot learning, and Tailored Knowledge Editing (TailoredKE) in multilingual settings, demonstrating significant advancements in cross-lingual knowledge transfer.

4. **Development of a Scalable Framework:** This framework employs retrieval-augmented techniques and dynamic layer selection to ensure that knowledge updates are consistently applied across languages. Tests are conducted separately during the evaluation phase to assess the effectiveness of the proposed methods.

Background

LLMs rely on transformer architectures, which allow them to efficiently process and generate text. Despite their remarkable capabilities, these models struggle to maintain the relevance of their internal knowledge, as facts can quickly become outdated. Traditional approaches to updating LLMs, such as fine-tuning, are computationally intensive and may lead to catastrophic forgetting. This research emphasizes the necessity for effective knowledge editing strategies, especially in multilingual contexts where changes in one language must accurately reflect in others.

Methodology

The methodology focuses on integrating advanced knowledge editing techniques within LLMs, emphasizing few-shot and zero-shot approaches along with TailoredKE. Key components include:

- **Few-Shot In-Context Learning (ICL):** This approach utilizes a few-shot paradigm where the model is presented with a small number of examples during inference to facilitate dynamic knowledge updates. By selecting examples through multilingual similarity retrieval, the model can effectively generalize updates across languages, enhancing its performance in multilingual contexts.

- **Zero-Shot Learning:** This technique allows the model to perform tasks without prior examples. By leveraging the inherent knowledge stored in the LLMs, zero-shot learning enables the model to handle knowledge editing tasks based on contextual cues provided in prompts, making it particularly useful in situations where training data is scarce or unavailable.

- **Tailored Knowledge Editing (TailoredKE):** This method focuses on the strategic selection of layers within the transformer architecture to optimize the knowledge editing process. By identifying which layers to modify, TailoredKE enhances the precision and effectiveness of updates, ensuring that knowledge changes are accurately reflected across various languages.

- **Extension of the MzsRE Dataset:** A significant aspect of this research involves expanding the MzsRE dataset through paraphrasing. This extension generates multiple linguistic variants of existing examples, enriching the dataset for cross-lingual tasks. The enhanced dataset provides a more comprehensive testbed for evaluating the effectiveness of few-shot and zero-shot approaches in multilingual knowledge editing.

Evaluation

The evaluation assesses the proposed methods' effectiveness in multilingual knowledge editing, focusing on two primary areas: multilingual knowledge editing and cross-lingual knowledge editing.

1. **Multilingual Knowledge Editing:** This section presents results from experiments using the MzsRE dataset with the LLaMA-2 backbone, comparing three different methods: Memit, TailoredKE (Targeted), and TailoredKE (Rephrase). The evaluation utilizes Exact Match (EM) metrics to assess consistency, efficacy, and generalization across various languages.

2. **Cross-Lingual Knowledge Editing:** This evaluation investigates the effectiveness of the developed knowledge editing techniques when edits are made in one language (English) and tested in other target languages.

Overall, the evaluation demonstrates that the integrated approach of using few-shot and zero-shot learning, along with tailored knowledge editing techniques, significantly enhances the performance of multilingual knowledge editing tasks. The findings emphasize the potential of these methods to address the challenges of static knowledge in LLMs, paving the way for more adaptable and responsive AI systems in diverse linguistic environments.

Conclusion

In conclusion, while LLMs have revolutionized NLP, their static nature limits their long-term usefulness, particularly in fast-evolving domains. Knowledge editing offers a promising alternative to costly retraining, enabling LLMs to incorporate new facts efficiently. The next challenge is to extend these techniques to multilingual and cross-lingual contexts, ensuring that updated knowledge is consistent across different languages. This thesis addresses the pressing need for effective multilingual knowledge editing in LLMs, offering scalable solutions that enable dynamic updates without extensive retraining. The findings underscore the potential of integrating ICL and TailoredKE to enhance the accuracy and reliability of LLMs in rapidly changing domains. Future research should continue to explore the challenges of knowledge editing in multilingual settings, paving the way for more adaptable and responsive AI systems.

Acknowledgements

I would like to express my sincere gratitude to my primary supervisor, Professor Aldo Lipani, for his invaluable guidance, support, and encouragement throughout my research. His insights and expertise at University College London have been crucial in shaping this thesis. I am also deeply grateful to Jerome Ramos, Professor Lipani's PhD student, for his continued assistance and helpful discussions during this process.

I would also like to extend my thanks to my internal supervisor, Professor Paolo Garza, for his ongoing support and advice, which has been a great source of motivation throughout my research journey.

Finally, I am thankful to University College London for providing the resources necessary for my studies and to the colleagues and friends who have been part of this journey, sharing their ideas and offering support along the way.

Table of Contents

List of Tables

List of Figures

Acronyms

LLM
Large Language Model

ICL
In-Context Learning

MEMIT
Mass-Editing Memory in Transformers

MLP
Multi-Layer Perceptron

GPT
Generative Pre-trained Transformer

LLaMA
Large Language Model Meta AI

MzsRE
Multilingual Zero-shot Relation Extraction

ReMaKE
Retrieval-Augmented Multilingual Knowledge Editor

Chapter 1

Introduction

Knowledge is a fundamental aspect of human intelligence and civilization [1]. Its organized structure allows us to represent real-world entities or explain concepts through symbolic means, enabling the expression of complex behaviors or tasks [2, 3, 4]. From birth, humans continuously acquire and adapt knowledge, applying lessons from this extensive reservoir in various situations. The study of knowledge—how it is acquired, stored, and interpreted—remains a compelling focus for researchers. This exploration is not only a technical endeavor but also a quest to replicate the intricate nature of human cognition, communication, and intelligence [5, 6, 7, 8, 9].

Large Language Models (LLMs) have become a cornerstone in natural language processing (NLP), being used across various domains such as translation, question answering, summarization, and more. These models, like GPT-4 [10], LLaMA-2 [11] , and others, exhibit remarkable capabilities by leveraging vast amounts of pre-trained knowledge. However, a critical challenge in maintaining the usability of these models lies in their static nature. Once trained, LLMs contain knowledge that can become outdated, incorrect, or incomplete. Retraining or fine-tuning these models to reflect new information is resource-intensive and often impractical, leading to the emergence of knowledge editing as a promising alternative [12, 13, 14]

LLMs function as vast repositories of knowledge, where facts, linguistic rules, and contextual information are embedded within their neural architectures. This embedded knowledge allows them to perform tasks such as answering factual questions or generating coherent and contextually appropriate text. However, one of the major challenges of using LLMs is maintaining the accuracy and relevance of this knowledge over time [15, 13, 14].

Although LLMs excel at generalization from training data, they are inherently static once trained. The world evolves—facts change, new information arises, and previously unknown events occur. For example, a model trained before 2020 will not know about the COVID-19 pandemic unless it is retrained with updated data.

Similarly, specific knowledge embedded in these models can become outdated, incorrect, or incomplete as the knowledge landscape changes over time.

This static nature raises a critical challenge: how can we update or correct knowledge in these massive models without retraining them entirely? Retraining is computationally expensive, time-consuming, and often impractical, particularly for models with billions of parameters [12, 15]. Moreover, even when retrained, it is not guaranteed that the model will incorporate the new knowledge effectively without losing existing valuable information [14].

In response to these challenges, knowledge editing has emerged as a promising solution. Knowledge editing allows for the injection or modification of specific pieces of information in an LLM without the need for complete retraining. This targeted approach ensures that models can stay up to date with current facts while preserving their overall performance and capabilities in areas that don't require updates. Methods like KN [16], MEMIT [17], and ROME [18] have been developed to perform efficient and scalable knowledge updates within these models.

While knowledge editing is well-explored in monolingual settings, especially English, most mainstream LLMs such as GPT-4 and LLaMA-2 are inherently multilingual. This capability allows them to process and generate text in multiple languages, yet the knowledge editing methods applied to these models have often been limited to a single language. In real-world applications, especially for globally deployed models, the ability to edit knowledge across languages is crucial. For instance, an edit made in English should ideally propagate to other languages, ensuring consistent behavior across multilingual environments [19].

Multilingual models face additional challenges related to entity alignment, linguistic nuances, and consistency across languages. Editing knowledge in one language and expecting consistent results in another remains a difficult task.

As discussed, the static nature of LLMs has led to the development of knowledge editing techniques aimed at updating or modifying specific pieces of knowledge within a model without needing to retrain the entire system. Knowledge editing focuses on altering the model's internal representations of factual information, allowing for precise and targeted changes.

Knowledge editing can be broadly categorized into several methods, each with distinct advantages and limitations:

- **Parameter-modifying methods** involve updating the model's internal parameters directly to reflect new knowledge. This approach allows for the injection of specific facts into the model's memory without disturbing unrelated knowledge. Techniques such as MEMIT and ROME have been developed to modify the parameters of models while maintaining their general performance.

- **Memory-based methods** store external knowledge in memory modules that can be accessed during inference without changing the model's core parameters.

This allows the model to reference new or corrected facts dynamically without risking catastrophic forgetting. Each of these approaches offers distinct trade-offs between computational cost, scalability, and precision.

One of the primary challenges in knowledge editing lies in maintaining locality, which refers to the ability to make a specific edit without unintentionally affecting other parts of the model's knowledge. Generality, or the model's ability to apply updated knowledge to semantically related queries, is another critical factor. Additionally, the reliability and portability of edits—whether the updated knowledge can generalize across different contexts and languages—pose significant hurdles for knowledge editing techniques.

While current knowledge editing methods represent significant advancements, they are primarily designed for monolingual scenarios. Extending these models to work in multilingual or cross-lingual contexts involves additional complexities, such as ensuring that the updated knowledge is accurately reflected across languages.

The primary challenge in multilingual knowledge editing is ensuring that changes made in one language are accurately reflected in other languages. This issue stems from language modeling gaps, where LLMs might perform well in one language (often English) but poorly in others, particularly in low-resource languages. When knowledge is edited in one language, the model may not generalize this update effectively across different languages, leading to inconsistencies.

For instance, when a fact is edited in English, a cross-lingual system must ensure that the edited knowledge is reflected accurately in languages like Chinese, Spanish, or French [20, 21]. One method for facilitating this is through ICL, which uses prompts to provide updated knowledge and allows the model to apply the changes across languages. However, ICL alone may not be sufficient for ensuring consistency across languages, particularly in low-resource settings.

A crucial component of multilingual and cross-lingual knowledge editing research is the availability of appropriate datasets. One such dataset is MzsRE, which is a multilingual extension of the Zero-shot Relation Extraction (ZsRE) dataset. MzsRE has been translated into 12 languages, including English, Chinese, Spanish, French, and more, and serves as a valuable resource for evaluating the cross-lingual capabilities of LLMs.

Datasets like Bi-ZsRE and MzsRE have been critical for evaluating the performance of knowledge editing methods in multilingual and cross-lingual contexts. These datasets allow researchers to test how well a model transfers knowledge from one language to another, ensuring that edits made in a source language (e.g., English) are consistently reflected in a target language (e.g., Chinese). [20]

As the need for multilingual and cross-lingual LLMs grows, so too does the demand for efficient and accurate knowledge editing methods. Several approaches have been developed to address the unique challenges of knowledge editing in multilingual settings, each with its own strengths and weaknesses.

One of the leading methods in this field is Retrieval-augmented Knowledge Editing. The ReMaKE [15] framework, for example, enhances the knowledge editing process by retrieving relevant information from a multilingual knowledge base and feeding it into the model as part of the prompt. By incorporating retrieved knowledge directly into the editing prompt, ReMaKE avoids the need for parameter updates and ensures that the edited knowledge is available across multiple languages.

ReMaKE outperforms traditional knowledge editing methods, particularly in multilingual settings where ensuring consistency across languages is a major challenge. In experiments, ReMaKE showed significant improvements over baseline methods like SERAC and MEND, particularly in low-resource languages.

The primary objective of this thesis was to extend existing knowledge editing methods to function effectively in multilingual and cross-lingual settings. Building on established techniques such as MEMIT, ICL, and ReMaKE, this research focuses on improving the scalability, accuracy, and consistency of knowledge updates across diverse linguistic contexts. The following key contributions highlight the advances made through this work:

- **Extension of the MzsRE Dataset with Paraphrasing:** One of the significant contributions of this thesis is the expansion of the MzsRE dataset, a multilingual extension of the zsRE dataset, to better accommodate cross-lingual knowledge editing tasks. Using the GPT API, approximately 20 sentences from the original dataset were paraphrased to generate multiple linguistic variants. These paraphrased sentences were then utilized in the ICL approach, enriching the dataset and providing a more comprehensive testbed for multilingual knowledge editing. This dataset extension allows for improved evaluation of how well knowledge edits generalize across paraphrases and alternate expressions in multiple languages.

- **Application of Few-Shot In-Context Learning (ICL) in Multilingual Settings:** Another important contribution is the adaptation of few-shot ICL to support multilingual knowledge editing. By incorporating multilingual similarity retrieval methods, the few-shot ICL approach was enhanced with semantically similar examples from various languages, significantly improving the model's ability to generalize knowledge updates across different linguistic contexts. The use of carefully selected few-shot examples, based on cosine similarity, proved to be the most effective strategy for achieving robust cross-lingual knowledge editing, particularly in low-resource languages where traditional methods struggle.

- **Evaluation of Knowledge Editing Techniques in Multilingual Contexts:** This thesis also provides a comprehensive evaluation of the adapted

knowledge editing techniques—MEMIT, ICL, and TailoredKE—in a multilingual context. The experiments conducted on the MzsRE dataset highlighted the advantages of combining layer selection with MEMIT for cross-lingual knowledge transfer. The results demonstrated that the few-shot ICL approach significantly outperforms zero-shot methods in both high-resource and low-resource languages, showcasing the potential of these techniques to enable accurate and scalable multilingual knowledge editing.

- **Development of a Scalable Framework for Cross-Lingual Knowledge Editing:** By addressing key challenges such as knowledge transfer across languages, the thesis presents a scalable framework for cross-lingual knowledge editing. This framework leverages retrieval-augmented techniques and dynamic layer selection to ensure that knowledge injected in one language is consistently applied across other languages. The integration of multilingual similarity retrieval with Sentence-BERT embeddings for selecting semantically similar examples further enhances the efficiency of the knowledge editing process in cross-lingual tasks.

Together, these contributions advance the field of multilingual knowledge editing by providing a robust framework that combines state-of-the-art techniques like MEMIT, ICL, and TailoredKE. The innovations in dataset expansion, layer selection, and retrieval-augmented few-shot learning push the boundaries of existing methods, offering new insights into how large language models can be updated and adapted for multilingual and cross-lingual tasks. This research paves the way for further exploration of scalable and accurate knowledge editing in diverse linguistic environments.

Chapter 2

Background

This chapter delves into the foundational concepts and developments in Large Language Models (LLMs) and their applications, particularly in multilingual contexts. In this section, we first explore the transformer architecture, the core framework behind modern LLMs, and how it enables these models to process vast amounts of information efficiently.

We then shift focus to multilingual language models, examining models such as GPT-4, LLaMA-2, and Mistral, which aim to bridge the gap between high- and low-resource languages. While these models offer powerful solutions for multilingual tasks, they also present unique challenges, particularly in maintaining consistent knowledge across languages. The chapter also discusses the mechanisms of knowledge storage and retrieval in LLMs, shedding light on how models internally organize information and respond to factual queries.

Finally, the section will cover knowledge editing techniques, which are crucial for updating and refining the knowledge stored in LLMs without full retraining. This includes a discussion on the mechanisms used to modify, add, or remove knowledge, and the particular difficulties of achieving these edits in multilingual and cross-lingual environments. These topics form the backbone of this thesis and provide the necessary background for understanding the challenges and innovations introduced in multilingual knowledge editing and cross-lingual transfer.

2.1 Large Language Models

2.1.1 Transformers for LLMs

LLMs are constructed upon the Transformer architecture, which was first introduced by Vaswani et al. (2017), and represent a significant departure from the recurrent models that have previously been used in the field of natural language processing

(NLP). The transformer architecture is founded on the self-attention mechanism, which enables the model to assess the relevance of distinct tokens within an input sequence. This mechanism enables the model to process tokens in parallel, thereby markedly enhancing its efficiency in comparison to older models, which processed sequences one token at a time [22].

As shown in Figure 2.1, the Transformer architecture consists of encoder and decoder stacks, each composed of multiple layers of multi-head self-attention and feed-forward neural networks. The self-attention module ensures that each token within a sequence can attend to every other token, effectively capturing long-range dependencies.

The *self-attention mechanism* (depicted in Figure 2.2, left) operates by computing attention scores for each token with respect to all other tokens in the input sequence. This enables the model to weigh the relevance of each token when generating the output. Additionally, the *multi-head attention mechanism* (Figure 2.2, right) involves running multiple parallel attention operations, allowing the model to capture different aspects of the relationships between tokens [22].

This parallelization and ability to understand context over longer distances have led to the widespread use of transformers in modern LLMs. Furthermore, each transformer layer consists of a feed-forward neural network (FFN), which introduces non-linear transformations to enhance the model's capacity to represent complex relationships. Transformers have laid the groundwork for LLMs such as BERT, GPT, and their multilingual versions, which aim to capture relationships not only within one language but across multiple languages [23, 24].

2.1.2 Multilingual Large Language Models

The continuous progress in LLMs has greatly influenced various aspects of NLP, enabling models to perform tasks such as translation, summarization, and sentiment analysis with remarkable precision. As the demand for global applications increases, the need for Multilingual Large Language Models (MLLMs) has become more crucial. These models are designed to work seamlessly across different languages, promoting inclusivity and improving access in diverse linguistic environments. Leading MLLMs, such as GPT-4 [10], LLaMA-2 [11], and Mistral [25], offer distinct features and capabilities that enhance multilingual language processing.

GPT-4

GPT-4, which was developed by OpenAI, is widely recognised as one of the most prominent multilingual language models. GPT-4 has been trained on a larger and more diverse corpus, thereby enabling it to handle multiple languages with a high

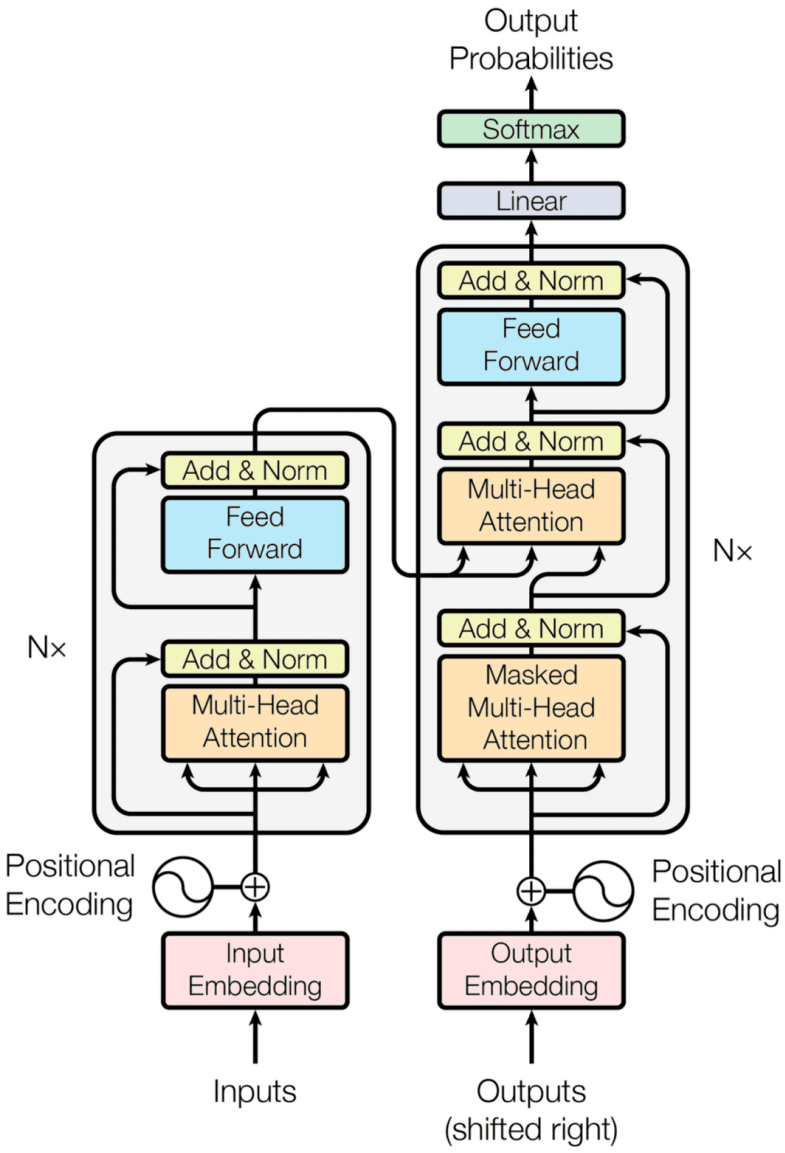

Figure 2.1: The Transformer model architecture, including the encoder and decoder stacks with multi-head self-attention and feed-forward layers [22].

degree of proficiency. GPT-4 displays robust capabilities in cross-lingual and multilingual tasks. Although GPT-4 continues to demonstrate superior performance in English, it has also exhibited notable advancements in the handling of low-resource languages and the ability to generalise across diverse linguistic structures.

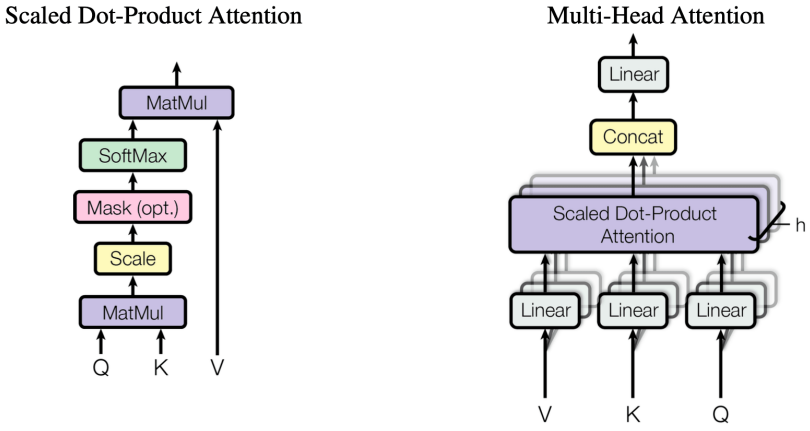

Figure 2.2: (Left) Scaled Dot-Product Attention mechanism, and (Right) Multi-Head Attention with multiple parallel attention layers [22].

GPT-4's strength lies in its ability to leverage large-scale, diverse data for multilingual training. This enables the model to transfer knowledge from high-resource languages, like English, to low-resource languages, which often suffer from limited training data. However, one of the ongoing challenges in GPT-4 is ensuring that knowledge edits made in one language are accurately reflected in others, particularly in less-resourced languages.

LLaMA-2

LLaMA-2 (Large Language Model Meta AI), released by Meta, represents another important advancement in multilingual modeling. LLaMA-2 has been optimized for performance across various tasks, including cross-lingual tasks, by training on a broad set of multilingual corpora. LLaMA-2 has been specifically designed to offer improved performance in low-resource languages compared to earlier models, thus helping bridge the gap between languages with abundant resources and those with limited ones.

One of LLaMA-2's key contributions is its ability to efficiently handle code-switching—a phenomenon common in multilingual environments where speakers mix languages in conversation. By training on data that includes such linguistic variations, LLaMA-2 enhances the model's robustness in dealing with real-world multilingual interactions. However, like GPT-4, LLaMA-2 faces challenges in maintaining consistency when edits or updates are made to the model's knowledge

base across different languages.

Mistral

Mistral, a recent multilingual model, focuses on efficiency and performance, particularly in multilingual settings. Mistral is designed to be a lightweight yet powerful model, optimizing both the speed of inference and resource efficiency. This makes it ideal for deployment in environments where computational resources are limited but multilingual support is critical.

Mistral places an emphasis on scalability and adaptability across languages, enabling users to fine-tune the model for specific cross-lingual tasks without the need for extensive computational overhead. It is particularly useful for real-time applications that require fast and reliable language processing across multiple languages. Mistral also incorporates recent advances in retrieval-augmented generation, which allows the model to access external knowledge bases dynamically, thus improving its accuracy in tasks involving factual information across languages.

While these models demonstrate impressive multilingual capabilities, there are still several challenges in multilingual knowledge representation and cross-lingual transfer learning. Ensuring that edits or updates to the model's knowledge base are propagated effectively across all languages remains a significant research challenge. Moreover, the issue of bias in multilingual models—where high-resource languages dominate model behavior—continues to be a concern. Future directions in the development of MLLMs like GPT-4, LLaMA-2, and Mistral will likely focus on improving the balance between languages, enhancing fine-grained knowledge editing capabilities, and reducing the computational footprint for real-time multilingual tasks.

2.1.3 Mechanism of Knowledge Storage in LLMs

The success of LLMs is not just due to the Transformer architecture but also to their ability to store vast amounts of world knowledge within their parameters. Researchers have explored how these models store and retrieve information, revealing that knowledge is distributed across layers in the model, with Figure 2.3 illustrating some of these research findings. Early layers tend to capture shallow, syntactic information, while deeper layers focus on semantic and factual knowledge [18].

For instance, studies suggest that phrase-level linguistic features are stored in lower layers, whereas semantic and factual knowledge reside in higher layers. The FFN layers, particularly, are thought to function as a form of key-value memory system, where specific neurons correspond to particular facts or knowledge items. This key-value system allows models to recall knowledge when relevant inputs are

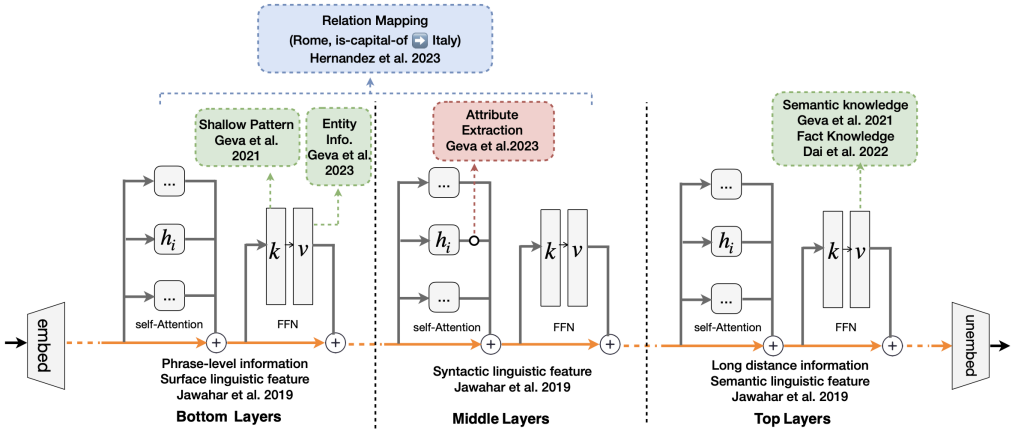

Figure 2.3: The mechanism of knowledge storage in LLMs.[17, 18, 16, 26].

presented [26].

LLMs like GPT-3 and GPT-4 demonstrate emergent behaviors such as the ability to answer factual questions by leveraging this internal knowledge base. However, the exact mechanism by which LLMs organize and store knowledge remains an area of ongoing research. Research into knowledge localization has found that certain neurons are activated by specific factual prompts, indicating that factual knowledge is distributed but still localized within specific components of the model [16].

2.2 Knowledge Editing in LLMs

Knowledge editing refers to the process of modifying, adding, or removing specific knowledge from an LLM without retraining the entire model. This task is critical as LLMs are often used in dynamic environments where facts and knowledge evolve over time. For example, as geopolitical events or scientific discoveries emerge, models need mechanisms to update their internal knowledge base without affecting their overall performance on unrelated tasks [27].

Current techniques for knowledge editing can be divided into three categories: resorting to external knowledge, merging new knowledge into the model, and editing intrinsic knowledge. These techniques draw from cognitive learning processes, such as recognition, association, and mastery, to systematically modify the model's knowledge base.

By employing methods such as causal tracing or localized gradient updates, researchers can identify specific neurons or layers responsible for storing particular pieces of information, allowing for targeted interventions [18]. This approach

ensures that changes to the model are minimal and do not degrade its general performance.

2.2.1 Mathematical Definition of Knowledge Editing

Given an LLM parameterized by θ, the objective of knowledge editing is to update the model so that it generates new, correct outputs for certain input prompts related to the knowledge being edited, without disrupting its overall performance. Formally, let the LLM be defined as a function $f_\theta(x)$, where x represents the input, and θ are the learned parameters of the model. The goal of knowledge editing is to update θ such that for a specific input x', the model outputs a desired target y', i.e.,

$$f_{\theta'}(x') = y',$$

where θ' is the updated parameter set after editing, and y' is the corrected or updated output. The key requirement is that the updated parameters θ' should maintain the model's performance on unrelated inputs x, so that:

$$f_{\theta'}(x) \approx f_\theta(x) \quad \text{for all} \quad x \neq x'.$$

In other words, knowledge editing seeks to make localized changes that only affect the specific knowledge being targeted while preserving the overall performance and integrity of the model.

Knowledge editing methods can generally be classified into three categories based on how they modify the model's internal or external representations of knowledge:

- **External Knowledge Methods:** In this approach, additional knowledge is provided at inference time through external resources such as databases or retrieval systems. These methods augment the model's output without directly altering its parameters. For instance, if the model encounters a prompt that contains outdated information, it queries an external source to provide the correct answer. This ensures that the original parameters θ remain unchanged.

- **Merging Knowledge into the Model:** This approach modifies the model's internal representations by combining new information with pre-existing knowledge. The goal here is to adjust the model's output layers or embeddings such that the new knowledge is integrated while keeping the model's overall architecture intact. Mathematically, this can be seen as finding a mapping function that adjusts the output probabilities conditioned on new facts:

$$P(y|x;\theta') = P_{\text{new}}(y|x),$$

where $P_{\text{new}}(y|x)$ reflects the updated knowledge embedded in the modified parameters θ'.

- **Intrinsic Knowledge Editing:** This method involves directly editing the parameters of the model to encode new or corrected knowledge. This can be mathematically formulated as a constrained optimization problem, where we aim to find an updated parameter set θ' that satisfies:

$$\theta' = \arg\min_{\theta} \mathcal{L}(f_\theta(x'), y') + \lambda \mathcal{L}(f_\theta(x), y),$$

where $\mathcal{L}$ is a loss function that penalizes deviations from the desired output for the edited input x', while λ controls the regularization to prevent deviations from the original outputs on unrelated inputs x.

Knowledge editing in LLMs holds promise in various fields, including AI-driven systems that need to remain up-to-date with real-time information or correct past mistakes. For example, an LLM used in medical diagnosis may require regular updates to incorporate new research findings, while a chatbot might need to be corrected for any biased or incorrect statements it may generate.

However, knowledge editing is a challenging task, particularly due to the distributed nature of knowledge in LLMs. There is always a risk that updating one piece of knowledge might inadvertently alter other unrelated facts stored within the model. Therefore, the design of robust methods that can isolate and edit specific knowledge with minimal side effects is a key research area. Techniques such as fine-tuned regularization and careful parameter control help mitigate these risks, but achieving optimal performance remains an open problem in the field.

2.2.2 Methods

The capabilities of LLMs have evolved to closely mimic human cognitive processes, especially when it comes to learning and knowledge acquisition. Inspired by the way humans learn, these stages can be applied to the process of editing LLMs, as illustrated in Figure 2.4. Research in education and cognitive science [**s1**, **s2**, **s3**] suggests that human knowledge acquisition occurs in three distinct phases: recognition, association, and mastery. This framework provides a useful lens through which to understand how knowledge editing functions in LLMs, as shown in Table 2.1.

- **Recognition Stage:** In the recognition stage, the model is introduced to new knowledge within a specific context, much like how humans first become familiar with new information. For example, by providing the model with sentences that demonstrate an updated fact, it can begin to recognize the knowledge that needs to be adjusted.

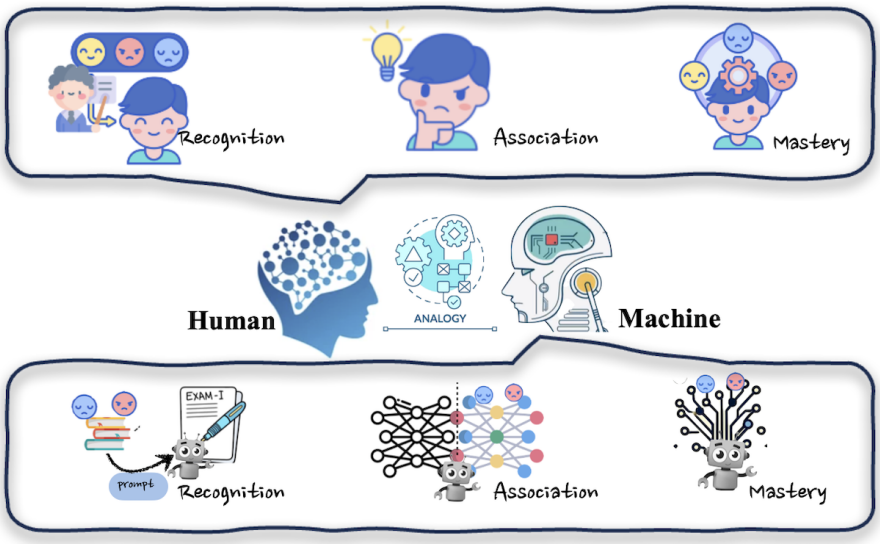

Figure 2.4: Applying Human Learning Phases to Knowledge Editing in LLMs: This figure illustrates the analogy between human learning phases and knowledge editing in LLMs, organizing current methods into the stages of recognition, association, and mastery.

- **Association Stage:** During this stage, the model starts linking the newly introduced knowledge with its existing knowledge base, much like humans connect new ideas to prior experiences. In this stage, the output or intermediate representation h may be adjusted by incorporating or substituting it with a knowledge representation h_{know}.
- **Mastery Stage:** The final stage, mastery, occurs when the model fully integrates the new information into its parameters and can reliably utilize it without external support, akin to a human achieving expertise. This process typically involves directly updating the model's weights, W, allowing it to independently handle tasks without additional interventions or external assistance.

2.2.3 Multilingual and Cross-lingual Editing in LLMs

MLLMs extend the capabilities of LLMs by supporting multiple languages. These models, such as mBERT, XLM-R, and BLOOM, are trained on multilingual corpora, enabling them to handle tasks across different languages [28, 29]. The challenge in

MLLMs is to maintain high performance in both high-resource and low-resource languages.

In multilingual and cross-lingual settings, LLMs must balance knowledge representation across languages. Recent research shows that multilingual models often develop language-independent neurons, which store abstract representations that can generalize across languages. This ability to share knowledge across languages is critical for tasks such as cross-lingual knowledge editing, where information must be updated in one language and reflected across others [30].

Multilingual models also encounter challenges related to language-specific knowledge conflicts, where facts or interpretations differ across cultural or linguistic contexts. Addressing these challenges requires advanced techniques for knowledge editing that can handle multiple languages simultaneously, without introducing bias or degrading performance in lower-resource languages [18].

2.3 Related Techniques

LLMs have inspired various techniques aimed at efficient model adaptation, knowledge augmentation, and continual learning. These techniques play a crucial role in enhancing model performance and updating factual knowledge without requiring full-scale retraining. This section provides an overview of related techniques in the domain of knowledge editing, parameter-efficient tuning, knowledge augmentation, continual learning, and machine unlearning.

2.3.1 Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning techniques, such as Adapters and Low-Rank Adaptation (LoRA), aim to reduce the computational costs associated with fully fine-tuning LLMs. These techniques focus on updating only a small subset of parameters, thereby maintaining overall model performance while minimizing resource usage.

Adapter-based Methods: Adapters introduce additional trainable parameters into the model. By attaching these adapters to various layers, the model can learn task-specific knowledge without modifying the core parameters, as discussed by Houlsby et al. (2019) [31].

LoRA: This method introduces low-rank matrix approximations to existing layers, enabling efficient updates with fewer trainable parameters. LoRA focuses on updating a limited number of weights while retaining most of the model's general-purpose knowledge [32].

While these techniques have been successful in task-specific fine-tuning, their direct application to knowledge editing remains underexplored. Parameter-efficient tuning could be adapted for more precise and localized knowledge updates.

2.3.2 Knowledge Augmentation

Knowledge-augmented methods supplement LLMs with external knowledge sources to address limitations such as missing or outdated information. The most common approach in this domain is Retrieval-Augmented Generation (RAG). RAG retrieves relevant documents from external knowledge sources, such as databases or the web, and combines them with model-generated outputs.

Input-level Retrieval Augmentation: Retrieved information is concatenated with the input prompt, enabling the model to generate more accurate responses based on updated context [33, 34].

Intermediate and Output Layer Augmentation: Retrieval-based components can also be incorporated into intermediate layers or blended with the output token distribution, allowing the model to correct misinformation during inference [35, 36].

While knowledge augmentation provides immediate solutions for generating accurate outputs, it does not update the internal knowledge of the model. Therefore, it is primarily useful for inference-time knowledge updates, leaving long-term knowledge storage unchanged.

2.3.3 Continual Learning

Continual learning techniques aim to enable LLMs to learn new information while retaining previously acquired knowledge. This approach is essential for models deployed in dynamic environments where new facts emerge frequently.

Memory-based Systems: These systems store knowledge in external memory modules that the model can access during inference, thus maintaining previously learned information without overfitting to new tasks.

Regularization-based Methods: These methods incorporate constraints during training to prevent catastrophic forgetting, ensuring that updates do not degrade the model's existing knowledge [37, 38].

Continual learning has become an important approach in managing the dynamic nature of knowledge. However, it remains computationally expensive and less efficient for fine-grained knowledge editing tasks.

2.3.4 Machine Unlearning

Machine unlearning is a growing area of research focused on removing specific knowledge from models without retraining. This technique is particularly useful in compliance scenarios, such as Right to be Forgotten regulations, where specific data needs to be erased. Selective forgetting methods identify and suppress the weights associated with specific knowledge to eliminate its influence on the model's outputs.

2.4 Knowledge Editing Models

This section explores several prominent approaches, each with unique strategies to enhance accuracy, scalability, and flexibility in knowledge editing across different tasks.

This section explores several prominent approaches, each with unique strategies to enhance accuracy, scalability, and flexibility in knowledge editing across different tasks. Table 2.1 provides a comparative overview of representative knowledge editing methods for LLMs. It highlights key aspects such as the area of the model being edited, whether additional training is required, and the ability to perform batch edits, helping to differentiate between the methods based on their functionalities and use cases.

2.4.1 Semi-Parametric Editing with a Retrieval-Augmented Counterfactual Model (SERAC)

SERAC [27] is a memory-based method developed for efficient and precise knowledge editing in large language models. In contrast to conventional gradient-based techniques that alter model parameters, SERAC preserves the original model by employing an external memory system. This system retains user-provided modifications, such as input-output examples or supplementary utterances, without necessitating any alterations to the model's fundamental architecture.

At the heart of SERAC is the ability to distinguish when an input requires modification and how to apply that modification. This is achieved through two main components: the scope classifier and the counterfactual model. The scope classifier evaluates whether a given input falls within the scope of the stored edits by estimating its relevance. If the input is deemed relevant, the counterfactual model predicts the appropriate output based on both the new input and the relevant edit stored in memory. When no applicable edits are found, the base model's original output is returned.

SERAC offers several advantages over existing knowledge editing techniques. By separating the process of identifying when an edit should be applied from how the model's behavior should change, SERAC minimizes the risk of unintended side effects. This approach allows SERAC to efficiently handle multiple simultaneous edits, a challenge for many other editing frameworks. The model has demonstrated superior performance in tasks like fact-checking, question-answering, and dialogue generation, where precise and scalable model edits are essential.

2.4.2 Knowledge Embedding (KE)

Knowledge Embedding (KE) methods embed factual knowledge into the parameters of LLMs, focusing on the attention mechanisms and feed-forward networks. KE achieves high performance in knowledge editing by introducing auxiliary models that can efficiently adapt the weight matrices in transformer models. This approach ensures that knowledge updates are reflected across layers without degrading the model's original performance. It strikes a balance between edit precision and computational efficiency [39].

2.4.3 Precise Model Editing in Transformers (PMET)

Precise Model Editing in a Transformers (PMET) introduces a novel approach to model editing by improving the precision of weight updates within transformer models. Existing techniques, such as ROME and MEMIT, rely on optimizing transformer layer hidden states to memorize target knowledge and update the weights of the FFN. However, these methods overlook that transformer layer hidden states contain information from multi-head self-attention (MHSA), FFN, and residual connections, leading to imprecise weight updates[40].

PMET improves upon this by simultaneously optimizing the hidden states of both MHSA and FFN. Despite this joint optimization, it restricts the actual weight updates to the FFN layer, leaving the MHSA weights unchanged. The rationale behind this is that MHSA primarily functions as a general knowledge extractor, encoding patterns of knowledge extraction rather than storing specific factual knowledge. As a result, PMET leverages the optimized FFN hidden states to perform more precise weight updates, thus enhancing the accuracy of knowledge edits while maintaining the model's overall reliability.

2.4.4 Knowledge Neurons (KN)

The Knowledge Neurons framework (KN) [16] aims to identify and manipulate specific neurons within a large-scale pretrained transformer model responsible for encoding factual knowledge. These neurons are critical in expressing relational facts stored in the model. The central idea is that factual knowledge in transformer models, like BERT, is distributed across FFNs. These FFNs can be viewed as key-value memory structures, where neurons act as keys storing specific pieces of information, and their activation allows the model to retrieve and express this knowledge.

To identify knowledge neurons, the authors propose a knowledge attribution method that tracks how individual neurons contribute to a model's predictions during a fill-in-the-blank task. This method is based on integrated gradients, a

technique that measures the contribution of each neuron to the output by examining how the model's predictions change when the neuron's activations are altered.

Two significant applications of knowledge neurons are: updating factual knowledge and erasing specific relations. By identifying and modifying the activations of particular neurons, the model's internal knowledge can be updated without retraining. For example, changing the knowledge about a fact (like the capital of a country) can be achieved by modifying a small number of key neurons responsible for encoding that information. Similarly, factual relations, such as personal information like birthplace, can be erased by suppressing the activations of related neurons. This neuron-level intervention demonstrates that knowledge neurons are not only useful for understanding how transformer models store factual knowledge but also for enabling fine-grained editing of this knowledge without the need for extensive retraining. These insights make knowledge neurons a promising tool for model interpretability and targeted knowledge editing.

2.4.5 In-Context Learning (ICL)

In-Context Learning (ICL) is an efficient approach for knowledge editing that operates during inference time. Rather than modifying the model's parameters, ICL allows knowledge to be injected into the model through prompt-based interactions. This makes it particularly useful for scenarios where frequent knowledge updates are required, as no model retraining is needed. ICL-based methods are gaining popularity for their flexibility in injecting and modifying knowledge across tasks without compromising model fluency [41, 42, 43].

2.4.6 Rank-One Model Editing (ROME)

Rank-One Model Editing (ROME) is a targeted knowledge editing model that uses rank-one updates to the feed-forward neural networks in transformers. By applying causal tracing, ROME identifies the precise layers and neurons that need modification, ensuring that factual knowledge can be edited with minimal disruption to other parts of the model. While powerful for single-fact updates, ROME struggles with bulk or multiple simultaneous edits [18].

2.4.7 Mass Editing Memory in a Transformer (MEMIT)

Mass Editing Memory in a Transformer (MEMIT) extends the capabilities of models like ROME by supporting bulk edits across multiple facts. It does this by utilizing a more distributed update mechanism that spans across several layers. This ensures that multiple factual updates can be performed simultaneously without harming

the overall performance of the model. MEMIT is highly effective for large-scale knowledge management [17].

2.4.8 Model Editing Networks with Gradient Decomposition (MEND)

MEND [44] is a model-editing approach designed to enable rapid, localized edits to LLMs using minimal computational resources. Unlike traditional fine-tuning, which can be computationally intensive and prone to overfitting when applied to single input-output corrections, MEND allows for efficient edits by leveraging gradient decomposition. This method employs small auxiliary networks to transform the fine-tuning gradient, making edits scalable even to models with over 10 billion parameters.

MEND operates by learning to modify the gradient obtained from standard fine-tuning. It uses a low-rank decomposition of this gradient, which reduces the complexity of parameterizing the transformation. These transformations are learned through auxiliary networks that take the raw gradient and output a targeted update to the model's weights, ensuring that the edits affect only the desired areas of the model's behavior while preserving performance on unrelated tasks. This approach allows MEND to maintain edit locality, reliability, and generality.

One of the key strengths of MEND is its ability to handle very large models, such as GPT and T5, without requiring retraining or access to the entire dataset during the editing process. By transforming the gradient in a computationally efficient manner, MEND can apply rapid edits even to models with billions of parameters, making it an ideal solution for scenarios where fast model updates are necessary without compromising the model's overall accuracy.

2.4.9 Transformer-Patcher (T-Patcher)

Transformer-Patcher [45] is a novel model editing technique developed to handle mistakes made by large transformer-based language models, especially in real-world scenarios where models are continuously deployed. Unlike previous approaches that focused on correcting single mistakes at a time, Transformer-Patcher introduces the concept of Sequential Model Editing (SME), which aims to fix errors as they occur, in an ongoing fashion. The approach works by adding and training a small number of neurons, referred to as patches, within the last FFN layer of a transformer model. These patches modify the model's behavior for specific problematic inputs while preserving its overall accuracy on irrelevant or already correct inputs.

One of the key strengths of Transformer-Patcher is its ability to maintain reliability, meaning that after an edit, the model produces the correct output for the modified input. Furthermore, the method ensures generality, as it enables the

edited model to generalize its corrections to similar inputs, such as paraphrased versions of the problematic input. Importantly, it achieves locality, ensuring that the edits do not degrade the model's performance on unrelated examples.

The underlying mechanism of Transformer-Patcher works by freezing the parameters of the original model and introducing patches into the last FFN layer to adjust the output accordingly. This method allows for efficient error correction without the need for retraining the entire model, making it highly suitable for dynamic and real-time industrial applications in natural language processing.

2.5 Advanced Techniques in Large Language Models: Prompt Engineering and In-Context Learning

As large language models (LLMs) like GPT-3 and GPT-4 have evolved, they now excel at performing a broad spectrum of tasks across domains. However, simply deploying these models with raw inputs does not always unlock their full potential. To effectively harness their capabilities for specific tasks, two key techniques are widely employed: *prompt engineering* and *in-context learning*. These techniques are essential for achieving optimal performance without the need for extensive task-specific fine-tuning. Instead, they allow for quick adaptation to diverse tasks by utilizing the deep language understanding that these models develop during pretraining.

Among these techniques, *prompt engineering* stands out as a powerful method for guiding LLMs toward desired outputs. It involves crafting clear, well-structured prompts that effectively frame the task for the model. By controlling how the task is presented, prompt engineering can lead to more accurate and task-specific responses without requiring additional training or modification of the model. This approach is particularly valuable because it leverages the general language knowledge embedded in LLMs, enabling efficient adaptation to new tasks with minimal computational effort [24, 58].

2.5.1 Prompt Engineering

Prompt engineering involves designing and crafting specific input text prompts that guide the model toward generating the desired output. The concept stems from the realization that language models are highly sensitive to the way instructions or tasks are presented. Even slight variations in the phrasing or formatting of a prompt can lead to significantly different outputs [58]. This sensitivity makes prompt engineering a powerful, yet sometimes challenging, technique to master.

Table 2.1: Comparison of key approaches for knowledge editing in LLMs. "No Training" indicates methods that do not require extra training, while "Batch Edit" refers to whether the method can handle multiple edits simultaneously.

Category/Method		Edit Area	Edit Function	No Training	Batch Edit
Association Stage	*MemPrompt[46]*	Memory+Retriever	Input → [Mem : Input]	✓	✓
	SERAC[27]	Memory+Classifier	Output → $\text{Model}_c(\text{x})$	✓	×
	MeLLo[42]	Memory+Retriever +Auxiliary Model	Input → [Mem : Input]	×	×
	IKE[41]	Memory+Retriever	Input → [Mem : Input]	✓	×
	ICE[43]	Prompt	Input → [Mem : Input]	✓	×
	PokeMQA[47]	Memory+Retriever	Input → [Mem : Input]	×	×
Recognition Stage	*CaliNET[48]*	FFN+params	Output head +params	✓	✓
	T-Patcher[45]	FFN+params	$h \rightarrow h + \text{FFN}_{add}(\text{x})$	×	✓
	REMEDI[49]	Auxiliary Model	$h \rightarrow \text{REMEDI}(x)$	×	✓
	GRACE[50]	FFN+codebook	$h \rightarrow \text{GRACE}(x)$	×	✓
	LoRA[51]	Attn or FFN	$h \rightarrow h + s : \text{LoRA}(x)$	×	✓
	MELO[52]	Attn or FFN	$h \rightarrow h + s : \text{LoRA}(x)$	×	✓
Mastery Stage	*FT-Constrained[53]*	Any	$\mathbf{W} \rightarrow \mathbf{W}'$	×	×
	ENN[54]	Any	$\mathbf{W} \rightarrow \mathbf{W}'$	×	×
	KE[39]	Attn or FFN +Auxiliary Model	$\mathbf{W} \rightarrow \mathbf{W}'$	×	×
	SLAG[55]	Attn or FFN +Auxiliary Model	$\mathbf{W} \rightarrow \mathbf{W}'$	×	×
	MEND[44]	FFN +Auxiliary Model	$\mathbf{W}_{down} \rightarrow \mathbf{W}'_{down}$	×	✓
	KN[16]	FFN	$\mathbf{W}_{down} \rightarrow \mathbf{W}'_{down}$	✓	×
	ROME[18]	FFN	$\mathbf{W}_{down} \rightarrow \mathbf{W}'_{down}$	✓	×
	MEMIT[17]	FFN	$\mathbf{W}_{down} \rightarrow \mathbf{W}'_{down}$	✓	✓
	PMET[40]	FFN	$\mathbf{W}_{down} \rightarrow \mathbf{W}'_{down}$	✓	✓
	MALMEN[56]	FFN	$\mathbf{W}_{down} \rightarrow \mathbf{W}'_{down}$	✓	✓
	BIRD[57]	FFN	$\mathbf{W}_{down} \rightarrow \mathbf{W}'_{down}$	×	✓

2.5.2 The Role of Prompts in LLMs

Large language models are trained to predict the next word in a sequence based on the context provided by the preceding words. During inference, the input (prompt) serves as the context, and the model generates a response based on this context. For example, if the prompt is "Translate the following sentence into French: 'Hello, how are you?'", the model generates the appropriate translation as "Bonjour, comment ça va?". The success of the response is directly influenced by how well the prompt communicates the task.

The core idea of prompt engineering is to create an input that clearly specifies the task for the model [**brown2020language**]. There are two main strategies in prompt engineering:

- **Zero-shot prompting**: In this method, the model is given the task in a single instruction without any examples. The model is expected to understand and perform the task solely based on the instructions. For instance, "Summarize the following text:" followed by a paragraph provides the model with no prior examples but asks for a summary directly.

- **Few-shot prompting**: In contrast to zero-shot, few-shot prompting provides the model with several examples of how the task is performed before asking it to generate the output for a new instance. For example, providing a few question-answer pairs followed by a new question allows the model to infer the structure and pattern from the provided examples [**brown2020language**]. This method can significantly improve performance on tasks where the model might struggle to understand the desired output format.

2.5.3 The Importance of Effective Prompts

Effectively designing prompts is crucial because it determines whether the model generates an accurate and relevant response. In many cases, slight modifications to the wording or structure of a prompt can lead to improved or deteriorated results. Researchers have found that inserting clarifying instructions, structuring the task in a clear and logical manner, and sometimes even including extraneous details (like listing "steps" or "instructions") can help the model better grasp complex tasks [59].

One interesting innovation that arose from prompt engineering is *chain-of-thought prompting*, which involves structuring prompts to guide the model through a reasoning process. This is particularly useful for tasks that require multi-step reasoning, such as solving math problems or performing logical deductions. In chain-of-thought prompting, instead of asking the model for a direct answer, the prompt encourages the model to explain the steps involved in reaching the answer

[59]. For example, in a math word problem, a chain-of-thought prompt might guide the model to first identify the variables, apply the necessary operations, and then generate the final answer. This step-by-step breakdown often leads to more accurate results in tasks that require complex thinking.

> Example of chain-of-thought prompting:
> *"If Tom has 3 apples and buys 2 more, how many apples does he have now? First, calculate how many apples Tom had initially. Then, add the number of apples he bought."*

By guiding the model through intermediate steps, prompt engineering enables models to perform tasks that were previously difficult to address through simple commands.

2.5.4 In-Context Learning (ICL)

In-context learning is a paradigm in which models "learn" from examples provided within the input prompt, rather than being explicitly trained on labeled data through gradient-based updates. Essentially, in-context learning allows models to adapt to new tasks on the fly, purely based on the context provided by a few examples. This approach leverages the model's general language understanding, enabling it to perform new tasks without altering its underlying parameters [**brown2020language**].

2.5.5 Mechanism of In-Context Learning

In in-context learning, the model is provided with a series of input-output pairs within the prompt. By observing these pairs, the model infers the pattern or structure of the task, allowing it to generalize to new inputs within the same prompt context. This method relies on the model's pretraining, during which it learns to recognize patterns across vast datasets [60].

An example of in-context learning would be as follows:

> *Translate the following sentences:*
> English: "The sky is blue." → Spanish: "El cielo es azul."
> English: "I am hungry." → Spanish: "Tengo hambre."
> English: "Good morning!" → Spanish:

In this example, the model uses the prior translations in the prompt to infer the task and provide the correct translation for the new sentence "Good morning!" without requiring additional training data.

2.5.6 Advantages of In-Context Learning

The primary advantage of in-context learning is its flexibility and efficiency. In traditional supervised learning, models must be fine-tuned on labeled datasets to perform well on specific tasks. However, with in-context learning, models can adapt to new tasks simply by observing a few examples provided at inference time [**brown2020language**].

Furthermore, in-context learning is particularly useful in scenarios where labeled data is scarce or unavailable. For instance, in low-resource language tasks or domain-specific tasks, it may be impractical to fine-tune a model on limited data. In such cases, providing a few labeled examples directly within the prompt can allow the model to generalize and perform the task with a high degree of accuracy.

In-context learning also allows models to handle a variety of tasks simultaneously without needing to be retrained for each new task. This opens up possibilities for rapid deployment of models in real-world applications, such as customer service, translation services, and content generation.

2.6 Applications and Impact

The combination of prompt engineering and in-context learning has led to numerous breakthroughs in NLP applications, particularly in tasks that benefit from minimal training data and flexible task adaptation. Some notable applications include:

- **Cross-lingual translation**: By providing examples of how sentences are translated between languages, models can perform translation tasks with high accuracy. This is particularly valuable in cases where traditional machine translation systems may struggle due to lack of training data in certain languages.
- **Question answering systems**: Carefully crafted prompts can guide models to answer questions more accurately by specifying the format or encouraging the model to focus on relevant information from a text.
- **Text summarization**: In tasks where users need concise summaries of documents or articles, prompts can be structured to instruct the model to provide a high-level overview, summarizing key points while ignoring irrelevant details.
- **Content generation and storytelling**: In the creative industries, prompt engineering is used to generate stories, articles, and even poetry. In this context, prompts can be designed to specify tone, style, or genre, giving creative control to the user while the model generates coherent and contextually relevant content.

The flexibility and adaptability of these techniques make them integral to the development of advanced AI applications that are highly responsive to user input.

2.7 Challenges and Future Directions

While prompt engineering and in-context learning offer powerful tools for interacting with large language models, they also present some challenges. One of the main issues is the sensitivity of models to slight variations in prompts. Small changes in phrasing or formatting can lead to drastically different results, making it difficult to consistently achieve the desired outcome [58]. Additionally, in-context learning relies on the assumption that the model can correctly infer the task from a limited number of examples, which may not always be the case for more complex or ambiguous tasks.

Another challenge is the lack of standardization in prompt design. Users often rely on trial and error to determine which prompts work best for a given task. Future research could focus on developing more robust and consistent methods for designing prompts, potentially incorporating automatic prompt generation techniques that optimize for task performance [59].

Finally, as LLMs continue to grow in size and complexity, the computational costs associated with these models remain a concern. More efficient methods for prompt processing and task adaptation may be necessary to scale these models for widespread use in industry.

2.8 Conclusion

Prompt engineering and in-context learning have revolutionized the way we interact with large language models. These techniques allow models to be applied to a wide variety of tasks with minimal training, making them versatile tools for natural language understanding and generation. As research in this area continues to evolve, we can expect further improvements in the efficiency and accuracy of these methods, paving the way for more advanced applications of LLMs in real-world scenarios.

Chapter 3

Methodology

3.1 Overview

The rapid advancement of large language models has enabled them to store and generate vast amounts of factual knowledge. However, as the world changes, these models often hold outdated or incorrect information. Addressing this issue without retraining the entire model is crucial, especially in multilingual environments where the knowledge needs to propagate across different languages. The focus of this thesis is on *multilingual knowledge editing*, which aims to update LLMs with new facts in a way that ensures consistency and accuracy across multiple languages. This research explores an efficient combination of *MEMIT (Mass-Editing Memory in Transformers)*, *in-context learning (ICL)*, and *retrieval-augmented techniques*, extending them to multilingual settings.

The *MEMIT* method is particularly useful for updating factual knowledge within specific layers of the model, allowing for precise control over what information is altered. This method helps inject new facts or correct outdated knowledge without significantly affecting unrelated information. MEMIT's ability to edit large amounts of information across different layers simultaneously makes it highly scalable [17].

To complement this approach, *ICL* [41, 42, 43] offers a flexible, non-intrusive way of influencing model behavior by including demonstration examples within the input context. ICL has proven to be an effective technique for knowledge editing in situations where direct parameter updates are either undesirable or impractical. This thesis adapts ICL to the multilingual setting, allowing for knowledge edits in multi languages. However, challenges remain in ensuring that knowledge edits performed in one language (e.g., English) can be effectively applied and queried in another [20, 21].

Addressing the challenges of *cross-lingual knowledge editing*, a retrieval-augmented

solution called *ReMaKE (Retrieval-augmented Multilingual Knowledge Editor)* is also incorporated. ReMaKE combines multilingual retrieval with in-context learning to improve the efficiency and scalability of knowledge edits across languages [15]. The *MzsRE dataset*, a multilingual extension of the zsRE dataset, is used in this research to evaluate the effectiveness of cross-lingual knowledge transfer across languages [20, 15, 21].

By combining these methodologies, this thesis presents a framework that addresses key challenges in multilingual knowledge editing, such as ensuring consistency across languages, maintaining generalization, and improving scalability without compromising accuracy.

3.2 Knowledge Editing with MEMIT

Mass-Editing Memory in Transformers (MEMIT) [17] is a knowledge-editing technique that modifies specific transformer parameters to update factual associations within a model's memory. MEMIT operates by identifying the critical path of Multi-Layer Perceptron (MLP) layers that mediate factual recall, and then modifying these layers to insert new factual memories. This process is illustrated in Figure 3.1, where MEMIT updates critical MLP layers based on causal mediation analysis.

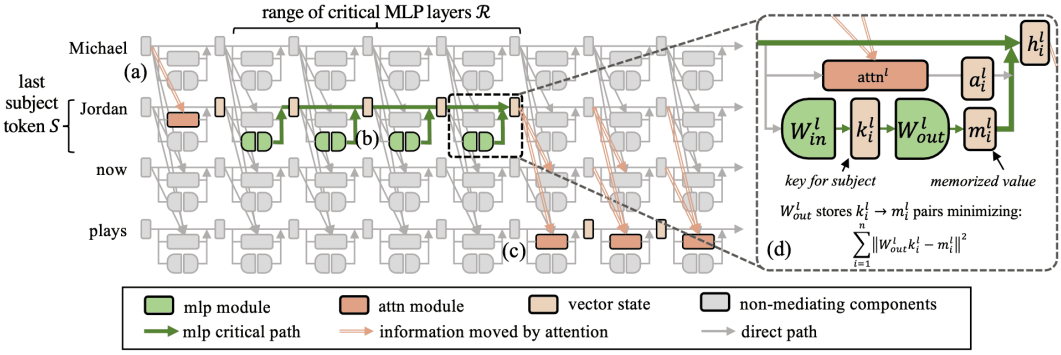

Figure 3.1: MEMIT adjusts the parameters of transformers involved in the key steps of MLP-driven factual recall. In the early layers, attention modules collect subject names into vector representations, while MLPs in crucial layers interpret these encodings and introduce memories into the residual stream. These memories are subsequently processed by attention modules to generate the final output. [17, 18]

MEMIT works by calculating the vector associations that we want the critical

MLP layers to remember and spreading these associations across multiple layers. Unlike single-edit methods, MEMIT allows for the simultaneous editing of thousands of memories by distributing the desired changes across the layers, ensuring that each memory is accurately reflected without overwriting unrelated information.

Figure 3.1 shows the core mechanism where early attention modules gather subject information, and critical MLP layers insert the memory into the residual stream. This ensures that the inserted knowledge is accessible across multiple contexts, improving the scalability and precision of the edits.

In this thesis, MEMIT is used to inject new knowledge into the model and update outdated facts. For example, when factual information (such as the current president of a country) needs to be updated, MEMIT locates the relevant parameters, modifies them, and verifies that the new fact is accurately represented across different contexts and queries. This is particularly beneficial for tasks involving factual knowledge that evolves over time, such as current events or updates in encyclopedic data.

One of the key advantages of MEMIT is its ability to scale. Traditional knowledge editing approaches, such as fine-tuning or gradient-based methods, require significant computational resources and often lead to overfitting or unintended side effects, such as forgetting previously stored knowledge. MEMIT overcomes these issues by focusing on parameter-specific updates, avoiding unnecessary alterations to other parts of the model. This ensures that only the target knowledge is modified, while unrelated facts remain unaffected.

To evaluate the effectiveness of MEMIT in a multilingual setting, the method was applied to update facts in models capable of generating responses in multiple languages. This ensures that the injected knowledge can be queried across different languages with consistent accuracy. The approach was tested on the *MzsRE* dataset, a multilingual extension of the zsRE dataset, which was used to validate the generalization and retention of the edited knowledge across various languages [21, 20, 15].

Overall, MEMIT serves as a powerful tool in this thesis for managing large-scale knowledge updates in multilingual LLMs, ensuring that new information is integrated without disrupting the model's existing knowledge base.

3.3 In-Context Learning for Knowledge Editing

In-Context Learning (ICL) is an innovative approach that enables language models to perform specific tasks by providing examples within the input context, without requiring any changes to the model's parameters. In ICL, task-specific instructions and examples are presented as part of the model's input, guiding it to generate the desired output. This makes ICL highly versatile and a non-intrusive method for

knowledge editing, as it avoids the need for parameter updates or fine-tuning [43].

Traditional knowledge editing techniques involve modifying model parameters to inject or update factual knowledge. However, this requires computationally expensive fine-tuning, especially for large models. In contrast, ICL offers a parameter-free approach to knowledge editing, where the language model learns to adjust its responses based on the provided examples or demonstrations in the input. This flexibility makes ICL a highly effective tool for temporary or context-specific knowledge updates, allowing the model to maintain accuracy for evolving knowledge without requiring any retraining or permanent modifications [41].

Figure 3.2 illustrates how ICL can be employed to inject new knowledge into a model by providing contextual demonstrations. These examples help the model learn and apply updated facts without modifying its internal parameters.

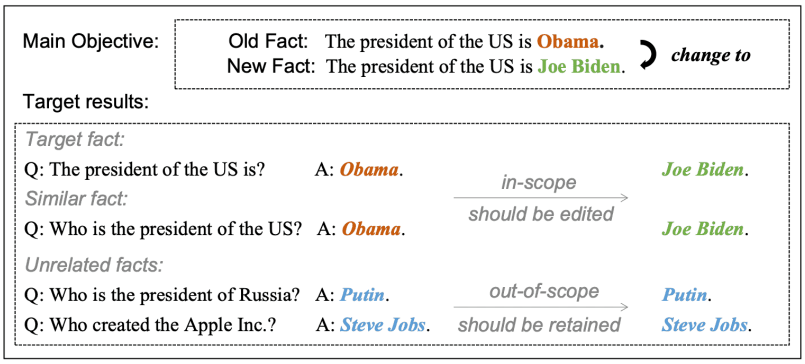

Figure 3.2: In-Context Learning workflow for knowledge editing. Demonstrations of new, updated, and retained facts guide the model to generate accurate outputs without parameter updates [41].

ICL operates by providing a series of demonstrations (or examples) that include the desired knowledge updates, guiding the model to adopt the new information during its inference process. These demonstrations act as references for the model to generate correct responses without altering its internal weights. This makes ICL particularly useful in black-box settings, where direct parameter modification is not possible.

The ICL-based knowledge editing process typically involves three types of demonstrations:

1. *Copying*: Demonstrations that provide the model with the exact new fact to be learned, ensuring that the model can replicate this fact in its responses.

2. *Updating*: These demonstrations focus on showing different rephrasings or variations of the fact to ensure that the model can generalize the new knowledge across different prompts and contexts.

3. *Retaining*: Demonstrations that present unrelated knowledge to guide the model in preserving its original knowledge outside the scope of the edited fact. This helps avoid over-editing or knowledge forgetting [42].

Figure 3.3 provides a visual example of the different types of ICL demonstrations used in knowledge editing. The demonstrations are formatted to ensure that new knowledge is injected, updated, and retained across various contexts without altering unrelated facts.

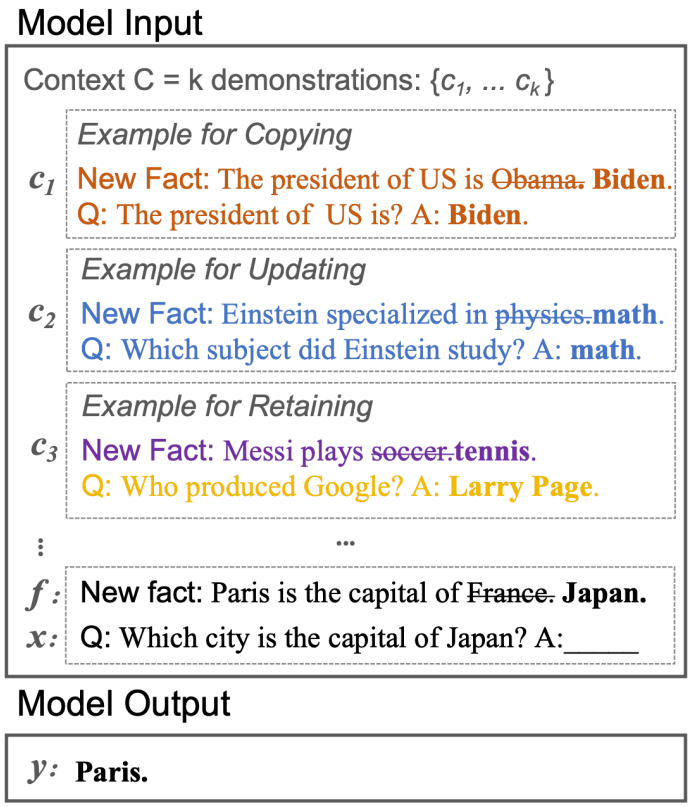

Figure 3.3: Example of in-context demonstrations for knowledge editing. Demonstrations include facts that are copied, updated, and retained to guide the model's output [41].

ICL has been shown to be highly effective for knowledge editing in large models

such as GPT-J, achieving a competitive success rate compared to gradient-based approaches, with the added benefit of fewer side effects such as over-editing or knowledge forgetting [41]. Moreover, ICL is scalable and applicable to larger language models, making it an efficient alternative to parameter-updating methods, especially in black-box or service-oriented model environments.

In this thesis, ICL is evaluated in the context of multilingual knowledge editing. By providing demonstrations in different languages, ICL can be adapted to ensure that updated knowledge generalizes across languages. This allows for knowledge updates to be effectively transferred between high-resource and low-resource languages, ensuring consistency and accuracy in the model's responses [21].

3.4 Interpretability-based Tailored Knowledge Editing (TailoredKE)

The original approach presented in the *Interpretability-based Tailored Knowledge Editing in Transformers* paper introduces a method called *TailoredKE* for editing factual knowledge in large language models. This method focuses on understanding the internal information flow of models and strategically selecting the layers to modify, in order to enhance the precision of knowledge edits and minimize over-editing. Unlike existing knowledge-editing methods that apply uniform layer modifications, TailoredKE tailors the edits to specific transformer layers based on the unique properties of the knowledge being modified.

3.4.1 Knowledge Editing through Layer Selection

TailoredKE builds upon insights from previous research, which identified that the middle layers of transformer models—particularly the feed-forward Multi-Layer Perceptrons (MLPs)—serve as key-value memories for storing factual knowledge [17]. By focusing on these layers, the method aims to modify parameters selectively, ensuring that only the relevant information is edited. This helps to prevent over-editing, where unrelated facts might be unintentionally altered.

One of the core innovations in TailoredKE is the *Dynamic Editing Window*, which selects specific layers for editing based on the characteristics of the knowledge being modified. Instead of fixing the edit layers across all tasks, as done in methods like MEMIT or ROME, TailoredKE observes the information flow related to the specific knowledge in question. By analyzing how entity representations evolve across layers, the method pinpoints the layers where the entity's attributes are recalled most effectively, and edits are applied only in these layers.

Figure 3.4 illustrates the overall process of TailoredKE, highlighting the three steps involved: strengthening the new memory, locating the key layers, and injecting

the new knowledge into the selected layers.

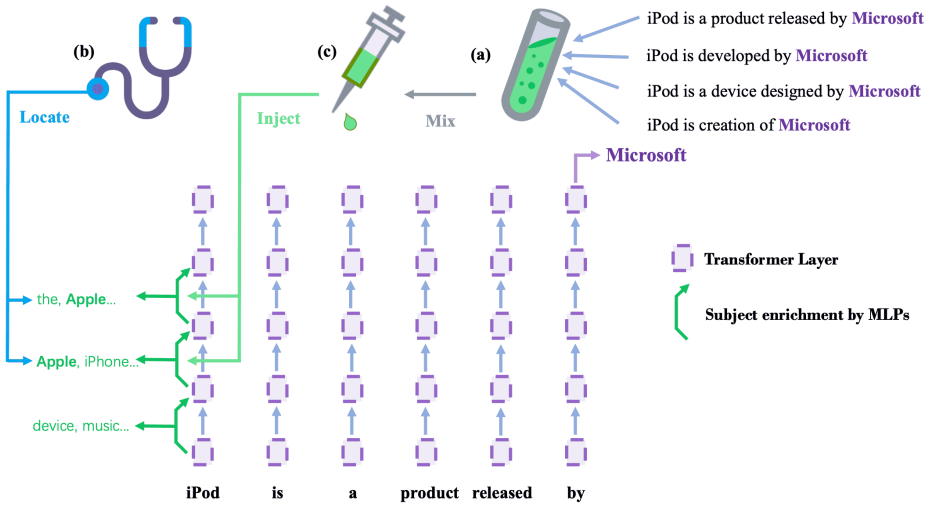

Figure 3.4: Overview of the TailoredKE process, involving the strengthening of the new memory, dynamic layer selection, and knowledge injection into selected layers.

3.4.2 Layer Selection and Entity Representation

TailoredKE's layer selection strategy is based on the idea that different layers in a transformer model recall different attributes of an entity. For example, shallow layers might associate an entity like “iPod” with generic attributes such as "device" or "music," while deeper layers might recall more specific associations like "Apple" or "iPhone." By observing this process, the method chooses layers that are most responsible for recalling the specific attributes being edited.

This approach reduces the likelihood of over-editing and ensures that the model can retain unrelated facts. For example, when editing the fact that "iPod is a product released by Microsoft," the method ensures that other related facts about "iPhone" or "Macbook" remain unaffected.

3.5 Multilingual Setting and Challenges

Adapting knowledge editing techniques to multilingual settings introduces several significant challenges, as models trained on multiple languages often display inconsistent behavior when edited in one language and queried in another. This section outlines the key difficulties encountered when extending the knowledge editing methods—such as MEMIT and TailoredKE—across languages and the strategies employed to address these challenges.

2. Generalization and Locality Issues: A significant issue with multilingual knowledge editing is the balance between generalization and locality. Generalization ensures that the edited knowledge applies across multiple linguistic contexts, while locality ensures that unrelated knowledge remains unaffected by the edit. In multilingual settings, this balance is difficult to achieve. The risk of over-editing is heightened, particularly when languages share lexical or syntactic similarities. For example, updating a fact about "Paris" in English might inadvertently alter related facts in French, where the representation of "Paris" shares attributes with the English version [21].

2. Data Scarcity and Low-Resource Languages: Another challenge is the lack of sufficient training data for low-resource languages. In such cases, even well-designed knowledge editing methods like MEMIT or TailoredKE may struggle to apply changes effectively across these languages. This is because the model has not learned robust representations for entities and relationships in low-resource languages, leading to poorer performance when attempting to edit or inject new knowledge into these languages [61].

3. Cross-lingual Transfer of Knowledge Edits: Ensuring that knowledge edited in one language is transferable to others is a complex task. Since different languages often have unique syntactic and semantic structures, a knowledge edit applied in English might not translate seamlessly to a language like Arabic, which has a distinct morphology. This cross-lingual inconsistency complicates the task of verifying that the knowledge injected or updated in one language is also correctly reflected in others [14, 42].

3.5.1 Addressing Multilingual Challenges

To overcome these challenges, the following strategies were employed:

1. Dynamic Layer Selection for Multilingual Models: Building upon the TailoredKE approach, the method was extended to include *Dynamic Editing Windows* tailored to each language. By observing how knowledge is represented and recalled in different languages, the layer selection process dynamically adjusts based on the unique structure and attributes of each language. This strategy helps ensure that edits applied in one language can be appropriately transferred and

localized in others without over-editing or loss of unrelated knowledge [17].

2. Use of Multilingual Datasets for Evaluation: The *MzsRE dataset*, a multilingual extension of the zsRE dataset, was employed to test cross-lingual generalization of the knowledge edits. This dataset was instrumental in identifying inconsistencies and guiding improvements in the knowledge editing process.

3. Retraining and Adaptation for Low-Resource Languages: To address the scarcity of data in low-resource languages, data augmentation techniques were used to generate paraphrased and translated versions of the knowledge to be injected. This strategy improves the robustness of the edits by exposing the model to more varied representations of the knowledge across languages. In cases where the model exhibited poor performance in certain languages, additional retraining was performed using synthetic data to strengthen the model's understanding of these languages [61].

3.6 Multilingual Similarity Retrieval Task

Multilingual Similarity Retrieval refers to the process of retrieving semantically similar text across multiple languages using machine learning models. This task is essential in many natural language processing (NLP) applications, including cross-lingual search engines, machine translation, multilingual knowledge management, and information retrieval. With the increasing globalization of digital content, systems must understand and retrieve semantically similar information across languages without requiring translations.

At its core, multilingual similarity retrieval focuses on determining the closeness or similarity of sentence pairs from different languages, which can be represented as embedding vectors in a high-dimensional space. The goal is to ensure that similar concepts expressed in different languages map to nearby points in this shared embedding space. Techniques such as *cosine similarity* are commonly used to measure how similar the embeddings of two sentences are, with a higher cosine similarity indicating greater semantic similarity.

To perform multilingual similarity retrieval, various pre-trained models have been developed to encode text from different languages into a common semantic space. These models typically use *transformer-based architectures*, which are highly effective in capturing contextual information and relationships between words and sentences. Some prominent models used for multilingual sentence embeddings include:

- **Sentence-BERT (SBERT)**: Sentence-BERT [62] is a modification of the BERT (Bidirectional Encoder Representations from Transformers) model. It has been fine-tuned for producing sentence-level embeddings that can be efficiently compared using cosine similarity. Multilingual versions of SBERT,

such as `paraphrase-xlm-r-multilingual-v1`, are widely used for tasks that involve matching sentences from different languages. These models ensure that semantically similar sentences across languages are positioned close to one another in the embedding space.

- **LASER (Language-Agnostic SEntence Representations)**: LASER [63], developed by Facebook AI, is another model designed for multilingual tasks. It supports over 90 languages and is particularly effective in cross-lingual settings where there is no overlap between languages in the training data. LASER encodes sentences from different languages into a single embedding space, allowing for direct comparison across languages.

- **XLM-R (Cross-lingual Language Model-RoBERTa)**: XLM-R [28]is a robust model for multilingual tasks, trained on a large corpus in 100 languages. XLM-R is particularly useful for similarity retrieval because of its ability to generate high-quality sentence embeddings. In the context of multilingual similarity retrieval, models like `distiluse-base-multilingual-cased-v2` leverage XLM-R to produce embeddings that are comparable across languages.

3.6.1 Cosine Similarity in Multilingual Retrieval

Once sentence embeddings are generated using models like SBERT, LASER, or XLM-R, the next step is to compute the similarity between sentences. *Cosine similarity* is the most common metric for this task, as it measures the cosine of the angle between two vectors (representing sentences). If the vectors are aligned closely, their cosine similarity will be near 1, indicating high similarity. Conversely, if they point in different directions, the cosine similarity will be near 0, indicating little to no similarity.

Given a query sentence in one language (e.g., English), the system can calculate its similarity with sentences in multiple other languages, enabling it to retrieve the most semantically similar sentences. This method eliminates the need for machine translation, making it efficient for multilingual systems.

Some applications of multilingual similarity retrieval are:

- **Cross-lingual Information Retrieval**: Allows users to submit queries in one language and retrieve relevant content in various other languages without the need for translation.

- **Multilingual Question Answering**: Helps match questions to the most semantically relevant answers, regardless of language, enhancing the usability of global-scale information systems.

- **Cross-lingual Document Matching**: Used in applications like plagiarism detection or document clustering, where documents written in different languages need to be compared based on their content.

Multilingual similarity retrieval plays a crucial role in enabling cross-lingual tasks by leveraging multilingual sentence embeddings and cosine similarity. It facilitates efficient comparison of text across languages without relying on translation models. As technology evolves, multilingual similarity retrieval will continue to advance, broadening its impact in various NLP applications.

3.7 Retrieval-augmented in-context learning

ICL is a non-intrusive method for supplying additional information to LLMs without altering their internal parameters. This approach works by appending relevant context to an existing prompt, which helps guide the language generation process. Additionally, retrieval-augmented ICL has been introduced, allowing the model to pull information from external databases when necessary. Off-the-shelf search engines are commonly employed to enhance this method [64, 65], locating semantically similar examples to improve LLM performance in few-shot learning scenarios. In cross-lingual cases, the search engine uses a sample from a low-resource language as a query to find the most semantically similar sample from a high-resource language. The retrieved high-resource language sample is then combined with the input to create a prompt for the LLM.

For example, Nie et al.[66] utilize semantically similar cross-lingual sentences as prompts to enhance sentiment classification in low-resource languages. While ICL is helpful for supporting cross-lingual tasks, the challenge of knowledge editing across different languages remains unexplored.

3.7.1 Zero-Shot and Few-Shot Knowledge Editing

In the context of ICL, two approaches are commonly used for knowledge editing:

- **Zero-shot knowledge editing**
- **Few-shot knowledge editing**

1. Zero-Shot Knowledge Editing

In the **zero-shot approach**, new knowledge is combined with the user-provided input, also referred to as the "test input," to create what is known as the "zero-shot prompt." This prompt serves as the primary input to guide the model in predicting the output $P(y_{l1}|x_{l1}, k_{i*l2})$. Unlike few-shot learning, the zero-shot method does

not rely on additional examples for guidance; instead, the model directly applies the new knowledge to the test input, depending solely on the context provided in the prompt.

The zero-shot approach is highly efficient, especially in cases where example data is either unavailable or minimal. By concatenating the new knowledge with the input, the model is tasked with predicting the most appropriate outcome based only on the provided context. This approach allows the model to generalize the application of new knowledge without additional guidance, making it ideal for quick, direct knowledge editing tasks.

The prediction process can be expressed with the formula:

$$P(y_{l1}|x_{l1}, k_{i*l2}),$$

where y_{l1} is the target output in the desired language $l1$, x_{l1} represents the test input, and k_{i*l2} is the new knowledge added to the prompt.

In practice, zero-shot knowledge editing is particularly useful for applying new information without needing to fine-tune the model or provide extensive training examples, leveraging the model's pre-existing generalization capabilities.

2. Few-Shot Knowledge Editing

In contrast, the **few-shot approach** incorporates additional context by introducing a set of bilingual examples $S = \{(s^1_{l1}, s^1_{l2}), ..., (s^q_{l1}, s^q_{l2})\}$, where s^j_{l1} and s^j_{l2} are the same statement in two different languages, $l1$ and $l2$. These bilingual examples are positioned between the new knowledge and the test input, resulting in a more comprehensive prompt, referred to as the "few-shot prompt."

The few-shot prompt is created by concatenating the "new knowledge," "bilingual examples," and "test input." This additional context helps the model to form stronger associations between the new knowledge and the input, thereby improving the accuracy of its predictions. The formula for predicting the output in this scenario is:

$$P(y_{l1}|x_{l1}, k_{i*l2}, S),$$

where S represents the bilingual examples added to provide further context and guide the model in applying the new knowledge.

Few-shot learning enhances the model's ability to generalize across languages and tasks by giving it more examples to work with, allowing it to better understand how the test input and the new knowledge relate to each other. This makes few-shot learning particularly effective in multilingual settings, where the model must handle complex relationships between languages.

Selecting Examples for Few-Shot Learning

For the few-shot setting, it is crucial to select appropriate examples that are semantically similar to the test input. In this approach, we use a multilingual similarity retrieval method to identify the most relevant examples from a training corpus spanning 12 languages. The examples are chosen based on their cosine similarity to the input sentences, using the multilingual Sentence-BERT model.

These semantically similar examples are incorporated into the few-shot prompt, improving the model's ability to predict the correct output by leveraging examples that closely match the input. By selecting high-quality examples, the model can learn more effectively during in-context learning, allowing for more accurate knowledge editing in cross-lingual scenarios.

The few-shot approach, with its reliance on carefully selected examples, significantly boosts the model's performance, particularly in scenarios where a deeper understanding of linguistic nuances is required.

3.8 Implementation Details

In this section, we outline the technical details involved in the implementation of the proposed multilingual knowledge editing methods—MEMIT, In-Context Learning (ICL), and the extended TailoredKE. This includes descriptions of the model architectures, training configurations, and the specific modifications made to adapt these techniques to multilingual settings.

3.8.1 Model Architectures

For our experiments, we used the following pre-trained LLMs:

- **GPT-J (6 billion parameters)**: GPT-J is an autoregressive language model pre-trained on English text. It was selected for its robust architecture and scalability, making it suitable for testing large-scale knowledge edits.

- **LLaMA-2 (7 billion parameters)**: LLaMA-2 is a smaller and efficient variant of large language models trained on diverse languages. This model was selected for its multilingual capabilities and was used for testing the multilingual extensions of TailoredKE. It provides a strong baseline for assessing the performance of knowledge editing in languages other than English.

- **Mistral (7 billion parameters)**: Mistral was used for its ability to handle dense multilingual tasks, offering robust performance across high- and low-resource languages.

3.8.2 MEMIT Implementation

The MEMIT method was implemented according to the original design, targeting specific MLP layers within the transformer architecture to update factual knowledge. MEMIT's primary advantage is its scalability, allowing for thousands of knowledge edits simultaneously.

In this implementation:

- We used causal mediation analysis to identify the key-value pairs stored in the middle layers of the transformer model responsible for factual recall.
- The layer updates were performed on the identified MLP layers, with the edits spread across multiple layers to prevent overfitting.
- MEMIT was applied to both high- and low-resource languages, with additional data augmentation for low-resource languages to enhance the model's ability to recall and apply the updated knowledge.

3.8.3 ICL Implementation

ICL was implemented by providing demonstrations of factual updates in the input context of the model. These demonstrations included examples of the new facts, paraphrases, and unrelated knowledge to guide the model in retaining its original knowledge. For multilingual tasks, the demonstrations were provided in multiple languages to ensure that the updated knowledge generalized effectively across different linguistic contexts.

The ICL prompts followed this format:

- **Demonstration 1:** An explicit statement of the updated fact.
- **Demonstration 2:** A paraphrase of the fact in the same language.
- **Demonstration 3:** The fact translated into another language.
- **Demonstration 4:** Unrelated facts to guide the model in retaining its previous knowledge.

3.8.4 Multilingual Similarity Retrieval Implementation

In this section, we will detail the implementation steps for finding the most similar sentence in Spanish, German, and French and others for each English sentence and concatenating them together using a pre-trained multilingual embedding model and cosine similarity. The approach uses Sentence-BERT (SBERT) for generating multilingual sentence embeddings and cosine similarity for measuring sentence similarity.

- **Model Selection:** We will use a pre-trained multilingual Sentence-BERT model from the Hugging Face library. Models such as paraphrase-xlm-r-multilingual-v1 or distiluse-base-multilingual-cased-v2 are capable of embedding sentences from different languages into a shared semantic space, allowing us to compare sentences from different languages directly.
- **Data Preprocessing:** The dataset consists of 742 English sentences, each associated with 20 sentences in Spanish, German, and French and other languages. We need to structure the data such that for each English sentence, we have its corresponding 20 sentences in the 12 target languages.
- **Generate Embeddings:** For each English sentence and its 20 associated sentences in each languages, we generate embeddings using the selected multilingual model.
- **Cosine Similarity Calculation:** Using cosine similarity, we compare the embedding of the English sentence with the embeddings of the 20 sentences in each target language (Spanish, German, and other languages). Cosine similarity provides a measure of how similar the two sentence vectors are.
- **Selecting the Most Similar Sentence:** For each English sentence, we select the sentence in each target language with the highest cosine similarity score. This represents the most semantically similar sentence across languages.
- **Concatenating the Sentences:** After selecting the most similar sentences from each language, we concatenate them together with the original English sentence to form the final result.

3.8.5 TailoredKE Implementation

The implementation of TailoredKE follows the methodology outlined in the original paper, with modifications made to accommodate the multilingual extension. Specifically:

Layer Selection: The Dynamic Editing Window (DEW) was extended to account for different languages. For each factual edit, we analyzed how the subject and object representations evolved across transformer layers in the context of multiple languages. Layers that recalled the factual information most accurately were selected for editing.

Knowledge Injection Process: The knowledge injection process was performed by applying the identified edits to the selected layers. Edits were applied using low-rank factorization, which efficiently updates the model's parameters while minimizing interference with unrelated knowledge. For each layer, the rank of the low-rank update was set to 4, as suggested by [17].

3.8.6 Extending TailoredKE to Multilingual Knowledge Editing

In this thesis, the TailoredKE approach is extended to handle *multilingual knowledge editing*. While the original method is designed for English, the extension introduces the ability to apply edits across multiple languages.

To achieve this, the multilingual extension adapts the Dynamic Editing Window to operate across different language-specific representations. For example, the layers responsible for recalling the attributes of "iPod" in English might differ from those in Chinese. By dynamically selecting the appropriate layers for each language, the method ensures that knowledge edits are applied consistently across languages, reducing the likelihood of inconsistency in knowledge recall.

By combining tailored layer selection with multilingual transfer capabilities, this thesis presents a robust solution to the challenges of editing factual knowledge in multilingual language models, ensuring both precision and scalability across languages.

3.8.7 Hardware and Software

The experiments were conducted on an NVIDIA RTX A6000 GPU with 50 GB of memory. The implementation of MEMIT, TailoredKE, and ICL was performed using the Hugging Face Transformers library, along with PyTorch for training and inference. Model checkpoints were saved after each fine-tuning step, and validation was conducted using the MzsRE dataset to monitor progress.

Chapter 4

Evaluation/Results

4.1 Evaluation Metrics

In this section, we describe the metrics used to evaluate the success of knowledge editing methods on multilingual datasets. These metrics are designed to measure the model's ability to inject new knowledge, generalize edits, maintain unrelated knowledge, and transfer knowledge across languages.

4.1.1 Efficacy

Efficacy is the most direct indicator of the success of knowledge editing. It measures the model's ability to correctly generate the newly edited knowledge when queried with the same prompts encountered during the editing process. A high efficacy score indicates that the edited knowledge has been successfully injected into the model.

$$\text{Efficacy} = \mathbb{E}_i \left[o_i^* = \arg\max_{o_i} f_{\theta^*}(o_i | p(s_i, r_i)) \right] \tag{4.1}$$

Where o_i^* is the new object, and f_{θ^*} represents the post-edit model, with s_i as the subject, r_i as the relation, and $p(s_i, r_i)$ being the input prompt.

4.1.2 Generalization

Generalization measures how well the model can apply the updated knowledge to paraphrased or alternate formulations of the original queries. A high generalization score indicates that the model can recognize and apply the edited knowledge across various linguistic contexts.

$$\text{Generalization} = \mathbb{E}_i \left[\mathbb{E}_{p \in \text{neighbour}(s_i, r_i)} \left[o_i^* = \arg\max_{o_i} f_{\theta^*}(o_i | p) \right] \right] \tag{4.2}$$

Here, neighbour(s_i, r_i) refers to paraphrases or alternate expressions of the query.

4.1.3 Specificity

Specificity evaluates how well the model retains unrelated knowledge after the edit. A high specificity score means that the knowledge editing process did not affect unrelated facts or concepts within the model.

$$\text{Specificity} = \mathbb{E}_i \left[\mathbb{E}_{p \in \text{irrelevant}(s_i, r_i)} \left[\arg\max_{o_i'} f_\theta(o_i'|p) = \arg\max_{o_i} f_{\theta^*}(o_i|p) \right] \right] \quad (4.3)$$

Where $f_\theta(o_i'|p)$ refers to the pre-edit model's prediction, ensuring that irrelevant knowledge is preserved.

4.2 Results

4.2.1 Multi Lingual Knowledge Edditing

This section presents the results of our experiments on the MzsRE dataset using the LLaMA-2 backbone. We compare three different methods: *Memit*, *TailoredKE*$_{Targeted}$ (which uses layer selection without rephrasing), and *TailoredKE*$_{Rephrase}$ (which includes sentence rephrasing but without layer selection). The goal is to assess how these techniques perform across multiple languages, with Exact Match (EM) metrics evaluated for consistency, efficacy, and generalization.

Experimental Setup

For each method, we conducted experiments on the following languages: English (EN), French (FR), Spanish (ES), Czech (CZ), German (DE), Dutch (DU), Portuguese (PT), Russian (RU), Thai (TH), Turkish (TR), Vietnamese (VI), and Chinese (ZH).

The baseline *Memit* method tests each language separately without rephrasing or layer selection. *TailoredKE*$_{Rephrase}$ introduces sentence rephrasing without layer selection, while *TailoredKE*$_{Targeted}$ applies selective layer editing without rephrasing.

Table 4.1 reports the EM scores for consistency across all languages.

Table 4.2 presents the efficacy results. *Memit* shows strong performance across most languages, with EM scores consistently above 90% in languages like French and Portuguese. *TailoredKE*$_{Rephrase}$ outperforms *Memit* in certain low-resource languages such as Russian and Vietnamese, indicating that sentence rephrasing can enhance the model's ability to accurately apply edits, even when training data for the target language is limited.

Table 4.1: Exact Match (EM) results for *Specificity* on the LLaMA-2 backbone obtained from testing in multiple languages.

Models	**EN**	**FR**	**ES**	**CZ**	**DE**	**DU**	**PT**	**RU**	**TH**	**TR**	**VI**	**ZH**
Memit	48.55	40.36	41.28	41.6	43.54	42.38	41.1	43.62	35.51	39.02	48.11	36.84
$TailoredKE_{Targeted}$	48.39	40.47	40.51	41.69	43.4	42.67	41.02	44.2	36.13	38.66	48.37	36.0
$TailoredKE_{Rephrase}$	48.21	40.55	40.65	41.89	43.31	42.49	40.8	43.69	35.96	38.97	48.03	36.05

Table 4.2: Exact Match (EM) results for *Efficacy* on the LLaMA-2 backbone obtained from testing in multiple languages.

Models	**EN**	**FR**	**ES**	**CZ**	**DE**	**DU**	**PT**	**RU**	**TH**	**TR**	**VI**	**ZH**
Memit	90.46	91.19	89.79	91.02	89.08	90.45	91.67	90.72	92.27	92.86	91.0	79.89
$TailoredKE_{Targeted}$	83.31	83.94	82.3	83.83	81.91	81.63	84.07	80.77	84.32	84.92	86.67	71.01
$TailoredKE_{Rephrase}$	**91.97**	**93.12**	**92.81**	**93.77**	**92.58**	**92.32**	**92.41**	**94.99**	**92.19**	**97.74**	**95.26**	**81.92**

Table 4.3 shows the EM results for generalization. *TailoredKE$_{Rephrase}$* consistently outperforms the other methods in languages like Czech, German, and Chinese, highlighting the importance of rephrasing in generalizing knowledge edits. *Memit* performs well in high-resource languages such as English and French but struggles slightly with low-resource languages, demonstrating the need for rephrasing techniques to aid in generalization.

The analysis reveals that while Memit performed well as a baseline method, especially in high-resource languages such as English and French, both TailoredKE$_{\text{Targeted}}$ and TailoredKE$_{\text{Rephrase}}$ outperformed it in several lower-resource languages.

TailoredKE$_{\text{Rephrase}}$ demonstrated superior efficacy and generalization, particularly in languages like Turkish and Czech, where rephrasing plays a critical role in ensuring the accurate application of knowledge edits. TailoredKE$_{\text{Targeted}}$, on the other hand, showed notable improvements in Vietnamese and Russian, highlighting the advantages of layer selection for these languages.

In conclusion, the combination of sentence rephrasing and layer selection offers clear advantages in enhancing the performance of knowledge editing across languages, particularly in those with fewer training resources. These findings suggest that incorporating both techniques can lead to more effective cross-lingual knowledge editing, and they will be further explored in subsequent sections of this thesis.

Table 4.3: Exact Match (EM) results for *Generalization* on the LLaMA-2 backbone obtained from testing in multiple languages.

Models	**EN**	**FR**	**ES**	**CZ**	**DE**	**DU**	**PT**	**RU**	**TH**	**TR**	**VI**	**ZH**
Memit	87.21	88.81	87.92	85.89	85.78	86.17	88.31	88.65	84.35	89.87	86.95	77.58
TailoredKE$_{Targeted}$	76.82	78.28	78.72	74.77	74.67	75.07	77.09	75.14	74.58	78.69	78.1	67.2
TailoredKE$_{Rephrase}$	**90.34**	**91.19**	**90.77**	**90.74**	**89.07**	**89.01**	**90.76**	**94.1**	**87.89**	**95.94**	**91.55**	**78.94**

4.2.2 Layer Selection Distribution Across Languages

In this section, we analyze the distribution of layer selections across various languages for the LLaMA-2 model. The layer selection refers to the specific transformer layers chosen during the knowledge-editing process, which could highlight how different languages engage distinct or overlapping regions of the model.

Layer Selection Distribution Analysis

This section presents the distribution of layer selections for each language, showing the number of times a particular layer (from Layer 4 to Layer 8) was chosen. This distribution is visualized in Figure 4.1, which depicts the layer-wise selection frequency for all languages.

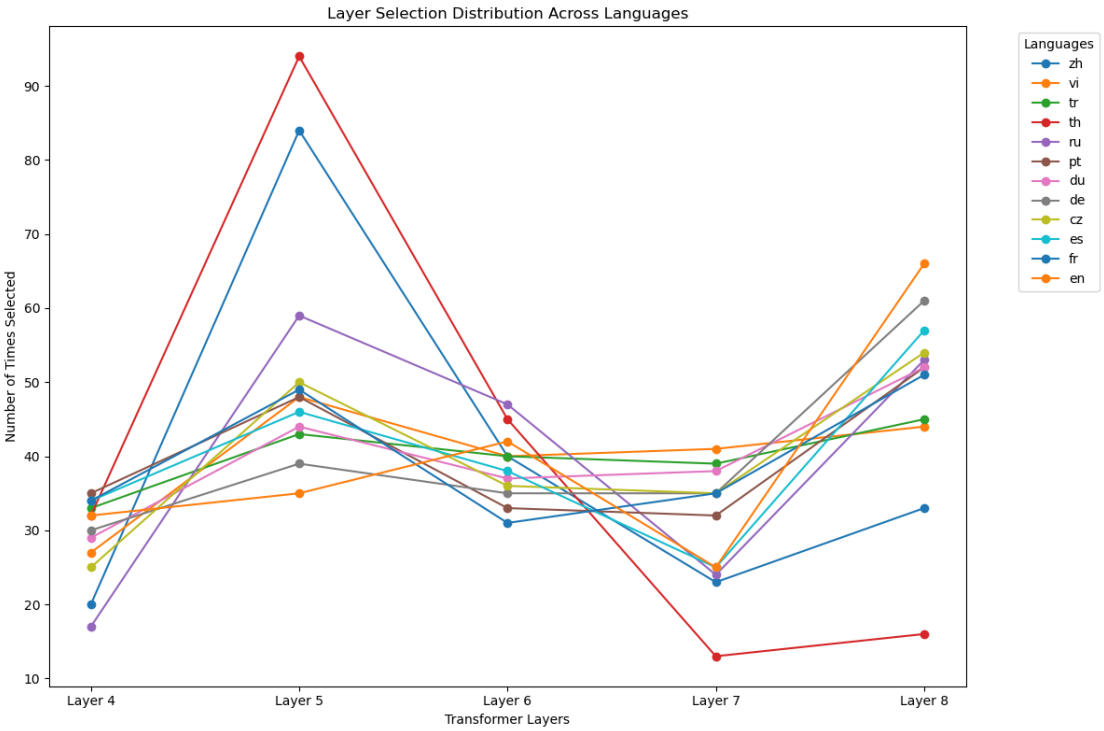

Figure 4.1: Layer Selection Distribution Across Languages for LLaMA-2. The graph shows how different transformer layers are selected for various languages.

Relevance of Language Families

A key question in this analysis is whether languages with similar roots (e.g., Romance languages such as Portuguese, Spanish, and French) display similar layer

selection behavior. By comparing the layer selection distributions, we observe that languages with common linguistic roots tend to exhibit some overlap in the layer selection patterns. For instance, in Romance languages like Portuguese, Spanish, and French, we see higher frequencies of selection for Layer 5 and Layer 8. However, distinct differences still exist, which may be attributed to language-specific nuances and varying resource availability for training the model.

Both Chinese (zh) and Thai (th) rely heavily on Layer 5, suggesting that middle layers of the transformer architecture are crucial for managing language features in these structurally complex languages.

On the other hand, languages from entirely different linguistic families, such as Chinese (zh) and Russian (ru), display more divergence in their layer selection, suggesting that the model engages different parts of its architecture depending on the language's structural characteristics.

These results show the relevance of language families in selecting layers for processing, as languages with similar linguistic features or roots tend to exhibit comparable layer selection patterns. This insight can be used in cross-lingual settings to enable faster edits, as knowing which layers are likely to be selected for certain languages can significantly improve the efficiency of knowledge editing across multiple languages. The implications of these findings and their application in cross-lingual knowledge editing will be further discussed in the next section.

4.3 Cross-Lingual Knowledge Editing

In this evaluation, we focus on cross-lingual knowledge editing, where the objective is to perform edits in one language (English) and test the model's performance in other target languages. We experimented with four different approaches to assess the effectiveness of various configurations for cross-lingual knowledge editing:

1. ***TailoredKE*$_{Rephrase}$ in source language without prompts**: Edit in English using rephrased sentences with no prompts or in-context learning during testing in the target language.

2. ***TailoredKE*$_{Rephrase}$ in source and target languages without prompts**: Edit using a combination of rephrased sentences from both English and the target language without any prompts in testing time.

3. ***TailoredKE*$_{Rephrase}$ in source language with zero-shot ICL for the target language**: Edit in English and test in the target language using zero-shot in-context learning.

4. ***TailoredKE*$_{Rephrase}$ in source language with few-shot ICL for the target language**: Edit in English and test in the target language using few-shot in-context learning.

4.3.1 Approach 1: *TailoredKE*$_{Rephrase}$ in Source Language Without Prompts

Initially, we used the *TailoredKE*$_{Rephrase}$ method to edit knowledge in English and tested in the target language without any prompts or in-context learning. This approach involved providing 20 rephrased sentences in English as context during the edit process, while the model predicted in the target language. The results were satisfactory, especially when compared to the results in the ReMaKe paper's original mode. However, the absence of prompts during testing limited the cross-lingual accuracy.

4.3.2 Approach 2: *TailoredKE*$_{Rephrase}$ in Source and Target Languages Without Prompts

In this approach, we combined rephrased sentences from both English (source) and the target language to edit knowledge, without any prompt in testing time. Interestingly, this approach did not lead to further improvements; in some cases, there was even a decrease in performance. This suggests that mixing sentences from both the source and target languages during the edit phase in a cross-lingual setting may introduce inconsistencies or noise, which could hinder the model's ability to accurately predict new knowledge. This result highlights that combining source and target language prompts during editing may not be an effective strategy for improving LLM performance in cross-lingual tasks.

4.3.3 Approach 3: *TailoredKE*$_{Rephrase}$ in Source Language with Zero-Shot ICL

Next, we applied *TailoredKE*$_{Rephrase}$ for editing in English and tested in the target language using zero-shot ICL. This method demonstrated a significant improvement in performance across all languages, including both low-resource and high-resource languages. The inclusion of zero-shot ICL allowed the model to better generalize and apply the edited knowledge in cross-lingual settings.

4.3.4 Approach 4: *TailoredKE*$_{Rephrase}$ in Source Language with Few-Shot ICL

Finally, we employed *TailoredKE*$_{Rephrase}$ in English, but used few-shot ICL during testing in the target language. This configuration yielded the best results across all approaches. The combination of accurate edits in the source language and the added context from few-shot prompts enabled more precise predictions in the target language. To select sentences for the few-shot ICL prompts, we used a

retrieval-based similarity model to identify the most semantically similar sentences, which further contributed to improved model performance.

From our experiments, which results showed in Table 4.5 for generalization, Table 4.4 for efficacy and Table 4.6 for specificity, it is clear that using few-shot ICL for cross-lingual knowledge editing provides the most robust results, outperforming both zero-shot and non-prompted approaches. While the zero-shot ICL method provided significant improvements, combining source and target language prompts during editing did not lead to further gains and, in some cases, resulted in performance drops. The use of $TailoredKE_{Rephrase}$ in the source language with carefully selected few-shot prompts proved to be the most effective method for achieving accurate cross-lingual knowledge edits.

Table 4.4: Exact Match (EM) results for *Efficacy* on the LLaMA-2 backbone obtained from editing in English and testing in cross_lingual setting.

Models	**FR**	**ES**	**DE**	**DU**	**CZ**	**PT**	**RU**	**TR**	**TH**	**VI**	**ZH**
Approach 1	56.55	53.83	59.95	57.71	56.99	54.79	44.95	50.94	49.95	62.08	47.96
Approach 2	54.44	51.12	57.68	56.54	55.37	52.84	43.89	49.78	48.69	61.12	46.58
Approach 3	73.52	72.26	77.43	74.56	71.35	70.01	65.48	69.73	68.69	79.12	66.79
Approach 4	86.73	85.37	88.28	83.12	81.23	80.35	75.14	85.33	78.69	83.28	78.01

Table 4.5: Exact Match (EM) results for *Generalization* on the LLaMA-2 backbone obtained from editing in English and testing in cross_lingual setting.

Models	**FR**	**ES**	**DE**	**DU**	**CZ**	**PT**	**RU**	**TR**	**TH**	**VI**	**ZH**
Approach 1	56.45	54.33	59.76	58.9	55.66	54.23	44.95	50.6	34.27	52.81	43.04
Approach 2	53.69	52.74	58.88	57.13	53.68	51.78	43.67	48.98	33.26	51.39	42.68
Approach 3	72.94	71.13	74.11	72.83	68.74	67.72	58.63	64.58	53.58	67.03	57.48
Approach 4	86.73	84.98	88.28	86.96	79.07	77.09	68.79	86.81	73.64	81.1	77.71

Table 4.6: Exact Match (EM) results for *Specificity* on the LLaMA-2 backbone obtained from editing in English and testing in cross_lingual setting.

Models	**FR**	**ES**	**DE**	**DU**	**CZ**	**PT**	**RU**	**TR**	**TH**	**VI**	**ZH**
Approach 1	40.5	40.88	43.31	42.24	41.72	40.96	43.72	38.89	34.27	38.13	36.6
Approach 2	40.37	40.46	42.36	41.26	40.03	39.85	42.51	37.77	33.59	37.78	35.89
Approach 3	68.51	67.32	69.3	66.35	64.34	64.93	70.03	65.31	59.02	62.16	60.38
Approach 4	73.65	67.86	69.71	68.34	66.93	67.03	71.62	66.82	59.93	63.09	61.41

Chapter 5

Conclusion

5.1 Overview

In this thesis, we addressed the challenges of multilingual knowledge editing for large language models (LLMs) by combining several innovative methodologies. The primary focus was on ensuring that knowledge edits performed in one language could be effectively transferred and applied in other languages, particularly in multilingual and cross-lingual settings. The approaches explored, including MEMIT, In-Context Learning (ICL), TailoredKE, and retrieval-augmented methods, were designed to achieve efficient knowledge updates while maintaining high accuracy and generalization across languages. Our experiments were conducted using state-of-the-art LLMs such as LLaMA-2, and validated on the MzsRE dataset, a multilingual extension of the zsRE dataset.

5.2 Summary of Methodology and Findings

The key contributions of this thesis can be summarized as follows:

5.2.1 Multilingual Knowledge Editing with MEMIT

MEMIT (Mass-Editing Memory in Transformers) proved to be an effective technique for editing factual knowledge within specific layers of the transformer models. By identifying critical MLP layers responsible for factual recall, MEMIT was able to inject new knowledge efficiently while minimizing interference with unrelated information. This method is scalable and capable of handling thousands of edits simultaneously, making it highly suitable for large-scale applications.

The multilingual extension of MEMIT was particularly successful, as it allowed knowledge edits to propagate across different languages. The experiments

demonstrated that MEMIT could consistently apply knowledge updates in both high-resource and low-resource languages. By spreading edits across multiple layers, MEMIT preserved the integrity of unrelated facts, ensuring that the model's overall performance remained robust across linguistic boundaries.

5.2.2 In-Context Learning (ICL) for Knowledge Editing

In-Context Learning (ICL) emerged as a powerful, non-intrusive method for knowledge editing, particularly in scenarios where direct parameter updates were not desirable. By providing task-specific demonstrations within the input context, ICL guided the model to generate accurate outputs without modifying its internal parameters. This flexibility made ICL highly effective for temporary or context-specific knowledge updates.

In multilingual settings, ICL demonstrated its potential to support cross-lingual knowledge transfer. By providing demonstrations in different languages, the model was able to generalize knowledge updates across both high-resource and low-resource languages. The use of zero-shot and few-shot ICL approaches further enhanced the model's ability to apply updated knowledge to various linguistic contexts. Few-shot ICL, in particular, significantly boosted performance by incorporating multilingual examples as prompts, leading to better generalization of edits across languages.

5.2.3 TailoredKE for Selective Layer Editing

TailoredKE introduced a novel method for precise knowledge editing by focusing on specific transformer layers. The dynamic layer selection mechanism allowed for more targeted edits, reducing the likelihood of over-editing and preserving unrelated facts. TailoredKE's strength lies in its ability to identify the layers most responsible for recalling factual information and applying edits selectively to those layers.

In the multilingual extension of TailoredKE, we observed that dynamic layer selection was crucial for ensuring that knowledge edits applied in one language could be appropriately transferred to other languages. This method allowed for a more consistent representation of knowledge across languages, improving the accuracy of cross-lingual knowledge recall.

5.2.4 Multilingual Similarity Retrieval Task

The multilingual similarity retrieval task, which involved retrieving semantically similar sentences across languages using models like Sentence-BERT, LASER, and XLM-R, was essential for improving the performance of few-shot ICL. By calculating cosine similarity between sentence embeddings, the system identified

the most relevant examples for each language, enhancing the model's ability to generalize knowledge edits across different linguistic contexts.

The use of cosine similarity in combination with pre-trained multilingual embeddings proved to be highly effective in matching sentences across languages. This method allowed the model to bypass the need for machine translation, making it more efficient in retrieving semantically similar information in a multilingual setting.

5.3 Evaluation Results

5.3.1 Cross-Lingual Knowledge Editing

The evaluation results highlighted the effectiveness of each approach in cross-lingual knowledge editing. While MEMIT showed competitive performance in high-resource languages, the introduction of ICL and TailoredKE significantly improved the model's ability to generalize knowledge edits across multiple languages.

The few-shot ICL approach provided the most robust results, particularly in low-resource languages where the availability of training data was limited. By selecting high-quality multilingual examples based on cosine similarity, few-shot ICL demonstrated superior efficacy, specificity, and generalization. This method outperformed zero-shot approaches and non-prompted editing methods, making it the most reliable technique for cross-lingual knowledge editing.

The evaluation also revealed that mixing source and target language prompts during the zero-shot ICL phase did not lead to performance improvements. In some cases, it even caused a slight decrease in accuracy, suggesting that combining language prompts can introduce noise, rather than enhancing the model's ability to apply knowledge edits across languages.

5.3.2 Generalization and Specificity

Both MEMIT and TailoredKE demonstrated strong generalization capabilities, allowing the model to apply knowledge edits across different paraphrased queries and linguistic contexts.

The generalization metric showed that sentence rephrasing, combined with selective layer editing, allowed the model to retain and apply knowledge edits more effectively than traditional fine-tuning methods. This capability is crucial for LLMs operating in multilingual environments, where consistency and accuracy across languages are essential.

5.4 Challenges and Future Work

Despite the promising results, several challenges remain in the field of multilingual knowledge editing. One of the key challenges is ensuring the scalability of these methods for larger models and more diverse languages. Low-resource languages, in particular, still pose difficulties due to the lack of sufficient training data. While data augmentation techniques helped mitigate some of these issues, further research is needed to improve the robustness of multilingual knowledge editing methods in low-resource languages.

Future work should also explore more sophisticated retrieval-augmented techniques, such as ReMaKE, to enhance the transferability of knowledge edits across languages. Additionally, improving the interpretability of layer selection and understanding how different languages interact with specific transformer layers will be critical for further advancements in this field.

In conclusion, this thesis has demonstrated that combining ICL, TailoredKE and zero-shot and few-shot prompts which offers a robust framework for multilingual knowledge editing in LLMs. The ability to perform scalable, precise knowledge edits while maintaining consistency and generalization across languages represents a significant advancement in the field. By addressing the challenges of cross-lingual knowledge transfer, this research provides valuable insights for future work on multilingual LLMs and their applications in diverse linguistic environments.

Bibliography

[1] Randall Davis, Howard E Shrobe, and Peter Szolovits. «What is a knowledge representation?» In: *AI Magazine* 14.1 (1993), pp. 17–33. DOI: 10.1609/AIMAG.V14I1.1029. URL: https://doi.org/10.1609/aimag.v14i1.1029 (cit. on p. 1).

[2] Yejin Choi. «Knowledge is Power: Symbolic Knowledge Distillation, Commonsense Morality, & Multimodal Script Knowledge». In: WSDM '22. Virtual Event, AZ, USA: Association for Computing Machinery, 2022, p. 3. ISBN: 9781450391320. DOI: 10.1145/3488560.3500242. URL: https://doi.org/10.1145/3488560.3500242 (cit. on p. 1).

[3] Hongming Zhang, Xin Liu, Haojie Pan, Haowen Ke, Jiefu Ou, Tianqing Fang, and Yangqiu Song. «ASER: Towards large-scale commonsense knowledge acquisition via higher-order selectional preference over eventualities». In: *Artificial Intelligence* 309 (2022), p. 103740. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2022.103740. URL: https://www.sciencedirect.com/science/article/pii/S0004370222000807 (cit. on p. 1).

[4] Christopher D Manning. «Human language understanding & reasoning». In: *Daedalus* 151.2 (2022), pp. 127–138 (cit. on p. 1).

[5] Karen L McGraw and Karan Harbison-Briggs. *Knowledge acquisition: Principles and guidelines*. Prentice Hall, 1990. ISBN: 978-0-13-517095-3 (cit. on p. 1).

[6] Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. «Linguistic Knowledge and Transferability of Contextual Representations». In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 1073–1094. DOI: 10.18653/v1/N19-1112. URL: https://aclanthology.org/N19-1112 (cit. on p. 1).

[7] Xu Han, Zhengyan Zhang, and Zhiyuan Liu. «Knowledgeable machine learning for natural language processing». In: *Communications of the ACM* 64.11 (2021), pp. 50–51. DOI: 10.1145/3481608. URL: https://doi.org/10.1145/3481608 (cit. on p. 1).

[8] Mohammad Hossein Jarrahi, David Askay, Ali Eshraghi, and Preston Smith. «Artificial intelligence and knowledge management: A partnership between human and AI». In: *Business Horizons* 66.1 (2023), pp. 87–99. ISSN: 0007-6813. DOI: https://doi.org/10.1016/j.bushor.2022.03.002. URL: https://www.sciencedirect.com/science/article/pii/S0007681322000222 (cit. on p. 1).

[9] Huajun Chen. «Large knowledge model: Perspectives and challenges». In: *CoRR* abs/2312.02706 (2023). DOI: 10.48550/ARXIV.2312.02706. URL: https://doi.org/10.48550/arXiv.2312.02706 (cit. on p. 1).

[10] Josh Achiam et al. «Gpt-4 technical report». In: *arXiv preprint arXiv:2303.08774* (2023) (cit. on pp. 1, 7).

[11] Hugo Touvron et al. «Llama 2: Open Foundation and Fine-Tuned Chat Models». In: (2023) (cit. on pp. 1, 7).

[12] Peng Wang et al. «EasyEdit: An Easy-to-use Knowledge Editing Framework for Large Language Models». In: *arXiv preprint arXiv:2308.07269* (2023). URL: https://arxiv.org/abs/2308.07269 (cit. on pp. 1, 2).

[13] Ningyu Zhang et al. «A Comprehensive Study of Knowledge Editing for Large Language Models». In: *arXiv preprint arXiv:2401.01286* (2024). URL: https://arxiv.org/abs/2401.01286 (cit. on p. 1).

[14] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. «Editing Large Language Models: Problems, Methods, and Opportunities». In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10222–10240. DOI: 10.18653/v1/2023.emnlp-main.632. URL: https://aclanthology.org/2023.emnlp-main.632 (cit. on pp. 1, 2, 34).

[15] Weixuan Wang, Barry Haddow, and Alexandra Birch. «Retrieval-Augmented Multilingual Knowledge Editing». In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 335–354. DOI: 10.18653/v1/2024.acl-long.21. URL: https://aclanthology.org/2024.acl-long.21 (cit. on pp. 1, 2, 4, 28, 29).

[16] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. «Knowledge Neurons in Pretrained Transformers». In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 8493–8502. DOI: 10.18653/v1/2022.acl-long.581. URL: https://aclanthology.org/2022.acl-long.581 (cit. on pp. 2, 11, 18, 22).

[17] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. «Mass-editing memory in a transformer». In: *arXiv preprint arXiv:2210.07229* (2022) (cit. on pp. 2, 11, 20, 22, 27, 28, 32, 35, 41).

[18] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. «Locating and editing factual associations in GPT». In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 17359–17372 (cit. on pp. 2, 10, 11, 15, 19, 22, 28).

[19] Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. «Multilingual large language model: A survey of resources, taxonomy and frontiers». In: *arXiv preprint arXiv:2404.04925* (2024) (cit. on p. 2).

[20] Jiaan Wang, Yunlong Liang, Zengkui Sun, Yuxuan Cao, Jiarong Xu, and Fandong Meng. «Cross-lingual knowledge editing in large language models». In: *arXiv preprint arXiv:2309.08952* (2023) (cit. on pp. 3, 27–29).

[21] Himanshu Beniwal, Mayank Singh, et al. «Cross-lingual editing in multilingual language models». In: *arXiv preprint arXiv:2401.10521* (2024) (cit. on pp. 3, 27–29, 32, 34).

[22] A Vaswani. «Attention is all you need». In: *Advances in Neural Information Processing Systems* (2017) (cit. on pp. 7–9).

[23] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding». In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423 (cit. on p. 7).

[24] Tom Brown et al. «Language Models are Few-Shot Learners». In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 1877–1901 (cit. on pp. 7, 21).

[25] Albert Q. Jiang et al. *Mistral 7B*. 2023. URL: https://mistral.ai (cit. on p. 7).

[26] Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. «Transformer Feed-Forward Layers Build Predictions by Promoting Concepts in the Vocabulary Space». In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 30–45. DOI: 10.18653/v1/2022.emnlp-main.3. URL: https://aclanthology.org/2022.emnlp-main.3 (cit. on p. 11).

[27] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. «Memory-based model editing at scale». In: *International Conference on Machine Learning*. PMLR. 2022, pp. 15817–15831 (cit. on pp. 11, 17, 22).

[28] Alexis Conneau et al. «Unsupervised Cross-lingual Representation Learning at Scale». In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747 (cit. on pp. 14, 36).

[29] BigScience Workshop et al. «Bloom: A 176b-parameter open-access multilingual language model». In: *arXiv preprint arXiv:2211.05100* (2022) (cit. on p. 14).

[30] Shijie Wu and Mark Dredze. «Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT». In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 833–844. DOI: 10.18653/v1/D19-1077. URL: https://aclanthology.org/D19-1077 (cit. on p. 15).

[31] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. «Parameter-efficient transfer learning for NLP». In: *International conference on machine learning*. PMLR. 2019, pp. 2790–2799 (cit. on p. 15).

[32] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. «Lora: Low-rank adaptation of large language models». In: *arXiv preprint arXiv:2106.09685* (2021) (cit. on p. 15).

[33] Patrick Lewis et al. «Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks». In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf (cit. on p. 16).

[34] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. «Retrieval Augmented Language Model Pre-Training». In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 13–18 Jul 2020, pp. 3929–3938. URL: https://proceedings.mlr.press/v119/guu20a.html (cit. on p. 16).

[35] Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Fei Sha, and William W. Cohen. «Mention Memory: Incorporating Textual Knowledge into Transformers Through Entity Mention Attention». In: *International Conference on Learning Representations (ICLR)*. 2022. URL: https://openreview.net/forum?id=OY1A8ejQgEX (cit. on p. 16).

[36] Yunzhi Yao, Shaohan Huang, Li Dong, Furu Wei, Huajun Chen, and Ningyu Zhang. «Kformer: Knowledge injection in transformer feed-forward layers». In: *CCF International Conference on Natural Language Processing and Chinese Computing*. Springer. 2022, pp. 131–143 (cit. on p. 16).

[37] James Kirkpatrick, Razvan Pascanu, Neil C Rabinowitz, et al. «Overcoming catastrophic forgetting in neural networks». In: *Proceedings of the National Academy of Sciences* 114 (2016), pp. 3521–3526. DOI: 10.1073/pnas.1611835114. URL: https://api.semanticscholar.org/CorpusID:4704285 (cit. on p. 16).

[38] Tom Mitchell, William Cohen, Estevam Hruschka, et al. «Never-ending learning». In: *Communications of the ACM* 61.5 (2018), pp. 103–115. DOI: 10.1145/3191513 (cit. on p. 16).

[39] Nicola De Cao, Wilker Aziz, and Ivan Titov. «Editing factual knowledge in language models». In: *arXiv preprint arXiv:2104.08164* (2021) (cit. on pp. 18, 22).

[40] Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. «Pmet: Precise model editing in a transformer». In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 17. 2024, pp. 18564–18572 (cit. on pp. 18, 22).

[41] Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. «Can we edit factual knowledge by in-context learning?» In: *arXiv preprint arXiv:2305.12740* (2023) (cit. on pp. 19, 22, 27, 30–32).

[42] Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. «MQuAKE: Assessing Knowledge Editing in Language Models via Multi-Hop Questions». In: *Conference on Empirical Methods in Natural Language Processing*. 2023. URL: https://api.semanticscholar.org/CorpusID:258865984 (cit. on pp. 19, 22, 27, 31, 34).

[43] Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. «Evaluating the Ripple Effects of Knowledge Editing in Language Models». In: *Transactions of the Association for Computational Linguistics* 12 (2023), pp. 283–298. URL: https://api.semanticscholar.org/CorpusID:260356612 (cit. on pp. 19, 22, 27, 30).

[44] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. «Fast model editing at scale». In: *arXiv preprint arXiv:2110.11309* (2022) (cit. on pp. 20, 22).

[45] Zeyu Huang, Yikang Shen, Xiaofeng Zhang, Jie Zhou, Wenge Rong, and Zhang Xiong. «Transformer-patcher: One mistake worth one neuron». In: *arXiv preprint arXiv:2301.09785* (2023) (cit. on pp. 20, 22).

[46] Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. «Memory-assisted prompt editing to improve GPT-3 after deployment». In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 2833–2861. DOI: 10.18653/v1/2022.emnlp-main.183. URL: https://aclanthology.org/2022.emnlp-main.183 (cit. on p. 22).

[47] Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. «PokeMQA: Programmable knowledge editing for Multi-hop Question Answering». In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Lun-Wei Ku, Andre Martins, and Vivek Srikumar. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 8069–8083. DOI: 10.18653/v1/2024.acl-long.438. URL: https://aclanthology.org/2024.acl-long.438 (cit. on p. 22).

[48] Qingxiu Dong, Damai Dai, Yifan Song, Jingjing Xu, Zhifang Sui, and Lei Li. «Calibrating Factual Knowledge in Pretrained Language Models». In: *Findings of the Association for Computational Linguistics: EMNLP 2022*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 5937–5947. DOI: 10.18653/v1/2022.findings-emnlp.438. URL: https://aclanthology.org/2022.findings-emnlp.438 (cit. on p. 22).

[49] Evan Hernandez, Belinda Z Li, and Jacob Andreas. «Inspecting and editing knowledge representations in language models». In: *arXiv preprint arXiv:2304.00740* (2023) (cit. on p. 22).

[50] Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. «Aging with grace: Lifelong model editing with discrete key-value adaptors». In: *Advances in Neural Information Processing Systems* 36 (2024) (cit. on p. 22).

[51] Suhang Wu, Minlong Peng, Yue Chen, Jinsong Su, and Mingming Sun. «Evakellm: A new benchmark for evaluating knowledge editing of llms». In: *arXiv preprint arXiv:2308.09954* (2023) (cit. on p. 22).

[52] Lang Yu, Qin Chen, Jie Zhou, and Liang He. «MELO: Enhancing Model Editing with Neuron-Indexed Dynamic LoRA». In: *arXiv preprint arXiv:2303.00001* (2023) (cit. on p. 22).

[53] Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X Yu, and Sanjiv Kumar. «Modifying Memories in Transformer Models». In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2021. URL: https://arxiv.org/abs/2012.00363 (cit. on p. 22).

[54] Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. «Editable Neural Networks». In: 2020. arXiv: 2004.00345 [cs.LG]. URL: https://arxiv.org/abs/2004.00345 (cit. on p. 22).

[55] Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. «Methods for Measuring, Updating, and Visualizing Factual Beliefs in Language Models». In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. Ed. by Andreas Vlachos and Isabelle Augenstein. Dubrovnik, Croatia: Association for Computational Linguistics, May 2023, pp. 2714–2731. DOI: 10.18653/v1/2023.eacl-main.199. URL: https://aclanthology.org/2023.eacl-main.199 (cit. on p. 22).

[56] Chenmien Tan, Ge Zhang, and Jie Fu. «Massive editing for large language models via meta learning». In: *arXiv preprint arXiv:2311.04661* (2023) (cit. on p. 22).

[57] Jun-Yu Ma, Jia-Chen Gu, Zhen-Hua Ling, Quan Liu, and Cong Liu. «Untying the Reversal Curse via Bidirectional Language Model Editing». In: (2024). arXiv: 2310.10322 [cs.CL]. URL: https://arxiv.org/abs/2310.10322 (cit. on p. 22).

[58] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. «Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing». In: *arXiv preprint arXiv:2107.13586* (2021) (cit. on pp. 21, 26).

[59] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. «Chain of thought prompting elicits reasoning in large language models». In: *arXiv preprint arXiv:2201.11903* (2022) (cit. on pp. 23, 24, 26).

[60] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. «Language models are unsupervised multitask learners». In: *OpenAI blog* 1.8 (2019), p. 9 (cit. on p. 24).

[61] Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. «Knowledgeable or Educated Guess? Revisiting Language Models as Knowledge Bases». In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli. Online: Association for Computational Linguistics, Aug. 2021, pp. 1860–1874. DOI: 10.18653/v1/2021.acl-long.146. URL: https://aclanthology.org/2021.acl-long.146 (cit. on pp. 34, 35).

[62] Nils Reimers and Iryna Gurevych. «Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks». In: *arXiv preprint arXiv:1908.10084* (2019) (cit. on p. 35).

[63] Mikel Artetxe and Holger Schwenk. «Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond». In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 597–610 (cit. on p. 36).

[64] Tianyu Gao, Adam Fisch, and Danqi Chen. «Making pre-trained language models better few-shot learners». In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers)*. Association for Computational Linguistics. Virtual Event, Aug. 2021, pp. 3816–3830 (cit. on p. 37).

[65] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. «REPLUG: retrieval-augmented black-box language models». In: *CoRR* abs/2301.12652 (2023). URL: https://arxiv.org/abs/2301.12652 (cit. on p. 37).

[66] Ercong Nie, Sheng Liang, Helmut Schmid, and Hinrich Schütze. «Cross-lingual retrieval augmented prompt for low-resource languages». In: *Findings of the Association for Computational Linguistics: ACL 2023*. Association for Computational Linguistics. Toronto, Canada, July 2023, pp. 8320–8340 (cit. on p. 37).