# MQA-KEAL: Multi-hop Question Answering under Knowledge Editing for Arabic Language

**Muhammad Asif Ali**[1], **Nawal Daftardar**[1,2], **Mutayyaba Waheed**[3],
**Jianbin Qin**[*,4], **and Di Wang**[*1,5]
[1]King Abdullah University of Science and Technology, KSA
[2]King AbdulAziz University, KSA; [3]University of Science and Technology, China
[4]Shenzhen University, China
[5]Center of Excellence for Generative AI, KAUST, KSA

## Abstract

Large Language Models (LLMs) have demonstrated significant capabilities across numerous application domains. A key challenge is to keep these models updated with latest available information, which limits the true potential of these models. Although, there have been numerous attempts for LLMs' Knowledge Editing (KE), *i.e.,* to update and/or edit the LLMs' prior knowledge and in turn test it *via* Multi-hop Question Answering under KE (MQA-KE), yet these studies are primarily focused on English language. In this paper, we extend MQA-KE for Arabic language. For this, we propose: **M**ulti-hop **Q**uestioning **A**nswering under **K**nowledge **E**diting for **A**rabic **L**anguage (MQA-KEAL). MQA-KEAL stores knowledge edits as structured knowledge units in the external memory. In order to solve multi-hop question, it first uses task-decomposition to decompose the question into smaller sub-problems. Later, for each sub-problem it iteratively queries the external memory and/or target LLM in order to generate the final response. In addition, we also contribute MQUAKE-AR (Arabic translation of English benchmark MQUAKE), as well as curate a new benchmark MQA-AEVAL for rigorous performance evaluation of MQA-KE for Arabic language. Experimentation evaluation reveals MQA-KEAL outperforms the baseline models by a significant margin. We release the codes for MQA-KEAL at https://github.com/asif6827/MQA-Keal.

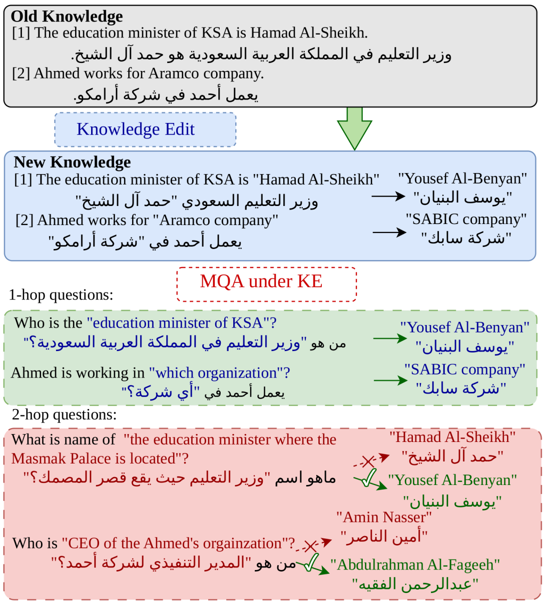
Old Knowledge
[1] The education minister of KSA is Hamad Al-Sheikh.
وزير التعليم في المملكة العربية السعودية هو حمد آل الشيخ.
[2] Ahmed works for Aramco company.
يعمل أحمد في شركة أرامكو.
Knowledge Edit
New Knowledge
[1] The education minister of KSA is "Hamad Al-Sheikh"
وزير التعليم السعودي "حمد آل الشيخ"
"Yousef Al-Benyan"
"يوسف البنيان"
[2] Ahmed works for "Aramco company"
يعمل أحمد في "شركة أرامكو"
"SABIC company"
"شركة سابك"
MQA under KE
1-hop questions:
Who is the "education minister of KSA"?
من هو "وزير التعليم في المملكة العربية السعودية؟"
"Yousef Al-Benyan"
"يوسف البنيان"
Ahmed is working in "which organization"?
يعمل أحمد في "أي شركة؟"
"SABIC company"
"شركة سابك"
2-hop questions:
What is name of "the education minister where the Masmak Palace is located"?
ماهو اسم "وزير التعليم حيث يقع قصر المصمك؟"
"Hamad Al-Sheikh"
"حمد آل الشيخ"
"Yousef Al-Benyan"
"يوسف البنيان"
Who is "CEO of the Ahmed's orgainzation"?
من هو "المدير التنفيذي لشركة أحمد؟"
"Amin Nasser"
"أمين الناصر"
"Abdulrahman Al-Fageeh"
"عبدالرحمن الفقيه"

Figure 1: An example illustration of multi-hop question answering under knowledge editing for Arabic and English language.

## 1 Introduction

Large Language Models (LLMs) have demonstrated immense potential across a wide range of natural language applications (Zhu et al., 2023; Huang et al., 2023a; Zhao et al., 2023; Hu et al., 2024; Yang et al., 2024d; Hong et al., 2024; Xu et al., 2023; Yang et al., 2024b,c; Su et al., 2023). A key challenge for these models is their limited adaptability to recent events and/or new data. For instance, training data for Llama-2 (Touvron et al., 2023a) only encompasses information about events till September 2022. This in turn restricts the true potential of these models to generate accurate responses about emerging events and questions beyond their training scope/timeline.

For this, numerous Knowledge Editing (KE) methods have been proposed that attempt to inject information about new facts, while avoiding massive costs associated with model re-training (Hu et al., 2021; Mitchell et al., 2021; Meng et al., 2022a,b; Zhang et al., 2024). However, these methods do not provide a comprehensive solution for the KE problem. For example, these models directly test the updated model for the edited knowl-

[*] J Qin and D Wang are co-corresponding authors.

*Proceedings of the 31st International Conference on Computational Linguistics*, pages 5629–5644
January 19–24, 2025. ©2025 Association for Computational Linguistics

edge without worrying about its impact on model's prior knowledge and/or facts explicitly correlated with the edits. An example illustration in this regard is shown in Figure 1, which emphasizes that if we edit the information about the *"Ahmed's workplace"*, corresponding knowledge/information about *"Ahmed's Boss/CEO"* also needs to be updated, a phenomenon widely known in literature as ***"ripple effect"***.

To overcome these limitations, recently there have been numerous research attempts in order to design and develop robust KE methods and corresponding evaluation benchmarks that allow testing KE at multiple hops centered around the edit, also known as Multi-hop Question Answering under Knowledge editing (MQA-KE). Existing research on MQA-KE is primarily classified into parameter-based (Hu et al., 2021; Shi and Lipani, 2023) and memory-based variants (Mitchell et al., 2022; Zhong et al., 2023; Gu et al., 2023; Cheng et al., 2024b,a), with memory-based methods outperforming the parameter-based methods. We observe, that majority of the existing solutions for MQA-KE and their evaluation benchmarks are peculiarly tailored for English language. While, recently LLMs have been extended to languages other than English, *e.g.,* AceGPT (Huang et al., 2023b) for Arabic language; Jias (Sengupta et al., 2023) a bilingual model supporting English and Arabic languages; and multi-lingual LLMs (Qin et al., 2024). There is a need to extend these methods to languages other than English.

In this work we extend existing work on MQA-KE to Arabic language. For this, we enumerate some of the key challenges as follows: *Firstly*, existing best-performing memory-based solutions are inadequate, because these methods save edits as unstructured text embeddings in a shared memory, making it non-trivial to retrieve the correct edit for a given question. This situation gets worse, especially when the number of fact edits grow beyond a certain limit. *Secondly*, there is a need for an effective mechanism that can effectively correlate the edit with its most relevant part and/or sub-part in the question in order to augment the end-performance of the model. *Thirdly*, there is a need for appropriate evaluation benchmarks for a rigorous evaluation of these systems for Arabic language.

Nevertheless, in this work we propose: **M**ulti-hop **Q**uestioning **A**nswering under **K**nowledge **E**diting for **A**rabic **L**anguage (MQA-KEAL), a novel approach, for MQA-KE in Arabic language. MQA-KEAL relies on following key components: (a) *"Structured Knowledge Retrieval"*, used to store the fact edits as a structured relational tuples in a shared memory. (b) *"Task-Decomposition"*, for decomposing the multi-hop questions into smaller sub-problems and/or knowledge units. (c) *"Iterative Traversal"* that traverses over the sub-problem to generate a list of responses as candidate answers, as well as filtering the candidate answers by leverage logic rules in order to come up with the intermediate and/or the final response.

For evaluation, we use: (i) MQUAKE-AR, an Arabic translation of an existing benchmark MQUAKE. (ii) MQA-AEVAL, a novel benchmark introduced in this work encompassing a wide range of single-hop and multi-hop questions primarily focused on Arabic Peninsula. Comprehensive experimental evaluation shows that MQA-KEAL outperforms the baseline models by a significant margin.

We outline the key contributions of this work as follows:

1. We propose MQA-KEAL, a novel approach for MQA-KE for Arabic language that initially decomposes multi-hop question into small sub-problems to generate candidate answers, later leverages logic rules to prune the candidates to come up with the final response.
2. We introduce two evaluation benchmarks, *i.e.,* MQUAKE-AR and MQA-AEVAL for MQA-KE for Arabic language.
3. We performed a comprehensive performance evaluation of MQA-KEAL, showcasing that the proposed model outperforms the baseline models by a significant margin.

## 2 Related Work

We classify the existing work on MQA-KE into: parameter based, and memory based methods.

The parameter based methods aim to fine-tune parameters of the large models in order to incorporate new knowledge and information. Usually, fine-tuning is a highly time-consuming process and is also highly vulnerable to catastrophic forgetting, *i.e.,* a phenomenon where model may forget and/or fail to retain its previous knowledge (Chen et al., 2020; Ding et al., 2024). In order to avoid higher computational costs parameter-efficient variants were introduced. These models use an auxiliary set of parameters for fine-tuning, *e.g.,* LoRA (Hu et al., 2021), Prompt Tuning (Shi and Lipani, 2023), QLoRA (Dettmers et al., 2024), MoRAL (Yang et al., 2024a).

The memory based methods on the other hand store the edit information in an explicit memory, later use retrieval methods to retrieve the edit that is most relevant to the question. Some examples include: SERAC by Mitchell et al. (2022), MeLLO by Zhong et al. (2023), PoKeMQA by Gu et al. (2023).

Generally, memory-based methods outperform the parameter-based methods. However, we observe, a key limitation of the memory-based methods is storing edits as embeddings learnt from unstructured text in a shared memory. This makes it challenging to disambiguate among different semantically relevant edits in the edit memory to retrieve the right fact edit. This situation exacerbates especially, when the number of edits in the edit memory grow beyond a certain limit. To overcome this MQA-KEAL stores edits as structured knowledge units, allowing relation-specific pruning *etc.*, helpful to perform the end-task in a performance-augmented way.

## 3 Preliminaries

In this section, we introduce the mathematical notation and formulate our problem definition.

### 3.1 Notation

For this work, we use knowledge graph triplets $(s, r, o)$ to represent the fact/knowledge, where $s$, $r$ and $o$ represent the subject, relation and object respectively. We use $e = (s, r, o \rightarrow o^*)$ to represent an individual fact edit. It emphasizes that object of relation $r$ with subject $s$ is updated from $o$ to $o^*$. We use $\mathcal{E} = \{e_1, e_2, \cdots, e_n\}$ to represent the collection of edits. We use $q \in \mathcal{Q}$ to represent multi-hop question. Answering $q$ requires multiple reasoning steps in order to compute the final response. We use $\mathcal{P} = \langle p_1, \cdots, p_n \rangle = \langle (s_1, r_1, o_1), \cdots, (s_n, r_n, o_n) \rangle$ to represent a chain of reasoning steps, with $o_n$ as the final response. For MQA-KE, if one of the fact undergoes an edit $e_i$, all subsequent facts in the knowledge chain needs to be updated. The resulting knowledge chain with updated path is represented as $\mathcal{P}^* = \langle (s_1, r_1, o_1), \cdots, (s_i, r_i, o_i \rightarrow o_i^*), \cdots, (s_n^*, r_n, o_n^*) \rangle$ with $o_n^*$ as the final answer. We use $f()$ to represent the target LLM. We use $\psi_i \in \Psi_I$ and $\psi_c \in \Psi_C$ to represent the implication and compositional rules respectively.

### 3.2 Problem Definition

The task of knowledge editing is to update and/or modify the knowledge in the LLMs without fine-tuning the entire model. Formally, given the LLM $f(\cdot)$, and a collection of fact edits $\mathcal{E} = \{e_1, e_2, \cdots, e_n\}$, we aim to augment the knowledge of $f(\cdot)$ using the facts edit information in $e_i \in \mathcal{E}$, such that it overrides the model's information about facts/knowledge correlated with $\mathcal{E}$, while keeping the other knowledge intact. Later, use updated model to generate and/or deduce the final response $o_n^*$ for the multiple-hop question $q$.

## 4 MQA-KEAL

**Overview.** In this paper, we propose **M**ulti-hop **Q**uestioning **A**nswering under **K**nowledge **E**diting for **A**rabic **L**anguage (MQA-KEAL), shown in Figure 2. MQA-KEAL first uses *"Structured knowledge Retrieval"* to store the fact edits in a shared memory as structured relational triplets. Later, it employs: *"Task Decomposition"*, to decompose the multi-hop question $q$ into sub-parts, and *"Iterative Traversal"* to traverse the sub-parts to generate the response for $q$. Note, to the best of our knowledge this work is amongst the initial attempts for knowledge editing and in turn testing it via multi-hop question answering for the Arabic languages.

### 4.1 Structured knowledge Retrieval

In order to successfully answer the multi-hop questions the retriever must be able to understand and comprehend multiple information units requisite to accurately answer the question. In order to overcome the limitations posed by existing work that store edits as embeddings learnt over unstructured text, we store and retrieve fact edits as embeddings learnt over structured relational triplets. Underlying reason in this regard is the fact that usually the key information within an individual edit $e_i$ may be primarily organized in form of relational triplets that allows relation-specific information filtering at later stages.

For example, the sentence: *"The president of Iran is Masoud Pezeshkian"* may be organized as: <Iran; president_is; Masoud Pezeshkian>. At the same time, we can sub-divide the multi-hop question into smaller information units and/or sub-problems (details in Section 4.2), and accordingly iteratively query the retriever with the sub-parts of the multi-hop question in order to generate the final response.

We also illustrate this phenomenon in Figure 3, where the upper half of the Figure shows that for

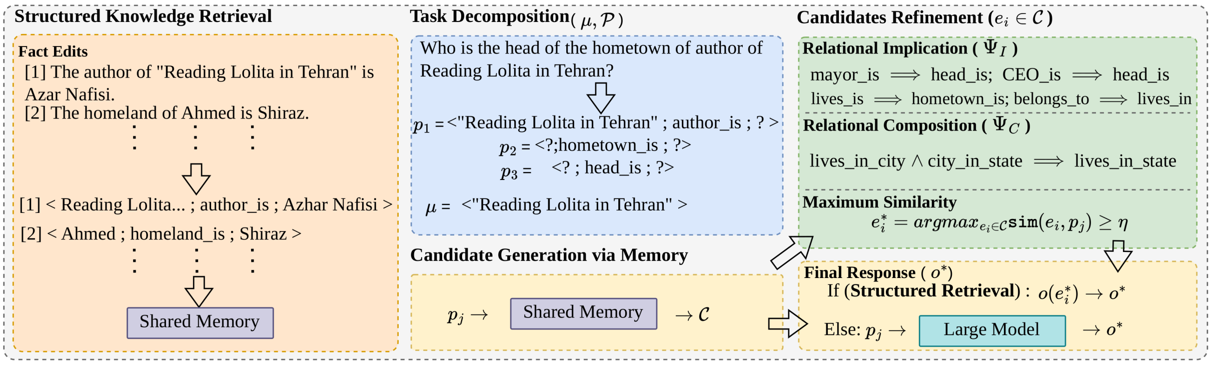
Structured Knowledge Retrieval
Fact Edits
[1] The author of "Reading Lolita in Tehran" is Azar Nafisi.
[2] The homeland of Ahmed is Shiraz.
[1] < Reading Lolita... ; author_is ; Azhar Nafisi >
[2] < Ahmed ; homeland_is ; Shiraz >
Shared Memory
Task Decomposition( $\mu, \mathcal{P}$ )
Who is the head of the hometown of author of Reading Lolita in Tehran?
$p_1$ = <"Reading Lolita in Tehran" ; author_is ; ? >
$p_2$ = <?;hometown_is ; ?>
$p_3$ = <? ; head_is ; ?>
$\mu$ = <"Reading Lolita in Tehran" >
Candidate Generation via Memory
$p_j \rightarrow$ Shared Memory $\rightarrow \mathcal{C}$
Candidates Refinement ($e_i \in \mathcal{C}$)
Relational Implication ( $\Psi_I$ )
mayor_is $\Longrightarrow$ head_is; CEO_is $\Longrightarrow$ head_is
lives_is $\Longrightarrow$ hometown_is; belongs_to $\Longrightarrow$ lives_in
Relational Composition ( $\Psi_C$ )
lives_in_city $\wedge$ city_in_state $\Longrightarrow$ lives_in_state
Maximum Similarity
$e_i^* = argmax_{e_i \in \mathcal{C}} \texttt{sim}(e_i, p_j) \geq \eta$
Final Response ( $o^*$)
If (Structured Retrieval) : $o(e_i^*) \rightarrow o^*$
Else: $p_j \rightarrow$ Large Model $\rightarrow o^*$

Figure 2: **Workflow of MQA-KEAL**. The left part of the Figure shows how we store fact edits. The central part illustrates task decomposition and candidate generation from the memory. The right part of the Figure shows candidate refinement and final response generation.

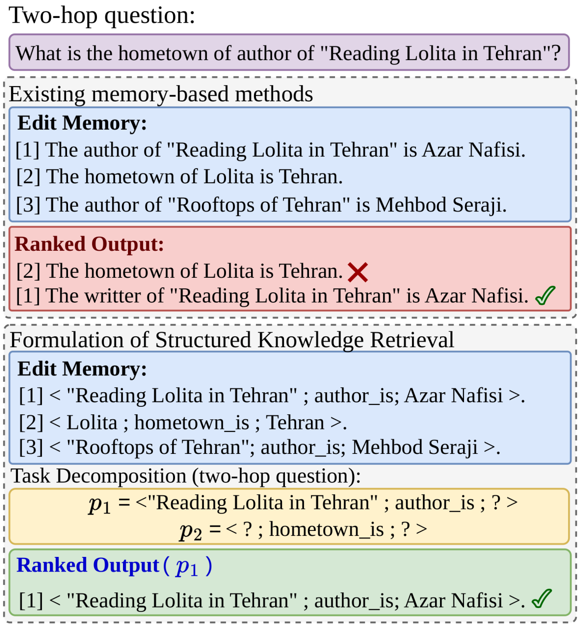
Two-hop question:
What is the hometown of author of "Reading Lolita in Tehran"?
Existing memory-based methods
Edit Memory:
[1] The author of "Reading Lolita in Tehran" is Azar Nafisi.
[2] The hometown of Lolita is Tehran.
[3] The author of "Rooftops of Tehran" is Mehbod Seraji.
Ranked Output:
[2] The hometown of Lolita is Tehran. ✗
[1] The writter of "Reading Lolita in Tehran" is Azar Nafisi. ✓
Formulation of Structured Knowledge Retrieval
Edit Memory:
[1] < "Reading Lolita in Tehran" ; author_is; Azar Nafisi >.
[2] < Lolita ; hometown_is ; Tehran >.
[3] < "Rooftops of Tehran"; author_is; Mehbod Seraji >.
Task Decomposition (two-hop question):
$p_1$ = <"Reading Lolita in Tehran" ; author_is ; ? >
$p_2$ = < ? ; hometown_is ; ? >
Ranked Output ( $p_1$ )
[1] < "Reading Lolita in Tehran" ; author_is; Azar Nafisi >. ✓

Figure 3: An example illustrating the limitation of existing memory-based methods that store knowledge edits as unstructured text, vs structured knowledge retrieval employed by MQA-KEAL.

the two-hop question: {*"What is the hometown of author of Reading Lolita in Tehran"?*}, the fact edit that is most semantically similar to the question comes out to be: {*"The hometown of Lolita is Tehran"*}. However, this retrieved fact does not guarantee whether we can use this information to successfully answer the question. On the other hand, lower-half of the Figure shows our formulation of structured knowledge retrieval, that emphasizes if we store the edits as relational triplets, and accordingly query the model by decomposing the multi-hop question into sub-parts, we can accurately yield the edits that are relevant to each sub-part of $q$ iteratively.

Formally, for edits $e_i \in \mathcal{E}$, we decompose the edits as relational triplets $e_i = < s, r, o >$ and use a retrieval model, *i.e.,* `Contriever` (Izacard et al., 2021), to embed and save these edits as a retrieval index. Later, during the model inference the index takes the query as input and generates top-$k$ facts most relevant to the input query.

**Example.** An example illustration of the structured memory retrieval module of MQA-KEAL is shown in the left half of Figure 2, where the unstructured text {*"The homeland of Ahmed is Shiraz"*}, is embed in the memory as {< *Ahmed; homeland_is; Shiraz* >}.

### 4.2 Task Decomposition

Task decomposition module of MQA-KEAL aims to decompose the multi-hop question $q$ in to smaller sub-components in order to come up with a reasoning path and/or chain ($\mathcal{P}$) that can be traversed iteratively in order to generate the final response.

For this, we leverage the instruction following abilities of the LLMs to decompose the multi-hop question $q$ using an in-context learning prompt. Formally, we use the target model $f(\cdot)$ to decompose the $q$ as follows:

$$\mu, \mathcal{P} =< p_1, p_2, \cdots p_n >= f(\mathrm{T}_{\text{relation}}, q) \qquad (1)$$

here, $\mathrm{T}_{\text{relation}}$ represents the in-context learning prompt used to decompose $q$, as outlined in Appendix E. The output of the model is a chain of reasoning path indicative of the key components in $q$, *i.e.,* $\mathcal{P}$, and the starting point of the path traversal, *i.e.,* $\mu$. Note, the start point $\mu$ is an entity and individual facts in $\mathcal{P}$ are organized in the form of relational triplets.

This formulation makes it convenient to iteratively traverse the $\mathcal{P}$ by either retrieving corresponding facts in the edit memory or querying the target LLM to come up with the final response.

**Example.** Continuing our previous example, the upper centre half of the Figure 2 shows how MQA-KEAL decomposes a 3-hop question: {*“Who is the head of the hometown of author of Reading Lolita in Tehran”?*} into $\mu$ = *“Reading Lolita n Tehran”* and $\mathcal{P} = \{p_1, p_2, p_3\}$ with $p_1$ = <*“Reading Lolita in Tehran”*; *author_is*; ?>, $p_2$ = <? ; *hometown_is*; ?>, and $p_3$ = <? ; *head_is*; ?>.

### 4.3 Iterative Traversal over $\mathcal{P}$

After decomposing the multi-hop question $q$ into multiple sub-problems, the iterative traversal part of MQA-KEAL iterates through the $\mathcal{P}$ one step at a time with $\mu$ as the starting point. During each step, it attempts to solve a smaller sub-problems with results to be used as starting point for the next round. For this, MQA-KEAL repeatedly iterates through multiple rounds of response generation using shared memory and target LLM.

Formally, for each sub-problem ($p_j \in \mathcal{P}$), the candidate generation modules looks for probable candidates $\{o_i^*\}$ for the answer. Given the fact, the end-goal of knowledge editing is to update the priorly contained knowledge of the LLMs. Thus, for response generation, we prioritize the responses retrieved from the from the edit memory, in case the information about a certain entity and/or facts has been updated (Section 4.3.1). For cases, the model is not able to generate substantial response from the edit memory, we resort back to querying the target model to generate the final response (Section 4.3.2).

#### 4.3.1 Response from Structured Retrieval

In contrast to the existing memory based methods (Zhong et al., 2023) that only consider topmost index semantically related to the input query, we retrieve top-$k$ edits as candidate answers, later refine these candidates via different pruning heuristics in order to generate the response $o_i^*$. We argue, this formulation of retrieving response by selecting multiple candidates as probable answers helps in overcoming the limitations posed by the memory-based systems and eventually helps in significantly augmenting the end-performance of MQA-KEAL.

Formally, for a given sub-problem ($p_j \in \mathcal{P}$), we look for the top-$k$ fact edits that are most relevant to the sub-problem, as follows:

$$\mathcal{C} = [e_1, \cdots, e_k] = k\text{-}\underset{e_i \subset \mathcal{E}:|e_i|=k}{\arg\max}\ \texttt{sim}(e_i, p_j) \quad (2)$$

where $k$-$\arg\max$ returns the indices of the top scored $k$ fact edits. `sim()` is used to compute the embedding similarity of the embedding vectors[1]. Finally, we consider $\mathcal{C}$ as the final set of candidates passed through the candidate filtering process, as detailed below.

**Algorithm 1** CANDIDATE FILTERING

```
Input: $\eta$ : thr; $\mathcal{C}$ : candidates; $p_j$; $\{p_1 \cdots p_n\} \in \mathcal{P}$
Output: $o_i^*$
 1: $o_i^* \leftarrow$ null; found $\leftarrow$ False
 2: for $e_i \in \mathcal{C}$ do
 3:     #1. Relational Implication ($\Psi_I$)
 4:     ## subject in alias check
 5:     if $s(p_j) \in [\texttt{alias}(s(e_i))]$ then
 6:         ## relational implication
 7:         if $\psi_i : r(p_j) \rightarrow r(e_i)$ then
 8:             $o_i^* \leftarrow o(e_i)$ & found $\leftarrow$ True
 9:         end if
10:     end if
11:     #2. Relational Composition ($\Psi_C$)
12:     if found == False then
13:         if $\psi_c : r(p_1) \wedge \cdots \wedge r(p_j) \rightarrow r(e_i)$ then
14:             $o_i^* \leftarrow o(e_i)$ & found $\leftarrow$ True
15:         end if
16:     end if
17: end for
18: #3. Maximum Similarity
19: if found == False then
20:     $e_i^* = \arg\max_{e_i \in \mathcal{C}_1} \texttt{sim}(e_i, p_j) \geq \eta$
21:     $o_i^* \leftarrow o(e_i^*)$
22: end if
23: return $o_i^*$
```

**Candidate Filtering.** The process-flow of candidate filtering is illustrated in Algorithm 1. It takes similarity threshold $\eta$, list of candidate answers $\mathcal{C}$, the sub-problem $p_j \in \mathcal{P}$ as input, and generates the final response $o_i^*$ as output for the sub-problem $p_j$.

For this, it initializes the variable $o_i^*$ to `null`, and uses a variable {`found`} initialized to `False`, used to keep track of the response generation. Later, for each candidate (*i.e.,* $e_i \in \mathcal{C}$), the Algorithm 1 iterates through three different stages, enumerated as follows:

*1. Relational Implication.* This is outlined in lines (3-10) in Algorithm 1. It considers the logical implication of relation pairs in the sub-problem ($p_j$) and candidate edit ($e_i$) in order to capture semantically related relations. We define relational implication as:

[1] Note, we use dot product as a metric indicative of similarity among embedding vectors.

*Definition: For two relations, $r_1$ implies $r_2$* (or $r_1 \implies r_2$) iff $\forall(s,o) \in r_1 \implies (s,o) \in r_2$ or $r_1 \subset r_2$.

Some examples in this regard include: `father_of` $\implies$ `parent_of`; `lives_in` $\implies$ `belong_to`, *etc.* For instance, if John is father of Tom, then John is also parent of Tom.

Formally, given two relations $r_1$ and $r_2$, we aim to compute whether $r_1 \implies r_2$. For this, we first perform subject entity disambiguation by analyzing if the subject of the $p_j$, *i.e.,* $s(p_j)$ matches with the alias names of subject in $e_i$, *i.e.,* [`alias`$(s(e_i))$], as outlined in lines (4-5). Later, in lines (6-7) we look for implication of relation pairs, *i.e.,* $r(p_j) \implies r(e_i)$ to assign the $o(e_i)$ as the answer $o_i^*$, as shown in line-8. Details about the implication rule extraction are explained in Appendix B.1.

*2. Relational Composition.* This is outlined in lines (11-17) in Algorithm 1. It aims to compute the relational composition along the knowledge path $p_j \in \mathcal{P}$. For this, we use horn rule to compute the relational composition, defined as follows.

*Definition: Horn rule is a special class of first-order logic rules that is composed of conjunctive predicates:* $\mathbf{r_b} = \{r_{b_1}, \cdots r_{b_n}\}$ *known as rule body or pre-condition, and a predicate $r_h$ as the rule head or consequence, represented as follows:*

$$r_{b_1}(s, z_1) \wedge \cdots \wedge r_{b_n}(z_{n-1}, o) \implies r_h(s, o)$$

Some exemplar horn rules regard include: `lives_in_city` $\wedge$ `city_in_continent` $\implies$ `lives_in_continent`. This rules aims to look for the candidate fact edits $e_i \in \mathcal{C}$ that are strongly correlated with the fact chain $\{p_1, p_2 \cdots p_n\}$.

As mentioned in lines (13-14), if the relational predicates satisfy the precondition along with the relational part of the edit satisfying the consequence of the horn rule $\psi_c$, we use corresponding $o(e_i)$ as the answer $o_i^*$. Further details about compositional rule extraction are explained in Appendix B.2.

*3. Maximum Similarity.* This step is outlined in lines (18-22) in Algorithm 1. It aims to capture fact edit exhibiting higher embedding similarity with the sub-problem. For this, we select the fact edit $(e_i)$ exhibiting similarity with $p_j$, compared against a threshold $\eta$, *i.e.,* `sim()` $\geq \eta$ to assign the corresponding $o(e_i)$ as the response $o_i^*$.

### 4.3.2 Response via target LLM

For the cases, MQA-KEAL is not able to generate substantial response from the structured knowledge retrieval, we resort to the information contained in the target LLM in order to generate the response. For this, we leverage the in-context learning abilities of the target LLM $f(\cdot)$, to generate a response for $p_j$, as follows:

$$o_i^* = f(\mathrm{T}_{\text{query}}, p_j) \quad (3)$$

where $p_j$ corresponds to the $j$-th sub-problem of the multi-hop question $q$, and $\mathrm{T}_{\text{query}}$ is the prompt used to query the target model (outlined in Appendix E).

The final response generated in this stage is used as an initial step for the subsequent steps to be traversed over the path $\mathcal{P}$, finally resulting in $o_n^*$ as the end response for the multi-hop question $q$.

## 5 Experimentation

### 5.1 Experimental Settings

**Datasets.** In order to evaluate the performance of MQA-KEAL, we translated existing widely used data sets for MQA under KE, *i.e.,* MQUAKE (Zhong et al., 2023), renamed as MQUAKE-AR (encompassing MQUAKE-T-AR and MQUAKE-CF-AR). Note, these data sets were translated using automated tools and manually validated by local Arabic language experts. Apart from this, we also curated a new data set, namely : **M**ulti hop **Q**uestion **A**nswering under knowledge editing for **A**rabic-region **EVAL**uation (MQA-AEVAL) that portray a more realistic setting for MQA under KE typically tailored for local Arab world. Further details about the datasets, translation process and statistics of the data sets are provided in Appendix C.1.

**Baseline Models.** We use existing best performing solutions for KE and MQA-KE as baselines. These include: (i) Fine-Tuning (FT) (Zhu et al., 2020), (ii) ROME (Meng et al., 2022a), (iii) MEMIT (Meng et al., 2022b), and (iv) MeLLo (Zhong et al., 2023). Further details about the baseline models are provided in Appendix C.2.

**Evaluation Metrics.** For performance evaluation, we use multi-hop accuracy (M-Acc) (Zhong et al., 2023), and hop-wise accuracy (H-Acc) (Gu et al., 2023). Further details about the evaluation metrics and their mathematical formulation are provided in Appendix C.4.

**Experimental Setup.** For experimentation, we use GPT-3.5-TURBO-INSTRUCT[2], as well as existing Arabic-centric LLMs, *i.e.,* AceGPT-13B (Huang et al., 2023b) and Jias-13B (Sengupta et al., 2023), as target LLMs. Further

[2] `https://platform.openai.com/`

| Method | MQUAKE-CF-AR | | | | | | MQUAKE-T-AR | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1-edited | | 100-edited | | 3000-edited | | 1-edited | | 100-edited | | 1868-editted | |
| | M-Acc | H-Acc | M-Acc | H-Acc | M-Acc | H-Acc | M-Acc | H-Acc | M-Acc | H-Acc | M-Acc | H-Acc |
| | | | | | | JIAS-13B | | | | | | |
| FT | 11.30 | 2.10 | 1.40 | 0.05 | 0.01 | - | 28.47 | 19.11 | 23.54 | 11.43 | 0.54 | 0.11 |
| ROME | 5.79 | 1.70 | 2.90 | 0.07 | 2.45 | 0.57 | 14.57 | 8.95 | 17.15 | 8.75 | 1.45 | 0.78 |
| MEMIT | 6.14 | 3.40 | 5.75 | 2.60 | 1.97 | 0.75 | 17.44 | 7.85 | 13.13 | 6.95 | 11.87 | 5.18 |
| MeLLo | 15.35 | 7.58 | 14.50 | 6.75 | 12.55 | 4.58 | 35.22 | 24.38 | 33.19 | 24.15 | 27.27 | 18.96 |
| MQA-KEAL | **24.69** | **13.15** | **22.05** | **14.47** | **18.17** | **14.32** | **47.21** | **35.90** | **45.27** | **36.30** | **42.89** | **34.15** |
| | | | | | | ACEGPT-13B | | | | | | |
| MeLLo | 17.83 | 9.78 | 15.32 | 4.97 | 13.25 | 4.54 | 67.35 | 45.44 | 57.12 | 41.21 | 39.17 | 33.21 |
| MQA-KEAL | **28.13** | **18.11** | **22.47** | **12.51** | **19.75** | **8.95** | **74.50** | **67.15** | **69.10** | **61.58** | **63.84** | **57.85** |
| | | | | | | GPT-3.5-TURBO-INSTRUCT | | | | | | |
| MeLLo | 21.20 | 7.5 | 18.70 | 9.52 | 15.07 | 5.69 | 72.37 | 61.19 | 67.47 | 58.43 | 45.41 | 34.87 |
| MQA-KEAL | **30.14** | **24.59** | **25.27** | **21.95** | **22.50** | **18.95** | **79.52** | **73.46** | **73.52** | **66.23** | **65.45** | **57.33** |

Table 1: **Experimental results for MQUAKE-AR.** The result of MQA-KEAL (*i.e.,* M-Acc and H-Acc) for different datasets and target LLMs compared against the baseline methods. For each target LLM, we boldface overall best scores with the second best underlined.

| Method | MQA-AEVAL | | | | | |
|---|---|---|---|---|---|---|
| | 1-edited | | 100-edited | | 229-edited | |
| | M-Acc | H-Acc | M-Acc | H-Acc | M-Acc | H-Acc |
| | | | JIAS-13B | | | |
| FT | 5.74 | 1.15 | 2.24 | 0.95 | 1.95 | 0.05 |
| ROME | 2.34 | 0.07 | 3.45 | 0.98 | 3.95 | 0.85 |
| MEMIT | 3.45 | 0.45 | 4.85 | 0.55 | 4.54 | 1.10 |
| MeLLo | 24.42 | 13.45 | 23.25 | 17.58 | 22.65 | 17.07 |
| MQA-KEAL | **37.64** | **29.45** | **35.32** | **28.53** | **34.38** | **24.45** |
| | | | ACEGPT-13B | | | |
| MeLLo | 27.50 | 20.21 | 25.13 | 17.59 | 24.17 | 18.01 |
| MQA-KEAL | **41.85** | **31.59** | **39.51** | **27.31** | **38.32** | **30.45** |
| | | | GPT-3.5-TURBO-INSTRUCT | | | |
| MeLLo | 35.57 | 23.41 | 33.51 | 28.87 | 32.64 | 26.45 |
| MQA-KEAL | **44.17** | **39.45** | **42.91** | **37.41** | **41.65** | **31.81** |

Table 2: **Experimental results for MQA-AEVAL.** We report the scores of MQA-KEAL (*i.e.,* M-Acc and H-Acc), compared against baseline models. We boldface overall best scores with second best underlined.

details about these LLMs are provided in Appendix C.3. Value of $\eta$ is set to 0.6 and 0.7 for MQUAKE-AR and MQA-AEVAL respectively. Similar to MeLLO (Zhong et al., 2023), for evaluation of MQA-KEAL, we used a batch of $k$ instances, *i.e.,* $k \in \{1, 100, 3000\}$ for MQUAKE-CF-AR, $k \in \{1, 100, 1868\}$ for MQUAKE-T-AR, and $k \in \{1, 100, 229\}$ for MQA-AEVAL. In Section 4.3.1, value of $k$=10 for top-$k$ candidate fact edits. All experiments were repeated for five rounds, and average scores are reported.

### 5.2 Main Results

The results of MQA-KEAL for MQUAKE-AR and different target LLMs are shown in Table 1. Here, we report M-Acc and H-Acc scores of MQA-KEAL compared against the baseline models. Analysing the results for the baseline models (*i.e.,* FT, ROME, MEMIT), we observe that widely used knowledge editing methods perform poorly on MQA-KE, which showcases the true knowledge augmentation potential of these models for LLMs.

Overall results in Table 1 show that MQA-KEAL consistently outperforms the baseline models by a significant margin across both metrics. For instance, for MQUAKE-CF-AR using GPT-3.5-TURBO-INSTRUCT as the target LLM MQA-KEAL improves the M-Acc scores by {42.1%, 35.1% and 49.3%} respectively for {1, 100 and 3000} edited cases. Likewise, for MQUAKE-T-AR, and GPT-3.5-TURBO-INSTRUCT as target LLM, the improvement in performance is {9.8%, 8.9% and 44.1%} respectively for {1, 100 and 1868}-edited cases. The results of MQA-KEAL with Jias-13B and AceGPT-13B as target LLM exhibit similar behaviors with our proposed model outperforming the best performing baseline models.

Analyzing the results of MQA-KEAL for newly proposed data (*i.e.,* MQA-AEVAL) in Table 2 shows a similar behaviour with MQA-KEAL yielding better performance than the baseline models. However, we observe the model performance on MQA-AEVAL is relatively lower compared to that of MQUAKE-T-AR. We enumerate some of the probable justifications in this regard as follows: (i) the instances in MQA-AEVAL are localized for local Arabic region which may not be adequately covered in LLMs' training corpora, as most of the existing LLMs are trained using knowledge directly acquired and/or translated from western regions (Naous et al., 2023); (ii) majority of the information in MQA-AEVAL (and corresponding reasoning path $\mathcal{P}^*$) is about recent events, beyond the cut-off of LLMs training data; and (iii) it is not easy to get aliases for entities for Arabic language, which limits the entity matching abilities of MQA-KEAL (line-5 in Algorithm 1).

We observe, amongst all data sets, MQA-KEAL

| Method | MQuAKE-T-Ar | | | | | |
|---|---|---|---|---|---|---|
| | 1-edited | | 100-edited | | 1868-edited | |
| | M-Acc | H-Acc | M-Acc | H-Acc | M-Acc | H-Acc |
| GPT-3.5-turbo-instruct | | | | | | |
| MQA-Keal (–I) | 71.31 | 67.45 | 65.38 | 61.57 | 59.55 | 52.10 |
| MQA-Keal (–C) | 77.44 | 71.56 | 72.87 | 64.33 | 62.58 | 55.19 |
| MQA-Keal (–IC) | 68.45 | 64.31 | 61.45 | 57.24 | 55.01 | 50.48 |
| MQA-Keal | 79.52 | 73.46 | 73.52 | 66.23 | 65.45 | 57.33 |

Table 3: **Ablation analysis for MQuAKE-T-Ar** under varying conditions and GPT-3.5-turbo-instruct as target LLM.

yields best scores for MQuAKE-T-Ar followed by MQA-Aeval and MQuAKE-CF-ar. This behavior is also consistent with the baseline models. Analysing the results for different target LLMs, we observe GPT-3.5-turbo-instruct yields best performance overall followed by AceGPT-13B and Jias-13B. A possible reason in this regard is the fact that AceGPT-13B being an instruction-tuned variant of Llama-2 (Touvron et al., 2023b) inherits its better task-decomposition abilities compared to that of Jias-13B. We also observe that with higher number of fact edits, the performance of the model decreases. This is attributable to multiple different factors as analyzed in Section 6.2. However, this effect is more pronounced for the baseline models compared to that of MQA-Keal.

# 6 Analyses

In this section, we perform a detailed analyses of MQA-Keal under different settings. This includes: (i) Ablation Analyses (ii) Error Analyses. Note, some additional experimental analyses are also reported in Appendix C.

## 6.1 Ablation Analyses

Ablation analysis aims to analyze the performance attributable to individual model components. For this, we report the performance of MQA-Keal for (i) –I (*w/o* implication rules), (ii) –C (*w/o* compositional rules), (iii) –IC (*w/o* both implication and compositional rules).

Corresponding results of MQA-Keal with GPT-3.5-turbo-instruct as target LLM and MQuAKE-T-ar data set are shown in Table 3. These results show that omission of implication rules have a more pronounced impact on the model performance compared to that of the compositional rules. For instance, compared with the complete model, the variant MQA-Keal(–I) exhibits {10.3, 11.1, 9.0}% reduction in performance; whereas, MQA-Keal(–C) results in {2.6, 0.8, 4.3}% decline in performance for {1, 100, and 1868}-edited cases respectively. This is understandable owing to the fact that overall implication rules have a broader coverage and are more likely to be satisfied compared to that of the compositional rules. Also, jointly omitting the implication and compositional rules exhibits a compounding effect, as evident in Table 3 with MQA-Keal(–IC) showing an accumulated decrease of {13.9, 16.4, 15.9}% in M-Acc scores for {1, 100, and 1868}-edited cases.

Also, comparing the results of MQA-Keal(–IC) with last row in Table 1, we observe that even if we omit the candidate filtering, the end-results of MQA-Keal are still better than MeLLo, *i.e.,* M-Acc = 55.01 for MQA-Keal(–IC) compared with M-Acc = 45.41 for MeLLo under 1868-edited cases. To summarize, these results show that the *"structured knowledge reterieval"* employed by MQA-Keal followed by candidate filtering offer a robust setting helpful in performing the end MQA-Ke task in a performance enhanced way.

## 6.2 Error Cases.

We analyzed a random sample of 50 error cases of MQA-Keal in order to understand and comprehend the limitations of the model. We categorize these error cases into following different categories: (a) errors by target LLMs, (b) errors by structured retrieval, (c) errors in task decomposition by LLM, and (d) miscellaneous errors.

For the variant of MQA-Keal with MQA-Aeval and AceGPT-13B as the target model, we observe that almost 11% of the errors were caused by the erroneous response generated by the target LLM (most probably because model is ignorant about some recent events beyond its training scope), 25% of the errors are caused by the structured fact retrieval, and 17% errors were caused by inappropriate task-decomposition by LLM. Rest of the errors were categorized to miscellaneous errors. This analysis shows that although our formulation of task decomposition along with structured knowledge retrieval employed by MQA-Keal were able to significantly improve the performance compared to the unstructured text embeddings, yet it did not completely eradicate the issue.

# 7 Conclusions and Future Work

In this work, we proposed MQA-Keal, a novel approach for Knowledge Editing and in turn test the edited knowledge via multi-hop question answering for Arabic language. Apart from that, we also contributed MQuAKE-ar an Arabic translated and humanly-validated version of existing MQuAKE (Zhong et al., 2023) data set; and MQA-AEval as new MQA-KE data set targeting information primarily on Arabian Peninsula. In the future, we aim to extend this work to multi-lingual LLMs, by unifying diverse concepts (Ali et al., 2020, 2021) and/or controlling the relative isomorphism of the vector spaces (Ali et al., 2023b,a).

## Limitations

Some of the core limitations of the proposed approach are outlined as follows:

- MQA-KEAL uses an iterative approach for generating response for multi-hop questions. Errors in the intermediate path may propagate and impact the final answer. Currently, the is no effective mechanism for recovery from errors in the intermediate path.
- Our work assumes, for the cases where there is no fact edit directly and/or indirectly correlated with a particular entity, MQA-KEAL entirely relies on the target output. For these cases, the end-result will be incorrect if the target model yields incorrect results.
- For performance comparison, MQA-KEAL uses corresponding target LLM to decompose the multi-hop question into smaller subproblems. For cases with target LLM exhibiting relatively inferior knowledge decomposition abilities, the end-result of MQA-KEAL is severely impacted.

## Ethics Statement

This work directly deals with updating the capability and/or editing the knowledge of large models. It has the potential for abuse, such as adding poisonous misinformation, malicious content, bias etc. Keeping in view these concerns, we highlight this work must not be used under critical settings.

## Acknowledgement

Di Wang is supported in part by the funding BAS/1/1689-01-01, URF/1/4663-01-01, REI/1/5232-01-01, REI/1/5332-01-01, and URF/1/5508-01-01 from KAUST, and funding from KAUST - Center of Excellence for Generative AI, under award number 5940.

## References

Muhammad Asif Ali, Maha Alshmrani, Jianbin Qin, Yan Hu, and Di Wang. 2023a. Gari: Graph attention for relative isomorphism of arabic word embeddings. *arXiv preprint arXiv:2310.13068*.

Muhammad Asif Ali, Yan Hu, Jianbin Qin, and Di Wang. 2023b. Gri: Graph-based relative isomorphism of word embedding spaces. *arXiv preprint arXiv:2310.12360*.

Muhammad Asif Ali, Yifang Sun, Bing Li, and Wei Wang. 2020. Fine-grained named entity typing over distantly supervised data based on refined representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7391–7398.

Muhammad Asif Ali, Yifang Sun, Bing Li, and Wei Wang. 2021. Fine-grained named entity typing over distantly supervised data via refinement in hyperbolic space. *CoRR*.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. 2020. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. In *Conference on Empirical Methods in Natural Language Processing*.

Keyuan Cheng, Muhammad Asif Ali, Shu Yang, Gang Lin, Yuxuan Zhai, Haoyang Fei, Ke Xu, Lu Yu, Lijie Hu, and Di Wang. 2024a. Leveraging logical rules in knowledge editing: A cherry on the top. *arXiv preprint arXiv:2405.15452*.

Keyuan Cheng, Gang Lin, Haoyang Fei, Lu Yu, Muhammad Asif Ali, Lijie Hu, Di Wang, et al. 2024b. Multi-hop question answering under temporal knowledge editing. *arXiv preprint arXiv:2404.00492*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Meng Ding, Kaiyi Ji, Di Wang, and Jinhui Xu. 2024. Understanding forgetting in continual learning with linear regression. *arXiv preprint arXiv:2405.17583*.

Bahare Fatemi, Siamak Ravanbakhsh, and David Poole. 2019. Improved knowledge graph embedding using background taxonomic information. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3526–3533.

Hengrui Gu, Kaixiong Zhou, Xiaotian Han, Ninghao Liu, Ruobing Wang, and Xin Wang. 2023. Pokemqa: Programmable knowledge editing for multi-hop question answering. *arXiv preprint arXiv:2312.15194*.

Yihuai Hong, Yuelin Zou, Lijie Hu, Ziqian Zeng, Di Wang, and Haiqin Yang. 2024. Dissecting finetuning unlearning in large language models. *arXiv preprint arXiv:2410.06606*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Lijie Hu, Liang Liu, Shu Yang, Xin Chen, Hongru Xiao, Mengdi Li, Pan Zhou, Muhammad Asif Ali, and Di Wang. 2024. A hopfieldian view-based interpretation for chain-of-thought reasoning. *arXiv preprint arXiv:2406.12255*.

Cynthia Huang, Yuqing Xie, Zhiying Jiang, Jimmy Lin, and Ming Li. 2023a. Approximating human-like few-shot learning with gpt-based compression. *Preprint*, arXiv:2308.06942.

Huang Huang, Fei Yu, Jianqing Zhu, Xuening Sun, Hao Cheng, Dingjie Song, Zhihong Chen, Abdulmohsen Alharthi, Bang An, Ziche Liu, et al. 2023b. Acegpt, localizing large language models in arabic. *arXiv preprint arXiv:2309.12053*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2021. Fast model editing at scale. *ArXiv*, abs/2110.11309.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. *ArXiv*, abs/2206.06520.

Tarek Naous, Michael J Ryan, Alan Ritter, and Wei Xu. 2023. Having beer after prayer? measuring cultural bias in large language models. *arXiv preprint arXiv:2305.14456*.

Sara Pieri, Sahal Shaji Mullappilly, Fahad Shahbaz Khan, Rao Muhammad Anwer, Salman Khan, Timothy Baldwin, and Hisham Cholakkal. 2024. Bimedix: Bilingual medical mixture of experts llm. *arXiv preprint arXiv:2402.13253*.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2024. Multilingual large language model: A survey of resources, taxonomy and frontiers. *arXiv preprint arXiv:2404.04925*.

Meng Qu, Junkun Chen, Louis-Pascal Xhonneux, Yoshua Bengio, and Jian Tang. 2020. Rnnlogic: Learning logic rules for reasoning on knowledge graphs. *arXiv preprint arXiv:2010.04029*.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, et al. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *arXiv preprint arXiv:2308.16149*.

Zhengxiang Shi and Aldo Lipani. 2023. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning. *arXiv preprint arXiv:2309.05173*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023a. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. In *Twenty-fourth international joint conference on artificial intelligence*.

Xilie Xu, Keyi Kong, Ning Liu, Lizhen Cui, Di Wang, Jingfeng Zhang, and Mohan Kankanhalli. 2023. An llm can fool itself: A prompt-based adversarial attack. *arXiv preprint arXiv:2310.13345*.

Shu Yang, Muhammad Asif Ali, Cheng-Long Wang, Lijie Hu, and Di Wang. 2024a. Moral: Moe augmented lora for llms' lifelong learning. *arXiv preprint arXiv:2402.11260*.

Shu Yang, Lijie Hu, Lu Yu, Muhammad Asif Ali, and Di Wang. 2024b. Human-ai interactions in the communication era: Autophagy makes large models achieving local optima. *arXiv preprint arXiv:2402.11271*.

Shu Yang, Jiayuan Su, Han Jiang, Mengdi Li, Keyuan Cheng, Muhammad Asif Ali, Lijie Hu, and Di Wang. 2024c. Dialectical alignment: Resolving the tension of 3h and security threats of llms. *arXiv preprint arXiv:2404.00486*.

Shu Yang, Shenzhe Zhu, Ruoxuan Bao, Liang Liu, Yu Cheng, Lijie Hu, Mengdi Li, and Di Wang. 2024d. What makes your model a low-empathy or warmth person: Exploring the origins of personality in llms. *arXiv preprint arXiv:2410.10863*.

Zhuoran Zhang, Yongxiang Li, Zijian Kan, Keyuan Cheng, Lijie Hu, and Di Wang. 2024. Locate-then-edit for multi-hop factual recall under knowledge editing. *arXiv preprint arXiv:2410.06331*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *Preprint*, arXiv:2303.18223.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *arXiv preprint arXiv:2305.14795*.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix X. Yu, and Sanjiv Kumar. 2020. Modifying memories in transformer models. *ArXiv*, abs/2012.00363.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *Preprint*, arXiv:2304.04675.

# A Background

## A.1 Knowledge Representation

We use graph triplets $(s, r, o)$ to represent the fact/knowledge, where $s$, $r$ and $o$ represent the subject, relation and object respectively. This is commonly used to represent the facts in the Knowledge graphs.

## A.2 Knowledge Editing (KE)

We use $e = (s, r, o \rightarrow o^*)$ to represent an individual knowledge update. It emphasize that the object of subject $s$ under relation $r$ is updated from $o$ to $o^*$. A collection of knowledge edits is represented by $\mathcal{E} = \{e_1, e_2, \cdots, e_n\}$.

## A.3 Multi-hop Question Answering under KE

A multi-hop question $q$ requires multiple reasoning steps in order to come up with the final answer/response. Generally, these reasoning steps formulate a chain of facts $\mathcal{C} = \langle (s_1, r_1, o_1), \cdots, r_n(s_n, r_n, o_n) \rangle$. The consecutive facts in $\mathcal{C}$ are chained together, i.e., $o_i$ from the proceeding step is $s_{i+1}$ for the subsequent fact, with $o_n$ as the final outcome. For multi-hop question answering under KE, if one of the fact $(s_i, r_i, o_i)$ in $\mathcal{C}$ undergoes an edit $(e_i \in \mathcal{E})$, the resulting chain for the subsequent facts need to be updated. The updated chain becomes: $\mathcal{C} = \langle (s_1, r_1, o_1), \cdots, (s_i, r_i, o_i \rightarrow o_i^*), \cdots, (s_n^*, r_n, o_n \rightarrow o_n^*) \rangle$, with $o_n^*$ as the final outcome. The end-goal of multi-hop question answering is to come up with the answer for $q$ based on edits in $\mathcal{E}$.

Multi-hop question answering under knowledge editing is a key challenge for LLMs. Some illustrative examples are shown below:

# B Rule Extraction

This work uses implication and compositional rules for filtering candidate response from structured memory for multi-hop questions. Details about the rule discovery process are explained as below:

## B.1 Implication Rules

For implication rule mining, we used translated the existing set of implication rules provided by (Wang et al., 2015; Fatemi et al., 2019). The translated rules were manually validated by local domain experts.

| Data | #Edits | 2-hop | 3-hop | 4-hop | Total |
|---|---|---|---|---|---|
| MQuAKE-CF-AR | 1 | 2454 | 855 | 446 | 3755 |
| | 2 | 2425 | 853 | 467 | 3745 |
| | 3 | | 827 | 455 | 1282 |
| | 4 | | | 436 | 436 |
| | All | 4879 | 2535 | 1804 | 9218 |
| MQuAKE-T-AR | 1 (All) | 1421 | 445 | 2 | 1868 |
| MQA-AEVAL | 1 (All) | 209 | 10 | 10 | 229 |

Table 4: Statistics of data sets.

## B.2 Compositional Rules

For compositional rule mining, we use RNN-Logic (Qu et al., 2020) as our mining tool, to extract a set of compositional logic rules $(\Psi_C)$ over Arabic Wikipedia[3]. After rule mining, we use rule's support threshold $(A_{\Psi_C})$ as a criterion for rule selection.

# C Additional Experimental Settings

## C.1 Dataset

For experimental evaluation, we curate an Arabic dataset named: MQA-AEVAL encompassing information about recent events in the Arabian Peninsula. Apart from that, we also use existing knowledge editing benchmarks under single-hop and multi-hop settings. These data sets were translated to the Arabic language followed by validation from native speakers. The statistics of the data set are given in Table 4, their details are as follows:
**(i) MQuAKE-AR.** MQuAKE-AR is the Arabic translated data of the o riginal MQuAKE data by (Zhong et al., 2023). We translate both components of MQuAKE, *i.e.,* MQuAKE-CF and MQuAKE-T. MQuAKE-CF-AR include 3,000 k-hop questions $(k \in 2, 3, 4)$ based on counterfactual editing. MQuAKE-T-AR is based on real-world knowledge changes to construct edit, but not given time scope. The statistics of data set is given in Table 4. For data translation, we use a semi-automated pipeline similar to the one used by (Pieri et al., 2024), *i.e.,* using a two step process: (i) iterative translation and scoring using LLMs (*e.g.,* ChatGPT), (ii) manual refinement of low scored samples as well as random samples from high-scoring samples.
**(ii) MQA-AEVAL.** Given that MQA-KEAL is focused on knowledge editing and corresponding multi-hop question answering for Arabic language. Thus, in order to rigorously test the performance of MQA-KEAL, we curated a new data set, namely: **M**ulti hop **Q**uestion **A**nswering under

[3] https://dumps.wikimedia.org/arwiki/

knowledge editing for Arabic-region EVALuation (MQA-AEVAL) The statistics of data set is given in Table 4. An instance of MQA-AEVAL is illustrated in the following example:

**Example.** An example instance of our newly generated data MQA-AEVAL is shown in Figure 4, given below.

| | |
|---|---|
| $\mathcal{E}$ | السعودية، وزير التعليم، حمد آل الشيخ ← يوسف البنيان |
| $q$ | من هو وزير التعليم في بلد جنسية مؤلف كتاب "حياة في الإدارة"؟ |
| $o$ | حمد آل الشيخ |
| $o^*$ | يوسف البنيان |
| $\mathcal{P}$ | (حياة في الإدارة، مؤلف، غازي القصيبي)<br>(غازي القصيبي، مواطن، السعودية)<br>(السعودية، وزير التعليم، حمد آل الشيخ) |
| $\mathcal{P}^*$ | (حياة في الإدارة، مؤلف، غازي القصيبي)<br>(غازي القصيبي، مواطن، السعودية)<br>(السعودية، وزير التعليم، يوسف البنيان) |

Figure 4: An example illustration of MQA-AEVAL.

### C.2 Baseline Models

The details about the baseline models are provided as follows:

**(i) Fine-tuning (FT).** It uses updated/edited knowledge to fine-tune the model parameters through gradient descent (Zhu et al., 2020).

**(ii) ROME.** ROME is based on the assumption that knowledge is stored in the feed-forward layers of transformer network. And, we can incorporate new knowledge in the model by simply locating and updating the parameters of these layers (Meng et al., 2022a).

**(iii) MEMIT.** MEMIT extends ROME by allowing a multi-edit scenario by editing multiple layers of model (Meng et al., 2022b).

**(iv) MeLLo.** MeLLo is a memory-based system to store the edited facts in an explicit memory, and prompts LLM to generate final response consistent with the edited facts (Zhong et al., 2023).

### C.3 Large Models

We use existing Arabic centric language models

**(a) GPT-3.5-TURBO-INSTRUCT.** GPT-3.5-TURBO-INSTRUCT is released by Open-AI[4], in November 2023. This model uses a context window of 16,385 tokens and is trained till on a training data with a cut-off date of September, 2021.

**(b) Jias-13B.** Jias-13B is a state-of-the-art Arabic-centric generative language model trained on a mixture of Arabic, English and programming languages text (Sengupta et al., 2023). We use foundation model with 13 billion parameters.

**(c) AceGPT-13B.** AceGPT-13B is an attempt to incrementally pre-train existing LLMs using Arabic data to incorporate Arabic grammar, culture and values (Huang et al., 2023b). We use 13 billion variant for foundation model.

### C.4 Evaluation Metrics

Details about the evaluation metrics and their mathematical formulation are provided as follows:

**(a) Multi-hop Accuracy (M-Acc).** M-Acc is used to compute the accuracy of the language models on multi-hop questions. For M-Acc, we use the same settings as Zhong et al. (2023). Formally, given a data instance $d = (\mathcal{E}, q, o, o^*, \mathcal{P}, \mathcal{P}^*)$, the calculation formula for M-Acc for the base model $f(\cdot)$ is as follows:

$$\mathbb{1}\left[\bigvee_{q \in \mathcal{Q}} [f(q) = o]\right]. \tag{4}$$

Likewise the M-Acc for the edited model $f^*(\cdot)$ is computed as:

$$\mathbb{1}\left[\bigvee_{q \in \mathcal{Q}} [f^*(q) = o^*]\right]. \tag{5}$$

**(b) Hop-wise Accuracy (H-Acc).** H-Acc is used to compute the correctness of the intermediate reasoning paths for MQA-KE. In order to compute H-Acc, we follow the same settings as outlined by Gu et al. (2023). Given edited knowledge path $\mathcal{P}^*$, we define H-Acc as:

$$\mathbb{1}\left[\bigwedge_{(s,r,o^*) \in \mathcal{P}^*} [f^*(s, r) = o^*]\right]. \tag{6}$$

## D Additional Experimental Results.

### D.1 Number of Hops

We also compute the M-Acc for MQA-KEAL under varying numbers of hops. For this, we report

[4] https://platform.openai.com/

the performance of MQA-KEAL for MQUAKE-T-AR. Corresponding results in Table 5 compared against the baseline models show that the baseline models experience a rapid decline in performance especially as the number of hops are greater than or equal to four ($\geq 4$). On the contrary MQA-KEAL yields relatively stable model performance with the increase in the number of hops.

| # Hops= | 2-hop | 3-hop | 4-hop |
|---|---|---|---|
| MeLLo | 85.67 | 75.67 | 23.45 |
| MQA-KEAL | 92.17 | 82.25 | 64.14 |

Table 5: M-Acc results for MQA-KEAL vs best performing baseline model using MQUAKE-T-AR and GPT-3.5-TURBO-INSTRUCTfor 1-edited cases under varying number of hops.

### D.2 Performance for English Language

We also compared the end-performance of MQA-KEAL for English language. Corresponding results of MQA-KEAL and MQUAKE-T data set compared against MeLLo (Zhong et al., 2023) are shown in Table 6. These results show MQA-KEAL outperforms MeLLo across both metrics (M-Acc, H-Acc) by a significant margin.

| **Method** | **MQUAKE-T** | | | | | |
|---|---|---|---|---|---|---|
| | 1-edited | | 100-edited | | 1868-edited | |
| | M-Acc | H-Acc | M-Acc | H-Acc | M-Acc | H-Acc |
| | GPT-3.5-TURBO-INSTRUCT | | | | | |
| MeLLo | 77.58 | 71.13 | 82.10 | 74.51 | 73.77 | 55.41 |
| MQA-KEAL | **90.13** | **82.85** | **87.02** | **81.15** | **79.73** | **71.78** |

Table 6: Performance comparison of MQA-KEAL compared against MeLLo (Zhong et al., 2023) for English language. For these results, we consider a batch of $k$ instances, *i.e.,* $k \in \{1, 100, 1868\}$ for MQUAKE-T. We boldface the best scores.

# E Prompts

## E.1 Prompts for Task Decomposition ($T_{\text{relation}}$)

Example Prompt for Task Decomposition

سؤال: من هو رئيس الدولة في البلد الذي تحمل فيه إيلي كيمبر الجنسية؟

نقطة البداية: إيلي كيمبر

مسار العلاقة: < إيلي كيمبر؛ مواطنة في؛ البلد > < البلد؛ رئيس الدولة هو؛ الشخص >

سؤال: ما هو مكان ولادة مؤسس الطائفة الدينية التي ينتمي إليها هيلاري من بواتييه؟

نقطة البداية: هيلاري من بواتييه

مسار العلاقة: < هيلاري من بواتييه؛ ينتمي إلى؛ الطائفة الدينية > < الطائفة الدينية؛ مؤسسها هو؛ الشخص > < الشخص؛ وُلد في؛ مكان الولادة >

سؤال: ما هي اللغة المستخدمة كوسيلة رسمية للتواصل في البلد الذي نشأت فيه الرياضة التي يلعبها ريان ثيريوت؟

نقطة البداية: ريان ثيريوت

مسار العلاقة: < ريان ثيريوت؛ يلعب الرياضة؛ الرياضة > < الرياضة؛ ينظمها؛ البلد > < البلد؛ اللغة الرسمية هي؛ اللغة >

سؤال: ما هو موقع مقر المؤسسة التي تعلم فيها مؤلف آجزيرة الكنزْ؟

نقطة البداية: جزيرة الكنز

مسار العلاقة: < جزيرة الكنز؛ مؤلفها هو؛ الشخص > < الشخص؛ تعلم في؛ الجامعة > < الجامعة؛ مقرها في؛ الموقع >

سؤال: ما هي الدولة المصدرة للرياضة التي يلعبها كول ألدريتش؟

نقطة البداية: كول ألدريتش

مسار العلاقة: < كول ألدريتش؛ يلعب المركز؛ المركز > < المركز؛ مرتبط برياضة؛ الرياضة > < الرياضة؛ نشأت في البلد؛ البلد >

سؤال: ما هي اللغات التي يجيدها رئيس سوريا؟

نقطة البداية: سوريا

مسار العلاقة: < سوريا؛ رئيسها هو؛ الشخص > < الشخص؛ يتحدث اللغات؛ اللغات >

إشارة:

- يجب أن تقتصر الإجابة على مسار العلاقة فقط دون أي كلمات أخرى.
- يُصاغ مسار العلاقة على النحو < الكائن؛ العلاقة؛ الهدف > ، ولا يجب أن يكون الكائن أو الهدف طويلًا جدًا.
- يرجى الحفاظ على جميع العلاقات أساسية وغير قابلة للتجزئة.
- يجب أن يكون الكائن في العلاقة هو الهدف في العلاقة السابقة. وعادةً ما يكون الكائن في العلاقة الأولى كيانًا مسمى.
- يجب تحديد الكيان المسمى في السؤال كنقطة انطلاق لمسار العلاقة.
- مطلوب تحديد كل من نقطة البداية ومسار العلاقة. تأكد من أن تنسيق الإخراج يتوافق مع المثال المذكور أعلاه.
- يرجى توليد مسار علاقة صالح يمكن أن يساعد في الإجابة على السؤال التالي:

### E.2 Prompts for Querying Target Model $\mathrm{T}_{\mathrm{query}}$

Example Prompt for Querying Target LLM

سؤال: ما هي عاصمة الدولة التي يحمل جنسيتها زوج إيفانكا ترامب؟
سؤال فرعي: إيفانكا ترامب، زوج؟
الإجابة المولدة: إيفانكا ترامب، زوج، جاريد كوشنر.
سؤال: من هو رئيس الدولة التي يحمل راين ويلسون جنسيتها؟
سؤال فرعي: رين ويلسون، بلد المواطنة؟
الإجابة المولدة: راين ويلسون، بلد المواطنة، الولايات المتحدة الأمريكية.
سؤال: من هي زوجة رئيس الدولة في الولايات المتحدة الأمريكية؟
سؤال فرعي: الولايات المتحدة الأمريكية، رئيس الدولة؟
الإجابة المولدة: الولايات المتحدة الأمريكية، الرئيس، دونالد ترامب.
سؤال: في أي قارة تقع دولة جنسية مؤسس الشركة المصنعة لجهاز آيفون ٥ ؟
سؤال فرعي: آيفون ٥ من إنتاج ؟
الإجابة المولدة: آيفون ٥ من إنتاج، شركة أبل.
يرجى إنشاء إجابة للسؤال التالي وسؤاله الفرعي: