Language Modeling with Editable External Knowledge

Anonymous ACL submission

Abstract

When the world changes, so does the text that people write about it. How do we build language models that can be easily updated to reflect these changes? One popular approach is retrieval-augmented generation, in which new documents are inserted into a knowledge base and retrieved during prediction for downstream tasks. Most prior work on these systems have focused on improving behavior during *prediction* through better retrieval or reasoning. This paper introduces ERASE, which instead improves model behavior *when new documents are created*, by incrementally deleting or rewriting other entries in the knowledge base each time a new document is encountered. In two new benchmark datasets evaluating models' ability to answer questions about a stream of news articles or conversations, ERASE improves accuracy relative to conventional retrieval-augmented generation by 7–13% (Mixtral-8x7B) and 6–10% (Llama-3-8B) absolute.[1]

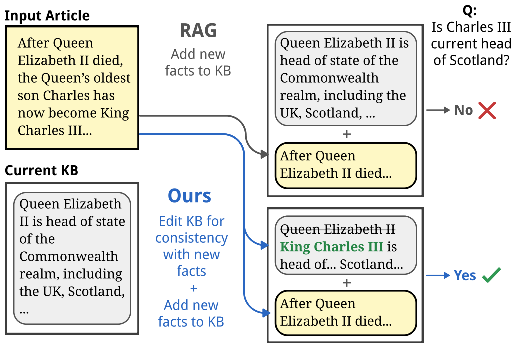

Input Article
After Queen Elizabeth II died, the Queen's oldest son Charles has now become King Charles III...
Current KB
Queen Elizabeth II is head of state of the Commonwealth realm, including the UK, Scotland, ...
RAG
Add new facts to KB
Ours
Edit KB for consistency with new facts
+
Add new facts to KB
Queen Elizabeth II is head of state of the Commonwealth realm, including the UK, Scotland, ...
+
After Queen Elizabeth II died...
~~Queen Elizabeth II~~
King Charles III is head of... Scotland...
+
After Queen Elizabeth II died...
Q:
Is Charles III current head of Scotland?
No ✗
Yes ✓

Figure 1: In standard retrieval augmented generation (RAG), new facts are simply added to an existing knowledge base $\mathcal{K}$. This can lead to stale facts in $\mathcal{K}$, which can in turn lead to incorrect predictions at inference time. In contrast, when ERASE reads a new input article, it not only adds new facts to $\mathcal{K}$, but also *updates* it. ERASE can edit or delete (not pictured) existing facts to keep $\mathcal{K}$ up to date, thereby enabling correct predictions at inference time. The same LM is used to update the memory and make predictions.

1 Introduction

The world—and the language we used to describe it—are constantly changing. Consider the example shown in Figure 1. After reading the article *After Queen Elizabeth II died, the Queen's oldest son Charles has now become King Charles III,* a knowledgeable reader might update an entire system of related beliefs, *e.g.,* that King Charles III is now also the new head of Scotland. How can we train language models and other software systems to reflect these changes?

Continual learning methods tackle the problem of a changing world by incrementally *training* on new information (Mitchell et al., 2018; Wang et al., 2024). But in language models, a simple (and often extremely effective) approach simply presents new information in models' inputs by leveraging either long-context methods (Tay et al., 2022) or retrieval augmented generation (RAG; Lewis et al., 2020a). which appends new documents to a knowledge base and retrieves a subset of relevant documents to condition on at prediction time (Guu et al., 2020; Lewis et al., 2020b).

An important limitation of current RAG approaches is that they sometimes retrieve *stale* documents that have been invalidated by new information. In Fig. 1, the article *After Queen Elizabeth II died...* would be appended to the existing knowledge base, which includes a fact about Queen Elizabeth's reign when she was alive, *e.g., Queen Elizabeth II is head of state of...Scotland.* When answering questions about the Scottish head of state, this document might be retrieved, leading the LLM to produce incorrect answers. Past attempts to address this issue have focused on improved *retrieval* methods, but not on ensuring accuracy and consis-

---

[1] Code and data will be made publicly available.

tency of the document collection itself.

This paper describes a method for retrieval-augmented generation that attempts to ensure that the external knowledge base always represents the current state of the world. This method, which we call ERASE (**E**nhancing **R**etrieval **A**ugmentation with **S**elf-consistent **E**diting; §3), achieves up-to-date knowledge by updating the knowledge base at *document insertion* time—*i.e.,* when new documents are read and added to the knowledge base—rather than at prediction time. Every time a new document is acquired, ERASE finds related documents in the knowledge base and decides whether to keep, edit, or delete them. These operations allow new information to be propagated and prevent stale information from being used for inference. In Figure 1, ERASE not only adds the new article to the knowledge base, but also *edits* the existing fact *~~Queen Elizabeth II~~* → ***King Charles III*** *is head of...Scotland*, thereby enabling correct prediction when this document is retrieved.

We evaluate ERASE's performance on question-answering (QA) tasks about a set of continually changing facts described by a stream of text. To do so, we introduce a new benchmark dataset, CLARK (**C**ontinual **L**earning **A**nd **R**evising **K**nowledge; §4), which contains two domains: (1) CLARK-NEWS, a factual QA domain consisting of a set of timestamped news articles paired with questions and timestamped answers; (2) CLARK-CONVERSATIONS, a long-conversation domain where facts about conversation participants evolve over the course of the conversation. The conversation domain contains both single-hop and multi-hop edits, the latter of which requires multi-hop inferences at the memory updating stage.

On this benchmark, ERASE outperforms standard RAG baselines and long-context models, giving 7–13% (Mixtral-8x7B) and 6–10% (Llama-3-8B) absolute improvements in accuracy compared to standard RAG on the factual QA domain and single-hop section of the conversation domain. On the multi-hop subset, we find that ERASE performs comparably to baselines, suggesting there is room for future work to improve multi-hop memory editing.

2 Background and Related Work

ERASE belongs to a growing body of work aimed at developing LM-based systems that can be updated after training. ERASE builds specifically on approaches that update LMs by modifying *inputs* rather than parameters—as discussed below, such methods are more flexible, and often more robust, than alternatives.

Long-context and retrieval-augmented generation: updating LMs via conditioning One simple and effective way to update LMs is simply to include new information in their input before inputs to the task of interest (e.g. by prepending a question about current events with a sequence of news articles). This approach begins to face challenges only when text containing new information is extremely long (e.g. comprising thousands of news articles). In these cases, it is neccessary either to use LMs specialized for very long input sequences, or to selecting a subset of inputs to condition on for each new query to the model (sometimes referred to as retrieval-augmented generation, or RAG).

Long-context models (Wang et al., 2020; Kitaev et al., 2020; Press et al., 2021; Su et al., 2024) focus on modifying LM architectures to allow long sequences to be processed efficiently, or extrapolation to long inputs. RAG methods, by contrast, dynamically retrieve relevant contexts tailored to individual queries (Guu et al., 2020; Lewis et al., 2020b). Previous work has explored auxiliary models that selectively choose when to perform retrieval (Mitchell et al., 2022b), or abstain from answering questions when retrieved sources present conflicting or outdated information (Chen et al., 2022; Zhang and Choi, 2023). Other work has examined augmenting LMs with *knowledge graphs* (Cai et al., 2023; Modarressi et al., 2024), structured relational knowledge bases that may be timestamped and whose nodes and edges may be updated. However, such structure can be difficult to construct and risks throwing away essential information; these methods are generally less used than unstructured knowledge bases.

Continual learning: updating LMs via fine-tuning A broader class of methods, applicable to a much broader class of machine learning models, study the problem of robustly performing **continual learning** under a non-stationary data distribution (Mitchell et al., 2018; Wang et al., 2024) via training objectives that ensure that new information is retained but old information is not forgotten (Jang et al., 2022; Mehta et al., 2023; Jang et al., 2023). Previous work on LMs has explored the use of continual pretraining (Jin et al., 2022), modified pretraining objectives (Xu et al., 2023), and synthetic data generation (Padmanabhan et al., 2023; Akyürek et al., 2024). Continual learning methods are computationally intensive and less widely used than RAG and related methods in language models.

Model editing: updating LMs with targeted interventions A final category of methods alter LM behavior by making targeted interventions to their parameters, either using specialized secondary "editing" models (Cao et al., 2021; Mitchell et al., 2022a) or performing closed-form updates (Meng et al., 2022, 2023). Current methods reliably update facts but not all their implications (Onoe et al., 2023; Hua et al., 2024), and are generally outperformed by retrieval- or fine-tuning-based methods.

Evaluating updates Few resources are currently available for evaluating models' ability to generate text about *changing* features of the world while attributing these changes to known source of information. The Entity Cloze by Date (ECBD) dataset contains entities from Wikidata along with cloze-style sentences (Onoe et al., 2022), and the LoCoMo dataset contains long conversations to measure long-term memory in models (Maharana et al., 2024); unlike CLARK, these datasets do not isolate entities whose properties *change* over time. Many datasets (Zhang and Choi, 2021; Chen et al., 2021; Meem et al., 2024; Dhingra et al., 2022; Kasai et al., 2023; Vu et al., 2023) have been released studying temporally-situated question answering; however, contexts in these datasets consist only of dates and not source documents. This makes it difficult to compare results across implementations: were improvements due to a better system, or simply due to a more complete set of documents in the knowledge base? In CLARK, we release both our questions and attributable source documents for those questions.

3 ERASE Method

We seek to develop a system that can generate text (e.g. for the question answering task depicted in Fig. 1) while updating its behavior in response to a continuous stream of documents describing a changing state of the world (e.g. the article about the death of Queen Elizabeth II, shown with a yellow background in Fig. 2). Informally, ERASE uses these documents to populate and edit a knowledge base that stores a collection of facts extracted from documents and represented as natural language strings (e.g. the identity of the new king, and the duration of Elizabeth II's reign, shown with gray backgrounds in Fig. 2). Importantly, the knowledge base records not just the content of each fact, but when it was first added, and (if relevant) when it ceased to be true. As new documents arrive, ERASE attempts to maintain the knowledge base in a *consistent* state—containing only facts that are currently true—by rewriting facts or marking them as false when contradictory facts are introduced by new articles (e.g. deleting facts about Elizabeth II's health and updating other references to the UK monarchy). During prediction, ERASE then operates like a normal RAG appach: retrieving true facts that are relevant to a given query.

More formally, we begin with a **language model** encoding a conditional distribution over strings $p_{\text{LM}}(\text{prediction} \mid \text{context})$. When a new **document** d_i is received with some **timestamp** τ_i, we update the **knowledge base** $\mathcal{K}$—each entry in $\mathcal{K}$ consists of both a **fact** f_j and a **fact history** $H_j = [(\tau_{j0}, v_{j0}), (\tau_{j1}, v_{j1}), \ldots]$, where each τ_{jk} is a timestamp and v_{jk} is a **truth value** indicating whether f_j was known to be true or false at time τ_{jk}. We then parse the new document into a sequence of facts f_j using the LM.

Unlike standard RAG methods, it is not in general necessary for facts extracted from documents to correspond one-to-one with facts in the knowledge base: knowledge base entries may also arise by editing old facts in response to new articles. To accomplish this, ERASE incorporates new documents into the knowledge base in three steps: **retrieval**, **updating**, and **adding**.

Step 1: Retrieve facts to edit.

$$R \leftarrow \texttt{Retrieve}(\mathcal{K}, d) \tag{1}$$

We retrieve a set of knowledge base entries $R = \{(f_{i_0}, H_{i_0}), \cdots (f_{i_m}, H_{i_m})\} \subset K$. Here we assume that the facts most likely to require *editing* in response to d are those most similar to d.[2] Following most modern RAG approaches (Lewis et al., 2020a), ERASE performs **dense vector retrieval**, using a learned embedding model $\mathcal{E}$ to assign documents and facts vector representations, then retrieve a set of m to optimize:

$$\texttt{Retrieve}(\mathcal{K}, d) = \underset{(f_j, H_j) \in \mathcal{K}}{\arg\text{ top-k}}\ \mathcal{E}(d)^\top \mathcal{E}(f_j) \tag{2}$$

---

[2] For efficiency, we retrieve facts relevant to the entire document in this step, rather than first parsing the document into facts, then retrieving facts relevant to each extracted fact.

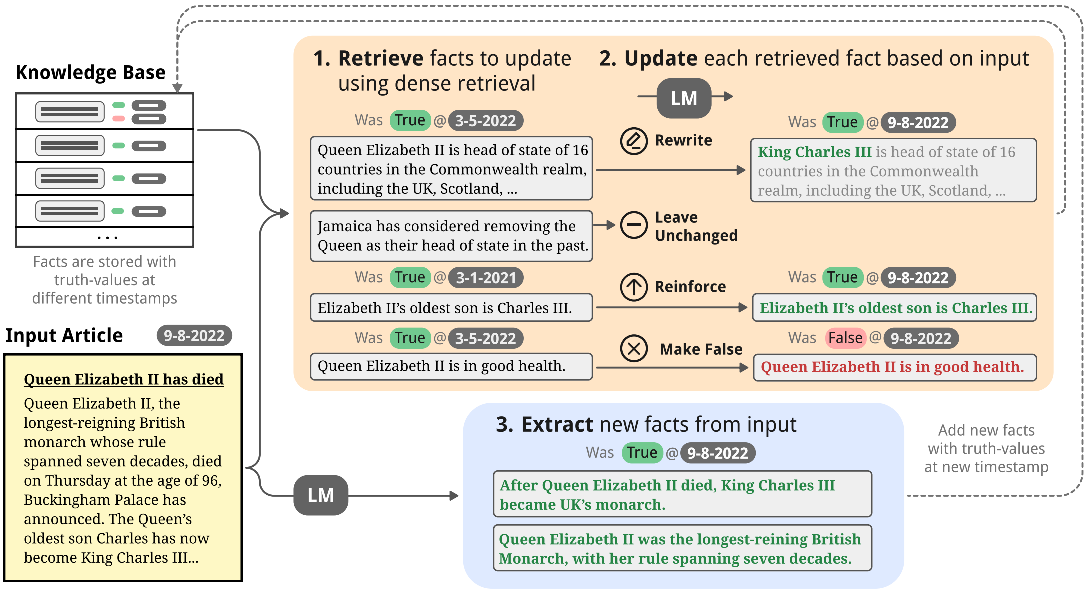

Knowledge Base
Facts are stored with truth-values at different timestamps
Input Article
9-8-2022
Queen Elizabeth II has died
Queen Elizabeth II, the longest-reigning British monarch whose rule spanned seven decades, died on Thursday at the age of 96, Buckingham Palace has announced. The Queen's oldest son Charles has now become King Charles III...
LM
1. Retrieve facts to update using dense retrieval
Was True @ 3-5-2022
Queen Elizabeth II is head of state of 16 countries in the Commonwealth realm, including the UK, Scotland, ...
Jamaica has considered removing the Queen as their head of state in the past.
Was True @ 3-1-2021
Elizabeth II's oldest son is Charles III.
Was True @ 3-5-2022
Queen Elizabeth II is in good health.
2. Update each retrieved fact based on input
LM
Rewrite
Leave Unchanged
Reinforce
Make False
Was True @ 9-8-2022
King Charles III is head of state of 16 countries in the Commonwealth realm, including the UK, Scotland, ...
Was True @ 9-8-2022
Elizabeth II's oldest son is Charles III.
Was False @ 9-8-2022
Queen Elizabeth II is in good health.
3. Extract new facts from input
Was True @ 9-8-2022
After Queen Elizabeth II died, King Charles III became UK's monarch.
Queen Elizabeth II was the longest-reining British Monarch, with her rule spanning seven decades.
Add new facts with truth-values at new timestamp

Figure 2: Overview of ERASE. We begin by retrieving existing facts relevant to input and prompting a LM to update them. We also extract facts from the input to add to our knowledge base.

Step 2: Update retrieved facts.

$$\forall (f_j, H_j) \in R,\ (f'_j, H'_j) \leftarrow \texttt{Update}(f_j, H_j, d, \tau)$$

$$\mathcal{K} \leftarrow \mathcal{K} \cup \{(f'_j, H'_j)\} \tag{3}$$

We update the knowledge base by modifying each retrieved fact $f_i \in R$ in one of the following ways:

- **Reinforce fact**: If the fact f is supported by d, we add $(\texttt{true}, \tau)$ to H. An example of such a case would be f = *Mary works in a warehouse* and d = *Mary came back from her job at UPS where she loaded and sorted packages all day*.

- **Keep fact unchanged**: If d is irrelevant to f or does not affect the truth value of f, then we do nothing and let $f' = f$ and $H' = H$. An example of such a case would be f = *Mary works in a warehouse* and f = *Mary took a jog in the park*.

- **Make fact false**: If f is contradicted by d, we add $(\texttt{false}, \tau)$ to H'. An example of such a case would be f = *Mary works in a warehouse* and d = *Mary got fired from her warehouse job*.

- **Rewriting**: Alternatively, if f is contradicted by d, we may *rewrite* it into a new expression f' that is inferrably true from d and the subset of retrieved facts $\subset R$ that have been *reinforced* or *kept unchanged*. We then replace the old KB entry (f, H) with a new KB entry $(f', [(\texttt{true}, \tau)])$.

For all operations above, we prompt a LM (the same LM that we use for evaluation) to classify each retrieved fact into one of *reinforce, no change, make false*.[3] We then iterate through all facts classified as *make false*, and ask the LM if it can rewrite the fact into a true expression. In this second phase, the LM is allowed to condition on facts that it classified as *reinforce* or *no change*, allowing it to potentially handle multi-hop edits. The full details of this procedure can be found in Appendix A.1.

Step 3: Add new facts.

$$\mathcal{K} \leftarrow \mathcal{K} \cup \texttt{Add\_facts}(T) \tag{4}$$

We add all new facts by conditioning on d and prompting the LM to extract atomic facts f. The prompt we use can be found in Appendix A.2. Analogously, Chen et al. (2023) used a *propositionizer* to decompose articles into propositions.

Prediction: To use an ERASE system after updating, generation is performed using a standard RAG pipeline described in step 1. We condition on both the retrieved facts and their corresponding

---

[3] The task in the first pass is similar to a fuzzy version of natural language inference classification. Inputs that make facts more likely (even if they do not exactly entail those facts) are classified as *support*, and inputs that make facts less likely (even if they do not exactly contradict those facts) are classified as *make false*.

history in context. The full prompt can be found in Appendix A.3.

4 Dataset

To construct datasets, we acquire a set of natural-language texts L_t, a set of ground truth world states W_t and a series of questions $q_0 \cdots q_n$ associated with W_t. We focus on questions that *update* over time: the set of questions we ask at each timestep are the same, but each question is associated with a list of timestamped answers $(q_i, \{(a_{i0}, t_{i0}), (a_{i1}, t_{i1}), \cdots\})$. We construct datasets in two domains where continual learning is useful: one about the evolving state of the world, and one about the evolving state of agents in a conversation. Samples from each dataset can be found in Figure 3. An overview of state transitions and questions in these two datasets can be found in Appendix C.

4.1 News Articles

World States In this domain, world states are expressed in the form `(subj, rel, obj)`: for instance, `(Elizabeth II, position held, monarch of the United Kingdom)`. We mine these triples from Wikidata.[4] As Wikidata is updated over time, each fact is also associated with a start and end date. To find changed facts, we extract `(subj, rel)` pairs for which there are at least two distinct fact relations at different timestamps between November 2021 and April 2024. Through this process, we obtain 1,174 triples for 10 unique relations, summarized in Table 8.

Documents For each world state `(subj, rel, obj, start_ts, end_ts)`, where the start and end timestamps are extracted from Wikidata, we obtain an English article confirming that fact between the start and end timestamps, validated by crowd workers. Through this process, annotators collected a total of **1149** articles.[5] See Appendix B.1 for details. These documents—rather than raw relation triples—are the input to ERASE.

Questions and Answers We automate the generation of questions and answers from W by writing templates for each relation and generating questions and answers from those templates. We generated a total of **1409** questions. The full list of templates can be found in Appendix B.1.

4.2 Synthetic Conversations

Following prior work (Maharana et al., 2024), we construct a synthetic conversation domain by placing two LLMs with different personas in conversation with each other. Conversations are engineered to reflect changing facts in the agents' simulated lives. A detailed overview of dataset construction can be found in Appendix B.2. To validate the LM generations, three authors manually examined 3 conversations (1008 questions) in total and got an average of 95% accuracy on these questions.

This synthetic domain allows us to rigorously control and evaluate forms of reasoning that may be hard to isolate in natural data like news articles.

World States We generate an independent world for each conversation. We model the world underlying a conversation as a Markov chain with states S, described by a list of `(subj, rel, obj)` relations, and allowable transitions $T(S)$. States S are defined by entities including people, companies, jobs, hobbies, along with mutable and immutable relations between them. Transitions $t \in T(S)$ change one or more relation in the state: for example, *Bob changed jobs to work at Google* changes the *employees* of Google, the set of *coworkers* of Bob, the set of *coworkers* of all Google employees, and the set of *coworkers* of all employees of Bob's former company, etc. At each timestep, we sample a transition from $T(S)$ uniformly at random. The full list of entities, relations, and transitions and their downstream effects can be found in Appendix B.2.

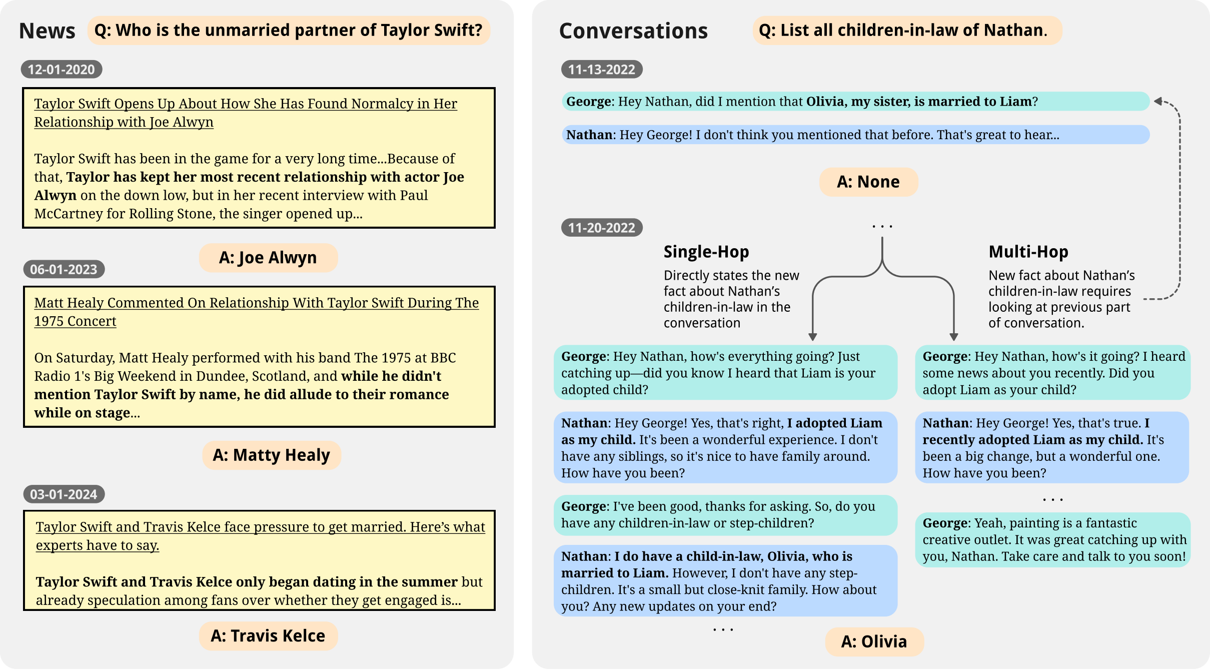
Conversations We generate conversations by sampling two people in the world p_1 and p_2 and prompting two LLMs with their corresponding personas and the initial world state S. We then generate twelve conversation "chunks"—separated by time—by sampling state transitions between *every other* chunk and having people converse about the facts that have changed after each transitions.

We also construct a challenge set of *multi-hop* updates in this domain, which require propagating changes to multiple downstream facts and reasoning about global coherence between facts. For example, Bob may mention that he has changed his job but may not mention that *Jane is no longer his coworker* or that *Mary (who works at Google) is*

---

[4] `https://www.wikidata.org/`, which is public domain. Its license can be found at `https://www.wikidata.org/wiki/Wikidata:Licensing`.

[5] Note 1149 $<$ 1174, meaning at least a few articles were shared across relations – these represent difficult cases where a single article makes multiple relation changes.



News **Q: Who is the unmarried partner of Taylor Swift?**

12-01-2020

Taylor Swift Opens Up About How She Has Found Normalcy in Her Relationship with Joe Alwyn

Taylor Swift has been in the game for a very long time...Because of that, **Taylor has kept her most recent relationship with actor Joe Alwyn** on the down low, but in her recent interview with Paul McCartney for Rolling Stone, the singer opened up...

A: Joe Alwyn

06-01-2023

Matt Healy Commented On Relationship With Taylor Swift During The 1975 Concert

On Saturday, Matt Healy performed with his band The 1975 at BBC Radio 1's Big Weekend in Dundee, Scotland, and **while he didn't mention Taylor Swift by name, he did allude to their romance while on stage**...

A: Matty Healy

03-01-2024

Taylor Swift and Travis Kelce face pressure to get married. Here's what experts have to say.

Taylor Swift and Travis Kelce only began dating in the summer but already speculation among fans over whether they get engaged is...

A: Travis Kelce

Conversations **Q: List all children-in-law of Nathan.**

11-13-2022

George: Hey Nathan, did I mention that **Olivia, my sister, is married to Liam**?

Nathan: Hey George! I don't think you mentioned that before. That's great to hear...

A: None

11-20-2022

. . .

Single-Hop

Directly states the new fact about Nathan's children-in-law in the conversation

George: Hey Nathan, how's everything going? Just catching up—did you know I heard that Liam is your adopted child?

Nathan: Hey George! Yes, that's right, **I adopted Liam as my child.** It's been a wonderful experience. I don't have any siblings, so it's nice to have family around. How have you been?

George: I've been good, thanks for asking. So, do you have any children-in-law or step-children?

Nathan: **I do have a child-in-law, Olivia, who is married to Liam.** However, I don't have any step-children. It's a small but close-knit family. How about you? Any new updates on your end?

. . .

Multi-Hop

New fact about Nathan's children-in-law requires looking at previous part of conversation.

George: Hey Nathan, how's it going? I heard some news about you recently. Did you adopt Liam as your child?

Nathan: Hey George! Yes, that's true. **I recently adopted Liam as my child.** It's been a big change, but a wonderful one. How have you been?

. . .

George: Yeah, painting is a fantastic creative outlet. It was great catching up with you, Nathan. Take care and talk to you soon!

A: Olivia

Figure 3: Sample data from our datasets. The News dataset consists of factual questions whose answers change over time, with the associated source inducing that change. The Conversations dataset consists of conversations between two personas with evolving life facts. The single-hop subset directly states all facts that are changed, while the multi-hop subset requires reasoning about previous chunks of conversation to infer all changes.

now his coworker. The LM must make multi-hop inferences to update the latter two facts.

We generate **100** conversations (50 single-hop, 50 multi-hop) in total. Conversations were on average **11045** tokens long in the single-hop subset and **11069** tokens long in the multi-hop subset. Detailed statistics may be found in Appendix Figure 7.

Questions and Answers Given a world state at time t, we query *all* facts about the world. Similar to the news setting, we automate generation of questions and answers through templates. We generate **140** questions per conversation.

5 Experiments

In our experiments, we present to a LM articles or conversational turns in chronological order, and periodically ask questions about the state of the world (as described by input documents) at that point in time.

5.1 Evaluation and Metrics

News articles We present the model with a stream of articles ordered by timestamp. As all answers are dated with a start and end timestamp, we always know which answer is true for a given timestamp.[6] We ask questions at regular intervals, at timesteps corresponding to when 20%, 40%, 60%, 80%, and 100% of the total world state changes have been revealed to the model. Because it is too expensive to ask every question at every timestep, we ask *all questions whose answers have changed* Q, then sample a subset of *questions whose answers have not changed* Q', such that $|Q'| = |Q|$. We design each question as a multiple choice question, where the model is asked to select between all answers that have been true for the question in the past, present, or future. This ensures that the negative options are sufficiently difficult, and allows us to probe for the models' updating capabilities. We report exact-match accuracies between the model-predicted answer to the true answer.

Conversation We evaluate each conversation independently, and report the mean and standard error of scores over each conversation. We stream in *chunks* of conversations into the model, and ask questions after each conversation chunk. Similarly to the news domain, we subsample questions whose answers have not changed, such that at each timestep we are asking the same number of ques-

---

[6] Note that this does not correspond to when these facts became true and false in the real world, but rather to when the article introducing the changed fact was written and read.

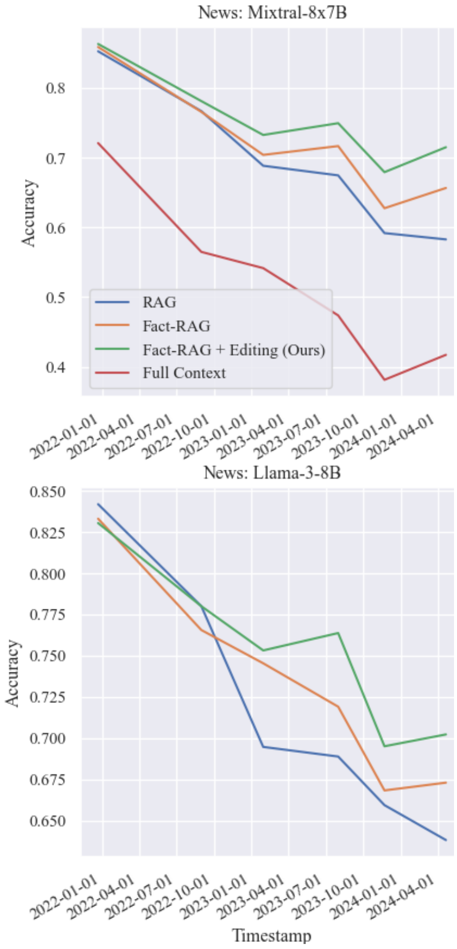


Figure 4: Mixtral-8x7B (top) and Llama-3-8B (bottom) results on the news article domain. ERASE outperforms RAG, RAG with fact-level granularity, and even long-context models, especially in later timesteps as more new information is learned.

tions whose answers have changed as those whose answers haven't changed. For questions that have multiple true answers (e.g. `List all siblings of Liam`), we measure the set equality between the generated and true sets of answers. Otherwise, we use the same exact match accuracy as we use for the news articles domain.

5.2 Models

We use a Mixtral 8x7b Instruct model (56B parameters; Jiang et al., 2024), queried using Together AI[7], and a local copy of Meta's Llama-3 8b Instruct model (8B parameters ; AI@Meta, 2024) run on one NVIDIA A100 GPU.[8] For all prompts during inference and update-time, we sample from the LM with temperature 0. We use GTR (T5-large; 770M parameters; Ni et al., 2022) as $\mathcal{E}$ to encode queries and documents for dense retrieval, both in the inference stage and the retrieval step of updating. We use a fast inner-product search datastructure for efficient retrieval (Douze et al., 2024). For prompting during the updating stage, we use the same LM that we are using for inference. We restrict the context window to 4096 for the news domain and 2048 for the conversation domain.[9] Inference and updating took a few hours to complete for both models and for all method. At inference time, we allow all models to perform zero-shot chain-of-thought, giving them an additional ability to reason about inconsistent facts at inference time.

5.3 Baselines

We compare ERASE to three baselines:

RAG RAG (Lewis et al., 2020a) stores and retrieves text at the granularity of *passages*. We save each article and conversation chunk as a separate passage in the knowledge base. For long articles and conversation chunks, we divide them into passages of length `context_window / 2`.

Fact-RAG To isolate the effects of *editing*, we benchmark against a version of RAG that stores and retrieves *facts* in the knowledge base, akin to Chen et al. (2023). We implement this baseline by prompting LMs to extract facts from passages, i.e. step 3 of ERASE, which outperformed the propositionizer from Chen et al. (2023).

Long context LMs Mixtral-8x7B has a long context window of 32k. We run an in-context learning baseline by conditioning Mixtral on all news articles or conversation chunks, presented in chronological order. These texts are timestamped, and Mixtral is able to condition on the most recent set of texts up to its context limit when making predictions. In the Conversations domain, this condition serves as a skyline since conversations fit completely into the context window.

6 Results

Figure 4 and Table 1 show results for the news and conversation domains respectively.

---

[7] `https://www.together.ai/`

[8] Llama-3 8b has knowledge cutoff of March 2023. Mixtral's has not been published, but appears to be around late 2022 or early 2023.

[9] Note this is smaller than the original context windows for these models, both to run our experiments efficiently, and to test out a (realistic) scenario where the total number of new world changes cannot fit into the context window of a language model.

		Data Subset					
			Single-hop			Multi-hop	
		0 updates	1 update	2+ updates	0 updates	1 update	2+ updates
Mixtral-8x7B	RAG (Lewis et al., 2020a)	$\mathbf{86.0}_{\pm \mathbf{0.7}}$	$56.7_{\pm 1.8}$	$50.9_{\pm 3.2}$	$\mathbf{84.5}_{\pm \mathbf{0.8}}$	$\mathbf{20.9}_{\pm \mathbf{1.4}}$	$20.0_{\pm 2.3}$
	Fact-RAG (Chen et al., 2023)	$82.7_{\pm 0.8}$	$51.5_{\pm 1.8}$	$52.7_{\pm 3.1}$	$81.8_{\pm 0.8}$	$18.0_{\pm 1.3}$	$\mathbf{30.2}_{\pm \mathbf{2.7}}$
	ERASE (Ours)	$82.0_{\pm 0.8}$	$\mathbf{59.1}_{\pm \mathbf{1.8}}$	$\mathbf{57.9}_{\pm \mathbf{3.1}}$	$81.5_{\pm 0.8}$	$\mathbf{20.1}_{\pm \mathbf{1.4}}$	$27.2_{\pm 2.6}$
	Full Context	$88.8_{\pm 0.6}$	$71.6_{\pm 1.6}$	$75.7_{\pm 2.4}$	$88.4_{\pm 0.6}$	$43.2_{\pm 1.7}$	$54.3_{\pm 2.8}$
Llama-3-8B	RAG (Lewis et al., 2020a)	$\mathbf{84.4}_{\pm \mathbf{0.7}}$	$57.8_{\pm 1.8}$	$55.2_{\pm 3.1}$	$\mathbf{83.6}_{\pm \mathbf{0.8}}$	$22.2_{\pm 0.1}$	$26.8_{\pm 2.6}$
	Fact-RAG (Chen et al., 2023)	$82.6_{\pm 0.8}$	$62.6_{\pm 1.7}$	$62.0_{\pm 3.0}$	$81.2_{\pm 0.8}$	$\mathbf{26.4}_{\pm \mathbf{1.6}}$	$\mathbf{32.1}_{\pm \mathbf{2.8}}$
	ERASE (Ours)	$82.0_{\pm 0.8}$	$\mathbf{65.3}_{\pm \mathbf{1.7}}$	$\mathbf{65.2}_{\pm \mathbf{2.9}}$	$81.0_{\pm 0.8}$	$\mathbf{26.5}_{\pm \mathbf{0.2}}$	$\mathbf{31.7}_{\pm \mathbf{2.7}}$

Table 1: Results on the synthetic conversation domain. Full context serves as a skyline in this domain as the full conversation fits into the context window. We compare against other retrieval-based methods. In **bold** are results that are the **statistically significantly best** out of all other methods in the same setting (model, data subset, # updates). While ERASE significantly improves single-hop edits in both models, it still struggles with multi-hop edits. Small LMs make errors in multi-hop reasoning during the overwriting stage, and suspect that as LMs improve multi-hop reasoning, we will see greater gains with ERASE.

* We merge 2+ updates as generally there is a long tail of questions with more updates. Only 27 questions total have 3+ updates.

ERASE improves over vanilla RAG with passage retrieval For both Mixtral and Llama-3 in both domains, we see significant improvements using ERASE over RAG, particularly as the number of edits increases. For example, in the news domain, at the final timestamp after reading all articles, Mixtral with ERASE is 13 points better than Mixtral with RAG, while Llama with ERASE is about 6 points better than Llama with RAG. We see similar trends on the single-hop subset of the conversation domain: for questions with 2+ updates, ERASE is 7 and 10 points better than RAG, using Mixtral and Llama respectively.

Editing existing facts improves beyond RAG with fact retrieval For both Mixtral and Llama-3, ERASE substantially improves performance over Fact-RAG as the number of edits increases, on both the news domain and the single-hop subset of the conversation domain. Improving knowledge base consistency helps, *even with step-by-step reasoning* at inference-time.

In the news domain, ERASE improves over long-context modeling In Figure 4, we plot Mixtral with its full context window on the news domain. Long-context models are unable to scale as more articles are added. However, we find that ERASE (and retrieval methods generally) are unable to compete against fitting full conversations in the context window Table 1. That said, the cost of conditioning on full conversations is greater than the cost of conditioning on simply retrieved facts, especially as the number of queries per conversation increases.[10]

Multi-hop retrieval and edits are still challenging Both LMs struggle with the multi-hop subset of the conversation dataset. We believe this isn't a drawback of fact editing itself, but of our implementation of it: a qualitative examination of failure cases (see Appendix D.1 for some examples) revealed that our retrieval model often failed to retrieve all downstream facts that need to be edited, and language models on the scale of Mixtral-8x7b and Llama-3-8b struggled with reasoning about multi-hop edits, failing to make those edits when necessary. A more powerful retrieval and editing model may be able to avoid these errors.

7 Conclusion

In this paper, we introduced ERASE, an approach for *editing existing facts* in a knowledge base when new documents are being inserted. We also introduced two datasets for testing the ability of models to update their knowledge, accompanied by documents that induce those changes. Editing existing facts brings significant improvements to RAG-based models. Even if future models become better at reasoning about inconsistencies with scale, fact editing is useful for amortizing the cost of reasoning about consistency *at insertion time*, rather than having to re-evaluate consistency each time a fact is queried. Future work can focus on improving any part of the update pipeline, particularly focusing on retrieving downstream facts (step 1) that will be affected by an input (which is different from retrieving simply *relevant* facts), and improving LM ability to perform multi-hop updates (step 2).

---

[10] Conditioning Mixtral on full conversations costs 7.3K tokens per query, whereas retrieval costs $\sim$ 1.7K tokens per query + a fixed cost of $\sim$ 42k tokens per conversation chunk. Generally in the real world that the number of queries far outflanks the number of documents generated about changes in the world. In our dataset without subsampling, full context would cost 102M tokens while ours would cost 28M tokens.

Limitations

As noted in Section 6, ERASE is still subpar for multi-hop updates, largely due to retrieval model's inability to retrieve all the necessary facts and the LMs' inability to reason about multi-hop edits. We believe that this limitation can be mitigated with better retrieval models and better LMs.

Second, because LMs have a tendency to hallucinate, allowing LMs to directly edit the knowledge base may introduce noise into the knowledge base. While our results found that the utility of propagation was greater than any hindrance due to such noise, this noise has the potential to snowball on long timescales as the number of new passages and edits grows beyond tens of thousands, hundreds of thousands, or millions. That said, we do not believe this limitation is inherent to knowledge-base editing: future work can explore more principled and rigorous approaches to editing with guarantees around what edits are made and to how many facts. Furthermore, we believe that for any approach to model editing, there is a natural tradeoff between noise and edit coverage.

Finally, having to process each document and update the knowledge base is less efficient than simply adding it to the retrieval store. We justify this cost by assuming that the number of insertions is far fewer than the number of queries. (For example, Forbes reports that 252,000 websites are created per day[11], while Google receives about 8.5 billion searches daily[12].) Thus, by shifting the cost of reasoning about consistency from query-time to insertion-time, ERASE is arguably *more efficient* in practice than RAG.

Ethical Considerations

Being able to interpretably edit models is useful for improving the safety and trustworthiness of models. If there is misinformation in the knowledge base, our method allows these facts to be corrected quickly and these corrections to propagate through the knowledge base. Our method magnifies the effect of each change, making it easy for system designers to keep knowledge up-to-date and remove any stale or incorrect knowledge. Conversely however, this could also empower malicious actors to insert false facts, which will also be propagated through the knowledge base. There will need to be safeguards in place to ensure that any inserted and propagated knowledge is from reliable sources, with potential vetting of each inserted article. One of the pros of ERASE is that we can see every LM operation occurring in real time: any update operation can be examined manually to ensure that the changes are desirable.

References

AI@Meta. 2024. Llama 3 model card.

Afra Feyza Akyürek, Ekin Akyürek, Leshem Choshen, Derry Wijaya, and Jacob Andreas. 2024. Deductive closure training of language models for coherence, accuracy, and updatability. In *Findings of the Association for Computational Linguistics*.

Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2023. Temporal knowledge graph completion: A survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6545–6553. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. *Preprint*, arXiv:2104.08164.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2307, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2023. Dense x retrieval: What retrieval granularity should we use? *Preprint*, arXiv:2312.06648.

Wenhu Chen, Xinyi Wang, William Yang Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

---

[11] `https://www.forbes.com/advisor/business/software/website-statistics/`

[12] `https://seo.ai/blog/how-many-people-use-google`

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Wenyue Hua, Jiang Guo, Mingwen Dong, Henghui Zhu, Patrick Ng, and Zhiguo Wang. 2024. Propagation and pitfalls: Reasoning-based assessment of knowledge editing through counterfactual tasks. *Preprint*, arXiv:2401.17585.

Joel Jang, Seonghyeon Ye, Changho Lee, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, and Minjoon Seo. 2023. Temporalwiki: A lifelong benchmark for training and evaluating ever-evolving language models. *Preprint*, arXiv:2204.14211.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. 2022. Towards continual knowledge learning of language models. *Preprint*, arXiv:2110.03215.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew Arnold, and Xiang Ren. 2022. Lifelong pretraining: Continually adapting language models to emerging corpora. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 1–16, virtual+Dublin. Association for Computational Linguistics.

Jungo Kasai, Keisuke Sakaguchi, yoichi takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. 2023. Realtime QA: What's the answer right now? In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *arxiv*.

Jannat Ara Meem, Muhammad Shihab Rashid, Yue Dong, and Vagelis Hristidis. 2024. Pat-questions: A self-updating benchmark for present-anchored temporal question-answering. *Preprint*, arXiv:2402.11034.

Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2023. Dsi++: Updating transformer memory with new documents. *Preprint*, arXiv:2212.09744.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36. ArXiv:2202.05262.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. 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. 2022a. Fast model editing at scale. In *International Conference on Learning Representations*.

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

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2018. Never-ending learning. *Commun. ACM*, 61(5):103–115.

Ali Modarressi, Abdullatif Köksal, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. 2024. Memllm: Finetuning llms to use an explicit read-write memory. *Preprint*, arXiv:2404.11672.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. 2022. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages

9844–9855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yasumasa Onoe, Michael Zhang, Eunsol Choi, and Greg Durrett. 2022. Entity cloze by date: What LMs know about unseen entities. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 693–702, Seattle, United States. Association for Computational Linguistics.

Yasumasa Onoe, Michael Zhang, Shankar Padmanabhan, Greg Durrett, and Eunsol Choi. 2023. Can LMs learn new entities from descriptions? challenges in propagating injected knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5469–5485, Toronto, Canada. Association for Computational Linguistics.

Shankar Padmanabhan, Yasumasa Onoe, Michael J.Q. Zhang, Greg Durrett, and Eunsol Choi. 2023. Propagating knowledge updates in lms through distillation. *Advances in Neural Information Processing Systems*, 36.

Ofir Press, Noah A Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6).

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and Thang Luong. 2023. Freshllms: Refreshing large language models with search engine augmentation. *Preprint*, arXiv:2310.03214.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *Preprint*, arXiv:2302.00487.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.

Yan Xu, Mahdi Namazifar, Devamanyu Hazarika, Aishwarya Padmakumar, Yang Liu, and Dilek Hakkani-Tur. 2023. KILM: Knowledge injection into encoder-decoder language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5013–5035, Toronto, Canada. Association for Computational Linguistics.

Michael Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michael Zhang and Eunsol Choi. 2023. Mitigating temporal misalignment by discarding outdated facts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14213–14226, Singapore. Association for Computational Linguistics.

A Prompts for ERASE

In this section, we list all prompts that we use for each step of our method.

A.1 Fact Updating

In practice, we implement these operations by performing *two passes* over the retrieved facts. In the first pass, we prompt the LM with the input d and each fact $f \in R$ and prompt it to *classify* the fact into one of *reinforce, no change, make false*. From this first pass, we divide the retrieved facts into two sets: R_{true}, comprising facts that remain true (*reinforce, no change*), and R_{false}, comprised of facts that have become false (*make false*). In the second pass, we iterate through R_{false}, and prompt the LM to rewrite the fact into a true fact (if possible), conditioned on the new document d and R_{true}. This serves a few purposes:

1. If f is only made partially false by d, we may retain information expressed in f but not d. For example, if f is *Mary and Bob work at UPS*, and d is *Mary got fired from UPS*, we may rewrite f as *Bob works at UPS*, rather than negating the entire fact.
2. Conditioning on R_{true} allows the LM to make *multi-hop* edits. For example, if f is *Mary is coworkers with Bob*, and d is *Mary changed workplaces to Amazon*, if R_{true} contains *Quinn works at Amazon*, then we can rewrite f as *Mary is coworkers with Quinn*.

First round: classifying facts as becoming more or less likely to be true.

```
[Input] [Timestamp: {ts}] {context} [End Input]

The fact "{fact}" was previously true. In light of the input, is "{fact}" likely still true as of {ts}? Begin by summarizing the changes we learned from the input, then reasoning briefly about them to give your final answer with "Answer: Reinforce" (if the input makes the fact more likely) or "Answer: Make False" (if the input makes the fact less likely) or "Answer: No Change" (if the input doesn't affect the fact, e.g. if the input is irrelevant to the fact). Assume that the fact is still true (keep true) if nothing in the input contradicts it.
```

Second round: extracting rewrites

```
[Input] [Timestamp: {ts}] {context}
Other True Facts at {ts}: {", ".join(still_true_facts)}
[End Input]

The fact "{fact}" was previously true but no longer. Given the above input and true facts, can you rewrite it into one that is true as of {ts}? Output your answer in form "rewrite: rewritten fact" or "no rewrite possible".
```

A.2 Fact Extraction

```
Extract all facts from the input text, with each fact on a new line and without bullet points or numbered lists. Facts should be simple, independent, standalone, and decontextualized. Break up long facts into smaller facts. Resolve all references (e.g. pronouns, definite articles, etc.) by copying full reference object everywhere it is referenced. Only include facts referring to the current world state (what is true *now*), as opposed to facts true in the past. If there are no facts, please output "No new facts." Do not include any other text.
```

A.3 Inference

Given a question `question` at timestep `ts` (and choices `answer_choices`), We first retrieve facts $f_i, [(\tau_{i0}, v_{i0}), (\tau_{i1}, v_{i1}), \cdots]$ from the knowledge base with similarity threshold > 0.7 to `question`. We then prompt a LM with the following:

```
Read the statements/passages below then answer the question below

***BEGIN STATEMENTS***
{f_i} ({v_{i0}} at {tau_{i0}}, {v_{i1}} at {tau_{i1}}, ...)
{f_j} ({v_{j0}} at {tau_{j0}}, {v_{j1}} at {tau_{j1}}, ...)
...
***END STATEMENTS***

Given the above statements are true and any prior knowledge you have, answer the following question at timestep {ts}?:
```

```
{question}

Briefly reason then answer with one of:
{answer_choices}.
```

For questions requiring list answers (e.g. list all the siblings of Rachel), we replace the last line with:

```
Briefly reason then answer with a JSON
list, ["item1", "item2", ...], of zero
or more of the following items: {
answer_choices}. If you include any of
the above items, make sure to copy their
 names exactly as is from the list. Your
 list may be empty, [], if none of the
answers are true.
```

B Dataset Construction Details

B.1 News Articles

We construct this dataset in three stages:

Extracting World States W. We retrieve `(subj,rel)` pairs from Wikidata for which there are at least two distinct fact relations at different timestamps, e.g. `(subj,rel,obj1,start_ts1,end_ts1)` and `(subj,rel,obj2,start_ts2,end_ts2)`. These timestamped facts are used to "represent" W. We filter for subjects `subj` located in English-speaking countries to ensure we can find English-language sources. We use SPARQL[13] to obtain a set of `(subj,rel)` pairs.

Obtaining Documents L. We annotate each timestamped relation, `(subj,rel,obj,start_ts, end_ts)` with a source written between `start_ts` and `end_ts` (preferably close to the `start_ts`) stating that the `(subj,rel,obj)` relation is true. We crowdsource annotations from Prolific in two stages. In the first stage, Prolific annotators were presented with an interface which scraped candidate news articles off of Google[14], and were asked to select sources which stated that the fact `(subj,rel,obj,start_ts, end_ts)` is true, but **did not** state that any succeeding fact, `(subj,rel,obj2,start_ts2, end_ts2)` where `start_ts2` $>$ `start_ts`, is true. In the second stage, we validated Prolific annotations from the first stage by presenting articles from the first round of annotations to annotators in the second round, and asking users whether those articles contained the fact in question. If second annotator does not affirm the fact is present in the article, we throw out the fact and the associated annotation. We do an additional third round of filtration with a language model, asking the language model to affirm that the text of an article contains `(subj,rel,obj,start_ts, end_ts)` but not any succeeding facts `(subj,rel,obj2,start_ts2, end_ts2)`. We only include articles and facts that pass all three rounds of annotation. We recruited English-speaking participants from the US for annotations for all annotations. The full set of instructions we give annotators can be found in Tables 2 and 3. Screenshots of the interface can be found in Figures 5 and 6.

Generating Question-Answers Pairs $(q, \{a\})$. We automate generation of questions and answers from W by writing templates for each relation and generating questions and answers from those templates. The full list of templates can be found in Table 4.

Prolific Details We recruited a total of 680 English-speaking prolific annotators from the United States, with each annotator spending an average of 16:50 minutes on the task ($\sim$ 7 minutes to read and understand instructions). We paid annotators an average of \$14.20 per hour. This task was deemed exempt from IRB review. No personally-identifiable information was collected or stored, and all prolific annotators were associated with an anonymous prolific ID.

B.2 Synthetic Conversations

We also construct this dataset in three stages:

Generating World States W. We model the underlying world and its transformations as a Markov chain with states S and a set of allowable transitions $T(S)$ determined by S. At each timestep, we randomly sample a transition from $T(S)$ uniformly at random. States S are described by a set of relations `(subj, rel, obj)`. The full list of entities types and relations for each entity type can be found in Table 5. To construct each world, we subsample 10 people and 5 companies, and randomly initialize their kinship and employment relations. Transitions $t \in T(S)$ change one or more relation in the

---

[13] https://www.w3.org/TR/sparql11-query/

[14] In particular, we set the to-be-matched parameter of the search to "news", i.e. https://www.google.com/?tbm=nws

Please read these instructions carefully and only proceed once you have understood them. Once you start the task, you will have 10 minutes to get through as many questions as possible.
For each question, you will be presented a fact. Please find a news article that implies that the fact is true, according to the below requirements:

1. The article implies the fact, such that a reasonable person, without any prior knowledge, can infer that the fact is true from reading the article.

   Example: For fact Emad Mostaque is CEO of Stability AI (was True from 2020 to 2024-03-23)

   Good Sources: This startup is setting a DALL-E 2-like AI free, consequences be damned: Article says "...Stability AI CEO and founder Emad Mostaque wrote in a blog post"

   Bad Sources: Artists can now opt out of the next version of Stable Diffusion: Cannot conclude fact from text of article

2. The article is a news article or blog post.

   Example: For fact Taylor Aylmer is a member of the Racing Louisville FC sports tea

   Good Sources: Team News: Aylmer to make first regular season start

   Bad Sources: Taylor Aylmer - Racing Louisville FC Midfielder - ESPN, Taylor Aylmer - Instagram

3. The fact is stated in the main body of the article text, not in a table, list, image, image caption, embedded tweet, etc.

   Example: For fact Taylor Aylmer is a member of the Racing Louisville FC sports team

   Good Sources: Team News: Aylmer to make first regular season start, Recap: Racing rallies to beat Orlando, keep playoff hopes alive: Fact is in a list at the end, not the main text

   Bad Sources: Jaelin Howell, Racing Louisville bring community together to help people with Down syndrome: Fact is in an image caption but nowhere in the main text

4. The article is a web page, not a PDF or other file format.

   Example: For fact Ali Shojaie is a IMS Fellow

   Good Sources: Ali Shojaie elected fellow of the Institute of Mathematical Statistics

   Bad Sources: IMS Carver Award 2023: Source is a PDF file, not a web page

5. The article is written in English.

   Example: For fact Emad Mostaque is CEO of Stability AI (was True from 2020 to 2024-03-23)

   Good Sources: This startup is setting a DALL-E 2-like AI free, consequences be damned

   Bad Sources: [Bengali article]: Article is not in English

6. Avoid articles that state that the fact is or is about to become false. These are generally written near or past the end date of a fact being true.

   Example: For fact Emad Mostaque is CEO of Stability AI (was True from 2020 to 2024-03-23)

   Good Sources: This startup is setting a DALL-E 2-like AI free, consequences be damned

   Bad Sources: Stability AI founder Emad Mostaque plans to resign as CEO, sources say: Article is about the fact being about to be false

If no listed articles satisfy these requirements, you have the option to either find a news article that satisfies the requirements (a google search link is provided for reference, you may need to manually adjust the query or date parameters) or selecting "cannot find source" if you cannot find any source in a reasonable amount of time.
There may also be a second fact that you need to avoid. If you see this fact in the article, do not select it as a source.
Tip: You may use "ctrl-f" (find tool) to quickly validate whether or not a fact is in the article.

Table 2: Instructions for round 1 of annotation for news article.

Please read these instructions carefully and only proceed once you have understood them. Once you start the task, you will have 12 minutes to get through as many questions as possible.
For each question, you will be presented a fact and a news article. Please confirm that the news article implies that the fact is true, and conforms to the below requirements:

1. The article implies the fact, such that a reasonable person, without any prior knowledge, can infer that the fact is true from reading the article.

   Example: For fact Emad Mostaque is CEO of Stability AI (was True from 2020 to 2024-03-23)

   Good Sources: This startup is setting a DALL-E 2-like AI free, consequences be damned: Article says "...Stability AI CEO and founder Emad Mostaque wrote in a blog post"

   Bad Sources: Artists can now opt out of the next version of Stable Diffusion: Cannot conclude fact from text of article

2. The article is written in English.

   Example: For fact Emad Mostaque is CEO of Stability AI (was True from 2020 to 2024-03-23)

   Good Sources: This startup is setting a DALL-E 2-like AI free, consequences be damned

   Bad Sources: [Bengali article]: Article is not in English

3. Avoid articles that state that the fact is or is about to become false. These are generally written near or past the end date of a fact being true.

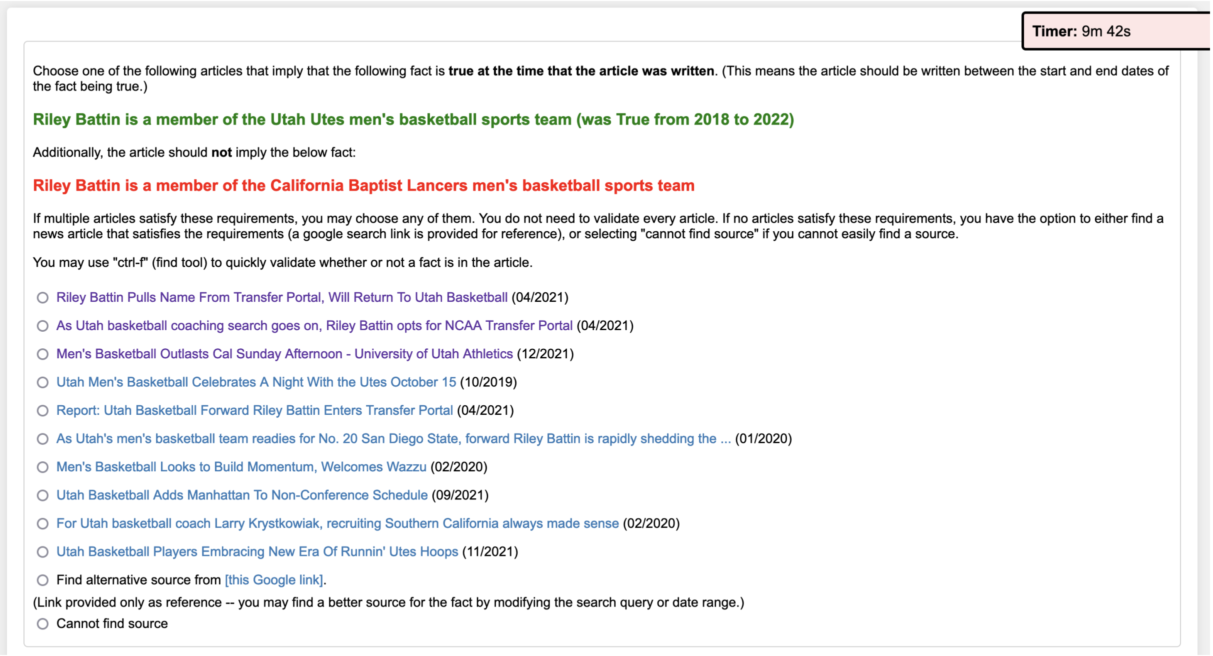
   Example: For fact Emad Mostaque is CEO of Stability AI (was True from 2020 to 2024-03-23)

   Good Sources: This startup is setting a DALL-E 2-like AI free, consequences be damned

   Bad Sources: Stability AI founder Emad Mostaque plans to resign as CEO, sources say: Article is about the fact being about to be false

If the provided article does not satisfy these requirements, you have the option to either find a news article that satisfies the requirements (a google search link is provided for reference, you may need to manually adjust the query or date parameters) or selecting "cannot find source" if you cannot find any source in a reasonable amount of time.
There may also be a second fact that you need to avoid. If you see this fact in the article, do not select it as a source.
Tip: You may use "ctrl-f" (find tool) to quickly validate whether or not a fact is in the article.

Table 3: Instructions for round 2 of annotation for news article.



Choose one of the following articles that imply that the following fact is **true at the time that the article was written**. (This means the article should be written between the start and end dates of the fact being true.)

Riley Battin is a member of the Utah Utes men's basketball sports team (was True from 2018 to 2022)

Additionally, the article should **not** imply the below fact:

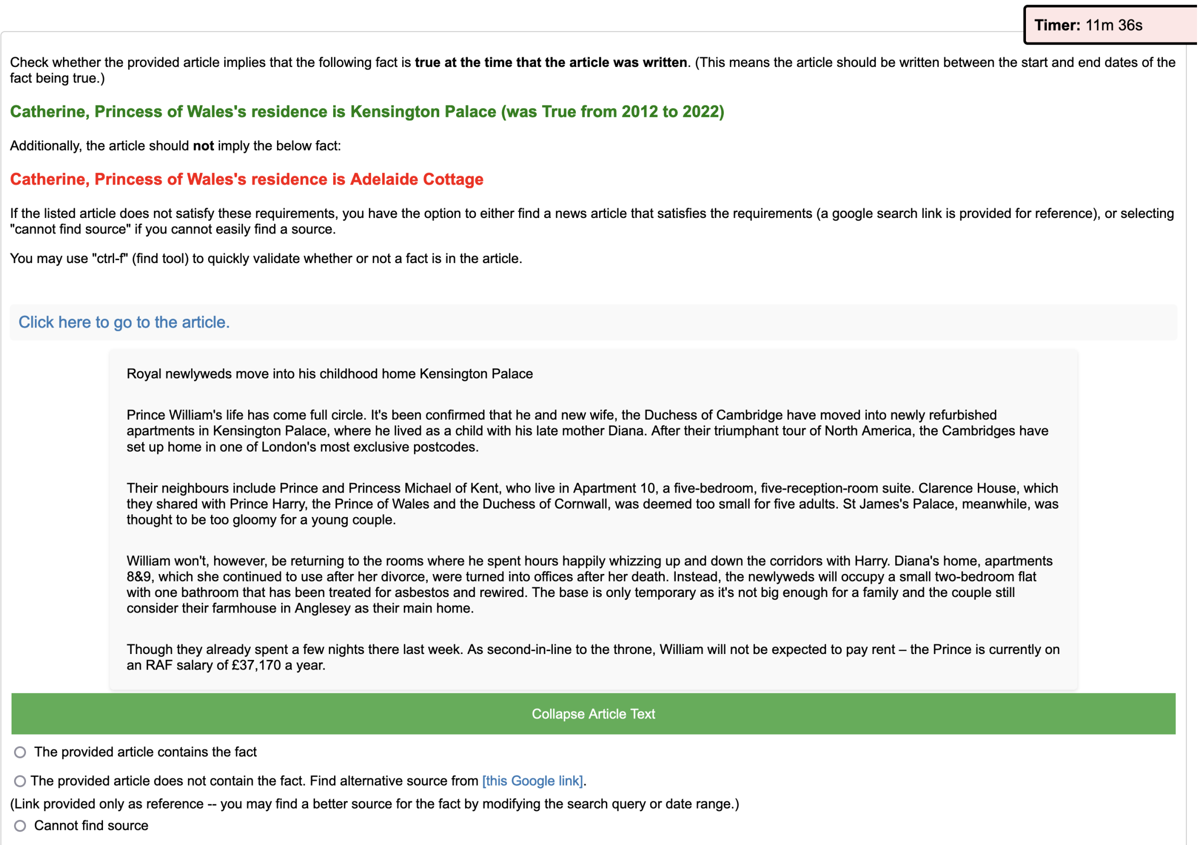
Riley Battin is a member of the California Baptist Lancers men's basketball sports team

If multiple articles satisfy these requirements, you may choose any of them. You do not need to validate every article. If no articles satisfy these requirements, you have the option to either find a news article that satisfies the requirements (a google search link is provided for reference), or selecting "cannot find source" if you cannot easily find a source.

You may use "ctrl-f" (find tool) to quickly validate whether or not a fact is in the article.

○ Riley Battin Pulls Name From Transfer Portal, Will Return To Utah Basketball (04/2021)
○ As Utah basketball coaching search goes on, Riley Battin opts for NCAA Transfer Portal (04/2021)
○ Men's Basketball Outlasts Cal Sunday Afternoon - University of Utah Athletics (12/2021)
○ Utah Men's Basketball Celebrates A Night With the Utes October 15 (10/2019)
○ Report: Utah Basketball Forward Riley Battin Enters Transfer Portal (04/2021)
○ As Utah's men's basketball team readies for No. 20 San Diego State, forward Riley Battin is rapidly shedding the ... (01/2020)
○ Men's Basketball Looks to Build Momentum, Welcomes Wazzu (02/2020)
○ Utah Basketball Adds Manhattan To Non-Conference Schedule (09/2021)
○ For Utah basketball coach Larry Krystkowiak, recruiting Southern California always made sense (02/2020)
○ Utah Basketball Players Embracing New Era Of Runnin' Utes Hoops (11/2021)
○ Find alternative source from [this Google link].
(Link provided only as reference -- you may find a better source for the fact by modifying the search query or date range.)
○ Cannot find source

Figure 5: Screenshot of round 1 of annotation for news article.



Timer: 11m 36s

Check whether the provided article implies that the following fact is **true at the time that the article was written**. (This means the article should be written between the start and end dates of the fact being true.)

Catherine, Princess of Wales's residence is Kensington Palace (was True from 2012 to 2022)

Additionally, the article should **not** imply the below fact:

Catherine, Princess of Wales's residence is Adelaide Cottage

If the listed article does not satisfy these requirements, you have the option to either find a news article that satisfies the requirements (a google search link is provided for reference), or selecting "cannot find source" if you cannot easily find a source.

You may use "ctrl-f" (find tool) to quickly validate whether or not a fact is in the article.

Click here to go to the article.

Royal newlyweds move into his childhood home Kensington Palace

Prince William's life has come full circle. It's been confirmed that he and new wife, the Duchess of Cambridge have moved into newly refurbished apartments in Kensington Palace, where he lived as a child with his late mother Diana. After their triumphant tour of North America, the Cambridges have set up home in one of London's most exclusive postcodes.

Their neighbours include Prince and Princess Michael of Kent, who live in Apartment 10, a five-bedroom, five-reception-room suite. Clarence House, which they shared with Prince Harry, the Prince of Wales and the Duchess of Cornwall, was deemed too small for five adults. St James's Palace, meanwhile, was thought to be too gloomy for a young couple.

William won't, however, be returning to the rooms where he spent hours happily whizzing up and down the corridors with Harry. Diana's home, apartments 8&9, which she continued to use after her divorce, were turned into offices after her death. Instead, the newlyweds will occupy a small two-bedroom flat with one bathroom that has been treated for asbestos and rewired. The base is only temporary as it's not big enough for a family and the couple still consider their farmhouse in Anglesey as their main home.

Though they already spent a few nights there last week. As second-in-line to the throne, William will not be expected to pay rent – the Prince is currently on an RAF salary of £37,170 a year.

○ The provided article contains the fact

○ The provided article does not contain the fact. Find alternative source from [this Google link].

(Link provided only as reference -- you may find a better source for the fact by modifying the search query or date range.)

○ Cannot find source

Figure 6: Screenshot of round 2 of annotation for news article.

`({subj}, employer, {obj})`	`Who is the employer of {subject}?`
	`Is {subject} an employee of {object}?`

`({subj}, chief executive officer, {obj})`	`Who is the CEO of {subject}?`
	`What company is {object} the CEO of?`
	`Is {object} the CEO of {subject}?`

`({subj}, chairperson, {obj})`	`Who is the chairperson of {subject}?`
	`What organization is {object} the chairperson of?`
	`Is {object} the chairperson of {subject}?`

`({subj}, head of state, {obj})`	`Who is the head of state of {subject}?`
	`Where is {object} the head of state of?`
	`Is {object} the head of state of {subject}?`

`({subj}, position held, {obj})`	`What government position does {subject} hold?`
	`Does {subject} hold government position {object}?`

`({subj}, member of sports team, {obj})`	`What sports team is {subject} a member of?`
	`Is {subject} a member of {object}?`

`({subj}, unmarried partner, {obj})`	`Who is the unmarried partner of {subject}?`
	`Who is the unmarried partner of {object}?`
	`Is {object} the unmarried partner of {subject}?`

`({subj}, residence, {obj})`	`Where does {subject} reside?`
	`Does {subject} reside in {object}?`

`({subj}, headquarters location, {obj})`	`Where is the headquarters location of {subject}?`
	`Is the headquarters location of {subject} in {object}?`

`({subj}, P463, {obj})`	`What organization is {subject} a member of?`
	`Is {subject} a member of {object}?`

`({subj}, member of political party, {obj})`	`What political party is {subject} a member of?`
	`Is {subject} a member of {object}?`

Table 4: Question-answer templates in the News domain

state. To be able to test the limits of our propagation, the set of transitions we define in this domain all change more than one relation: for example, "*Bob changed jobs to work at Google*" changes the *employees* of Google, the set of *coworkers* of Bob, the set of *coworkers* of all Google employees, and the set of *coworkers* of all employees of Bob's former company, etc. The full list of transitions and their downstream effects can be found in Table 6.

Generating Conversations L. We generate conversations by sampling two people in the world p_1 and p_2 and prompting two LLMs with their corresponding personas and initial facts. We then generate twelve conversation "chunks" as follows: We begin by sampling the next transition we want to make in the world. The transition corresponds to a natural language string that corresponds to only a single relation. However, we know that each transition is associated with multiple changing relations. To be able to infer the *downstream* changes of a single relation changing, we need to know auxiliary facts related to the *object* of the changed relation. In the multi-hop subset of this dataset, we mention auxiliary facts in the *prior* conversation chunks, while only mentioning the immediate transition (on a single relation) in the current chunk (*without* mentioning any downstream changes). Thus, to make the correct downstream inferences on this subset, the system must retrieve and reason across facts from prior conversation chunks.

For the singlehop subet, we mention *all downstream effects* in the same conversation chunk that a transition is made.

Generating Question-Answers Pairs $(q, \{a\})$. Given a world state at time t, we query *all* facts about the world. Similar to the news setting, we automate generation of questions and answers through templates. Templates in this setting can be found in Table 7.

C Dataset Statistics

The breakdown of changes in each of our datasets can be found in Table 8 for news articles and Figure 7 for conversations. The breakdown of questions for conversations can be found in Table 9.

D Qualitative Analysis

D.1 Error Analysis: Conversations (Multihop)

Prototypical examples of multihop edit errors can be found below:

Retrieval Errors Sometimes, the full list of facts that need to be updated are not retrieved. For example:

Input conversation chunk:

> *2023-11-01*
>
> *Katie: Hey Olivia! How have you been? Guess what? I've changed my job to Library Assistant at Central Public Library! What's new with you?*
>
> *Olivia: Hey Katie! That's amazing news! Congrats on the new job as a Library Assistant at Central Public Library!*
>
> ...

Retrieved facts:

- *The role of General Practitioner at HealthFirst Medical Clinic is a full-time job.*
- *The role of Library Assistant at Central Public Library is a full-time job.*
- *Rachel has a full-time job as a Medical Assistant at HealthFirst Medical Clinic.*
- *Rachel works at HealthFirst Medical Clinic as a Medical Assistant.*
- *The work hours of a Library Assistant at Central Public Library are from 9 to 17.*
- *Katie works full-time at the Urban Development Project.*
- *Olivia works full-time at the airport.*
- *The salary for a Library Assistant at Central Public Library is $80,000.*
- *Peter works from 9 to 17 at the Central Public Library.*
- *Diana and Liam both have full-time jobs at the Urban Development Project.*
- *The salary of an Archivist at Central Public Library is $130,000.*

Entity Type	Possible Relations
Person	spouse, parents, children, job, company, hobbies, coworkers, work location, boss, salary, industry, is-employed-full-time, work hours, workplace, siblings, parents-in-law, children-in-law, step-parents, step-children, equipment necessary for hobbies
Company	employees, jobs, head, location, industry, workplace type
Job	company, salary, is-full-time, work hours
Hobby	equipment necessary for hobby

Table 5: Full list of entities and relations defining each world state in the Conversation domain.

Transition type	Downstream effects
`person.job` changes from `job1` to `job2`	person.company, person.coworkers, person.work-location, person.boss, person.salary, person.industry, person.is-employed-full-time, person.work-hours, person.workplace, job1.company.employees, job2.company.employees
`person.spouse` changes from `person1` to `person2`	person.parents-in-law, person.parents.children-in-law, person.children.step-parents, person.step-children, person1.spouse, person1.parents-in-law, person1.parents.children-in-law, person2.spouse, person2.parents-in-law, person2.parents.children-in-law, person2.children.step-parents, person2.step-children
`person` adopts `child`	person.children, child.parents, child.siblings, child.spouse.parents-in-law, person.children-in-law, child.step-parents, person.spouse.step-children, person.children.siblings
`person` gets a new hobby `hobby`	person.equipment-necessary-for-hobbies
`job.salary` changes	for all people that have that job: person.salary
`job.work-hours` changes	for all people that have that job: person.work-hours

Table 6: Full list of possible state transitions in the Conversation domain. Note the set of available transitions may vary depending on the underlying state.

`({subj}, spouse, {obj})`	`Who is the spouse of {subj}?`
	`Who is the spouse of {obj}?`

`({subj}, job, {obj})`	`What is the job of {subj}?`
`({subj}, company, {obj})`	`Which company does {subj} work at?`
`({subj}, hobbies, {obj})`	`List all known hobbies of {subj}.`
`({subj}, coworkers, {obj})`	`List all known coworkers of {subj}.`
`({subj}, work location, {obj})`	`In which city does {subj} work?`
`({subj}, boss, {obj})`	`Who is the head of {subj}'s workplace?`
`({subj}, salary, {obj})`	`What is the salary of {subj}?`
`({subj}, industry, {obj})`	`What industry does {subj} work in?`
`({subj}, is-employed-full-time, {obj})`	`Does {subj} work full-time or part-time?`
`({subj}, work-hours, {obj})`	`What are the work hours of {subj}?`
`({subj}, workplace, {obj})`	`What type of workplace does {subj} work out of?`
`({subj}, parents, {obj})`	`List all parents of {subj}.`
`({subj}, children, {obj})`	`List all children of {subj}.`
`({subj}, siblings, {obj})`	`List all siblings of {subj}.`
`({subj}, parents-in-law, {obj})`	`List all parents-in-law of {subj}.`
`({subj}, children-in-law, {obj})`	`List all children-in-law of {subj}.`
`({subj}, step-parents, {obj})`	`List all step-parents of {subj}.`
`({subj}, step-children, {obj})`	`List all step-children of {subj}.`
`({subj}, necessary equipment for hobby, {obj})`	`List all equipment {subj} needs for their hobbies.`

Table 7: Question-answer templates in the Conversation domain

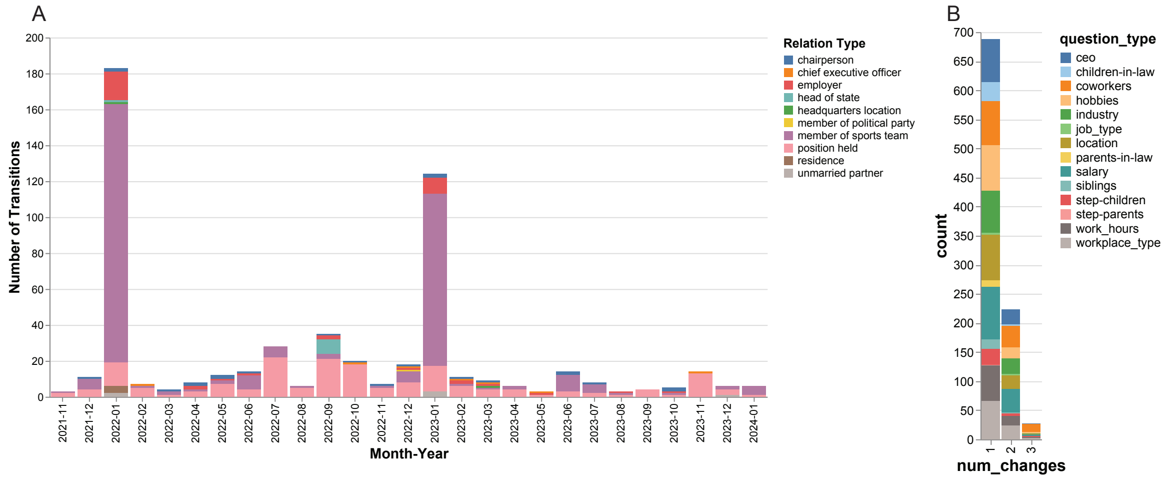


Figure 7: Distribution of changed relation types in the (A) News and (B) Conversation domains. (A) depicts the number of transitions per year of each month, while (B) depicts the number of relation types that underwent that number of changes across all conversations.

Relation type	`# (s, r)`	`# (s, r, o)`
Member of sports team	284	382
Position held	164	382
Employer	38	77
Chairperson	20	42
Head of state	9	18
CEO	6	13
Unmarried partner	5	12
Residence	4	8
Headquarters	2	4
Member of political party	1	2
Total	533	1174

Table 8: Breakdown of changed relation types in the News domain, categorized by number of unique `(subj, rel)` pairs and `(subj, rel, obj)` triples.

Question Topic	# Yes/No	# Multiple Choice	# MC Choices
Boss	140	74	26
Coworkers	481	-	-
Industry	-	74	26
Is employed full-time	82	-	-
Salary	158	80	11
Work hours	110	64	10
Work Location	274	72	20
Workplace	140	74	26
Total	1385	438	

Table 9: Distribution of generated questions in the Synthetic Conversation domain, categorized by question topic and type.

The retriever at step 1 failed to retrieve the following facts to update. At inference time, these facts were retrieved and affected the LM's answer on questions:

- *Katie's working hours are from 10 to 15.*
- *Katie's salary is $100000.*
- *Diana and Liam work with Katie.*

Updating Errors Sometimes, facts that needed to be updated were retrieved, but were not updated by the LM. For example,

Input conversation chunk:

> *2023-11-26*
>
> *Katie: Hey Olivia! How have you been? I heard you changed your job to Safety Officer at Urban Development Project! How's the new role treating you?*
>
> *Olivia: Hey Katie! Yes, I did change my job to Safety Officer at Urban Development Project! The new role has been quite a change, but I'm really enjoying it so far. How have you been?*
>
> ...

Retrieved fact:

- *Olivia works from 9 to 17 at her job.*

Model-outputted update (and reasoning):

```
Summary of changes:


    Olivia changed her job to Safety Officer at Urban Development Project.

    No information is provided about Olivia's work schedule.
```

```


Reasoning:

The input does not provide any
information that would suggest
Olivia's work schedule has
changed. The conversation is
about Olivia's new job, Katie's
hobbies, and their plans for
social activities. There is no
mention of Olivia's work
schedule or any changes that
would affect it. Therefore, the
fact "Olivia works from 9 to 17
at her job" is likely still true
 as of 2023-11-01.

Answer: No Change
```

Note that this fact should've been flagged for potential updating and forwarded to the rewrite step (where the LM has an opportunity to retrieve more facts to confirm or deny this fact). However, the LM does not classify this fact as *make false* because it is not directly contradicted by the input.

E Use of AI Assistants

Code was written with Co-pilot turned on. GPT* models were also consulted for creating acronyms for the method and dataset names.